Structured Machine Learning for Mapping Natural Language to Spatial Ontologies

Parisa KORDJAMSHIDI

Supervisory Committee:
Prof. dr. Adhemar Bultheel, chair
Prof. dr. Marie-Francine Moens, supervisor
Prof. dr. ir. Hendrik Blockeel
Prof. dr. Luc De Raedt
Prof. dr. ir. Johan Suykens
Prof. dr. ir. Tinne Tuytelaars
Prof. dr. ir. Luc Van Gool
Prof. dr. Dan Roth
(University of Illinois)
Dr. ir. Martijn Van Otterlo
(Radboud University Nijmegen)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

June 2013

Celestijnenlaan 200A box 2402, B-3001 Heverlee (Belgium)

D/2013/7515/82
ISBN 978-94-6018-696-7

Dedicated to my father and to the memory of my mother.

Dedicated to my dear Hamid.

The face of my beloved is reflected in my cup
Little you know why with wine, I always myself align.
Eternal is the one whose heart has awakened to love.
This is how eternal records my life define.
(Hafez, 1325-1390)

Acknowledgement

I found the PhD to be a great and funny caricaturist working on your whole personality. If you are unlucky, he suddenly points to your tiny bit oversized nose and draws it as if nothing is visible but a big nose in your face, and simplifies your eyes as two tiny dots, as if you have seen nothing through them in your life.

Struggling with such a cruel artist, one should be lucky to have people, I call them *relatives*, who never lose their belief in you, some relatives who know all the corners of your personality better than yourself, some relatives who simply seem to be passengers passing by while making deep inspirations that remain forever in you, and some relatives whose 'being': understanding, peaceful mind and smile, is just a great phenomenon in your life; and I was lucky to have such relatives.

This thesis would not have been possible without the help, support and patience of my supervisor, Professor Marie-Francine Moens whom I gratefully thank. She, being hardworking and ambitious, has been a great motivating example for me when dealing with my research problems. She is great at pointing out the novel and important research problems and I am thankful to her for introducing me to such an interesting research topic.

It was a great luck to have Martijn van Otterlo helping me in proceeding with my research. I enjoyed working with him and, the productive research, interesting discussions that we had about my research and about various things that he reads and knows are unforgettable. His support and the way he gave value to all kinds of ideas was important to me and always gave me self-confidence.

I would like to specially thank Professor Dan Roth. From the beginning of my PhD, I have been reading and thinking about his elegant and important research and his great ideas. It was incredible to have him as a jury member and to discuss with him about my research.

I would like to thank other members of my jury committee Professor Tinne Tuytelaars, Professor Hendrik Blockeel, Professor Luc Van Gool and Professor Johan Suykens for spending time on evaluating my thesis, and for providing useful comments and suggestions. I would like to thank Professor Adhemar Bultheel for chairing my PhD committee. My special thanks go to Professor Luc De Raedt, it was a great honor to have his feedback and advise on my research.

I would like to mention more people who, in one way or another, have had influence on my work. I would like to thank Professor Maurice Bruynooghe, he has been my first link to the DTAI group and the CS department in Leuven. I enjoyed a lot working with Joana Hois, I started the most exciting period of my PhD when I got to know her and her research. The first CoSLI workshop and later the Bremen visit were amazing, I could not have had these great experiences and memories if would not know her. It was nice to have known Steven Bethard and organize the SpRL workshop with him. To my experience, his calm and realistic attitude was very inspiring. It was a valuable experience working for a short while with Professor Paolo Frasconi whom I would like to thank.

I would like to thank my colleagues and friends particularly the people who spent time reading my research texts and also parts of this thesis: Kurt De Grave, Nima Taghipour, Jan De Belder, Thomas Provoost, Mathias Verbeke, Anton Dries, also for his help whenever I had technical problems. I would like to thank Bernd Gutmann for his help during the time I was working and learning about Problog, Nfoil, etc, and Niels Landwehr for his quick feedbacks in the same period. Moreover, thanks to Bogdan Moldovan, for his companion in the coffee breaks.

I would like to thank my dear friend Sabine Drebusch for her open and kind cooperation in annotating the very first dataset for SpRL and also Tigist Kassahun for annotating my main dataset.

It was my good fortune to meet Professor Ross D. King during my PhD. It was great to have the chance to talk to him about my research, and to receive his advise on how to do research.

And finally, the unconditional love, belief and support of my family in the hard moments, many times, has been the only reason that I could hold on to achieve my ambitions. I would like to thank and express my love to my spouse Hamid for his unconditional love and belief in me and his great patience at all times, and to my dear daughter Berelian for her incredible understanding. I gratefully thank my father who always has been an example and a great advisor for me in dealing with the challenges in my life. I would like to thank my sister Maria

who is alway a kind and warm support for me. Being so organized and precise, as her, is alway a dream that I never could achieve. I would like to thank my brother Eghbal, who is a real friend to me, and from my childhood on has been a great influence and motivation for me to choose the direction of my studies. Maybe if he was not supporting my real interests I would have written a thesis in medicine instead of computer science! And my younger sister Parnian who is like a mirror, I am amazed how she can read my odd thoughts! And my younger brother Aref who follows the same path of computer science: do not worry, just be Intelligent, it is only Artificial!

Parisa Kordjamshidi

Leuven, June 2013

Abstract

Natural language understanding is one of the fundamental goals of artificial intelligence. An essential function of natural language is to talk about the location, and translocation of objects in space. Understanding spatial language is important in many applications such as geographical information systems, human computer interaction, the provision of navigational instructions to robots, visualization or text-to-scene conversion, etc.

Due to the complexity of spatial primitives and notions, and the challenges of designing ontologies for formal spatial representation, the extraction of the spatial information from natural language still has to be placed in a well-defined framework. Machine learning has not systematically been applied to the task, and no established corpora are available. In this thesis I study the problem from cognitive, linguistics and computational points of view, with a primary focus on establishing a supervised machine learning framework.

This thesis makes five main research contributions. The first is the design of a spatial annotation scheme to bridge between natural language and formal spatial representations. In this scheme the universal and commonly accepted cognitive spatial notions and multiple well-known qualitative spatial reasoning models are applied.

The second is the definition of a novel computational linguistic task that utilizes the annotation scheme to map natural language to spatial ontologies. For this task I have built rich annotated corpora and an evaluation scheme.

The third is a detailed investigation of the linguistic features and structural characteristics of spatial language that aid the use of machine learning in extracting spatial roles and relations from annotated data. The learning methods used are discriminative graphical models and statistical relational learning.

The fourth is the proposal of a unified structured output learning model for ontologies. The ontology components are learnt while taking into account the ontological constraints and linguistic dependencies among the components. The ontology includes roles and relations, and multiple formal semantic types.

The fifth is the proposal of an efficient inference approach based upon constraint optimization. It can deal with a large number of variables and constraints, and makes building a global structured learning model for ontology population, feasible. To test the approach I have performed an empirical investigation using my spatial ontology.

The application of my proposed unified learning model for ontology population is not limited to the extraction of spatial semantics, it could be used to populate any ontology. I argue therefore that this work is an important step towards automatically describing text with semantic labels that form a structured ontological representation of the content.

Beknopte samenvatting

Het begrijpen van natuurlijke taal door een machine is één van de fundamentele doelstellingen van de kunstmatige intelligentie. Een essentiële functie van natuurlijke taal betreft het communiceren van de locatie en translocatie van objecten in de ruimte. Inzicht in ruimtelijke taal is belangrijk voor vele toepassingen zoals geografische informatiesystemen, human computer interactie, het verstrekken van navigatie-instructies aan robots, visualisatie en tekst-naar-scene conversie.

Vanwege de complexiteit van ruimtelijke primitieven en begrippen, en de uitdagingen bij het ontwerpen van een ontologie voor de formele ruimtelijke representatie moet de extractie van de ruimtelijke informatie uit natuurlijke taal in een welomschreven kader geplaatst worden. Machinaal leren is nog steeds niet systematisch toegepast op de extractietaak, en er zijn nog geen beschikbare corpora om de leeralgoritmen te trainen. In dit proefschrift bestudeer ik de problemen vanuit de cognitieve, taal- en computationele invalshoeken, met een primaire focus op het ontwikkelen van een kader voor gesuperviseerd machinaal leren.

Dit proefschrift draagt bij tot vijf belangrijke onderzoeksrealisaties.

De eerste realisatie betreft het ontwerp van een ruimtelijke annotatie die een overbrugging vormt tussen de natuurlijke taal en een formele ruimtelijke voorstelling. In dit ontwerp worden universele en algemeen aanvaarde cognitieve ruimtelijke begrippen en meerdere bekende kwalitatieve modellen van ruimtelijk redeneren toegepast.

De tweede realisatie is de definitie van een nieuwe computationeel linguïstische taak die de annotatie gebruikt om natuurlijke taal te koppelen aan een ruimtelijke ontologie. Voor deze opdracht heb ik rijk geannoteerde corpora publiek beschikbaar gesteld en een evaluatieprocedure opgesteld.

De derde realisatie is een gedetailleerd onderzoek van de taalkundige en

structurele kenmerken van ruimtelijke taal die het gebruik van machinaal leren voor het extraheren van ruimtelijke rollen en relaties uit de geannoteerde data mogelijk maakt. De leermethoden gebruiken methoden van discriminatieve grafische modellen en statistisch relationeel leren.

De vierde realisatie is het voorstel voor een geïntegreerd kader voor het machinaal leren van de gestructureerde output voorgesteld door een ontologie. De modellen voor het toekennen van de ontologiecomponenten worden geleerd, rekening houdend met de ontologische beperkingen en taalkundige afhankelijkheden tussen de componenten. De ontologie bevat rollen en relaties, en meerdere formele semantische types.

Tenslotte de vijfde realisatie betreft een efficiënte aanpak op basis van de optimalisering van randvoorwaarden. Het voorgestelde model kan omgaan met een groot aantal variabelen en beperkingen, en maakt het opbouwen van een globaal gestructureerd leermodel voor ontologiepopulatie haalbaar. Om de aanpak te testen heb ik een empirisch onderzoek met behulp van mijn ruimtelijke ontologie uitgevoerd.

De toepassing van het voorgestelde geïntegreerd leermodel voor ontologiepopulatie is niet beperkt tot de extractie van ruimtelijke semantiek uit tekst, het kan ook worden gebruikt voor de populatie van elke andere ontologie. Ik pleit dan ook dat dit werk een belangrijke stap is in de richting van het automatisch beschrijven van tekst met semantische descriptoren die een gestructureerde ontologische representatie van de inhoud vormen.

List of Symbols

Symbol	Description
A	Subset of random variables
$\mathcal{C}$	Set of templates
C	Set of candidates for a template
d	Unknown distribution function
D	Parameter dimension
E	Set of training examples
$\hat{Er}(h)$	Expected error of hypothesis h
f	Joint feature function
h	Mapping from the input x to the output y/hypothesis
K	Dimension of the feature vector
$K(p)$	Dimension of the feature vector of a template $\mathcal{C}_p$, or dimension of a feature vector related to a subset of variables
l	Convex upper bound on the expected loss
$\mathbf{l}$	Set of labels
ℓ	Likelihood function
N	Number of examples in the training data
p	Distribution function
P	Number of templates
S	Natural language sentence
S	Set of decompositions in DecL
$\mathcal{S}$	Working set constraints for cutting plane
T	Length of a sequence in linear-chain model, The number of input components in general
$\mathcal{T}$	The number of iterations when relevant for algorithms
w	Instantiation of the weight vector W
w^*	Optimal parameter values (i.e. weights)
w_A	Assignment to a subset of the weight vector
W	Weight vector i.e. model parameters in Max-Margin models
$\mathcal{X}$	Input domain of the prediction function

$x \in \mathcal{X}$	Element of input domain
X	Random variable that takes values in input domain
$\mathcal{Y}$	Structured output domain of the prediction function
$y \in \mathcal{Y}$	Element of output domain
Y	Random variable that takes values in output domain
x_A, y_A	An assignment to a subset of random variables
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Graph, $\mathcal{V} = \{nodes\}, \mathcal{E} = \{edges\}$
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$	Factor graph, $\mathcal{V} = \{variable\ nodes\}, \mathcal{F} = \{factor\ nodes\},\ \mathcal{E} = \{edges\}$
Z	Normalization factor
Δ	Loss function
ζ	Set of constraints in CCMs
θ	Model parameters in probability distributions
ρ	Parameter set for constraints in CCMs
ϕ	Input feature function
Ψ	Potential function/compatibility function/local function
$<, >$	Dot product of two vectors
$\triangleq$	Equal to by definition
$\odot$	Element-wise product of two vectors
$\|.\|$	1-norm of a vector

	Specific frequently used symbols
$sp, sp(x)$	indicator function of the set of spatial indicators, used also for the set itself
$lm, lm(x)$	indicator function of the set of landmarks, used also for the set itself
$tr, tr(x)$	indicator function of the set of the trajectors, used also for the set itself
sp	the label of spatial indicator
tr	the label of trajector
lm	the label of landmark

When relevant, the label values shown with bold symbols.
A same notation is used for the sets and their indicator functions.

Abbreviations

CCM	Constrained conditional model
CRF	Conditional random field
DecL	Decomposed learning
GUM	General upper model
HMM	Hidden Markov Model
IBT	Inference based training
IE	Information extraction
ILP	Integer linear programming
LAL	Link and label model
LO	Learning only
L+I	Learning plus inference
MAP	Maximum a posteriori
POS	Part-of-speech tagging
QSR	Qualitative spatial representation and reasoning
SpQL	Spatial qualitative labeling
SpRL	Spatial role labeling
SRL	Semantic role labeling
SVM	Support vector machine
RCC	Regional connection calculus
TPP	The preposition project

Contents

List of Figures

List of Tables

Prologue

Chapter 1

Introduction

1.1 Spatial Information Extraction from Natural Language

To explain the main goal of this thesis and the challenges we begin with an example:

Example. *In the afternoon bring me the book on AI which is on your table.*

The first question is, *What does it mean to extract spatial information from such a sentence?*
We use the research overview about spatial semantics made in [168] to clarify the possible answers to this important question. The spatial semantics in the language can be characterized in at least in three ways. Firstly according to a *form class*, that is, for example, to extract the *prepositional phrase* or what is called *locative expression* in the above sentence. This is a useful language clue, if the sentence is assumed to contain only spatial information, for example, when giving a robot only navigational instructions in which the phrases have only a spatial interpretation. But in a free context the prepositional phrase can express temporal information like in the above example by the phrase "In the afternoon". The second approach defines the spatial semantics *notionally* according to predefined cognitive spatial primitives and notions. In the above example, we may consider "book" as a *trajector*, i.e. an object whose location is described, and "table" as a *landmark*, i.e. an object that is used as a reference to describe the location of the trajector. The preposition "on" restricts the *region*

in which the book is placed with respect to the table. The third approach defines spatial semantics through its *communicative function*, that is the spatial information is the part of the sentence which answers a *where* question, such as "where is the book? on your table".

And here the second question raises, *What kind of spatial primitives, notions and formal semantic representation should be used, if we aim to deal with an unrestricted language?*

If we aim to represent the spatial information in any type of language in a formal machine understandable form then we need to represent the functional spatial information notionally. In other words, we need to recognize the parts of the sentence that express some spatial information first, such as: "book", "on", "table"; then represent them using cognitive primitives that clarifies their functionalities, for example "book" plays the role of a *trajector*, "table" plays the role of a *landmark*, etc. We may represent this level of abstraction as e.g. *on(book, table).* Then we may inspect a formal spatial meaning representation, for example, *Externally-connected(book,table)*, to connect the extracted notions to some formal machine understandable semantics between regions, points or any other spatial primitives. The advantage of such a representation will be that the extracted information can be manipulated by machines, for example, for automatic spatial reasoning, which is not possible over natural language directly.

And the last question, in the most interest of this thesis, is *How can we computationally deal with this problem which can be entitled as "automatic mapping of natural language to a formal spatial representation appropriate for automatic spatial reasoning"?*

1.2 Motivation

On a high level, one of the ambitious goals of artificial intelligence is language understanding. One of the essential functions of natural language is to talk about objects and their relative or absolute location and translocation in a particular space according to a frame of reference. Understanding the spatial language or more technically spatial information extraction from natural language is important for many applications including geographical information systems, human computer interaction, providing navigational instructions to the robots and visualization or text-to-scene conversion.

For example, if the sentence mentioned in Section 1.1, is an instruction given to a robot, then the robot should be able to recognize that neither "AI" nor "afternoon" are landmarks. Moreover it should extract the main spatial elements of that sentence and connect their relation to a predefined meaning representation model, for example, indicating an externally connected region relationship, to be able to recognize and grab the "book" which is "on" the "table" when there are several books placed in a variety of locations with respect to that table.

1.3 Context

This thesis is placed at the intersection of three main research fields, *natural language processing*, *spatial knowledge representation (the ontology of the space)* and *machine learning*. As a matter of fact, the research presented in this thesis is multidisciplinary, and it is the first effort to make a systematic connection between the above three fields in a theoretically sound and computationally tractable setting.

Due to the complexity and variety of spatial primitives and notions, and the difficulty in reducing them to a small number of concepts, formalizing and reasoning over spatial information has lagged behind compared to dealing with similar notions such as temporal information [160, 45]. Therefore the succeeding steps such as mapping natural language to a formal spatial semantic representation, as defined in a spatial ontology commonly accepted by cognitive semantics, has not been acted upon actively. Hence, machine learning models which are the most successful and dominant methods for a variety of computational linguistic tasks, have not yet been systematically applied to the extraction of spatial semantics. Subsequently, no established corpora are available for computational purposes. All these issues are sufficient reasons for the current situation in which the extraction of spatial semantics has no well-defined framework yet compared to other computational linguistic tasks.

In this thesis we study this problem from *cognitive*, *linguistics* and *computational* points of view with a primary focus on establishing a *supervised machine learning framework* for it.

The spatial primitives and notions in this thesis are built upon related cognitive linguistic studies [168, 167, 53, 146]. The formal spatial representations that we exploit are based on spatial knowledge representation and reasoning models [45, 160, 24, 118, 65, 163].We consider machine learning solutions [97] for the problem of mapping to its spatial semantics. The machine learning approaches that we focus on are classified in the three main possibly overlapping categories of probabilistic graphical models [78, 142], statistical relational models [37, 42, 32] and structured output prediction models [155, 103, 25, 150, 111]. We pay particular attention to the structured output prediction models in the presence of global constraints [20, 130, 91] and relational features [42, 149, 44, 14] in relational data domains [142].

The connection between natural language and qualitative spatial reasoning models has been studied from a cognitive and linguistic point of view [54, 55, 56]. Large linguistically motivated spatial ontologies are used and these connections have been formalized with a fine-grained ontology engineering [8]. But machine learning techniques to learn the mapping between natural language and the spatial ontologies have not been systematically applied. However, due to the ambiguity and uncertainty in the language, machine learning models are best situated to solve this problem.

The machine learning models have been applied to extract very domain dependent spatial concepts such as extracting toponyms and *toponym resolution* [56, 132] which is in the interest of geographical information systems, and considering locations just as *named entities* [100] which is in the interest of text analyzers, or in a more technical linguistic sense, recognizing the locatives in the frame of particular verbs as *adjuncts* in *semantic role labeling* [90, 61]. In visual contexts, a few research works consider more generic spatial objects such as *trajectors* and *landmarks* in a multimodal setting but considering mostly visual features rather than linguistic ones [66, 151].

Using *machine learning models* for establishing a connection between language and adequate spatial ontologies is very challenging. The state-of-the-art challenges in the *structured output prediction*, *relational learning* in the presence of *correlated output* variables, and contemplating *relational features* and *global constraints* are the main concerns in this task compared to the classical applications of machine learning [37].

1.4 Challenges and Research Questions

In this section we point to the emerging challenges and research questions that we enquire in the relevant areas: spatial ontologies, computational linguistics and machine learning.

- **Chal 1.** The choice of the spatial primitives and notions to be extracted from unrestricted natural language, independent from domain for a computational model is challenging. For example, the spatial primitives for describing the objects in a room, entities in a children story or geographical places can be largely different. Moreover, a detailed account of each domain leads to the complexity and the infeasibility of the computational models.

Q 1. What kind of spatial primitives are adequate to consider for starting a realistic and feasible machine learning effort?

- **Chal 2.** There are various spatial knowledge representation and reasoning models each of which focuses on one specific spatial aspect in detail. On the one hand, mapping language to well-defined spatial representation and reasoning models, grants the possibility of performing spatial reasoning on the extracted information, on the other hand there is a gap between the expressivity of natural language and spatial representation models. The "book on table" can be assumed as a relation about two *externally connected* regions, but a similar expression applies to a book which is on a small shelf on the table and so the book is disconnected from the table. In the latter case a formal representation such as *above* can be more helpful and *externally connected* is an over-specification of the relation.

Q 2. Do we represent the semantics using spatial reasoning formalisms? Which formalisms should we use and how can we deal with the existing gap?

- **Chal 3.** Language in general is ambiguous and polysemous. To clarify, in the above example, what can help the machine to recognize that "book" is a trajector of "table" not of "AI", and that the preposition "on" in the phrase "on AI" does not have a spatial sense while the first "on" in the sentence has.

Q 3. What kind of computational approaches, which features and structural clues can help disambiguating spatial semantics? Can we do this with learning from annotated data?

- **Chal 4.** There is no data available annotated with generic spatial notions and qualitative formal representations to be used in supervised machine learning models.

Q 4. Which linguistic corpora are relevant and appropriate for annotation? What kind of formal semantics are expressed in the linguistic corpora?

- **Chal 5.** After designing our spatial ontology, we encounter the computational challenges for the extraction of spatial entities and their relationships. Each sentence is a structured input[1] and can contain an arbitrary number of spatial entities and relationships each of which can have different and multiple semantic types. In addition, the type assignments should obey ontological relationships and constraints. In the

[1] Often sentences are assumed to have a sequential or a tree structure depending on the computational task at hand.

above example the spatial relation is composed of three components and it can have topological semantics (externally connected regions) as well as directional semantics (above). This means, there are some constraints that we expect according to the common sense background knowledge or linguistic structure of the language, and these can help the machine making accurate predictions. For example, we know that when an object is "on" something, then it can not be "about" or "in" it.

Q 5. What kind of machine learning models are appropriate for such a structured input and output task? How we can contemplate the relational features and global structure of the spatial language and background knowledge in a learning framework?

- **Chal 6.** Given the designed spatial ontology, given the structured output prediction models which by definition have the capacity to consider the output correlations and relational features, we encounter the difficulty of a large number of variables and a large number of constraints over them which makes inference during training and inference during prediction highly inefficient. For example, in the above sentence there are 14 words that can make basically 14^3 ternary relations (without pruning), and these candidates can have multiple semantics depending on the ontology of the spatial semantics that are used.

Q 6. How we can make efficient inference to exploit these particular correlations, say, ontological constraints, language structure and properties during joint prediction and joint training when we have a large number of possible roles and relations each of which can be of multiple types?

1.5 Contributions of the Thesis

Given the above mentioned challenges in the context of our research, we end up with the following contributions:

- The first contribution of this thesis is proposing a **spatial annotation scheme**. This scheme covers universal and commonly accepted spatial primitives and notions and applies the well-known formal semantics represented in the qualitative spatial representation and reasoning models that grant automatic spatial reasoning. This scheme covers static as well as dynamic spatial notions. The semantic gap between the natural language and formal spatial representation models is alleviated by labeling the spatial relations with multiple calculi models to be able to cover all semantic aspects represented in natural language.

- The second contribution is defining a **novel computational linguistic task** according to the proposed scheme for mapping natural language to a spatial ontology consisting of two layers, i.e. *spatial role labeling* (SpRL) and *spatial qualitative labeling* (SpQL), for which we establish annotated *corpora* and an evaluation setting for the first time. The former layer considers the extraction of the spatial roles and relations, while the second layer considers the assignment of multiple semantic types to the spatial relations. The introduced task has now become a **benchmark** for a computational linguistic challenge.

- The third contribution is an extensive investigation of the **linguistic features** and **structural characteristics** of spatial language. In this study *discriminative graphical models* and *statistical relational learning* frameworks are used to extract the spatial roles and relations in the SpRL layer. This investigation, by designing a variety of models, classifying the error types and, performing a cross domain evaluation indicates the importance of considering contextual and relational features, long distance dependencies and background knowledge about the spatial language.

- The forth and major contribution of this thesis, which lies in the machine learning field, is proposing a **unified structured output learning framework for ontology population**. In this framework we learn a model to map natural language to the full spatial ontology encompassing both SpRL and SpQL layers. We learn the extraction of spatial roles, relations and the multiple semantic types of the relations jointly, while contemplating the ontological constraints and dependencies among these output components.

- The fifth contribution, is proposing **efficient inference** approaches based on constraint optimization techniques that deal with a large number of variables and constraints in the global structured learning model for ontology population. An empirical evaluation of the spatial ontology task assesses the influence of the relational features and global constraints when they are used during training and prediction.

1.6 Outline of the Thesis

This thesis is organized in three main parts. **Part I** discusses the spatial annotation scheme, data and the spatial ontology population task, **Part II** considers the spatial role labeling layer of the ontology, and **Part III** investigates the structured output prediction models for mapping natural language to the full spatial ontology. In the following we give a brief outline of each part.

In this part, following this introductory chapter, in **Chapter 2** the foundations of various aspects of this research including structured machine learning, natural language processing and ontological concepts, pointing to the ontology of the space, are provided.

In **Part I**, the spatial annotation scheme, the data and the spatial ontology population task are defined. In **Chapter 3**, the spatial annotation scheme is illustrated by examples and the corpora which are annotated according to the spatial scheme as well as other relevant corpora are introduced. In **Chapter 4**, the deliberated spatial ontology consisting of two semantic layers of spatial role labeling (SpRL) and spatial qualitative labeling (SpQL) is introduced. The spatial ontology population is defined as the targeted machine learning task. The features and the global constraints that are considered in various machine learning models through this thesis are clarified. The evaluation metrics for measuring the performance of the learning models are defined.

In **Part II**, the spatial role labeling layer of the spatial ontology is investigated. **Chapter 5**, provides the first machine learning model for the spatial role labeling layer. The importance of this layer as an independent computational linguistic task compared to semantic role labeling is argued. The extraction of the spatial roles and the relations via linear chain conditional random fields as well as skip-chain models are described in a multi-sequence tagging model. The preposition disambiguation as an assisting task for spatial role labeling is applied and the The Preposition Project (TPP) dataset is exploited as an additional resource to aid spatial role labeling. An extensive error analysis and cross domain evaluation is provided. In **Chapter 6**, given the relational nature of the spatial role labeling task, the statistical relational learning models, particularly kLog [42], are considered. A relational model of SpRL is represented using an entity relationship diagram. Various models such as spatial predicate classification, pipelining the learning of the spatial role and spatial relation predicates and, sequence tagging are programmed declaratively using kLog's logical language. The underlying graph kernel is used to exploit the sentence level contextual features. Experimental results and analysis are provided.

In **Part III**, a unified structured learning model for ontology population is proposed, and efficient inference during training and prediction on the basis of constraint optimization techniques, is investigated. The proposed model is instantiated and empirically studied for the case of spatial ontology population. In **Chapter 7**, the relational learning problem of ontology population is formalized in the framework of structured output prediction. The inference during prediction and the loss-augmented inference during training are both formulated as the objectives of constrained optimization problems. The notion of templates is used to formalize the relational structure of the objective function in a basic model called Link-And-Label model. The objective function is augmented with a component-based loss function during training. The proposed communicative inference is described to decompose the large space of output variables and efficiently solve the formulated objectives for each example. Other relevant decomposed training approaches are discussed and compared. In **Chapter 8**, the spatial ontology population is formulated in the proposed Link-And-Label framework. The templates of the model are defined based on the predefined spatial ontology and are unrolled to build a multinomial objective function for each example. The first order constraints are grounded to a linear form. An extensive empirical investigation is performed over various models from very local models to a global model including the joint training and prediction of both SpRL and SpQL layers using the proposed communicative inference approach and applying linear programming relaxation techniques.

The last part of this thesis contains **Chapter 9**, which draws the conclusions of this thesis and points to future work.

Chapter 2

Foundations

This thesis is a multidisciplinary work at the crossing point of natural language processing, spatial information representation and ontologies, and structured machine learning. In this chapter we provide the foundations to these relevant areas and clarify the context of this work to situate the contribution of the thesis. In Section 2.1 a background to structured machine learning models as well as relational learning and inference techniques, is provided. In Section 2, a background to natural language processing and the tasks which are used through this thesis is provided. In Section 4, the basic notions about ontology as a medium for meaning representation for natural language and the ontology of the space are introduced.

2.1 Structured Machine Learning

Structured learning mostly refers to learning problems with a strong interdependence among output variables. In this context the term *structured output prediction* is often used to highlight this interdependence. This stands in contrast to the simpler approaches of classification, where input data are mapped to, for instance, single labels. Many real world applications of machine learning in domains such as natural language processing, computer vision, robotics and computational biology involve learning from structured input data to predict structured outputs. Examples are tasks such as parsing of natural language sentences [27], classifying web documents considering the relationships between webpages via hyperlinks [149], or named entity recognition when the similarity between distant words is considered when labeling them [142]. In these problems output variables are interdependent and the predicted assignments should obey certain structural characteristics. For instance, in parsing the predicted output should have a tree structure. Considering the interdependencies and structural constraints over the output space easily leads to intractable training and prediction situations. There is a body of research for designing algorithms that deal with structures such as trees, sequences, sets or any arbitrary structural dependencies [20, 27, 155, 142, 103, 150]. In some structured learning models, the expected structural characteristics of the output variables are imposed on the prediction of the learning models as a number of hard/soft constraints by applying constraint optimization techniques [20, 126]. Moreover, increasing attention goes to relational learning [38, 44, 148, 37] in which the structural dependencies hold between groups of variables [142] and therefore the data is (can be) represented in a relational form. For example in named entity recognition, the words of a sentence are treated as entities that have relationships with each other. The entities are labeled as *person*, *location*, *organization*, etc. [143, 126]. The entities are extracted and connected to each other and placed in a relational database. Exploiting relational dependencies between groups of objects helps parameter tying for achieving more efficient structured learning models in such problems.

We base our work on the most recent research achievements and successful learning models for the extraction of spatial information from text. Our models are related to two main types of structured learning techniques: a1) Graphical models, particularly conditional random fields (CRFs) which are one of the most successful graphical models applied in several relevant tasks in natural language processing and other domains; a2) The generalized linear models for structured output learning namely structured support vector machines (SVMs) and structured perceptrons. In addition to these, we consider two main learning frameworks that can be aligned with arbitrary learning techniques: b1) Constraint conditional models (CCMs) which provide an explicit formalization for using declarative constraints and apply constraint optimization techniques as a way of exploiting background knowledge for decision making in structured output prediction; b2) The statistical relational learning framework, which aims at providing a relational and/or logical representation of the domain to formalize the learning from relational data, exploiting background knowledge, and even logical reasoning during statistical learning.
In the following subsections we provide the background that is relevant for this thesis on the representation, training and inference algorithms in the aforementioned learning techniques and frameworks.

2.1.1 General Framework

The supervised learning models in this thesis have the following general setting [27],

- We have an input domain $\mathcal{X}$ and an output domain $\mathcal{Y}$.
- There is a fixed but unknown underlying probability distribution $d(x, y)$ over $\mathcal{X} \times \mathcal{Y}$, and N training examples $E = \{(x^i, y^i) \in \mathcal{X} \times \mathcal{Y} : i = 1 \dots N\}$ are drawn independently and identically from a distribution $d(x, y)$.
- Given the set of examples, the task is to find a function $h : \mathcal{X} \to \mathcal{Y}$ that maps inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$. Function h is called a hypothesis and $h(x)$ will approximate y on new samples from the distribution $d(x, y)$.

Given the general setting, the main characteristics of a learning algorithm are the additional assumptions that are made about the hypothesis class $\mathcal{H}$ of functions h and the criterion for selection of h from $\mathcal{H}$. The structured learning problem can be viewed and categorized in various ways, we use the following categorization made in [103].

- **Probabilistic parameter learning.** In this thesis, from this category we consider a *discriminative* formulation rather than a *generative* one [142]. Hence, the conditional distribution of the output is considered rather than its joint distribution. Let $d(y|x)$ be the (unknown) conditional distribution of the output variables for a problem to be solved. For a parameterized conditional distribution $p(y|x; W)$ with parameters $W \in \mathbb{R}^D$, probabilistic parameter learning is the task of finding a point estimate of the parameter w^* that makes $p(y|x; W)$ closest to $d(y|x)$ for every $x \in \mathcal{X}$. In these models the distribution function p is assumed to be from an exponential family and the parameters are estimated based on the penalized maximum likelihood of the training examples where in the conditional distribution case, the following log likelihood is used as the basis:

  $$\ell(W) = \sum_{i=1}^{N} log\ p(y^i|x^i; W) \tag{2.1}$$

  and the function h has the following form given the optimal parameter w^*,

  $$h(x; w^*) = \arg\max_{y \in \mathcal{Y}} p(y|x; w^*). \tag{2.2}$$

- **Loss-minimizing parameter learning.** In these models, we define a loss function $\Delta(y, \hat{y})$ which measures the cost of proposing an output $\hat{y}$ when the true output is y. In general, for any arbitrary loss, $\Delta(y, \hat{y}) = 0$ if $y = \hat{y}$. The standard loss function for classification is 0/1 loss, $\Delta(y, \hat{y}) = \mathbb{1}(y \neq h(x))$, where $\mathbb{1}(.)$ denotes the indicator function, that is $\mathbb{1}(true) = 1$ and $\mathbb{1}(false) = 0$. Then loss minimizing parameter learning is the task of finding a parameter value w^* such that the expected prediction risk (often regularized)

$$Er(h) = \sum_{x,y} d(x,y)\Delta(y, h(x; W)), \tag{2.3}$$

is minimized. Since the distribution d is usually unknown, the expected loss of the hypothesis can not be calculated explicitly therefore the empirical loss of the function h on the training set E is minimized instead,

$$\hat{Er}(h) = \frac{1}{N} \sum_{i=1}^{N} \Delta(y_i, h(x_i; W)), \tag{2.4}$$

and assuming a generalized linear class for h, it will have the following form (^{T}, indicates the matrix transpose),

$$h(x; w^*) = \arg\max_y {w^*}^T \cdot f(x,y), \tag{2.5}$$

where w^* is an optimal parameter vector and f is a feature map over x and y.

We will apply approaches to solve learning problems in both probabilistic parameter learning and loss-minimization parameter learning. However, each approach can be described with different possible views. Particularly, if the *Bayes optimal* concept is considered, it is provable that maximizing the likelihood and finding the *Bayes optimal* solution, which outputs the most likely y under the distribution d for each input x, can not outperform minimizing the expected loss $\hat{Er}(h)$ over the space of all possible functions [27]. However, the distinction between the two approaches mentioned above can be based on the type of the applied basis functions (e.g. linear vs. exponential/log linear). In the following section, the particular algorithms that we use in these classes of models, are described.

2.1.2 Conditional Random Fields

A conditional random field (CRF) is an *undirected graphical model* or *Markov random field*, conditioned on a set of observations X to predict a set of output variables Y [144]. As in other graphical models, the probability distributions over the set of random variables are factorized according to an underlying graph. The conditional distribution over the large number of variables is represented by a product of local functions that each depend on only a small number of variables. This *factorization* of the global probability distribution makes learning and inference feasible. We define $\mathcal{V} = X \cup Y$ and consider probability distributions over these sets of random variables. We denote an assignment to X by x, an assignment to a set $A \subset X$ by x_A, and similarly for Y. A CRF generally defines a probability distribution $p(y|x)$ as follows:

$$p(y|x) = \frac{1}{Z(x)} \prod\nolimits_A \Psi_A(x_A, y_A), \tag{2.6}$$

for any choice of factors $\mathcal{F} = \{\Psi_A\}$, where $\Psi_A : \mathcal{V}^n \rightarrow \mathbb{R}^+$. $\Psi_A(x_A, y_A)$s are called *potential, local or compatibility* functions, and $Z(x)$ is the normalization factor, n is the number of variables involved in the factor, and:

$$Z(x) = \sum_y \prod\nolimits_A \Psi_A(x_A, y_A) \quad \text{and} \quad \Psi_A(x_A, y_A) = \exp\left\{ \sum_{k=1}^{K(A)} w_{Ak} f_{Ak}(x_A, y_A) \right\}, \tag{2.7}$$

where each $\{f_{Ak}(x_A, y_A)\}_{k=1}^{K(A)}$ is a set of real-valued feature functions and $W_A = \{w_{Ak}\} \in \mathbb{R}^{K(A)}$ is a parameter vector associated to each factor. Obviously the family of the distributions over the random variables is assumed to be exponential. Graphically, the factorization in CRFs is represented by a factor graph [76]. A factor graph is a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{F}, \mathcal{E})$ containing two sets of nodes $\mathcal{V}$ and $\mathcal{F}$ which are variable nodes and factor nodes, respectively. $\mathcal{E}$ is the set of edges in the factor graph. A variable node $v_s \in \mathcal{V}$ is connected to a factor node $\Psi_A \in \mathcal{F}$ if v_s is an argument of Ψ_A. Figure 2.1 shows an example of a factor graph. The black boxes show the factor nodes and the circles show the variable nodes. The gray variable nodes are the inputs and the white ones are outputs. The correlated variables are connected to each other via the factor nodes. Hence, in this figure the sequential relationships between the output variables as well as the correlation between each output node with current observation x are modeled. Finally the conditional probability is the following:

$$p(y|x) = \frac{1}{Z(x)} \prod_{\Psi_A \in \mathcal{G}} \exp\left\{ \sum_{k=1}^{K(A)} w_{Ak} f_{Ak}(x_A, y_A) \right\}. \tag{2.8}$$

Theoretically, the structure of graph $\mathcal{G}$ is arbitrary, however the most commonly used CRF model used in natural language processing has been the linear-chain

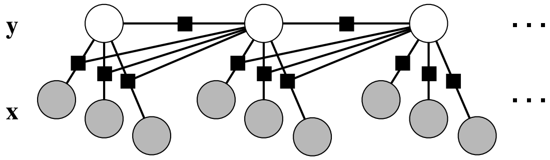


Figure 2.1: Graphical representation of a linear-chain CRF in which the transition score depends on the current observation [142].

CRF, which we describe here.

Linear-chain CRF. When modeling sequential relationships, for example between words in a natural language sentence, the CRF graph $\mathcal{G}$ will be a linear-chain in the form of a (often first-order) Markov chain [78, 142]. In a setting where each word in a sentence is tagged by a label, the dependency between the label of each word and the label of its previous word in the sentence can be considered (see figure 2.1). Considering sequential relationships can increase the learning model's performance. In the linear-chain CRF, the conditional probability $p(y|x)$ is computed as

$$p(y|x) = \frac{1}{Z(x)} \prod\nolimits_{t=1}^{T} \Psi_t(y_{t-1}, y_t, x_t), \tag{2.9}$$

where $X = (x_1, \ldots, x_T)$ is a sequence of observations and $Y = (y_1, \ldots, y_T)$ is the corresponding set of labels assigned to X. The potential function $\Psi_t(y_{t-1}, y_t, x_t)$ captures the degree to which the assignment y_t to the output variable fits the transition from y_{t-1} and x_t. The potentials typically factorize according to a set of features $f = \{f_k(.)\}$ such that $\Psi_t(y_{t-1}, y_t, x_t) = \exp\{\sum_{k=1}^{K} w_k f_k(y_{t-1}, y_t, x_t)\}$, where $W = \{w_k\}$ is the parameter set of the linear chain model which has only one factor. A forward-backward algorithm can be used to compute the marginal distributions and the Viterbi algorithm to compute the most probable sequence label assignment.

Templates in general CRFs. In relational domains, practical models rely extensively on parameter tying. For example, in the linear-chain model, the same weights are used for the factors $\Psi_t(y_t, y_{t-1}, x_t)$ at each position in the sequence. To denote this, the factors of $\mathcal{G}$ are partitioned into a set of templates $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_P\}$, where each $\mathcal{C}_p$ is a clique template whose parameters are tied. This notion of clique template is also used in [148, 145, 38]. Each clique template $\mathcal{C}_p$ is a set of factors which has a corresponding set of sufficient statistics $\{f_{pk}(x_p, y_p)\}$ and parameters $\theta_p \in \mathbb{R}^{K(p)}$. Then the CRF can be written as

$$p(y|x) = \frac{1}{Z(x)} \prod_{\mathcal{C}_p \in \mathcal{C}} \prod_{\Psi_c \in \mathcal{C}_p} \Psi_c(x_c, y_c; \theta_p), \tag{2.10}$$

where each factor is parametrized as

$$\Psi_c(x_c, y_c; \theta_p) = exp\Bigg\{ \sum_{k=1}^{K(p)} w_{pk} f_{pk}(x_c, y_c) \Bigg\}, \tag{2.11}$$

and the normalization function is

$$Z(x) = \sum_y \prod_{\mathcal{C}_p \in \mathcal{C}} \prod_{\Psi_c \in \mathcal{C}_p} \Psi_c(x_c, y_c; \theta_p). \tag{2.12}$$

For example, in a linear-chain CRF, one clique template $\mathcal{C} = \{\Psi_t(y_t, y_{t-1}, x_t)\}_{t=1}^{K}$ is used in the model. However, in many real world applications such as relation extraction tasks, certain long-distance dependencies between entities play an important role. A CRF model called *skip-chain* CRF [142] accounts for the probabilistic dependencies between distant labels. These dependencies are represented by augmenting the linear-chain CRF with factors dependent on the labels of the words in arbitrary positions in the sentence. The features on skip edges can incorporate information from the context of both endpoints, so the strong evidence of one endpoint can influence the label at the other endpoint. The skip-chain CRF model, includes two clique templates one is the sequential one for connecting neighboring positions in the sequence and the other connects arbitrary positions according to some pre-defined conditions. If we assume $\mathcal{I} = \{(u, v)\}$ is the set of all pairs of positions for which there are skip edges then the probability of a label sequence y given input x is

$$p_\theta(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, x_t) \prod_{(u,v) \in \mathcal{I}} \Psi_{uv}(y_u, y_v, x_u, x_v), \tag{2.13}$$

where Ψ_t are factors for sequential relations and Ψ_{uv} are factors over skip edges. These factors are defined as

$$\Psi_t(y_t, y_{t-1}, x_t) = \exp\{\sum_{k=1}^{K_1} w_{1k} f_{1k}(y_t, y_{t-1}, x_t)\} \tag{2.14}$$

and

$$\Psi_{uv}(y_u, y_v, x_u, x_v) = \exp\{\sum_{k=1}^{K_2} w_{2k} f_{2k}(y_u, y_v, x_u, x_v)\}, \tag{2.15}$$

where $\theta_1 = \{w_{1k}\}_{k=1}^{K_1}$ are the parameters of the linear-chain template and $\{f_{1k}\}$ is the related set of feature functions or sufficient statistics. Similarly,

$\theta_2 = \{w_{2k}\}_{k=1}^{K2}$ are the parameters of the skip-chain templates, and $\{f_{2k}\}$ is its related set of feature functions or sufficient statistics. The full set of model parameters is $\theta = \{\theta_1, \theta_2\}$. In the general CRFs various approximate inference approaches in two main categories of *variational methods* such as *loopy belief propagation* and *Markov Chain Monte Carlo (MCMC)* methods such as *Gibbs sampling* are used [144].

2.1.3 Structured Support Vector Machines

As pointed out before, this approach is based on finding the hypothesis h that minimizes the loss over the training data i.e. expected risk,

$$h^* = \arg\min_{h \in \mathcal{H}} \hat{Er}(h). \tag{2.16}$$

If our set of hypotheses, $\mathcal{H}$, is large enough, we will be able to find h that has zero or very small empirical risk. However, simply selecting a hypothesis with lowest risk is generally not a good idea and leads to over fitting. To alleviate this problem the following regularized risk is minimized instead of $\hat{Er}(h)$,

$$\mathcal{R}(h) + \frac{Cost}{N} \sum_{i=1}^{N} \Delta(y^i, h(x^i)), \tag{2.17}$$

the first term $\mathcal{R}(h)$ of which is the regularizer, prevents the learning models from over-fitting on the training data by penalizing functions according to their complexity (e.g. the degree of the polynomial) [156], *Cost* is a constant penalizing the training error.

In structured learning, given ys with arbitrary complex structures for each x, a function $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is learnt to assign a score to each input-output pair instead of a direct mapping from $\mathcal{X}$ to $\mathcal{Y}$. In this way, the prediction is performed by maximizing g with parameter vector W over y for a given input x [155],

$$h(x; W) = \arg\max_{y \in \mathcal{Y}} g(x, y; W), \tag{2.18}$$

where

$$g(x, y; W) = \langle W, f(x, y) \rangle, \tag{2.19}$$

g is assumed to be a linear *discriminant* function over the joint features of the input and output $f(x, y)$; W denotes a weight vector and $\langle , \rangle$ denotes a dot product between two vectors. The basic idea for learning the function g parametrized by W is that it should roughly fulfill the following constraint for each training data point,

$$g(x^i, y^i; W) \geq g(x^i, y; W), \qquad \forall y \in \mathcal{Y}. \tag{2.20}$$

These constraints mean the function g should assign a larger score to the ground-truth output y^i than all other possible wrong ys for each input x^i. A more sophisticated variation is to impose the difference of the scores to be larger than a margin and having the loss between a wrong y and the ground truth y^i as a lower bound,

$$g(x^i, y^i; W) \geq g(x^i, y; W) + \Delta(y^i, y), \quad \forall y \in \mathcal{Y}. \tag{2.21}$$

These inequalities are referred to as *learning constraints* (not to be confused with the constraints over the output structure discussed in Section 2.1.5). In the structured output case, a large number of possible ys per x^i leads to a large number of learning constraints when searching for the optimum W based on the above idea. A solution to this problem is to consider only the most violated y for each x. In other words, at each training iteration the following inference is solved per training example,

$$\arg\max_{y \in \mathcal{Y}} (g(x^i, y; W) - g(x^i, y^i; W) + \Delta(y^i, y)). \tag{2.22}$$

Hence discriminative structured prediction algorithms such as structured perceptron [25, 27], max-margin Markov networks [150] and structured SVMs [155] need a solution for this inference task during training and then the learning is to minimize $l(W)$ which is a convex upper bound on the loss $\hat{Er}(h)$ over the training data [130]:

$$l(W) = \sum_{i=1}^{N} \max_{y \in \mathcal{Y}} (g(x^i, y; W) - g(x^i, y^i; W) + \Delta(y^i, y)), \tag{2.23}$$

the inner maximization is referred to as *loss-augmented inference.* This learning approach forms the basis of the structured support vector machines (SSVM), but there are a number of formulations for max-margin optimization in SSVMs as suggested in [155]. We show the following 1-slack margin rescaling formulation in which the margin is rescaled by the loss,

$$\begin{array}{ll} \min_{W,\xi} & \frac{1}{2}\|W\|^2 + \frac{Cost}{N}\sum_{i=1}^{N} \xi_i \\ s.t. & \forall i : \xi_i \geq 0, \\ \forall i, \forall y \in \mathcal{Y} : & \langle W, f(x^i, y^i) - f(x^i, y)\rangle \geq \Delta(y^i, y) - \xi_i, \end{array} \tag{2.24}$$

where ξ_is are the slack variables to allow errors in the training set particularly when the training data is not linearly separable and $Cost > 0$ is a constant that controls the tradeoff between the training error minimization and margin maximization. Referring back to the formula 2.17, in this formulation the $\|W\|^2$ is the regularization term which is the $l2$ norm of the weight vector. And instead of minimizing the expected loss, the convex upper bound in formula 2.23 is minimized (i.e. expected ξ_is) along with the margin maximization.

Algorithm 1 Cutting-plane for SVM-struct

```
 1: Given training data: $E = (x^i, y^i)_{i=1}^N$; $Cost$, $\epsilon$
 2: $\mathcal{S}_i \leftarrow \emptyset \ \ \forall i = 1, \ldots, N$
 3: repeat
 4:     for $i = 1$ to $N$ do
 5:         $H(y) \triangleq \Delta(y^i, y) + W^T f(x^i, y) - W^T f(x^i, y^i)$
 6:         compute $\hat{y} = \arg\max_{y \in \mathcal{Y}} H(y)$
 7:         compute $\xi_i = \max\{0, \max_{y \in S_i} H(y)\}$
 8:         if $H(\hat{y}) > \xi_i + \epsilon$ then
 9:             $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{\hat{y}\}$
10:             $W \leftarrow$ optimize primal over $\cup_i \mathcal{S}_i$
11: until no $\mathcal{S}_i$ has changed during iteration
```

Algorithm 1 shows the cutting plane algorithm of SVM-struct[1].The cutting plane algorithm suggested in [155] takes the most violated ys by finding the maximum a posteriori (MAP) of H in line 6. H is the objective of the loss-augmented inference defined in line 5. The algorithm adds the most violated examples one by one building a working subset of constraints ($\cup_i \mathcal{S}_i$, S is the constraint set associated to the ith training example) at each iteration, and updates the weight vector W (line 10) by working on the primal formulation in this algorithm.

2.1.4 Structured Perceptron-based Model

The variation of the structured perceptrons that has a very similar basis to the SSVM suggested by [27] is shown in Algorithm 2. This algorithm minimizes the same convex upper bound $l(W)$ of the structured loss. In the simplest case there is no regularization term included and the training is performed by a sub-gradient algorithm. Beginning with a weight vector W initialized with zeros, the structured perceptron algorithm iterates through each element of the training set, updating the weight vector after processing each training instance. The training set is processed repeatedly until convergence. In each update step t, if the most violated y is not the correct answer, the difference between the feature vectors of the ground-truth and the model's prediction is added to the weight vector W.

There are more effective variations of this algorithm, such as averaged perceptron and voted perceptron [25, 27]. In the averaged perceptron, the final weight vector W is the average over all model weights W_t at each iteration t, that

[1] http://www.cs.cornell.edu/people/tj/svm_light/svm_struct.html

Algorithm 2 Sub-gradient-descent for structured perceptron

```
Given training data: $E = (x^i, y^i)_{i=1}^N$; step sizes $\eta_t$
$W \leftarrow \mathbf{0}$
for $t = 1$ to $\mathcal{T}$ do
    for $i = 1$ to $N$ do
        $\hat{y} \leftarrow \arg\max_{y \in \mathcal{Y}} \Delta(y^i, y) + W^T f(x^i, y) - W^T f(x^i, y^i)$
        $W \leftarrow W + \eta_t (f(x^i, y^i) - f(x^i, \hat{y}))$
```

is $W = 1/\mathcal{T} \sum_{t=1}^{\mathcal{T}} W_t$, where $\mathcal{T}$ is the number of iterations. This heuristic regularizes the parameters of the model and compensates for not having an explicit regularization term in the training objective. Similarly, the voted perceptron assigns a score to each model of iteration t according to its true predictions on the training and uses the aggregation of the weighted models at prediction time.

2.1.5 Constrained Conditional Models

As mentioned above, one essential property that the structured output prediction models should provide is that the predicted outputs obey certain structural characteristics. The main idea behind constrained conditional models (CCMs) is that the structural characteristics, that are often very complex when modeled as training features, can be considered only at the prediction time to avoid complicating the training process. Hence these structural features are formally treated separately as a number of constraints, and the constraint optimization techniques are used to solve the prediction time inference in CCMs. This idea has been used for about a decade [124] and is well formalized recently in [20]. In this model, in addition to the set of feature functions, $f = \{f_k(.)\}$ in the above described learning models, a set of constraints, $\zeta = \{\zeta_m(.)\}$ is defined over each input and output. A constrained conditional model is formally characterized by two weight vectors, W and ρ, and finds a solution to the following optimization problem:

$$\arg\max_y \sum_{k=1}^{K} w_k f_k(x, y) - \sum_{m=1}^{M} \rho_m \zeta_m(x, y). \tag{2.25}$$

where each constraint $\zeta_m \in \zeta$ is a boolean function that indicates if the joint assignment violates a constraint. ρ is the penalty incurred for violating the constraints, M is the number of constraints and K is the number of features over the given pair of input and output. When constraints should be strictly fulfilled, they are assigned an infinite penalty and are called, *hard constraints*. Hard constraints represent unfeasible assignments to the optimization problem.

Constraints that can be occasionally violated are called *soft constraints*. For these models often integer linear programming (ILP) techniques described in Section 2.1.7 or other techniques such as A^* search are used for decision making at prediction time. A useful property of these models is that the constraints can be formulated in a declarative way and there is a software that provides the facilities of implementing such models [120]. These models are becoming popular in various natural language processing tasks [22, 140, 110].

2.1.6 Relational Learning

In traditional machine learning the data is represented in a flat attribute-value format which is also called a *propositional* format. However, in many real world problems the data has a rich relational structure. In these domains, the data in its most common and practical form is usually described in terms of entities and relationships and it is organized and stored in a relational database. A relational database contains a number of tables that are connected to each other via various *keys*. Each table is called a *relation* and each *tuple* in a table contains the attributes that describe an entity or describe the relationship between more than one entity. For example, a database of an organization such as a university contains information about students and professors which are among the main entities and about the relationships between them such as advisor and advisee relations. Another powerful and flexible framework for representing and storing relational data is to use the knowledge base technology by which rich knowledge (mostly in first order logic form) can be stored, manipulated and reasoned over. However, the expressiveness and flexibility in knowledge base formalisms is provided by trading off the efficiency compared to databases.

Relational learning applies to learning from *relational data* which we briefly define as the data with a structure that is organized in relations, which are for example a set of tuples in tables in a database or instances of a first order logical predicate in a knowledge base.

A useful characteristic of relational representations for the learning models is that functional dependencies are organized among groups of data elements (i.e. tables), and hence a *first order representation* of the data can represent such relational dependencies for parameter tying and efficient learning.

To learn from relational data, researchers with a focus on relational databases use the entity-relationship (E/R) model as a first order representation of the data. The probabilistic relational models (PRM) [44], relational Markov networks (RMN) [148] and relational dependency networks (RDN) [101] are examples of such relational learning frameworks. On the other hand researchers with a focus on knowledge base notions use first order logic formalisms to represent relational data. In this case, sometimes the logical learning and inference are used and combined with probabilities in the framework of probabilistic logical languages to deal with uncertainty inherent in learning and inference. Examples are PRISM [131] and Problog [33]. These approaches are referred to as *logical relational learning* to highlight their ability in combining the ideas from the two subfields of machine learning and knowledge representation [32]. Sometimes in spite of using first order logical representations, the logical inference is replaced by probabilistic inference such as done in Markov logic networks [38]. A recently developed relational learning framework called, kLog [42], is used in Chapter 6, is based on the ideas of Datalog [46] and considers both techniques in relational databases and knowledge base formalisms. It has the capabilities of logical inference and deduction as well as statistical learning from relational data.

Although structured learning, discussed in Section 2.1, and learning from relational data can be assumed as two orthogonal aspects of learning models, these two aspects often are concurrent. We find it useful to distinguish between these two aspects and shortly describe their natural concurrence here.

In a relational data domain, the entities to be labeled are related to each other in complex ways and their labels are not independent. For example, in hypertext classification, the labels of the pages (i.e. entities) that have a link to each other are highly correlated. However, each entity can be classified independently, ignoring the correlations between its label and the label of other entities. These correlations are considered only implicitly when applying the relational features such as the attributes of the other linked pages when labeling a page. But explicitly considering the correlations between the unknown labels of the linked pages and classifying them collectively can yield a better prediction [149]. Hence, most of the relational learning models consider this challenging problem which is referred to as collective classification or, more generally, structured output prediction. For example in PRMs, a relational version of Bayesian networks, is used to define a joint probabilistic model for a collection of related entities and similarly in the case of RMNs or MLNs, a relational version of Markov networks is used for structured output prediction.

In structured learning tasks, often the data has a relational nature in which firstly there are dependencies between entities that we would like to model and secondly each entity has a rich set of features for learning [142] and a relational organization of the data can facilitate parameter tying towards obtaining efficient learning models for structured output prediction. In addition to the above mentioned frameworks based on relational databases and declarative languages based on knowledge representation formalisms, there is a tendency for designing functional imperative languages for flexible modeling of learning from relational data such as FACTORIE [92] (based on CRFs) and Learning Based Java (LBJ) [120] (based on CCMs).

The term *relational learning* in its technical usage often refers to the learning models in which the data and knowledge, the subject to the learning, are represented in a well-defined relational representation, for example in a first order logical or relational database form. In these frameworks to exploit classic machine learning approaches, usually the relational data first should be represented in attribute-value representations and second a propositional inference model should be constructed on which statistical learning and various inference techniques can be applied. The solutions to the former problem, that is generating the necessary input in a feature-vector format for the statistical learner, are called *propositionalization* [75]; and the solutions to the latter, that is dynamically constructing propositional models from relational data to make inference over the actual objects and their relations in the domain [164], are called *knowledge based model construction* (KBMC).

Finally, on a less technical but more conceptual level, we use the term *relational learning* in Chapter 7 to refer to learning models that exploit the relational structure of the data and the background knowledge modeled conceptually independent from any relational formal language. The relational learning term has been used with the same sense also in the previous works where the focus is on structured learning in a relational data domain and for natural language processing [142, 126].

2.1.7 Approximate Inference

To predict the output in structured output prediction models, an inference problem must be solved for finding the best y given a trained model. However, in inference-based-training techniques as well as probabilistic models, the inference over the $\mathcal{Y}$ space is also performed during training. In inference-based models, this is an explicit prediction step using an intermediately trained model. This inference is what we referred to as loss augmented inference in Formula 2.22 and it is repeated over all training examples in each training iteration. Although exploiting the structure is crucial for accurate structured prediction and learning, usually representing complex interactions makes the inference intractable. In practice, we often need to use approximate inference by giving up one of the ideal characteristics of an exact inference approach. Sebastian Nowozin and Christoph H. Lampert provide a thorough categorization of the trade-offs that one needs to consider in solving real world problems [103]. These trade-offs are about giving up in one or more of the following aspects in solving a novel structured prediction problem. These aspects are *generality* (by considering a tractable subclass of the problem at hand with specific structural restrictions), *optimality* (by considering iterative algorithms that are without optimality guarantees), *worst-case complexity* (by considering efficient algorithms that are intractable in the worst case such as branch and bound algorithms), *integrality* (by enlarging the feasible set in a way that inference is simplified) and *determinism* (by using randomness for achieving tractable solutions). Fienley et al. describe approximation methods in two general classes of undergenerating and overgenerating algorithms which are also classes of solutions that trade optimality and integrality as the main idea [40].

Undergenerating. Undergenerating methods approximate the $\arg\max_{y \in Y}$ by $\arg\max_{y \in \underline{Y}}$ where $\underline{Y} \subseteq Y$. For example in the context of undirected graphical models (Markov random fields), the greedy search and loopy belief propagation algorithms are in this class of approximation algorithms.

Overgenerating. On the other hand, overgenerating methods approximate $\arg\max_{y \in Y}$ by $\arg\max_{y \in \bar{Y}}$, where $\bar{Y} \supseteq Y$. For example LP-relaxation and graph-cut algorithms belong to this class of algorithms.

However, overgenerating methods such as LP-relaxations have been shown to work well in the framework of structured output learning so we also use LP-relaxation, i.e. the relaxation of an integer linear programming as the basic approach for inference in some of our models. The advantage of this approach in our problem is that we have a straightforward linear formulation of the hard constraints over the output space and integrating these constraints is straightforward and efficient in the ILP formulation. We describe this technique briefly.

LP-relaxation

In this framework, inference is viewed as an optimization problem and the exact formulation of the problem and the constraints over the $\mathcal{Y}$ space are formulated. In LP-relaxation, the problem is first formulated as an integer linear program with the following form,

$$\begin{aligned} &\arg\max_{y} \quad c^T y \\ &\text{subject to } Ay \leq b \\ &y \text{ is integer,} \end{aligned} \tag{2.26}$$

where y represents the target variables, c^T is the transposed coefficient matrix of the linear objective, A is the matrix of the coefficients of the linear (in)equality constraints and b is the vector of the constants. By removing the constraint of y being integer from the formulation 2.26, an integer linear programming relaxation is obtained. It is proven that if the constraint matrix A is *totally unimodular* [125], the solution to this problem for an integral b is integral.

Definition A matrix A is totally unimodular if the determinant of every square submatrix of A is $+1$, -1, or 0.

Integer linear programming is commonly used in natural language processing (NLP) tasks. Even for non-linear objective functions and complicated finite feasible sets, we can introduce additional auxiliary variables to make the necessary connections by imposing linear constraints and formulate the problem as a linear program. This formulation has been used even in NLP tasks for which the total unimodularity does not hold and often acceptable approximations are provided by this technique [125]. To apply this technique we need to formulate a linear objective function for the inference (prediction/loss-augmented) in terms of the output labels.

2.2 Natural Language Processing

Developing systems that understand natural language has been a long term ambition of artificial intelligence. The ultimate goal of natural language processing (NLP) is to make computers understand statements written in human languages. Current NLP technology considers various tasks for lexical, syntactic and semantic analysis which are briefly overviewed in this section.

2.2.1 Morphological and Lexical Analysis

The lexicon of a language is its vocabulary, which includes its words and expressions. Morphology is the identification, analysis and description of structure of tokens or words. A token is a set of characters that form a linguistic unit with meaning. Often words are accepted as being the smallest units of analysis and so as being the tokens. In lexical analysis, the aim is to divide the text into paragraphs, sentences and words. Usually this step is performed prior to more complex NLP tasks. In this step, individual words are analyzed into their components and non-word tokens such as punctuation marks are separated from the words, a survey on English morphology is provided by Jurafsky and Martin in Chapter 3 of their book [62].

2.2.2 Syntactic Analysis

Part-of-speech tagging. First, we note that in language usually there are two types of lexical classes: open and closed classes. Closed lexical classes contain a fixed set of words such as the class of articles, prepositions, auxiliary verbs and pronouns. The open classes, on the contrary do not have a fixed word membership such as the class of nouns, verbs, adjectives and adverbs. The task of part-of-speech (POS) tagging, is to assign the lexical category of the words in the sentence. This can be a difficult task due to the ambiguities in the language and because a similar word form can have different POS tags in different contexts. The POS tags for the sentence

The vase is on the ground on your left.[2]

are the following:

DT/The NN/vase VBZ/is IN/on DT/the NN/ground IN/on PRP$/your NN/left ./.

where the tags are based on the Penn Treebank standard tag set[3], for example the tag DT indicates that *The* is a determiner and tag NN indicates that *vase* is a noun.

Parsing. It is the syntactic processing of the language in which a flat input sentence is converted into a hierarchical structure that corresponds to the units of meaning in the sentence. It plays an important role in natural language systems for two reasons: first semantic processing must operate on sentence constituents. If there is no syntactic parsing step, then the semantic system must decide on its own constituents. If parsing is done, on the other hand, it constrains the number of constituents that the semantic analysis can consider. Second, syntactic parsing is computationally less expensive than semantic processing. Thus it can play a significant role in reducing the overall complexity of an NLP system. The two main tasks for identifying the syntax are first part-of-speech tagging and producing the constituent-based parse tree, and second the dependency structure representation using dependency-based parsing which we describe briefly, see also [62].

Constituent-based parsing. The constituent-based parse tree of the above sentence is represented in figure 2.2[4]. S is the root node and the leaf nodes contain the lexical tokens in the sentence (The, vase, is, ...). In the branch

[2] http://cogcomp.cs.illinois.edu/demo/pos/results.php

[3] http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

[4] http://nlp.stanford.edu/software

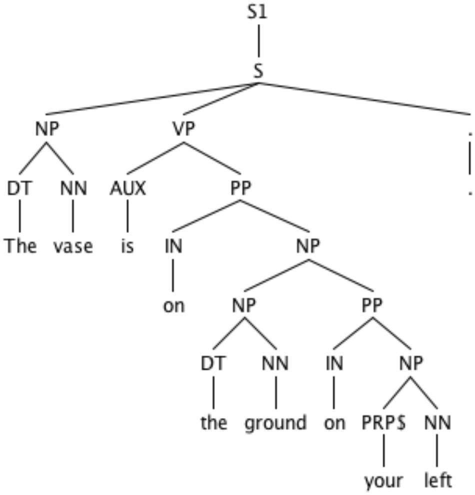


Figure 2.2: Example of a constituent-based parse tree including POS tags.

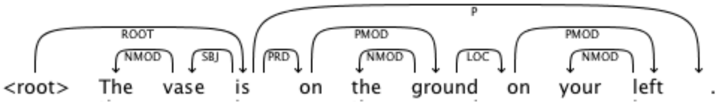


Figure 2.3: Example of a dependency-based parser including dependency relations.

nodes, for example the abbreviation NP is for noun phrase. VP is for verb phrase, and the verb here is an auxiliary verb (AUX). DT stands for the determiner, in this instance for the definite article "the". The tags are again according to the Penn Tree bank tag set.

Dependency-based parsing. The dependency tree of the same sentence is represented in figure 2.3[5]. The basic idea here is to show the links between the lexical items by binary asymmetric relations called dependencies. A dependency relation holds between a head and a dependent. Hence the main challenge of a dependency grammar is determining the criteria for establishing the relations and distinguishing the head from the dependent. The notion of head plays an important role even in the constituent-based frameworks which use the syntactic head. To identify a syntactic head there are some criteria, the main one is that

[5] http://barbar.cs.lth.se:8081/parse

the head of a group of words has the same syntactic and semantic category as the group and often can replace that group. For detailed specifications see [102]. In the example of figure 2.3, the source of an edge is the head and the destination is the dependent. For instance, *is* is the root and does not depend on any other word. *vase* is the subject of *is* and the edge between these two words is labeled by SBJ. *The* is the noun modifier and dependent on *vase* so the edge between them is labeled by NMOD. A detailed description of the labels and abbreviations for the dependency trees is in the manual of Stanford's dependency parser. [6]

2.2.3 Semantic Analysis

The most complex natural language processing tasks are *semantic analysis* and language understanding, which is the subject of interest in this thesis. The term semantic analysis is used more often to refer to discovering sentence level semantics. More course grained semantics considering the influence of the other sentences in the analysis of the meaning of one sentence in a text is referred to as *discourse integration*. When the situational background knowledge is added to the ingredients of the semantic analysis then it is referred as *pragmatic analysis*. However, apart from the granularity level and the task-dependent goals and challenges, the representation of the semantics is always an issue. There are two main approaches for semantic analysis (see also [62]): a) *meaning representation* considering formal languages; b) considering *lexical semantics*. In real world applications, these two main general frameworks can be used as the basis of the more pragmatic tasks that consider the semantics, namely *information extraction* and *mapping to ontologies (ontology population)*. However, information extraction can be considered as a simple case of mapping language to ontologies. We describe these tasks and approaches briefly in the following.

Meaning Representation

The idea behind the notion of meaning representation is that the meaning of the linguistic expressions can be captured in formal structures. Hence, a formal language is used to represent meaning. In this framework, first order logic, description logic, semantic networks, conceptual dependency diagrams or frame-based representations are used.

[6] http://nlp.stanford.edu/software/dependencies_manual.pdf

Lexical Semantics

A different view on semantic analysis compared to meaning representation is to consider rich word semantics drawn from linguistic studies of words. It considers the word senses in the language, the frames of the individual verbs, and the semantics and roles that they allow. We point to some important related tasks that apply this view.

Word sense disambiguation. Often the first step in the semantic processing systems is to look up the individual words in a dictionary (or lexicon) and extract their meanings. Many words have several meanings, and it may not be possible to choose the correct one just by looking at the word itself. The process of determining the correct meaning of an individual word is called *word sense disambiguation* or *lexical disambiguation*. It is done by associating, with each word in the lexicon, information about the contexts in which each of the word's senses may appear. For example in Chapter 5 we use such a task to recognize the sense of the prepositions, particularly their spatial semantics in a sentence.

Semantic role labeling. The choice of semantic roles is not an easy and straightforward type of abstraction. In this respect, *thematic* roles are one attempt to capture the semantic commonality between different verbs. For example the role of *AGENT* is carried by the subject of many verbs such as *break* and *open*. The objects of these two verbs i.e. *BrokenThing* and *OpenedThing* are directly affected, and their thematic role is *theme*. The goal of using thematic roles and semantic roles in general is to have a shallow meaning representation that makes inference about the meaning feasible. The problem with thematic roles is the difficulty of coming up with a standard set of roles. It is difficult to provide a formal definition of the roles of *AGENT, THEME, INSTRUMENT*, etc. To deal with this problem one solution has been to define *generalized semantic roles* that abstract over the specific thematic roles. In this case, the roles that act like patient and agent roles called *PROTO-AGENT* and *PROTO-PATIENT* are used. In addition to proto-roles, many computational models define semantic roles particular for a set of verbs or nouns. The two most commonly used lexical resources that use some alternative version of semantic roles are PropBank which uses both proto-roles and verb-specific semantic roles, and FrameNet which uses frame-specific semantic roles. For a more in depth discussion about the roles and variations see [62].

Semantic role labeling annotates every verb in the sentence with a structure called semantic frame usually according to the ProbBank frames [106]. A semantic frame consists of a predicate label that indicates the sense of the verb, and a number of arguments called semantic roles. A semantic role indicates the

The vase	is	on	the ground on your left	.
agent[A0]	V:be	-	patient[A1]	-

Table 2.1: Example of the output of a semantic role labeler.

The meeting	is	on	Monday	.
agent[A0]	V:be	-	temporal[AM-TMP]	-

Table 2.2: Example of the output of a semantic role labeler.

role of the labeled phrase in the sentence with respect to the predicate which is the verb. In table 2.1, for example the verb *is* has predicate label *be*. Semantic roles, *The vase*, with label A0, is an argument of the predicate and has the agent role with respect to *is*. The phrase *the ground on your left* with the label A1, is the second argument of *is* and has the role of patient with respect to the predicate. In the second example in table 2.2, in addition to the main role of agent, there is another secondary role of AM-TMP which shows *Monday* has temporal information. These additional roles are called adjuncts[7].

Information Extraction

Information extraction is usually defined as the extraction of limited types of semantics from unstructured data (here text) and as the storage of the extracted information in some structured format, for example relational databases [98]. The most well-known IE task is named entity recognition (NER). In this task the goal is to find the names of the people, places and organizations that are mentioned in ordinary news texts. A step further in IE is relation extraction and classification. For example, relations such as employee-of or part-of are recognized that hold between the identified entities.

Jantje/PERSON is going to Spain/LOCATION tomorrow/DATE for holidays.

Another linguistic task which can be classified in the IE category of tasks is temporal information extraction and the extraction of temporal relations between events [68]. It is worth mentioning that semantic role labeling also can be viewed as a kind of information extraction that extracts linguistically oriented semantics where the verb in the sentence has a pivotal role. The spatial information extraction, which is the topic of this thesis, also can be considered

[7] http://cogcomp.cs.illinois.edu/demo/srl/

as an information extraction task, which particularly targets the extraction of spatial semantics.

2.3 Ontologies

Here we introduce some terms and provide definitions about the concepts related to ontologies.

Formal Ontology is concerned with the systematic enumeration and classification of the various kinds of entities represented within a given conceptualization of the world, together with an account of their properties and relationships [45].

To distinguish ontologies from knowledge bases, ontologies are described as a scheme for the knowledge base. In every knowledge base there is a kind of conceptualization, either explicitly or implicitly. This conceptualization is what is referred to as ontology [49]. Hence, knowledge bases can be built by extracting the relevant instances from information to populate the corresponding ontologies. This process is known as *ontology population* or *knowledge markup* [166]. Depending on the available knowledge and information resources in a specific domain, ontologies can be hand crafted or even be learnt automatically. In this respect, the task of *learning ontology* from text can be defined as the process of deriving high-level concepts and relations in addition to the axioms from the available information to build an ontology.

2.3.1 Ontology Components

There are five types of components that make up an ontology, namely, *terms, concepts, taxonomic relations, non-taxonomic relations*, and *axioms*. Terms are used to form concepts and the concepts which are related to each other make the relation layer of the ontology. Relations represent the interaction between the concepts in the ontology.

Taxonomic relations can make a hierarchy of concepts. This implies the discovery of the `is-a(X,Y)` relationships, also referred to as hyperny/hyponym. Non-taxonomic relations describe more complex interactions between the concepts and extraction of these relations need syntactic and dependency analysis of the textual information. The meronymy relations such as `part-of(X,Y)` or `contain(X,Y)`, attributes such as `property(X,Y)` or other relations such as thematic roles, possession, and causality are more complex to handle in learning and population of the ontologies. Finally, by generalizing over the relations, the axioms are produced. Ontologies may contain axioms for validation and enforcing constraints. Depending on the relational and axiomatic richness and the formality of representation in an ontology, a spectrum of different kinds of ontologies emerge. In one end, the term *lightweight ontology* is referred to the ontologies that make little or no use of the axioms. At the other end, the term *heavyweight ontology* referrs to the ontologies that intensively use axioms in their specifications [47].

2.3.2 Meaning Representation via Mapping to Ontologies

We consider mapping natural language to ontologies as a general framework for semantic representation. Mapping to a knowledge representation scheme supported by an ontological scheme, can be seen as an extensive and deep information extraction paradigm which is considered in modern information systems.

Referring back to the semantic analysis of natural language described in the previous section, here we can make its connection to the ontologies more clear. In fact, one of the most widely accepted methodologies for meaning representation using formal semantics, is *model-theoretic semantics* [105]. In *model-theoretic semantics*, syntactically correct utterances in a language are assigned a semantic interpretation in terms of truth values with respect to a certain world model. This world model is based on an ontology represented by a formal language. Interpreting the meaning of the textual units needs a detailed world model that uses as few as possible primitives, and it enables the world modeler to build descriptions of complex objects and process them in a computational fashion.

Understanding natural language text can be modeled as matching the text and the pre-defined ontology. In the most pragmatic and shallow case, this will be a task similar to information extraction with a number of named entities and their relationships. When having a rich set of concepts and relations, mapping the text to an ontology will be a kind of meaning representation generation. This is similar to knowledge-based systems which model *understanding* through representing the outcome of the input text analysis as a set of well-formed structures in a formal artificial language. The crucial point in designing ontologies, in order to have any explanatory power, is that the building blocks of the meaning representation language must be interpreted in terms of an independently motivated model of the world. *The process of NL analysis is then interpreted as putting lexical, syntactic, and prosodic units of the source text in correspondence with elements of the text meaning representation language* [105]. This is the idea that is followed in modern information extraction systems and particularly in the semantic web model [166]. In this thesis when extracting spatial semantics, we go beyond a basic information extraction task by viewing this task as a mapping of text to spatial ontologies.

2.3.3 Space in Ontology

Spatial information usually refers to the information about the physical location of objects or entities in a space. Formalizing spatial semantics and also considering the way those are expressed in natural language is an extremely active and challenging research area [10]. The ontology of space is not granted as a settled down question though it has been the subject of a very old philosophical debate. The two concrete elements of an ontology of space are spatial entities constituting the space and the primitive spatial notions expressed over these entities. These elements are in fact interdependent, some notions are more difficult to express over some kind of entities [160]. There is a large body of research on representing the spatial notions in a formal qualitative model appropriate for automatic spatial reasoning. These models are referred to as *qualitative spatial reasoning* models (QSR). Another research domain in which spatial primitives have been formalized and practically used is the domain of geographical information systems. These systems mostly consider geographical entities and toponymy. A general ontological view on the space aims to integrate all the current formalizations about the space and connect them to natural language [5] via an extensive ontology engineering. The SFB/TR8 project is the main active project in this direction,

We see the ontological modeling of space as particularly necessary within the SFB for facilitating qualitative spatial reasoning in general, for achieving interoperability over the different spatial calculi used within the participating projects, and for ontologically grounding the spatial expressions found in natural language, Bateman et al. [9] p. 1.

Part I

Spatial Information Extraction

Outline

Extraction of spatial information from natural language is a key task for applications that are required to answer questions or reason about spatial relationships between entities. The research fields of natural language processing, computer vision and robotics, human machine interaction and geographical information systems (GIS) concern this type of information in various tasks. Examples include systems that perform text-to-scene conversion, generation of textual descriptions from visual data, robot navigation tasks and giving directional instructions.

Contrary to the large body of past research on spatial cognition, spatial language and spatial reasoning models, the extraction of spatial information from natural language is not well-developed yet and does not have a well-defined framework. This is in contrast to the extraction of temporal information, and it is due to the complexity of the spatial entities and notions. There is neither a systematic research work in this field from a computational linguistics (CL) point of view, nor a systematic application of the current dominating approach in CL which is statistical machine learning. The main obstacles for employing machine learning in this context have been the lack of an agreement on a unique semantic representation model and the gap between expressiveness of natural language and the existing formal spatial representations. These explain the lack of annotated data on which machine learning can be employed to learn and extract the spatial relations. These shortages are the main motivations of the work presented in this part.

In this part of the thesis, in Chapter 3, we establish a framework for spatial information extraction in the form of a spatial annotation scheme to be used by the machine learning models and we prepare rich corpora according to this scheme for further machine learning investigations.

In chapter 4 we define the task of spatial information extraction as ontology population for mapping natural language to spatial ontologies based on the proposed annotation scheme and the corpus analysis. This novel task is the basis for our computational linguistic models for extraction of spatial information in the whole thesis. Given the rich structural characteristics of the spatial language and the relationships in the spatial ontologies, this task forms a highly challenging semantic extraction task which is interesting for structured and relational machine learning models. The following publications are the basis of this part.

Related publication

Kordjamshidi, P., van Otterlo, M., Moens, M. F. (2010). Spatial role labeling: Task definition and annotation scheme. In Calzolari, N. (Ed.), Khalid, C. (Ed.), Bente, M. (Ed.), *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10).* Malta, 19-21 May 2010 (pp. 413-420) European Language Resources Association (ELRA).

Kordjamshidi, P., van Otterlo, M., Moens, M. F. (2010). From language towards formal spatial calculi. In Ross, R. (Ed.), Hois, J. (Ed.), Kelleher, J. (Ed.), Proc. of 1st Workshop COSLI'10. *Computational Models of Spatial Language Interpretation (COSLI).* Mt.Hood/Portland, OR, USA, 15-August 2010 (pp. 17-24).

Kordjamshidi, P., Hois, J., van Otterlo, M., Moens, M. F. (2011). Machine learning for interpretation of spatial natural language in terms of QSR. *International Conference on Spatial Information Theory (COSIT1). Extended abstract, Vol. Spatial Information Theory (technical report). COSIT.* Belfast, 12-16 September (pp. 1-5).

Kordjamshidi, P., Bethard, S., Moens, M. F. (2012). SemEval-2012 task 3: Spatial role labeling. **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012): Vol. 2.* SemEval-2012. Montreal- Canada, 7-8 June (pp. 365-373) ACL.

Kordjamshidi, P., Hois, J., van Otterlo, M., Moens, M. F. (2013). Learning to interpret spatial natural language in terms of qualitative spatial relations. In: Tenbrink T. (Ed.), Wiener J. (Ed.), Claramunt C. (Ed.), *Representing Space in Cognition: Interrelations of Behavior, Language, and Formal Models* Oxford University Press.

Chapter 3

Spatial Annotation Scheme

Given the large body of the past research on various aspects of spatial information, the main obstacles for employing machine learning for extraction of this type of information from natural language have been: a) the lack of an agreement on a unique semantic model for spatial information; b) the diversity of formal spatial representation models ; c) the gap between the expressiveness of natural language and formal spatial representation models and consequently; d) the lack of annotated data on which machine learning can be employed to learn and extract the spatial relations. These items draw the direction of the contributions on which this chapter is built.

In this chapter we introduce a spatial annotation scheme for natural language that supports various aspects of spatial semantics, including static and dynamic spatial relations. The annotation scheme is based on the ideas of *holistic spatial semantics* as well as *qualitative spatial reasoning* models. Spatial roles, their relations and indicators along with their multiple formal meaning are tagged using the annotation scheme producing a rich spatial language corpus. The goal of building such a corpus is to produce a resource for training the machine learning methods for mapping the language to formal spatial representation models, and to use it as ground-truth data for evaluation.

We describe the foundations and the motivations for the concepts used in designing our spatial annotation scheme in Section 1. We illustrate the scheme and its XML and relational representation via examples, in Section 2. The investigated corpora, annotated data and the annotation challenges are described in Section 3. We conclude in Section 4.

3.1 Annotation Scheme: Foundation and Motivation

In the proposed annotation scheme two main aspects of spatial information are considered. Firstly the cognitive aspects and the way that spatial concepts are expressed in the language, and secondly the formal models that are designed for spatial knowledge representation and reasoning independent from natural language. A scheme which covers these aspects will be able to connect natural language to formal models and make spatial reasoning based on text feasible. Here we describe the main elements related to these aspects that form the basis of the proposed scheme. We point to the challenges in making a flexible connection between these two sides of spatial information.

3.1.1 Holistic Spatial Semantics

One part of our proposed scheme is based on the *holistic spatial semantics* theory. An approach to spatial semantics that has the utterance (itself embedded in discourse and a background of practices) as its main unit of analysis, rather than the isolated word, is characterized as *holistic*. Such an approach aims at determining the semantic contribution of each and every element of the spatial utterance in relation to the meaning of the whole utterance. One major advantage of such an approach is that it does not limit the analysis to a particular linguistic form, form class (e.g. prepositions), or theoretically biased grammatical notion. The main spatial concepts considered in this theory are the following.

Trajector: The entity whose location or position is described. It can be static or dynamic; persons, objects, or events. Alternative common terms include local/figure object, locatum, referent, or target.

Landmark: The reference entity in relation to which the location or the motion of the trajector is specified. Alternate terms are reference object, ground, or relatum.

Region: This concept denotes a region of space which is defined in relation to a landmark. By specifying a value such as interior or exterior for this category, the trajector is related more specifically and more precisely with respect to the landmark.

Path: It is a most schematic characterization of the trajector of actual or virtual motion in relation to a region defined by the landmark. In cognitive semantics this concept is used in two different ways, rich path or minimal path. The minimal path is represented by its *beginning*, *middle* and *end*, similar to the distinction *source/medium/goal*. The minimal path is enriched when its information is combined with region or place.

Motion: This concept also can be characterized in a rich or minimal way. In its minimal way, motion is treated as a binary component indicating whether there is perceived motion or not. The minimal representation of motion allows a clear separation from the path and direction, while the rich one conflates it with these.

Direction: It denotes a direction along the axes provided by the different frames of reference, in case the trajector of motion is not characterized in terms of its relation to the region of a landmark.

Frame of reference: In general, a frame of reference defines one or more *reference points*, and possibly a coordinate system based on axes and angles. Three reference types can typically be grammaticalized or lexicalized in English: intrinsic, relative, and absolute [81]. Recently, more detailed distinctions were presented in [152], where spatial reference frames are represented and systematically specified by the spatial roles locatum, relatum, and (optional) vantage together with a directional system.

However, how these theoretical concepts are applied to linguistic descriptions, is a controversial question. The answer to this question has many challenges such as dealing with polysemy, characterizing the semantic and phonological poles of the language [168]. In the holistic approach a many-to-many mapping between semantic concepts and form classes is allowed [167]. For example, in general a specific word can contribute to expressing the concept of landmark as well as region or even path.

3.1.2 Qualitative Spatial Representation

The second part of our suggested scheme is based on qualitative spatial reasoning (QSR) models. QSR models are designed based on more logical, geometrical or algebraic spatial semantics independent from natural language. However the *cognitive adequacy* of these models has been an important concern. *Cognitive adequacy* refers to the degree in which a set of concepts and relationships, and the computational inference over them is consistent with the mental conceptualization of humans and the way that a human reasons about those concepts and their relationships [119]. Two important reasons for paying attention to the qualitative approach are a) this model is closer to how humans represent and reason about commonsense knowledge; b) it is flexibile in dealing with incomplete knowledge [118].

Three main aspects of spatial information are topological, directional and distal information which are somehow complementary information that could specify the location of the objects under consideration. Other aspects are size, shape, morphology, and spatial change (motion). Most of the qualitative spatial calculi focus on a single aspect, e.g., topology, direction, distance but recently there are combinatory models and tools that are able to reason based on multiple calculi models [118, 163]. Here we briefly describe the main aspects of the spatial information that are the basis of our spatial meaning representation in the proposed scheme and qualitative calculi models are available for them.

Topological Relations

Distinguishing topological relationships between spatial entities is a fundamental aspect of spatial knowledge. Topological relations are inherently qualitative and hence suitable for qualitative spatial reasoning. In reasoning models based on topological relations, the spatial entities are assumed to be regions rather than points, and regions are subspaces of some topological space [118]. A set of jointly exhaustive and pairwise disjoint relations, which can be defined in all topological models based on *parthood* and *connectedness* relations, are DC, EC, PO, EQ, TPP, NTPP, TPP^{-1}, NTPP^{-1}. The best known approach in this domain is the Region Connection Calculus by Randell et al. [115] known as the RCC-8 model that we use to represent the topological relationships expressed in the language. RCC is heavily used in qualitative spatial representation and reasoning. The above relation symbols are abbreviations of their meanings (see Fig. 3.1): disconnected $\mathsf{DC}(\mathsf{a},\mathsf{b})$, externally connected $\mathsf{EC}(\mathsf{a},\mathsf{b})$, partial overlap $\mathsf{PO}(\mathsf{a},\mathsf{b})$, equal $\mathsf{EQ}(\mathsf{a},\mathsf{b})$, tangential proper-part $\mathsf{TPP}(\mathsf{a},\mathsf{b})$, non-tangential proper-part $\mathsf{NTPP}(\mathsf{a},\mathsf{b})$, tangential proper-part inverse $\mathsf{TPP}^{-1}(\mathsf{a},\mathsf{b})$, and non-tangential proper-part inverse $\mathsf{NTPP}^{-1}(\mathsf{a},\mathsf{b})$, which describe mutually exclusive and exhaustive overlap and touching relationships between two (well-behaved) regions in the space. The cognitive adequacy of this model is discussed in [119].

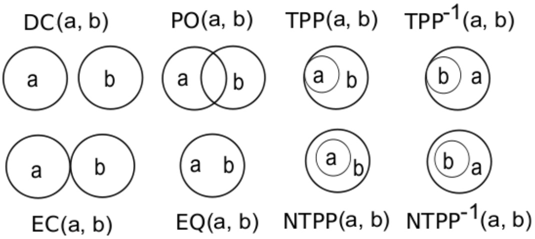


Figure 3.1: The RCC-8 relations.

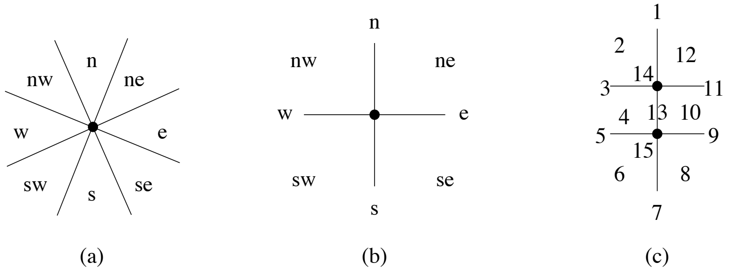


Figure 3.2: Directional relations between points: (a) Cone-based model; (b) Projection-based model; (c) Double-cross model [118].

There are other topological models such as 9-intersection given by Egenhofer [39] which is based on interior, exterior, and boundary of regions.

Directional Relations

Direction or orientation is also frequently used in linguistic descriptions about spatial relations between objects in qualitative terms, for example the expressions such as *to the left* or *in the north* are more often used than *45 degrees*. The frame of reference discussed in the previous section is an important feature to characterize directional relations. Absolute directions are in the form of {S(south), W(west), N(north), E(east), NE(northeast), SE(southeast), NW(northwest), SW(southwest)} in a geographical space. Relative directions are {Left, Right, Front, Behind, Above, Below} and used in a local space. These are only different in terminology compared to the former set of relations and can be adapted and used in qualitative direction calculus such as the cone-base, projection-based and double-cross models [118] (see figure 3.2). The double cross model (figure 3.2.c) assumes an additional axis and considers a perspective point in addition to the reference point.

Distal Relations

Along with the topology and direction, distance is one of the most important aspects of the space. Distance is a scalar entity and can be represented *qualitatively* such as *close*, *far* or *quantitatively* such as *two meters far*. Distances are also categorized as being either *absolute* or *relative*. The absolute distance describes the distance between two entities and the relative distance describes the distance between two entities compared to a third one. The computational models for distances often consider spatial entities as points. For more information about the various models for distal reasoning see [118, 163].

3.1.3 The Gap between Natural Language and Formal Models

Although QSR models are often cognitively adequate, spatial language contains ambiguity and polysemy. Moreover, linguistic spatial expressions can express complex spatial semantics considering various aspects of the space at the same time. In contrast to natural language, formal spatial models focus on one particular spatial aspect and specify its underlying spatial logic in detail [55]. Therefore there is a gap between the level of expressivity and specification of natural language and spatial calculi models [7]. Huge spatial ontologies are needed to be able to represent the spatial semantics expressed in the linguistic expressions. In the work of [54] the alignment between the linguistic and logical formalizations is discussed. Since these two aspects are rather different and provide descriptions of the environment from different viewpoints, constructing an intermediate, linguistically motivated ontology is proposed to establish a flexible connection between them. Generalized Upper Model (GUM) is the state-of-the-art example of such an ontology [6, 122]. The GUM-Space ontology is a linguistically motivated ontology that draws on findings from empirical cognitive and psycholinguistic research as well as on results from theoretical language science. GUM contains 73 spatial modalities that are distinguished in the GUM-Space ontology based on their hierarchical dependencies [8]. However, we realized, mapping to an intermediate linguistic ontology with a fairly large and fine-grained division of concepts is to some extent difficult because firstly it implies the need for a huge labeled corpus, secondly the semantic overlap between the included relations in the large ontologies makes the learning model more complex. In addition, although the logical reasoning is computationally possible using an ontology such as GUM, the kind of spatial reasoning which is provided by calculi models is not feasible. Hence to perform actual spatial reasoning another layer of bridging between the GUM representation and calculi models is required [54]. Therefore, we use a layer of formal representation models in our proposed scheme instead of linguistically motivated ontologies. However, to alleviate the above explained gap we propose to map the linguistic expressions to multiple calculi. This issue is reflected in our annotation scheme and will be discussed in the following section.

```
(1) TRAJECTOR(idT,token)
(2) LANDMARK(idL,token,path)
(3) SPATIAL_INDICATOR(idI,token)
(4) MOTION_INDICATOR(idM,token)
(5) SR(idS,idI,idT,idL,idM)
(6) SRType(idS,id_gtype,gtype,stype,sp_value,f_o_ref)
```

Table 3.1: Relational representation of the annotation scheme.

3.2 Annotation Scheme: Relational Representation

We design an annotation scheme for tagging natural language with spatial roles, relations and their meaning. We take into account the cognitive-linguistic spatial primitives according to the theory of holistic spatial semantics as well as spatial relations according to the well-known qualitative spatial representation models described in Section 3.1.2. Table 3.1 shows the relational representation of the proposed spatial scheme. We describe these relations and the used terminology in the following.

In all these relations a *token* can be a word or a set of words. Each token that identifies a spatial role is assigned a unique *key*. Each token can play multiple roles as trajector or landmark in the sentence, thereby participating in various spatial relations. Each token is assigned a new identifier for each role that it plays. As it is shown in table 3.1,

In relation `(1)`, `idT` is an identifier that identifies a *token* that plays the role of *trajector*.

In relation `(2)`, `idL` is an identifier that identifies a *token* that plays the role of *landmark*. Each landmark is related to a *path* which characterizes a path or a complex landmark with a value in `{BEGIN,MIDDLE,END,ZERO}`. `ZERO` value is assigned when the path is not relevant.

In relation `(3)`, `idI` is an identifier that identifies a token that indicates the existence of a spatial relation and is called *spatial indicator*. According to the HSS theory [167], the relationship between trajector and landmark is not expressed directly but mostly via the region or direction concepts. We abstract from the semantics of these bridging concepts and tag the tokens which define constraints on the spatial properties- such as the location of the trajector with respect to the landmark- as a *spatial indicator* (e.g. *in*, *on*). A spatial indicator signals the existence of a spatial relation independent from its semantics.

In relation (4), idM is an identifier that identifies a token (a word here) that indicates the existence of any kind of motion with a spatial influence in the sentence.

In relation (5), we present a complex relation which links all the elements that are a part of a whole *spatial configuration* containing the identifiers of the above mentioned relations. This relation, which is named as SR, is identified by the identifier idS to be used in describing its semantic properties in relation (6). We refer to this relation as *spatial relation* later.

In relation (6), the type of the semantics of the *spatial configuration* is determined regarding the involved components. Since all of these components (trajector, landmark, etc.) contribute to the semantics of the relation, the fine-grained semantics are assigned to the whole *spatial configuration* which was identified by idS. We allow multiple semantics to be assigned to one spatial configuration, hence the additional identifier id_gtype is used to identify each related type. All the above mentioned elements are related to the cognitive elements of the spatial configuration but this relation is about the *formal representation* of the semantics which we now clarify in detail.

Formal semantics. As discussed in Section 3.1.3, to cover all possible semantic aspects of a linguistic expression about a spatial configuration, we allow multiple semantics to be assigned to it. We revisit this issue later in Chapter 4. For each spatial relation/configuration, we assign one or more general types which have one of the values {REGION,DIRECTION,DISTANCE}. With respect to each general type a specific type is established. The specific type of a relation that is expressed by the configuration is stated in the stype attribute. If the gtype is REGION then we set stype with topological relations in a formalism like RCC8 [141] (any other topological model might be used here). If an indicator of direction is observed then the stype can be {ABSOLUTE,RELATIVE}. The absolute and relative direction values are also according to Section 3.1.2. In case the gtype of the spatial relation is DISTANCE then it is classified as {QUALITATIVE,QUANTITATIVE}. For qualitative distances we use a predefined set of terms including far, near, etc., and for quantitative distances the numbers and values in the text form the key distance information. Finally, each spatial relation given its general type identifier is tagged by a frame of reference f_o_ref with a value in {INTRINSIC,RELATIVE,ABSOLUTE}.

3.2.1 Annotation Approach

Semantic annotation of a corpus is a challenging, and ambiguous task [99]. We have investigated several kinds of spatial descriptions to facilitate the annotation process, and we have defined guidelines to make the task easier and less ambiguous. Below we list a set of questions which annotators should ask themselves while annotating. The annotations are performed at the sentence level. The annotators use their understanding of explicit words and their senses. The questions are:

1. Is there any direct (without commonsense implications) spatial description in the sentence?
2. Which words are the indicators of the spatial information?
3. Which words are the arguments of those spatial indicators (semantically connected)?
4. Which tokens have the role of trajector for the spatial indicator and of *what* is the spatial description described?
5. Which tokens have the role of landmark for the spatial indicator? (Is there a landmark?)
6. Is there a complex landmark? if so, can we describe it in terms of a point in a path (beginning, middle, end)?
7. Is there a motion? and if so, which tokens refer to the motion indicator?
8. What is the frame of reference?
9. Given a predefined set of formal spatial relations, which formal relation describes the semantics the best?
10. Is one formal semantic type enough for a rough visualization/schematization of the meaning of the spatial relation, and locating the objects in the space?
11. Do we need multiple annotations to capture the semantics of the relation?

To aid dealing with ambiguities in the annotation task we categorize the spatial descriptions into *complex* and *simple* descriptions. The annotation guidelines and examples are described first in the simple case and later extended to complex cases. The answers to questions $8, 9, 10$ require the selection of a formal spatial representation which can be with multiple choices.

3.2.2 Simple Descriptions

We define a *simple description* as a spatial description which includes one target, at most one landmark and at most one spatial indicator. For answering the first question mentioned in the previous section we consider the conventional specifications of the location or change of location (i.e. translocation) of an entity in space as a spatial description such that conversational implications are excluded. For example, the answer *He is washing the dishes* to the question *Where is he?* could – with some inference – imply *He is in the kitchen*, but we do not consider that here. Examples of simple descriptions are:

Example 1.

a. There is a meeting on Monday.
b. There is a book on the table.

Example 1.*a.* has the same structure of a spatial description with the preposition "on" but "on Monday" is a temporal expression, so there is no spatial description, but in Example 1.*b.*, there is a spatial description about the location of a book. In case there is a spatial description in the sentence, its components are tagged according to the aforementioned definitions.

Trajector

The following sentences show the way *trajector* should be annotated.

Example 2.

a. She is at school.
<TRAJECTOR id='1'> She </TRAJECTOR>
b. She went to school.
<TRAJECTOR id='1'> She </TRAJECTOR>
c. The book is on the table.
<TRAJECTOR id='1'> The book </TRAJECTOR>
d. She is playing in her room.
<TRAJECTOR id='1'> She </TRAJECTOR>
e. Go left!
<TRAJECTOR id='0'> NIL </TRAJECTOR>

When the trajector is implicit as in example 2.*e.* "NIL"is added as trajector.

Landmark

A *landmark* is tagged according to its aforementioned definition. The source of ambiguity here is that sometimes an explicit landmark is not always needed, for example in the case of directions. The second more difficult case is when the landmark is deleted by ellipsis and it is implicit. In such cases we annotate the landmark by NIL.

Example 3.

a. The balloon passed over the house.
<LANDMARK id=’1’ path=’ZERO’>the house</LANDMARK>
b. The balloon passed over.
<LANDMARK id=’1’ path=’ZERO’>NIL</LANDMARK>
c. The balloon went up.
<LANDMARK id=’1’ path=’ZERO’>NIL</LANDMARK>
d. The balloon went over there.
<LANDMARK id=’1’ path=’ZERO’>there</LANDMARK>
e. John went out of the room.
<LANDMARK id=’1’ path=’BEGINNING’> the room </LANDMARK>
f. John went through the room.
<LANDMARK id=’1’ path=’MIDDLE’>the room</LANDMARK>
g. John went into the room.
<LANDMARK id=’1’ path=’END’>the room</LANDMARK>
h. John is in the room.
<LANDMARK id=’1’ path=’ZERO’>the room</LANDMARK>

In example 3.*c.* we have a relative direction, and thus an implicit landmark should be there. In example 3.*d.* “there”should be resolved in preprocessing or postprocessing and the annotators should not concern the reference resolution here. Another special case happens when there is a motion with spatial effect and the landmark is like a path and the indicators indicate a relation in some part of the path. In that case a path attribute is set; see the examples 3.*e.* to 3.*h.*

Spatial Indicator

The spatial terms, or spatial indicators, are mostly prepositions but can also be verbs, nouns and adverbs or a combination of them. We annotate each signal of the existence of the spatial information in the sentence as spatial indicator.
Example 4.

a. He is in front of the bush.
<SPATIAL-INDICATOR id=’1’ > in front of</SPATIAL-INDICATOR>
b. Sit behind the bush.
<SPATIAL-INDICATOR id=’1’ > behind </SPATIAL-INDICATOR>
c. John is in the room.
<SPATIAL-INDICATOR id=’1’ > in </SPATIAL-INDICATOR>

Motion Indicator

These are mostly the prepositional verbs but we leave it open for other semantical categories like adverbs, etc. In this scheme we just tag them as indicators but a further extension is to map them to motion verb classes.

Example 5.

a. The bird flew to its nest.
<MOTION-INDICATOR id=’1’ > flew to</MOTION-INDICATOR>

We tag the token “flew to”as the indicator because the preposition affects the semantics of the motion.

Spatial Relation and Formal Semantics

The spatial configuration’s components recognized by the annotators should be put in relations called *spatial relations* (SR). In a simple description it is often easy because we have one trajector, one/zero landmark and one spatial indicator, so these constitute at least one clear coarse spatial relation to be tagged. If a motion indicator is present which is related to the spatial relation and the location of the trajector then the identifier of the motion also is added in the spatial relation. Each spatial relation is associated with a number of formal semantics, for example, when it implies both topological and directional information. The difficulty of annotation is how to fill in the semantic attributes. In other words the mapping between linguistic terms and formal relations like RCC is not always clear and easy. We discuss this later in this chapter. For each type of relation we add a new frame of reference as an attribute. For example, the frame of reference is more relevant for the directional relationships compared to topological relationships. Hence, it makes more sense to assign this concept according to each specific annotated type of semantics.

Example 6.

a. She is at school.
<TRAJECTOR id=’1’ > She</TRAJECTOR>
<LANDMARK id=’1’ path=’ZERO’>school</LANDMARK>
<SPATIAL-INDICATOR id=’1’ > at </SPATIAL-INDICATOR>
<SR id=’1’ trajector=’1’ landmark=’1’ spatial-indicator=’1’ motion-indicator=’NIL’/>
<SR id=’1’ SRtype id=’1’ general-type=’REGION’ specific-type=’RCC8’ spatial-value=’TPP’ frame-of-reference=’INTRINSIC’ />
b. She went to school.
<TRAJECTOR id=’1’ > She</TRAJECTOR>

<LANDMARK id=’1’ path=’END’> school </LANDMARK>
<SPATIAL-INDICATOR id=’1’ > to </SPATIAL-INDICATOR>
<MOTION-INDICATOR id=’1’ > went to </MOTION-INDICATOR>
<SR id=’1’ trajector=’1’ landmark=’1’ spatial-indicator=’1’ frame-of-reference=’INTRINSIC’ motion-indicator=’1’/>

<SR id=’1’ SRtype id=’1’ general-type=’REGION’ specific-type= ’RCC8’ spatial-value=’TPP’ frame-of-reference=’INTRINSIC’ />

c. The book is on the table.
<TRAJECTOR id=’1’ > The book </TRAJECTOR>
<LANDMARK id=’1’ path=’ZERO’> table </LANDMARK>
<SPATIAL-INDICATOR id=’1’ > on </SPATIAL-INDICATOR>
<SR id=’1’ trajector=’1’ landmark=’1’ spatial-indicator=’1 motion-indicator=’NIL’/>
<SR id=’1’ SRtype id=’1’ general-type=’REGION’ specific-type=’RCC8’ spatial-value=’EC’ ’ frame-of-reference=’INTRINSIC’ />
d. She is playing in her room.
<TRAJECTOR id=’1’> She </TRAJECTOR>
<LANDMARK id=’1’ path=’ZERO’> her room </LANDMARK>
<SPATIAL-INDICATOR id=’1’ > in </SPATIAL-INDICATOR>
<MOTION-INDICATOR id=’1’ > playing </MOTION-INDICATOR>
<SR id=’1’ trajector=’1’ landmark=’1’ spatial-indicator=’1 motion-indicator=’1’/>
<SR id=’1’ SRtype id=’1’ general-type=’REGION’ specific-type=’RCC8’ spatial-value=’TPP’ frame-of-reference=’INTRINSIC’/>

3.2.3 Complex Descriptions

In this section we illustrate how our scheme is able to handle complex spatial descriptions. In [3] three classes of complex description forms are identified to which we point here:
I: Complex locative statements are locative phrases with more than one landmark. The explanations are about one target, meanwhile some relations can be inferred between landmarks, but for the annotation – annotators should not do additional reasoning steps – only what is explicitly expressed in the sentence should be tagged. Therefore the annotation in example 7, is a straightforward annotation of various possible spatial relations.

Example 7.

The vase is in the living room, on the table under the window.
<TRAJECTOR id=’1’> The vase </TRAJECTOR>
<LANDMARK id=’1’ path=’ZERO’> the living room </LANDMARK>
<LANDMARK id=’2’ path=’ZERO’> the table </LANDMARK>

```
<LANDMARK id=’3’ path=’ZERO’>the window </LANDMARK>
<SPATIAL-INDICATOR id=’1’ > in </SPATIAL-INDICATOR >
<SPATIAL-INDICATOR id=’2’ > on </SPATIAL-INDICATOR >
<SPATIAL-INDICATOR id=’3’ > under </SPATIAL-INDICATOR>
<SR id=’1’ trajector=’1’ landmark=’1’ spatial-indicator=’1’ motion-indicator=’NIL’ />
<SR id=’1’ SRtype=’1’ general-type=’REGION’ specific-type=’RCC8’ spatial-value=’NTPP’ frame-of-reference=’INTRINSIC’ />
<SR id=’2’ trajector=’1’ landmark=’2’ spatial-indicator=’2’ motion-indicator=’NIL’/>
<SR id=’2’ SRtype=’1’ general-type=’REGION’ specific-type=’RCC8’ spatial-value=’EC’ frame-of-reference=’INTRINSIC’ />
<SR id=’3’ trajector=’1’ landmark=’3’ spatial-indicator=’3’ motion-indicator=’NIL’/>
<SR id=’3’ SRtype=’1’ general-type=’DIRECTION’ specific-type=’RELATIVE’ spatial-value=’BELOW’ frame-of-reference=’INTRINSIC’ />
```

II: Path and route descriptions are possibly the most important for multimodal systems. In this kind of descriptions a *focus shift* can happen. It means that the speaker explains one target referring to some landmarks, but at some point explains another object or landmark, i.e. the focus shifts to another entity as trajector. Annotators should recognize this focus shift and annotate the rest of the phrases by the new trajector. The following example shows such an expression, but here we only tagged the spatial indicators and not the motion indicators to simplify its representation.

Example 8.

The man came from between the shops, ran along the road and disappeared down the alley by the church.

```
<TRAJECTOR id=’1’ > the man </TRAJECTOR>
<LANDMARK id=’1’ path=’BEGINNING’> the shops </LANDMARK>
<LANDMARK id=’3’ path=’END’> the alley <LANDMARK/>
<TRAJECTOR id=’2’ > the alley </TRAJECTOR >
<LANDMARK id=’4’ path=’ZERO’> the church </LANDMARK>
<SPATIAL-INDICATOR id=’1’ > between </SPATIAL-INDICATOR >
<SPATIAL-INDICATOR id=’2’ > along </SPATIAL-INDICATOR>
<SPATIAL-INDICATOR id=’3’ > down </SPATIAL-INDICATOR>
<SPATIAL-INDICATOR id=’4’ > by </SPATIAL-INDICATOR>

<SR id=’1’ trajector=’1’ landmark=’1’ spatial-indicator=’1’ motion-indicator=’NIL’/>
<SR id=’1’ SRtype=’1’ general-type=’Region’ specific-type=’RCC8’ spatial-value=’IN’ frame-of-reference=’INTRINSIC’ motion-indicator=’NIL’/>
```

```
<SR id=’2’ trajector=’1’ landmark=’2’ spatial-indicator=’2’ motion-indicator=’NIL’/>

<SR id=’2’ SRtype id=’1’ general-type=’Region’ specific-type=’RCC8’ spatial-value=’EC’ frame-of-reference=’INTRINSIC’/>
<SR id=’3’ trajector=’1’ landmark=’3’ spatial-indicator=’3’ frame-of-reference=’RELATIVE’ motion-indicator=’NIL’/>
<SR id=’3’ SRtype id=’1’ general-type=’Direction’ specific-type=’Relative’ spatial-value=’Below’ frame-of-reference=’RELATIVE’ />
<SR id=’4’ trajector=’2’ landmark=’4’ spatial-indicator=’4’ frame-of-reference=’INTRINSIC’ motion-indicator=’NIL’/>
<SR id=’4’ SRtype=’1’ general-type=’Region’ specific-type=’RCC8’ spatial-value= ’DC’ frame-of-reference=’INTRINSIC’ />
```

III: Sequential scene descriptions are linked descriptive phrases. After each description usually an *object focus shift* happens.

Example 9.

Behind the shops is a church, to the left of the church is the town hall, in front of the town hall is a fountain.

```
<TRAJECTOR id=’1’> church </TRAJECTOR>
<LANDMARK id=’1’ path=’ZERO’> shops </LANDMARK>
<SPATIAL-INDICATOR id=’1’ > behind </SPATIAL-INDICATOR>
<TRAJECTOR id=’2’ > town hall </TRAJECTOR>
<LANDMARK id=’2’ path=’ZERO’> church </LANDMARK>
<SPATIAL-INDICATOR id=’2’ > to the left of </SPATIAL-INDICATOR>
<TRAJECTOR id=’1’> fountain </TRAJECTOR>
<LANDMARK id=’2’ path=’ZERO’> town hall </LANDMARK>
<SPATIAL-INDICATOR id=’3’ > in front of </SPATIAL-INDICATOR>
<SR id=’1’ trajector=’1’ landmark=’1’ spatial-indicator=’1’ frame-of-reference=’INTRINSIC’ motion-indicator=’NIL’/>
<SR id=’1’ SRtype=’1’ general-type=’Direction’ specific-type=’Relative’ spatial-value=’Behind’ frame-of-reference=’INTRINSIC’ />
<SR id=’2’ trajector=’2’ landmark=’2’ spatial-indicator=’2’ frame-of-reference=’INTRINSIC’ motion-indicator=’NIL’/>
<SR id=’2’ SRtype=’1’ general-type=’Direction’ specific-type=’Relative’ spatial-value=’Left’ frame-of-reference=’INTRINSIC’ />
<SR id=’3’ trajector=’3’ landmark=’3’ spatial-indicator=’3’ motion-indicator=’NIL’/>
<SR id=’3’ SRtype=’1’ general-type=’Direction’ specific-type=’Relative’ spatial-value=’Front’ frame-of-reference=’RELATIVE’/>
```

In addition to the complex descriptions mentioned in [3], the following examples show some additional special characteristics. The next example contains one indicator *for* for two relations.

Example 10.

John left Boston for New York.
<TRAJECTOR id=’1’> John </TRAJECTOR>
<LANDMARK id=’1’ path=’BEGIN’>Boston </LANDMARK >
<LANDMARK id=’2’ path=’END’> New York </LANDMARK >
<SPATIAL-INDICATOR id=’1’ > for </SPATIAL-INDICATOR>
<MOTION-INDICATOR id=’1’> left </MOTION-INDICATOR >
<SR id=’1’ trajector=’1’ landmark=’1’ spatial-indicator=’NIL’ motion-indicator=’1 />
<SR id=’1’ SRtype id=’1’ general-type=’Direction’ specific-type=’Relative’ spatial-value=’NTPP’ frame-of-reference=’ABSOLUTE’ />
<SR id=’2’ trajector=’1’ landmark=’2’ spatial-indicator=’1’ motion-indicator=’1’ />
<SR id=’2’ SRtype=’1’ general-type=’Direction’ specific-type=’Relative’ spatial-value=’NTPP’ frame-of-reference=’ABSOLUTE’/>

In example 11 the focus shift is ambiguous. The phrase *on the left* can refer to the door or to the table. If more information is available (for example in a multimodal context other information could come from video input) then we could estimate the likeliness of each alternative. In general, if an annotator is not sure about the reference then all the true relations are added. For machine learning purposes, this is still a correct annotation because no additional inference is performed and both meanings can be extracted for the same sentence.

Example 11.

The table is behind the door on the left.
<TRAJECTOR id=’1’>The table </TRAJECTOR >
<LANDMARK id=’1’ path=’ZERO’>the door </LANDMARK >
<SPATIAL-INDICATOR id=’1’ > behind </SPATIAL-INDICATOR >
<SPATIAL-INDICATOR id=’2’ > on the left </SPATIAL-INDICATOR >
<SR id=’1’ trajector=’1’ landmark=’1’ spatial-indicator=’1’ motion-indicator=’NIL’/>
<SR id=’1’ SRtype=’1’ general-type=’Direction’ specific-type=’Relative’ spatial-value=’BEHIND’ frame-of-reference=’RELATIVE’ motion-indicator=’NIL’/>
<SR id=’2’ trajector=’1’ landmark=’NIL’ spatial-indicator=’2’ frame-of-reference=’RELATIVE’ motion-indicator=’NIL’/>
<SR id=’2’ SRtype=’1’ general-type=’Direction’ specific-type=’Relative’ spatial-value=’LEFT’ frame-of-reference=’RELATIVE’ />
<TRAJECTOR id=’2’ >The door </TRAJECTOR >
<SR id=’3’ trajector=’2’ landmark=’NIL’ spatial-indicator=’2’ frame-of-reference=’RELATIVE’ motion-indicator=’NIL’ />
<SR id=’3’ SRtype=’1’ general-type=’Direction’ specific-type=’Relative’ spatial-value=’LEFT’ frame-of-reference=’RELATIVE’ />

In example 12, there are one trajector, three landmarks and three indicators. The landmarks are geographically related, but the annotators should not use their background about this geographical information.

Example 12.

He drives within New England from Boston to New York.

```
<TRAJECTOR id=’1’ > He </TRAJECTOR >
<LANDMARK id=’1’ path= ’ZERO’> New England <LANDMARK >
<LANDMARK id=’2’ path=’BEGIN’> Boston </LANDMARK >
<LANDMARK id=’3’ path=’END’> New York </LANDMARK >
<SPATIAL-INDICATOR id=’1’ > within </SPATIAL-INDICATOR>
<SPATIAL-INDICATOR id=’2’ > from </SPATIAL-INDICATOR>
<SPATIAL-INDICATOR id=’3’ > to  </SPATIAL-INDICATOR >
<MOTION-INDICATOR id=’1’> drives </MOTION-INDICATOR>
<SR id=’1’ trajector=’1’ landmark=’1’ spatial-indicator=’1’ motion-indicator=’1’/>
<SR id=’1’ SRtype=’1’ general-type=’Region’ specific-type=’RCC8’ spatial-value=’NTPP’ frame-of-reference=’ABSOLUTE’/>
<SR id=’2’ trajector=’1’ landmark=’2’ spatial-indicator=’2’ motion-indicator=’1’ />
<SR id=’2’ SRtype=’1’ general-type=’Region’ specific-type=’RCC8’ spatial-value=’NTPP’ frame-of-reference=’ABSOLUTE’/>
<SR id=’3’ trajector=’1’ landmark=’2’ spatial-indicator=’3’ motion-indicator=’1’/>
<SR id=’3’ SRtype=’1’ general-type=’Region’ specific-type=’RCC8’ spatial-value=’NTPP’ frame-of-reference=’ABSOLUTE’/>
```

Another possibility is having one indicator but with various roles. In example 13, "cross" is a motion indicator and also spatial indicator.

Example 13.

The car crosses the street.

To map the relations to formal representations, the ontology of the objects and also shape information about the objects are necessary for the machine to learn from. We do not discuss these issues here further, but just show two examples.

Example 14.

The room is at the back of the school.
The tree is at the back of the school.

In the first sentence the semantics of the spatial indicator *at the back of* is about an interior region of the school whereas in the second sentence it is about an exterior region.

3.2.4 Adding a Temporal Dimension

In the suggested scheme for each relation a time dimension can be easily added. Temporal analysis of sentences can be combined with spatial analysis to assign a value to the temporal dimension of each relation and the interpretation is the time instant at which the spatial relation holds. Looking back to example 10, in the first spatial relation, the temporal dimension is related to *yesterday*.

Example 16.

John left Boston for New York yesterday.

```
<TIME-INDICATOR id=’1’> yesterday </TIME-INDICATOR >
<SR id=’1’ trajector=’1’ landmark=’1’ spatial-indicator=’1’ motion-indicator=’1’
frame-of-reference=’ABSOLUTE’ time-indicator=’1’/>
```

The analysis of temporal expressions could be done separately and only the time-indicator attribute is added to related spatial relations.

3.3 Data Resources

We performed a broad investigation to find possible data resources to be used as training data by supervised machine learning models for the extraction of spatial information. As, to our knowledge, such data were not publicly available so far, we have built a corpus, based on the aforementioned annotation scheme we refer to it as CLEF which is used as a benchmark for the **SemEval-2012 shared task**. The main experimental investigations in the thesis are over editions of this corpus. In addition to the main corpora we annotated very small datasets from different domains and used these in cross domain evaluations in Chapter 5. We also point to a few datasets which were indirectly relevant for our task. The detailed information is given in the following sections and the relevant statistics are provided in Tables 3.2 and 3.3.

3.3.1 Corpus Collection

The main annotated corpus for the whole scheme is a subset of **IAPR TC-12 image Benchmark** [50] referred as **CLEF**. It contains 613 text files that include 1213 sentences in total. The original corpus was available without copyright restrictions. The corpus contains 20, 000 images taken by tourists with textual descriptions in up to three languages (English, German and Spanish). The texts describe objects, and their absolute and relative positions in the image. This makes the corpus a rich resource for spatial information. However the descriptions are not always limited to spatial information. Therefore they are less domain-specific and contain free explanations about the images. An essential property of this corpus is not only that it contains a large enough number of spatial language texts for learning, but also that it has additional (non-linguistic) spatial information, i.e. images, from which a qualitative spatial model can be built that can be related to the textual information. Hence, an additional advantage of this dataset is providing the possibility for further research on combining spatial information from vision and language. The first column in table 3.2 shows the detailed statistics about the spatial roles in this data. The average length of the sentences in this data is about 15 words including punctuation marks with a standard deviation of 8. The textual descriptions have been indexed and annotated with the spatial roles of trajector, landmark, and their corresponding spatial indicator. Separate roles are assigned to phrases and the head words of the phrases. Both verbs and verb phrases are indexed and annotated, particularly when they participate in forming the spatial configurations, and this is mostly the case for dynamic spatial relations. Each sentence with a spatial relation is additionally annotated as DYNAMIC or STATIC, and each spatial relation is annotated with a GUM-Space modality which are not used in this thesis. Moreover, every sentence with a spatial relation is annotated with the aforementioned formal semantics in Section 3.2.

At the starting point two annotators including the author and another non-expert annotator, annotated 325 sentences for the spatial roles and relations. The goal was to realize the disagreement points and prepare a set of instructions in a way to achieve highest-possible agreement. From the first effort an inter-annotator agreement of 0.89 for Cohen's kappa was obtained [17]. This very first version of annotations is used in the experiments in Chapter 5. We refer to it as **SemEval-0** version. We continued with a third annotator for the remaining 888 sentences. The annotator had an explanatory session and received a set of instructions and previously annotated examples as a guidance to obtain consistent annotations. This version is referred to as **SemEval-2012** version and is used as a benchmark in the workshop with this name. The data has a minor revision in its latest edition and is enriched with the QSR annotations. This version is referred to as **SemEval-1**. In SemEval-1 for the directional relations such as *on the left*, the landmark is assumed to be implicit while the word *left* was annotated as landmark in the previous versions. Such expressions, in fact, express *left* of some implicit object depending on the frame of reference. This edition is used and reported on, in Chapter 8.

	CLEF	**GUM** (Maptask)	**Fables**	**DCP**
#Sentences	1213	100	289	250
#Spatial relations	1706	112	121	222
#Trajectors	1593	65	106	199
#Landmarks	1462	69	95	188
#Spatial indicators	1468	112	121	222
#nonSpatial prepositions	695	10	743	587

Table 3.2: Data statistics on the occurrence of spatial components in different corpora; The CLEF corpus is used for SemEval-2012.

Spatial relations	1706

Topological	EQ	DC	EC	PO	PP
1040	6	142	462	15	417

Directional	BELOW	LEFT	RIGHT	BEHIND	FRONT	ABOVE
639	18	159	103	101	185	71

Distal
82

Table 3.3: Data statistics of the QSR additional annotations on SemEval-2012, referred to as SemEval-1.

The statistics about formal spatial semantics of the relations are shown in table 3.3. In the current corpus only 50 examples are annotated with more than one general spatial type. For example, *"next to"* is annotated as a topological relation DC in terms of RCC-8 and as a distance relation CLOSE in terms of a relative distance calculus:

(1) *Two people are sitting next to her.*

```
trajector: people
landmark: her
spatial-indicator: next to
general-type: region/distance
specific-type: RCC-8 / relative-distance
spatial-value: DC / close
path: none
frame-of-reference: none
```

2D vs. 3D annotations. Although the textual data used is accompanied by images, the qualitative spatial annotation for CLEF was based on the text itself.

This was done to focus on information that can actually be extracted from the language itself. Nevertheless, human imagination about a described scene can interfere with the textual description, which has resulted in some variations. As an example, take the following sentence and its annotation:

(2) *Bushes and small trees (are) on the hill.*

```
trajector: bushes
landmark: the hill
spatial-indicator: on
general-type: region
specific-type: RCC-8
spatial-value: EC
path: none
frame-of-reference: none
```

This 3-D projection of the description of a 2-D image is annotated as externally connected. In the 2-D image, however, a partial overlap may also be adequate. In contrast, a 2-D map (with an allocentric perspective) of the described scene would lead to a non-tangential proper part annotation. This example illustrates what we have said before; that RCC-8 alone is – quite naturally – not enough to capture adequately all necessary spatial information, and that in a more general approach additional (and combinations of) qualitative spatial calculi have to be used.

Dynamic vs. static annotations. In the CLEF data set 25 of the relations are annotated as DYNAMIC, the others as STATIC. If a dynamic situation is annotated with a (static) RCC-8 relation, the qualitative relation can be regarded as a snapshot of the situation. This is shown in the following example:

(3) *People are crossing the street.*

```
trajector: people
landmark: road
spatial-indicator: crossing
general-type: region / direction
specific-type: RCC-8 / undefined
spatial-value: EC / undefined,
path: middle
frame-of-reference: none
```

Hence, the annotations refer to time slices for the (linguistic) explanation of the (static) image. This allows a mapping from dynamic descriptions to (static) RCC-8 relations mainly by including the path feature and the relative situation of the trajector with respect to an imaginary path related to the landmark. Allowing RCC-8 annotations for dynamic descriptions is also supported by the conceptual neighborhood graphs [43]; Every topological change, i.e. movements of regions with respect to each other and their changing relations, can be split into a sequence of adjacent RCC-8 relations according to the neighborhood graph [65]. The annotated RCC-8 relation thus reflects one relation out of this sequence, i.e. one moment in time of the topological change. However, we may not predict if the annotations refer to a time slice that reflects the start, intermediate, or end point of the path or the motion process. For instance, it is shown that linguistic expressions seem to focus primarily on the end point of the motion [116].

3.3.2 Other Linguistic Resources

In this part we briefly point to the other relevant resources for spatial information extraction from language, which we used to perform some additional experiments.

- **TPP dataset** Since the spatial indicators are mostly prepositions, the preposition sense disambiguation is an important relevant task to our problem. Fortunately, for this specific task, there is a standard test and training data provided by the SemEval-2007 challenge [85]. It contains 34 separate XML files, one for each preposition, totaling over 25,000 instances with 16,557 training and 8,096 test example sentences; each sentence contains one example of the respective preposition.

- **GUM (Maptask) dataset** Another relevant small corpus is the general upper model (GUM) evaluation data [6], comprising a subset of a well-known Maptask corpus for spatial language. It has been used to validate the expressivity of spatial annotations in the GUM ontology. Currently, the dataset contains more than 300 English and 300 German examples. We used 100 English samples in the GUM (Maptask) corpus. The following example shows the GUM-annotation for one sentence:

  (4) *The destination is beneath the start.*

  ```
  SpatialLocating(locatum:destination,
  process:being,placement: GL1
  (relatum:start,
  hasSpatialModality:UnderProjectionExternal)).
  ```

Here, *relatum* and *locatum* are alternative terms for landmark and trajector. *Spatial modality* is the spatial relation mentioned in the specific spatial ontology. Although complete phrases are annotated in this dataset, we only use a phrase's headword with trajector (**tr**) and landmark (**lm**) labels and their spatial indicator (**sp**). Using this small corpus to evaluate our approach for a very domain-specific corpus, including only instructions and guidance for finding the way on a map, is beneficial.

- **DCP dataset** The dataset contains a random selection from the website of *The Degree Confluence Project.*[1] This project seeks to map all possible latitude-longitude intersections on earth and have people who visit these intersections provide written narratives of the visit. The main textual parts of randomly selected pages are manually copied, and up to 250 sentences are annotated. Approximately 30% of the prepositions are spatial. This percentage represents the proportion of spatial clauses in the text. The webpages of this dataset are similar to travelers' weblogs but include more precise geographical information. The richness of this data enables broader applicability for future applications. Compared to CLEF, this dataset includes less spatial information, and the type of text is narrative rather than descriptive. It also contains more free (unrestricted) text. Moreover, the spatio-temporal information contained in this data has recently been used to extract discourse relations [57].

- **Fables dataset** This dataset contains 59 randomly selected fable stories[2], which have been used for data-driven story generation [93]. The dataset contains a wide scope of vocabulary and only 15% of the prepositions have a spatial meaning, making it the most difficult corpus for our system. We annotated 289 sentences from this corpus for cross-domain experiments.

There is another small dataset about *Room descriptions* prepared by Tenbrick et al. in [135]. We had a limited access to 124 sentences of this corpus that contains directional and topological descriptions for an automatic wheelchair about the objects in a room. The full dataset which contains pictures of the room can help preparing multimodal analyses which is not the focus of this thesis.

[1] http://confluence.org/

[2] http://homepages.inf.ed.ac.uk/s0233364/McIntyreLapata09/

3.4 Related Work

In recent cognitive and linguistic research on spatial information and natural language, several annotation schemes have been proposed such as ACE[3], GUM[4], GML[5], KML[6], TRML[7] which are described and compared to the SpatialML scheme in [88]. The most systematic work in this area regards the SpatialML [89] scheme which focuses on geographical information. SpatialML uses PLACE tags to identify geographical features. SIGNAL, RLINK and LINK tags are defined to identify the directional and topological spatial relations between a pair of locations. Topological spatial relations in SpatialML are also connected to RCC8 relations. However, SpatialML considers static spatial relations and focuses on geographical domains. The corpus which is provided along with SpatialML scheme contains rich annotation for toponymy but not for learning spatial links and especially links between arbitrary objects. GUM, also aims at organizing spatial concepts that appear in natural language from an ontological point of view. The formulated concepts are very expressive, but the ontology is large and more fine-grained than what could be effectively learnable from a rather small corpus. An interesting new XML scheme based on SpatialML and GUM was proposed in [134], targeting spatial relations in the Chinese language. It also deals with geographical information and defines two main tags of geographical entity and spatial expression. In [112], a spatio-temporal markup language for the annotation of motion predicates in text informed by a lexical semantic classification of the motion verbs, is proposed. The interesting point is that the proposed scheme seems suitable for tagging dynamic spatial relations, based on motions in space and time. However, the focus is on motion verbs and their spatial effects and not on spatial language in general. There is another spatial annotation scheme proposed in [112] in which the pivot of the spatial information is the spatial verb. The most recent and active research work regards the ISO-Space scheme [113] which is based on this last scheme and SpatialML. The ISO-Space considers detailed and fine-grained spatial and linguistic elements, particularly motion verb frames. The detailed semantic granularity considered there, makes the preparation of the data for machine learning more expensive, and there is no available data for machine learning annotated according to that scheme yet. Our proposed scheme is closely related to the SpatialML scheme, but more domain independent considering more universal spatial primitives and cognitive aspects. It is relevant to the ISO-Space scheme but the pivot of the relation is not necessarily the verb, and a general notion of spatial indicator is

[3] Automatic content extraction

[4] General upper model

[5] Geography markup language

[6] Keyhole markup language

[7] Toponym resolution markup language

used as the pivot of each spatial configuration. Spatial information is directly related to the part of language that can be visualized. Thus, the extraction of spatial information is useful for multimodal environments. One advantage of our proposed scheme is that it considers this dimension. Because it abstracts the spatial elements that could be aligned with the objects in images/videos, it can be used for annotation of audio-visual descriptions as shown in [15]. Our scheme is also useful in other multimodal environments where, for example, natural language instructions are given to a robot for finding the way or objects.

There are a few sparse efforts towards creating annotated data sets for extraction of some limited elements of our scheme. For example in [83] the Chinese version of Aesops Fables has been labeled in terms of trajector, landmark and spatial expressions and turned into an evaluation database for the extraction of spatial relations. It has been applied in a very limited machine learning setting, only a binary classifier was used so far for the extraction of the trajector. In [134] texts from a Chinese encyclopedia concerning geographical information is annotated using the XML scheme we have mentioned. GUM also is accompanied by an evaluation corpus containing a limited set of 600 sentences in German and English. It should be mentioned that from the linguistic point of view, FrameNet frames [41] are a useful linguistic resource which can be very helpful for identifying spatial components in the sentence. Spatial relations can be seen, to some extent, as a part of the frame-based semantic annotation. There are various semantic frames which are related to spatial roles and semantics. Frames like LOCATIVE RELATION, SELFMOTION, PERCEPTION, BEING LOCATED seem most related to spatial semantics. Hence, using these semantic frames requires making a connection between the general spatial representation scheme and the specific frames that could be related to each word. Therefore defining a tag set is important to have a unified spatial semantic frame for spatial semantics and to integrate partial annotations that tend to be distributed over different layers [77]. Towards this direction a corpus is annotated (in German) for walking directions [133]. The preprocessed texts are annotated on the following three levels: *pos lemma* (part-of-speech and lemma), *syn dep* (dependency relations) and *sem frame* (frames and semantic roles). For tagging walking directions on the semantic frame level, annotation was carried out using FrameNet frames. However, the available resources and corpora are very limited for a broad machine learning research on this area, hence we provide an annotated dataset according to the proposed scheme which we described in this chapter and which has been used as the first benchmark for spatial information extraction from natural language in SemEval2012.[8]

[8] http://www.cs.york.ac.uk/semeval-2012/task3/

3.5 Conclusion

The first contribution of this chapter is proposing a spatial annotation scheme on the basis of related research. The advantages of the proposed scheme compared to other existing schemes are: a) It is based on the concepts of two layers of cognitive spatial semantics and formal spatial representation models; b) This scheme is domain-independent and useful for real world applications and it is rather flexible to be extended in both mentioned layers to cover all aspects of spatial information; c) It is easily applicable for annotating spatial concepts in image data and multimodal settings; d) It supports static as well as dynamic spatial relations; e) Using multiple formal semantic assignments it bridges the gap between the natural language spatial semantics and formal spatial representation models.

For each of the cognitive and formal semantic aspects, we exploit the most commonly accepted concepts and their formalizations to establish an agreeable setting for spatial information extraction. Extraction of the spatial information accruing to this scheme facilitates automatic spatial reasoning based on linguistic information.

The second contribution of this chapter regards corpora preparation according to the proposed scheme and assessing the available resources for the goal of this thesis which is a machine learning investigation for spatial information extraction from natural language. The noticeable points about the selected data are: a) It is free text about various topics containing spatial and non spatial information; b) The descriptions are related to photographs implying that they contain rich spatial information; c) Having the aligned photographs in a spatially annotated corpus provides a potential for learning in multimodal settings combining information from both language and vision and also grounding language in perception in further research. A part of our prepared data has been used as a benchmark in SemEval-2012 shared task on *spatial role labeling* [69] and it is available for follow up research in this field. Providing such a benchmark is an important step towards persuasion and progress on spatial information extraction as a formal computational linguistic task and spotting the practical problems towards enriching both the corpora and the proposed task which is hard to achieve without any practice.

The two important semantic layers of the scheme and the formal task of their recognition are discussed in detail in Chapter 4 of the thesis.

Chapter 4

Task Definition: from Language to Spatial Ontologies

In this chapter we define the main framework for mapping natural language to spatial ontologies. Having a tendency for being pragmatic yet our proposed framework is based on the theoretical cognitive and linguistic foundations as well as on the cognitively adequate formal spatial models. These theoretical foundations were described in Chapter 3. The task is formulated as an ontology population to be performed via machine learning models. We aim at learning to assign the segments in the sentence to the concepts in the ontology. The considered concepts form an ontology based on the aforementioned spatial annotation scheme. We highlight the distinction between two *spatial role labeling* (SpRL) and *spatial qualitative labeling* (SpQL) layers in the ontology. We describe the structural characteristics of the two layered ontology to be exploited in the learning models.

The two layers of the semantics are explained in Section 1. We formulate the general machine learning task encompassing the two layers of SpRL and SpQL in Section 2. The constraints and features are described in Section 3. The methodology and the metrics that we employ for evaluation of all models are explained in Section 4. We conclude in Section 5.

4.1 Two Layers of Semantics

The gap between the semantics expressed in natural language and the formal semantics considered in spatial calculus models are discussed in Section 3.1.3. Due to this gap, learning how to map the spatial information in natural language onto a formal representation is a challenging problem. To overcome the complexity of this problem in a systematic way, our spatial scheme is divided in to two abstraction layers of cognitive-linguistic and formal models [7, 72, 70]:

1. A layer of **linguistic conceptual representation** called spatial role labeling (SpRL), which predicts the existence of spatial information at the sentence level by identifying the words that play a particular spatial role as well as their spatial relationship [74];
2. A layer of **formal semantic representation** called spatial qualitative labeling (SpQL), in which the spatial relation is described with semantic attribute values based on qualitative spatial representation models (QSR) [45, 71].

Establishing a connection between the two layers is a complex task. In spite of spatial calculi which focus on a single spatial aspect [118], spatial language often conveys multiple meanings within one expression [18]. In our conceptual model we argue that mapping the language to multiple spatial representation models could solve the problem of the existing gap to some extent. Because various formal representations capture the semantics from different angles, their combination covers various aspects of spatial semantics needed for locating the objects in the physical space. Hence, the SpQL has to contain multiple calculi models with a practically acceptable level of generality. Moreover, mapping to spatial calculi forms the most direct approach for automatic spatial reasoning compared to mapping to more flexible intermediated ontologies discussed in Section 3.1.3. However, we believe that this two layered model does not yield sufficient flexibility for ideal spatial language understanding. As in any other semantic tasks in natural language additional layers of *discourse* and *pragmatics* must be worked out, which is not the focus of this thesis.

4.2 Task Definition as Ontology Population

Our main task is to map a given sentence X composed of a number of words $x_1 \dots x_n$ to the predefined spatial ontology shown in Figure 4.1.a. The task is to label the words in the sentence with spatial roles (SpRL), detect the spatial

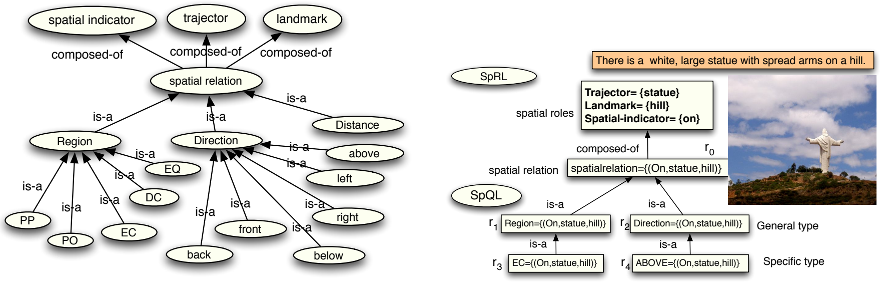


Figure 4.1: **(a)** Spatial ontology; **(b)** Example of the SemEval-1 benchmark.

relations, and label the spatial relations with their spatial semantics including the course-grained in addition to fine-grained semantic labels. The words can have multiple roles and the relations can have multiple semantic assignments. The labels are assigned according to the relationships and constraints that we discuss in the following sections. The considered spatial ontology here is only a lightweight [166] ontology, but pinpoints to the main challenges in the recognition of ontological label structures in text.

4.2.1 Spatial Role Labeling (SpRL)

In the *spatial role labeling* (SpRL) layer, the cognitive-linguistic spatial semantics are considered. Figure 4.1.b shows the sentence, *There is a white large statue with spread arms on the hill.*, which is labeled according to the nodes in the spatial ontology in Figure 4.1.a. In the SpRL step the goal is to identify the words that play a spatial role in the sentence and classify their roles, moreover to recognize the link between the spatial roles and extract the spatial relations. In the example sentence, we need to extract a spatial relationship signaled by *on* that holds between *statue* and *hill*. The word *statue* has the role of trajector (**tr**). The word *hill* has the role of landmark (**lm**). These two spatial entities are related by the spatial expression *on* that is the spatial indicator.

These spatial roles are the three main nodes in our ontology. We refer to these nodes as *single label*s. A *single label* refers to an independent concept in the ontology. The spatial configuration that we consider in the whole thesis considers the link between the three roles which is labeled as a *spatial relation*, also called a spatial triplet. We refer to these kind of nodes in the ontology as *linked label*s. *Linked label*s show the connection between the concepts in the ontology. For example here the spatial relation is a linked label that shows a **composed-of** relationship with the composing labels of spatial roles. There is one spatial relation in the above sentence, <*on*$_{\mathbf{sp}}$ *statue*$_{\mathbf{tr}}$ *hill*$_{\mathbf{lm}}$>. In general, there can be a number of spatial relations in each sentence. Although the spatial indicators are mostly prepositions, but in general the *sense* of the prepositions depends on the *context*. The first preposition *with* in the example sentence states the possession of the arms, so <*with statue arms*> is not a spatial relation.

The trajectors and landmarks can be implicit, meaning that there is no word in the sentence to represent them. In some linguistic spatial expressions, there is no need to express the spatial information based on any landmark [168]. In these cases we use the term *undefined* instead of the roles to keep the triplet representation consistent. For example, in the sentence *Come over here* where the trajector *you* is only implicitly present, the spatial triplet is represented as <*over*$_{\mathbf{sp}}$ *undefined*$_{\mathbf{tr}}$ *here*$_{\mathbf{lm}}$>. The other SpRL elements such as motion, path and frame of reference are considered only very marginally in this thesis due to the static descriptions in our main dataset and the lack of examples for these concepts.

Spatial relations can be inferred by spatial reasoning too. For instance, in the example of *The book is on the table behind the wall.* The spatial relations <*on*$_{\mathbf{sp}}$ *book*$_{\mathbf{tr}}$ *table*$_{\mathbf{lm}}$> and <*behind*$_{\mathbf{sp}}$ *table*$_{\mathbf{tr}}$ *wall*$_{\mathbf{lm}}$> are extracted directly from the sentence but the relation <*behind book wall*> can be inferred by spatial reasoning. Such inferred relations are not considered in this task because they make the semantic annotation of the data more difficult and less consistent.

4.2.2 Spatial Qualitative Labeling (SpQL)

In the *spatial qualitative labeling* (SpQL) layer, the goal is to map the entire spatial configuration that is extracted from the SpRL layer to a formal semantic representation. As we simplified the first layer by ignoring a number of concepts such as shape, size and motion, in this layer also a number of simplifications are considered for the sake of feasibility of the learning task given the available resources and data. Our representation of the spatial semantics is based on multiple spatial calculi [118, 18]. Figure 4.1.a, shows the semantics that are mostly considered in this thesis. The three general types of regional (i.e topological), directional and distal cover all coarse-grained aspects of space (ignoring shape and size) and qualitative spatial calculi are available for them. Henceforth, we map extracted cognitive linguistic elements to multiple qualitative representations including these three categories.

We assume that trajector and landmark are 'interpreted' as two regions. As a consequence, we can map static as well as dynamic spatial expressions, although dynamicity is not directly covered in RCC (unless neighborhood graphs are used as a form of spatial change over time). Since the mapping can point to relations from different calculi (e.g., RCC and an orientation calculus), this is more suited to achieve the required level of expressivity for a spatial expression. Given multiple representations over the linguistic spatial information, qualitative (even probabilistic) spatial reasoning will be feasible over the produced output. The learned relations could be considered as probabilistic constraints about most probable locations of the entities in the text.

As can be seen in Figure 4.1.a., the fine-grained semantics are considered according to the spatial scheme described in the last chapter. However, for mapping to topological relations in RCC-8, we observe that in the textual descriptions, mostly salient objects are chosen as landmarks. Hence, the inverse proper part relations occur less frequently (cf. [65]). This motivates that we combine all variations of proper part including *{TPP,NTPP,TPPI, NTPPI}* into one class *{PP}*, resulting in what we call RCC-mod. In other words we used five categories related to topological relations in our ontology. With regard to the directional relationships, since our data contains image descriptions, relative directions occur in the linguistic captions rather than the cardinal and absolute directions. Hence we consider learning relative directional information in the ontology. For the distal information, since the corpus contains a small set of examples with distal information, a general class of distance is used in the applied spatial ontology.

In the example of Figure 4.1a., spatial roles compose the relation (on,statue,hill) and then this relation is labeled with *regional* and *EC* together with *directional* and *above*. This sentence has no distal information.

4.3 Constraints and Features for the Machine Learning Models

As described in Chapter 2 about the spatial language, the lexical, syntactic and semantic features of language can help the extraction of spatial semantics. There is also linguistic and commonsense background knowledge on the spatial language to be exploited when designing an intelligent model for automatic spatial semantic extraction. In this section we aim to specify all types of information that can be useful for the machine learning models that we design through this thesis. We divide these characteristics in two categories of *Features* and *Constraints*. The benefit of the distinctions between these two types of information is extensively discussed in [20]. Generally, constraints capture global/structural characteristics of the problem while the features capture local properties. Considering global properties during training is more complex than considering local properties. The complexity is due to the involvement of complex correlations between variables. Moreover, learning models need a large number of training examples in order to capture the global properties of the problem. Global characteristics also can be modeled in the form of features, but separating them from the features helps the models to treat them more efficiently. For example, global constraints are considered sometimes only during prediction time rather than during the training as in CCM models.

4.3.1 Constraints

The spatial language has a number of structural characteristics that we exploit for automatic extraction of spatial roles and relations. These global constraints should hold among the predicted output labels by the models. For instance:

- Each spatial relation is composed of three spatial roles that some of them can be undefined. It implies the following constraints that are referred to as *composed-of constraints*.
    - If there is a trajector or landmark in the sentence then there is a spatial indicator too.
    - If a spatial indicator is detected in the sentence then we impose the extraction of a spatial triplet, possibly with undefined roles.
- Avoiding spatial reasoning imposes the following structural properties, also referred to as *spatial reasoning constraint*: each word with a fixed spatial role is only connected to one indicator. Note that a same word can be connected to a different spatial indicator while having a different spatial role.
- An object can play only one role of trajector or landmark with respect to one specific spatial indicator. We refer to this property as *multilabel constraint.*
- The number of spatial relations in a sentence can be restricted. This restriction can be set according to the statistics in the annotated data and proportional to the length of the sentence. We refer to this type of constraint as a *counting constraint.*

The above mentioned constraints are about the SpRL layer of the ontology and the sentence level characteristics of the spatial language. We refer to the SpRL layer constraints as *horizontal constraints.* Considering the ontological structure of the nodes in the SpQL layer, we introduce the following constraints:

- If an *is-a* relationship holds between two semantic nodes (labels) l_1 and l_2 in the ontology, that is l_1 is-a l_2, then each instance of l_1 should be an instance of l_2. We refer to these as *is-a constraints*.

- Each arbitrary triplet can be assigned zero or more fine-grained semantic labels. If the triplet is recognized as spatial, then there should be at least one fine-grained semantic label assigned to it. We refer to these as *null-assignment constraints*.

- Given a general type-label, each spatial relation is associated to only one fine-grained label under that type. This property is due to the mutual exclusivity of the formal spatial semantics. We refer to these as *mutual-exclusivity constraints*.

These last three type of constraints relate to the structure of the in-depth and fine-grained spatial semantics and properties of the formal models, hence we refer to these as *vertical constraints*.

In addition to the above mentioned constraints between the *predicted outputs*, there are often constraints that can be employed without knowing the output labels, but which are defined according to expert knowledge and can be integrated in a *preprocessing* phase. We refer to these type of constraints as *input constraints*. Employing the input constraints technically is trivial but their application is important for the feasibility of the training in many structured learning problems. To clarify this we can consider the formulated ontology population problem in which usually there are a large number of possibilities for assigning the components in the input to the concepts in the output. In this problem, it is always beneficial to prune possibilities using relevant background knowledge. For example, we know that in the English language, the spatial indicators are mostly prepositions and can be selected based on the POS tags. They are mostly tagged as IN and TO by parsers. Since prepositions belong to a closed lexical category, we also could collect a lexicon for prepositions according to our corpus.
Trajectors are mostly singular (NNS) or plural nouns (NN), or the words that are labeled as subject (SBJ) in the dependency tree. The landmarks also are mostly singular, plural or proper nouns (PRN). These linguistic features are extracted by a syntactic and dependency parser. As discussed before, spatial roles can be *undefined* too. These kind of constraints are usually used before passing the inputs to the training and prediction models to reduce the output search space. We formalize these properties when they are used in our designed models according to the models' parameters in Chapters 8, Chapter 6 and partially in Chapter 5.

4.3.2 Features

In addition to the global characteristics of spatial language there are a number of linguistically motivated features that we use to train our learning models. We use natural language processing tools to process the sentences and extract these features. These *input features* can describe single words or even composed components of a sentence like pairs of words or phrases. First we describe the input features related to single words in the context of a sentence. We assume that words are indexed based on their sequential position in the sentence. We call the features that are assigned to a single word, *local features*. Although for some local features the context of a word in the sentence has to be considered, we use the term local as far as the features are assigned as a property of one single word. The following local features are assigned to each single input component (word). We refer to the input features by the ϕ symbol, and each class of features is indexed by a relevant name, $\phi_{local}(x_i)$.

- **Word-form** $\phi_{wf}(x_i)$: lexical information is indicative and important for spatial semantics. So the word form itself is used as a feature. For example, the two expressions, *the meeting in the afternoon* and *the meeting in the office*, differ only in one word, but the term *afternoon* can not be a landmark, while the term *office* can be.

- **Part-of-speech tag** $\phi_{pos}(x_i)$: POS tags are informative features for recognizing spatial roles, e.g. the IN tag (i.e preposition) gives a higher chance to a word for being a spatial indicator, and the NN tag (i.e. noun) gives a higher chance for being a trajector or landmark. Therefore, POS tags can be used as distinguishing features for training a model. For example, *He is on the left.* compared to *He left you.*, the two similar word forms *left* can be distinguished using their part of speech tags. *left* in the first sentence is a noun (NN) and has a spatial meaning, but in the second it is a verb (VB) and does not carry any physical locative information.

- **Semantic role** $\phi_{srl}(x_i)$: since semantic role models are learned from large corpora, they can provide information for spatial role labeling. For example, a preposition which is labeled as locative (LOC) by a semantic role labeler has a high probability of being a spatial indicator.

- **Dependency relation** $\phi_{dprl}(x_i)$: the assigned labels to the words by the dependency parsers are useful for spatial role labeling. For example, the SBJ label meaning that a word is the subject of the sentence gives a higher chance to that word for being a trajector. Or the label of NMOD, meaning that a word is a noun modifier, often indicates that the word does not carry any spatial role. The relations represented by the dependency trees could be directly exploited in finding the spatial links too. We exploit only the dependency labels for the sake of reduced complexity.

- **Subcategorization** $\phi_{sub}(x_i)$: subcategorization shows the sister nodes of the parent of a word in the parse tree. For example, for the preposition *on* in the phrase *on your left*, this feature is IN-NP. This provides information about the structural context of each word. Given this feature the learning model will know, for instance, whether a specific preposition is a part of a prepositional noun phrase or of a verb phrase. Hence the learning model can learn the frequent contextual patterns of the spatial prepositions.

- **Spatial context of the preposition** $\phi_{spc}(x_i)$: in spite of the verbs and nouns, spatial terms and prepositions are usually a closed lexical class of words in many languages. Hence a list of terms such as directions *left, right* and other spatial terms can be easily collected. Here, for each spatial indicator the existence of a spatial term in its neighborhood, is used as a feature. This feature helps to recognize the spatial sense of the prepositions which are in a spatial context. Moreover, it helps to detect the undefined landmarks. Normally, the undefined landmarks occur in the context of the directional phrases such as *on the left*. In these cases the landmark is implicit and depends on the spatial frame of reference. However if a directional phrase is followed by *of* then it means the phrase attributes an explicit landmark, for example in the phrase *on the left of the room*. In other words, if the spatial context contains the second preposition *of* then the possibility of having an explicit landmark is higher. So we define a feature with three dimensions to distinguish whether a spatial indicator has a spatial context with a second preposition, whether it has a spatial context without a second preposition or no spatial context at all.

- **Undefined** $\phi_{und}(x_i)$: to stress that a word is a dummy one a binary feature is used that indicates this. This feature is used in some models for technical ease.

In addition to the local features, we use a number of relational pairwise features, between the words including,

- **path** $\phi_{path}(x_i, x_j)$: the parse tree path between a word x_i that is a candidate of trajector/landmark and a word x_j that is a candidate for spatial indicator.

- **before** $\phi_{before}(x_i, x_j)$: this feature indicates whether the position of a word which is a candidate for trajector/landmark is before a word which is a candidate for spatial indicator. Also whether the position of the candidate trajector is before the candidate landmark.

- **distance** $\phi_{dis}(x_i, x_j)$: the relative distance between trajector/lanmark candidate x_i and the candidate spatial indicator x_j is defined as,

$$distance = \frac{\#\text{Nodes on the path between } x_i \text{ and } x_j}{\#\text{Nodes in the parse tree}},$$

  and the integer value of the inverted distance is used as a nominal feature.

All the above mentioned features which are mostly nominal, are turned into binary vectors. If there is no preprocessing and candidate selection phase for the roles, the relational features should be computed for all possible pairs of words in the sentence. We find it useful to make a distinction between *relational features* and *contextual features*. A *contextual feature* is a feature which in spite of not being local, is assigned to a single component without mentioning the other related component(s). For example the above mentioned *spatial context* is such a feature since it is a binary feature that indicates whether a spatial term exists in the neighborhood of a preposition, but there is no explicit relation referring to the identifiers of its neighborhood.

For the extraction of the linguistic features we use the LTH[1] tool that produces features in the CoNLL-08 format[2]. In the statistical machine learning models, features are usually represented in a feature vector and applied for training as well as prediction. However, in the structured machine learning models that we consider in this thesis, not only the vectors of features but also the structural constraints of spatial language and spatial semantics need to be taken into account. These issues will be discussed in the later chapters where models are proposed for performing the task we described in this chapter.

4.4 Evaluation Methodology

In this thesis we use classic machine learning evaluation metrics. The evaluations are based on 10-fold cross validation. In all the designed learning models the evaluations are provided per each node in the ontology if it is relevant and predicted for that specific model. More precisely, we evaluate the predictions for the single labels including the nodes trajectors, landmarks, spatial-indicators, for the linked labels including pairs of spatial indicator-trajector and spatial indicator-landmark, for the linked label of spatial relation and, for the semantic

[1] http://barbar.cs.lth.se:8081/

[2] http://barcelona.research.yahoo.net/dokuwiki/doku.php?id=conll2008:format

type linked labels in the ontology such as region, etc. The evaluation metrics of precision, recall and F1-measure are used which are defined as:

$$recall = \frac{TP}{TP+FN}, \quad precision = \frac{TP}{TP+FP}, \quad F1 = \frac{2 * recall * precision}{(recall + precision)}$$

where,

TP = the number of predicted components that exactly match the ground truth,
FP = the number of predicted components that do not match the ground truth,
FN = the number of ground truth components that do not match the predicted components.

In the evaluation of linked labels, a prediction is true when all the composing single labels are according to the ground truth. These values are counted per test sentence and are summed up over all the sentences in the test set for each fold. The precision, recall and F1 are calculated for each fold separately and afterwards averaged over the 10-folds (i.e. macro averaging) per label. The evaluations will be reported based on the performance of models over the two layers. For the SpRL layer each label and linked label are reported separately. For the SpQL layer each linked label is evaluated separately and then the weighted averages over all SpQL linked labels in the ontology are reported. Because the number of examples is highly variable among SpQL linked labels, each metric value for an SpQL linked label is weighted with the proportion of its examples when calculating the final value of each evaluation metric (micro-averaging).

4.5 Conclusion

The contribution of this chapter is its novel view on the spatial information extraction as an ontology population task. The considered concepts form a *lightweight spatial ontology* based on the aforementioned spatial annotation scheme and are adapted according to the corpus analysis and statistics in the data. We highlighted the distinction between two *spatial role labeling* (SpRL) and *spatial qualitative labeling* (SpQL) layers in the ontology. Distinguishing between the two layers provides a modular and flexible platform for future extensions of the cognitive concepts and their formal representation. Moreover, we contributed to characterizing linguistically motivated local features and global structural and ontological properties of the spatial language that can be exploited in machine learning models.

According to the structural characteristics, ontological relations and the relational features of the data we conclude that relational and structured learning models are well fitted solutions for this problem. We need a model which is able to learn from annotated data, use relational features and can consider the relationships between the output variables in the spatial ontology encompassing the ontological relationships as background knowledge. The distinguishing feature of the formulated task is that it is a unique computational linguistic *semantic* task that comes along rich *structural* characteristics. This property makes this task interesting for structured machine learning models. Designing such models is the roadmap of the rest of this thesis which brings us further, to more contributions on this topic.

Part II

Spatial Role Labeling

Outline

In Part I of the thesis we defined a new framework and corpora for learning to map natural language to spatial ontologies. In addition, the motivation behind the proposed ontology and its two layers of semantics was described. In this part, we investigate the extraction of the first layer of the ontology which is called *spatial role labeling* (SpRL).

The main contribution of this part of the thesis is designing the first machine learning models for this newly introduced task and extensive experimental investigation of its feasibility and its computational challenges using structured and relational machine learning frameworks.

In Chapter 5 we use graphical models, particularly conditional random fields which are among the best performing approaches in various natural language processing tasks. We apply a variety of CRF models and combine the two tasks of spatial role labeling and preposition disambiguation for recognizing spatial prepositions for SpRL.

Given the relational nature of the SpRL problem using models that are able to work with a formal relational representation and learning is justified. In Chapter 6, we provide various formulations of relational learning for SpRL using the kLog framework. The advantage of kLog compared to other relational learning frameworks is that by using an entity relationship diagram it provides a clear first order representation of the data model and by using a relational logical language it provides the flexibility and ease for programming for different learning models. It facilitates relational feature engineering and exploiting background knowledge. kLog uses a powerful *convolution* kernel that helps using extensive contextual features.

The experimental results of both frameworks in the two chapters are promising for dealing with SpRL and provide us with an extensive analysis about the challenges in the problem itself and the advantages and the disadvantages of the proposed models in these frameworks. This part of the thesis has been published in the papers mentioned below.

Related publication

Kordjamshidi, P., van Otterlo, M., Moens, M. F. (2011). Spatial role labeling: Towards extraction of spatial relations from natural language. *ACM Transactions on Speech and Language Processing, 8 (3), article 4, 36 p.*

Kordjamshidi, P., Frasconi, P., van Otterlo, M., Moens, M. F., De Raedt, L. (2011). Spatial relation extraction using relational learning. *Latest Advances in Inductive Logic Programming. ILP.* Windsor Great Park, United Kingdom, 31 July-3rd August (pp. 1-6).

Kordjamshidi, P., Frasconi, P., van Otterlo, M., Moens, M. F., De Raedt, L. (2012). Relational learning for spatial relation extraction from natural language. In Muggleton, S. H. (Ed.), Tamaddoni-Nezhad, A. (Ed.), Lisi, F. (Ed.), *Inductive Logic Programming: LNCS, Vol. 7207. (pp. 204-220).* Berlin Heidelberg: Springer.

Chapter 5

Graphical Models

Given the general task defined in Chapter 4 and the prepared corpus described in Chapter 3, in this chapter the spatial role labeling layer for mapping to spatial ontologies is investigated extensively. The main contribution of this chapter is the first computational investigation of the spatial role labeling (SpRL) problem. In this investigation the discriminative graphical models known as conditional random fields (CRF) are applied. CRFs have been applied successfully in many natural language processing and information extraction tasks. In line with the general ontology population task and mapping the language to spatial ontologies in the previous chapter, here we highlight the SpRL layer and its value to be considered as an independent computational linguistic task for extraction of spatial semantics from language. Finer grained investigations in this chapter are: a) Modeling multi-sequence tagging for multiple role assignments and relation extraction; b) Using template based CRFs to model long distance dependencies for training the SpRL model; c) Exploiting the standard *preposition disambiguation* task through which additional knowledge about spatial prepositions is considered; b) Cross domain evaluation of the SpRL models; c) Formulating the types of errors and providing an extensive error analysis in extraction of such semantics.

In Section 5.1, we describe our view on the spatial role labeling task. In Section 5.2, we provide the formal problem definition and input output representation. In Section 5.3, we describe the linear-chain and skip-chain CRF models that we apply, in addition to the preposition disambiguation subtask. In Section 5.4, we present the experimental study including cross domain and cross model results and error analysis. In Section 5.6, we conclude.

5.1 Spatial Role Labeling

As an independent computational linguistic task, SpRL considers the extraction of generic spatial semantics from natural language. From the SpRL point of view, language is a medium to explain the space and the location of objects and their movements, therefore different lexical categories including nouns, prepositions, verbs and other categories can paly a role in this direction.

We define **spatial role labeling** as *the automatic labeling of words or phrases in sentences with a set of spatial roles.* The roles take part in one or more *spatial relations* expressed by the sentence. The *spatial indicator* (typically a preposition) establishes the type of spatial relation, and other constituents express the participants of the spatial relation (e.g., entities' locations). The following sentence is an example:

$$Give\ me\ the\ [gray\ \ book]_{\mathbf{tr}}\ [on]_{\mathbf{sp}}\ [the\ \ big\ \ table]_{\mathbf{lm}}.$$

Our spatial role set consists of *trajector* (**tr**), *landmark* (**lm**) and *spatial indicator* (**sp**) and *none* (**nrol**) [63, 168, 73]. The above sentence contains several subsequences labeled with these roles. These are defined in Chapter 3.

Other conceptual aspects, such as *motion indicators*, indicate specific spatial motion information (usually specified in terms of *verbs*); *frame of reference* and the *path* of a motion are influencing concepts for spatial semantics and roles [168]. However, we restrict our focus to prepositions conveying spatial information.

Spatial role labeling is a special type of *semantic role labeling*, and, as with semantic roles, the spatial relations supported by the roles contribute to the recognition of the semantic frame of a sentence [90]. In semantic frame labeling, a predicate is identified and disambiguated, and its role arguments are recognized. In spatial role labeling, the spatial indicator is identified (instead of the verb predicate) and disambiguated, and its semantic role arguments including the trajector and landmark, are found.

However, differences between these two tasks exist. In spatial role labeling, the roles are more specific regarding their semantics; there is no direct correspondence between the semantics structure based on traditional semantic frames (*patient*, *agent*) and the spatial semantics structure of a sentence. In the above example, the FrameNet frame of `Giving` provides the *semantic type `Locative_relation` as the `Place` where the `Donor` gives the `Theme` to the `Recipient`.* The location refers to the place where the *giving* is performed, and not the location of the book, mentioned in the prepositional phrases which is important for spatial role labeling.

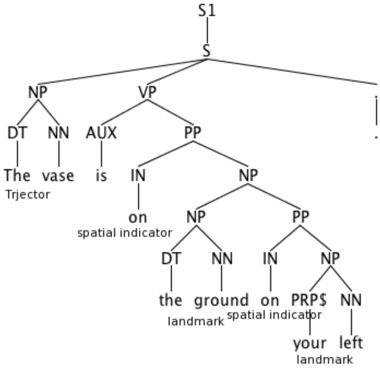


Figure 5.1: Parse tree labeled with spatial roles.

General spatial relation extraction presents many challenges concerning task-specific ambiguities and difficulties. There is not always a direct mapping between a sentence's grammatical structure and its spatial semantic structure. This issue is more challenging in complex spatial expressions that convey several spatial relations. The simple example below shows that grammatical dependencies cannot always identify spatial dependencies and connections:

The vase is **on** *the ground* **on** *your left.*

The dependency tree relates the first appearance of *"on"* to the words *"vase"* and *"ground"*. This process produces a valid spatial relation connecting the right trajector to the right landmarks. If we systematically follow the grammatical clues and information, then the second appearance of *"on"* connects the *"ground"* and *"your left"*, producing a less meaningful spatial relation in terms of trajector, landmark and spatial indicator (*"ground on your left"*), Figure 5.1 shows the related parse tree. When confronted with more complex relations and nested noun phrases, deriving "spatially valid" relations is not straightforward and highly dependent on the *lexical meaning* of words. However, recognizing the right prepositional phrase (PP) attachment during syntactic parsing can improve the identification of spatial arguments. Other linguistic phenomena, such as *spatial-focus-shift* and *ellipsis of trajector and landmark* [84], make extraction more difficult. *Spatial motion detection* and *recognition of the frame of reference* are additional challenges that are not treated here.

It should be noticed that, both the formal and informal (pragmatic) spatial expression meanings in natural language are highly dependent on lexical details, on the ontological structure of spatial information spaces, and on the embedding of extracted information into existing spatial knowledge.

5.2 Problem Statement

The spatial role labeling task finds *spatial relations* in natural language sentences, each of which includes a *spatial indicator* and its *arguments*. We assume that the sentence is a priori partitioned into a number of segments. The segments could be words, phrases or arbitrary subsequences of the sentence. More formally, let S be a sentence defined as a sequence of T segments:

$$S = \big\langle x_1, x_2, \ldots, x_T \big\rangle.$$

The target is to extract the spatial relations as a set of triplets with the following form, from each sentence:

$$\Big\langle \ x_{\text{spatial\_indicator}}, \quad x_{\text{trajector}}, \quad x_{\text{landmark}} \ \Big\rangle,$$

where $x_{\text{spatial\_indicator}}$, $x_{\text{trajector}}$ and x_{landmark} are three distinct segments of S, denoting the parts of S that represent the *spatial indicator* and its *trajector* and *landmark* arguments, respectively. For any spatial relation, the value of the trajector (or landmark) can be "undefined", meaning that no segment in S represents the trajector (or landmark). In those cases, we call the trajector (or landmark) *implicit*, as in the sentence *"Come over here"*, where the trajector *"you"* is only implicitly present. To formulate a learning setting for this problem we define a set of roles: roles $= \{\text{trajector}, \text{landmark}, \text{spatial\_indicator}, \text{none}\}$ and build a setting based on sequence tagging. The goal is to assign each segment in the sentence one or more of these roles and moreover recognize the link between the roles. Given a sentence S, the set of all *spatial indicators* of S is denoted *sp*. It is induced by the indicator function *sp* defined over all segments x of S: [1]

$$sp(x) = \begin{cases} 1 & \text{if } x \text{ is a spatial indicator} \\ 0 & \text{otherwise.} \end{cases}$$

We assume that spatial indicators overlap with neither each other nor trajectors and landmarks. In other words, for any sentence S, if x and x' are two segments of S, then $sp(x) = 1$ and $sp(x') = 1$ imply that $x \cap x' = \emptyset$. This is a realistic assumption according to our annotation scheme. Because

[1] We use the same symbols for denoting the sets and their indicator functions through this thesis which is a commonly accepted notation.

trajectors and landmarks are spatial indicator *arguments*, we define two indicator functions relative to a given spatial indicator s in sentence S.

$$tr_s(x) = \begin{cases} 1 & \text{if } x \text{ is a trajector of s} \\ 0 & \text{otherwise} \end{cases}, \; lm_s(x) = \begin{cases} 1 & \text{if } x \text{ is a landmark of s} \\ 0 & \text{otherwise.} \end{cases}$$

The set of *trajectors* (*landmarks*) with respect to the spatial indicator s is denoted tr_s (lm_s), induced by indicator functions tr_s and lm_s defined over all segments in S. For a spatial indicator s, its trajector and landmark cannot overlap with each other or s itself (though they can be undefined, as mentioned earlier). Although we have defined spatial indicators, trajectors and landmarks as arbitrary segments of a sentence, we focus on single words, each as one segment. However, a phrase in the sentence commonly plays a role, and we thus assume that the *head word* of the phrase is the role-holder. This is our first assumption which is often made in similar tasks [96]. The word level assignments can be leveraged to phrase level using NLP tools as preprocessing or post-processing. However, the phrase boundaries also can be learnt jointly along with the target roles. For example, in *"the huge blue book"*, *"book"* is the head word, and *"huge"* and *"blue"* are modifiers. In our data, the labeling scheme reflects this fact and only assigns roles to head words and labels the remaining words (e.g., modifiers) as “none". Hence, a sentence is hereafter assumed to be a sequence of words.

Since each word can have multiple spatial roles, we design a multi-sequence tagging setting in which each training sentence in the ground-truth data turns to a number of sequences, each of which is annotated with respect to one candidate spatial indicator with all possible trajectors and landmarks. Our second assumption, which is a realistic one in the English language, is that the spatial indicators are mostly prepositions, hence we can consider this subset of words as spatial indicator candidates. A sentence can thus provide multiple examples, up to the number of its candidate spatial indicators. We formally define each sentence in the corpus as a sequence of words $\langle x_1, \ldots, x_n \rangle$. Let k be the number of spatial indicator candidates in a sentence S; S then induces k examples $e_1 \ldots e_k$, where examples e_i and e_j have the same spatial indicator candidate for no i and j. Each e_i ($i = 1 \ldots k$) is a sequence $\langle (x_1, l_1), \ldots, (x_n, l_n) \rangle$ in which each word x_i ($i = 1 \ldots n$) is tagged such that: i) At most, one x_j gets a label l_j = spatial_indicator; ii) Some words get a label trajector or landmark, if they are a trajector or landmark of the candidate spatial indicator x_j; and iii) The remaining words get a label none. If a preposition is not spatial, all words in the sequence are tagged with none. As an illustration, consider the following sentence, which produces two example sequences:

A	girl	and	a	boy	are	sitting	at	the	desk	in	the	room
nrol	**tr**	**nrol**	**nrol**	**tr**	**nrol**	**nrol**	**sp**	**nrol**	**lm**	**nrol**	**nrol**	**nrol**
nrol	**nrol**	**nrol**	**nrol**	**nrol**	**nrol**	**nrol**	**nrol**	**nrol**	**tr**	**sp**	**nrol**	**lm**

The sentence is labeled twice, each time with a different indicator. Using our indicator functions, we have

$$sp = \{\text{at,in}\},\ tr_{\text{at}} = \{\text{girl,boy}\},\ lm_{\text{at}} = \{\text{desk}\},\ tr_{\text{in}} = \{\text{desk}\},\ lm_{\text{in}} = \{\text{room}\}.$$

The spatial relations for this sentence are the triples produced by the following:

$$\{\text{at}\} \times \{\text{girl, boy}\} \times \{\text{desk}\} = \big\{\langle\text{at, girl, desk}\rangle, \langle\text{at, boy, desk}\rangle\big\},$$

$$\{\text{in}\} \times \{\text{desk}\} \times \{\text{room}\} = \big\{\langle\text{in, desk, room}\rangle\big\}.$$

An example with an implicit trajector is the following sentence:

Go	under	the	bridge
nrol	**sp**	**nrol**	**lm**

In this case, we derive the spatial relation using

$$sp = \{\text{under}\}, \qquad tr_{\text{under}} = \emptyset, \qquad lm_{\text{under}} = \{\text{bridge}\},$$

which results in $\langle\text{under, undefined, bridge}\rangle$ as the corresponding spatial relation. In our learning model given a corpus of sentences tagged with spatial indicators, trajectors and landmarks, we produce a multitude of sequences as training examples, and construct (i.e. learns) an automated spatial relation extraction method based on tagging the multi-sequences and the extraction of the relations. We employ the same model for unseen data.

5.3 Approach

In analogy with semantic role labeling (SRL) in which words are classified based on a known *predicate* (a verb), in SpRL the spatial indicator is the pivot (i.e. predicate) of the spatial arguments. A spatial indicator can be from various lexical word classes, although the most dominant form is the *preposition*. In SRL, one can start from a verb and find roles related to it, but in SpRL, one must first find the *sense* of the pivot (i.e. the preposition). Sometimes, a proposition has a *spatial sense*, but that same preposition might not have a spatial sense in a different context.

Given the defined set of roles in the previous section, the set of spatial relations in a sentence S, denoted SR, is defined thus (where x, x', x'' are words in S),

$$\mathrm{SR} = \big\{\langle x, x', x''\rangle \mid x \in sp,\ x' \in tr_x,\ x'' \in lm_x\big\}.$$

In this definition, three functions should be estimated. First, the function sp is needed; it takes a word in the sentence as an input and estimates whether it is a spatial indicator. We employ a general binary classifier; for spatial indicators, for example we learn a function $\hat{sp}$ representing the probability that a word x is spatial, given its set of features in the context of sentence S. In other words, to determine the value of the indicator function sp, we compute (using $r = \{\text{spatial}, \text{nonspatial}\}$),

$$sp(x) = \begin{cases} 1 & \text{if spatial} = \arg\max_r \hat{sp}(r \mid x, S) \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

Indicating which words in the sentence have the trajector or landmark role requires two other functions, *given that we know that some word s is a spatial indicator*. We can train a multi-class classifier that classifies words in a sentence into $r' = \{\text{trajector}, \text{landmark}, \text{none}\}$. We call this function $\hat{R}$, and it takes a spatial indicator and tags words with these roles. We use a probabilistic classifier and the roles are predicted as follows,

$$r_{x,s} = \arg\max_{r'} \hat{R}(r' \mid x, s, S)), \tag{5.2}$$

where x is a word in sentence S, s is the pivot spatial indicator. By finding the best role assignments which is referred to as finding the maximum a posteriori (MAP), we obtain the sets of roles with respect to each spatial indicator,

$$lm_s(x) = \begin{cases} 1 & \text{if } r_{x,s} = \text{landmark} \\ 0 & \text{otherwise,} \end{cases} \qquad tr_s(x) = \begin{cases} 1 & \text{if } r_{x,s} = \text{trajector} \\ 0 & \text{otherwise.} \end{cases}$$

From Equations 5.1 and 5.2, we see that a pipelined model can be a solution for this task. We can first find words that potentially carry a spatial sense (i.e. being a spatial indicator or not, $sp(s) = 1$), and we then find the corresponding trajectors and landmarks for each pivot. The general structure of our pipeline approach consists of the following steps, outlined in subsequent sections:

- **Finding spatial indicators:** The first task consists of labeling parts of an input sentence S that play the spatial pivot role or finding the preposition with spatial sense. Section 5.3.1 describes this step, which utilizes TPP data to learn the labeling task.

- **Finding spatial arguments:** The second task consists of classifying parts of an input sentence S that play the landmark or trajector roles, *given a (spatial) pivot.*

In an additional *relation extraction* phase, we assemble the results of the previous two steps to form spatial relation triplets with spatial indicators and their trajector and landmark arguments (see also Algorithm 3). This step is straightforward and involves no learning. We also investigate an alternative approach in which we tackle both steps jointly:

- **Finding spatial indicators and their arguments jointly:** In this task, we do not use a separate preposition disambiguation step but instead learn to tag all words in a sentence jointly. The examples in the dataset are used to train a model that assigns the spatial indicator, trajector, and landmark roles simultaneously and the spatial relations are constructed based on the extracted roles afterwards. Section 5.3.3 describes this approach.

The remainder of this section describes the features and algorithms we designed and implemented for the spatial relation recognition task.

5.3.1 Learning Spatial Indicators

According to the aforementioned formalization, the set *sp* contains only prepositions and $sp(x) = 1$ holds only for prepositions with spatial sense. The sense of prepositions, as the main candidates for spatial indicators, can be disambiguated by machine learning methods, as a large corpus exists for it [153, 85]. We consider prepositions because of their importance and the feasibility of the disambiguation task [1].

The *locatives* recognized by SRL might help recognizing the spatial prepositions, but this is often not the case. The following two examples stem from the preposition disambiguation dataset (TPP) [85].

(i) *He saw Owen redden with pleasure and*
*laughed flinging an arm **about** his shoulders ...*

(ii) *This project compares assumptions incorporated into*
*social policies **about** these obligations ...*

Table 5.1 shows the labels assigned by a part-of-speech (POS) tagger, a dependency parser, and SRL to the preposition *"about"*. The parse tree, the dependency tree and even the semantic role labeler could not distinguish between two senses of the preposition *"about"*. We therefore propose to *learn* these senses from a corpus labeled with senses (TPP) provided for the

Prep	**POS**	**DepRel**	**SRL**	**sense**
about(i)	IN	NMOD	Arg1	spatial
about(ii)	IN	NMOD	Arg1	topic

Table 5.1: Assigned labels by a POS tagger, dependency tree and SRL to *"about"* with two different senses.

preposition disambiguation task (SemEval07) [85], featuring the category *SpatialSense* among others.

More specifically, the function $\hat{sp}$ is trained to perform the preposition disambiguation task described in the previous section in Equation 5.1. It uses the following linguistically motivated features and the preposition contextual features that we aim to classify:

- The *preposition* itself
- By exploiting the dependency parser:
  - The words directly dependent on the preposition (*head1*)
  - The words on which the preposition is directly dependent (*head2*)
- For the predicates which have a dependency relation with the preposition:
  - All words that are arguments of the predicate other than the preposition are added using a semantic role labeler

For all extracted words satisfying the above conditions, the following features are also included: *lemma*, *part-of-speech* tag (POS), type of *dependency relation* (DPRL), *semantic role labels* and, for predicates, the sense of the predicate (if assigned).

To identify the spatial prepositions, we use the TPP data provided for the preposition disambiguation task, SemEval07 [85]. We extract the features from the training and test data and use a *maximum entropy* and a *Naive Bayes* classifier to disambiguate the prepositions' sense. This process results in a binary classification of a preposition's spatial or nonspatial sense.

5.3.2 Trajector and Landmark Classification

We utilize the first step of preposition sense disambiguation, described in the previous section, to recognize the spatial indicators first, after which its arguments (trajectors and landmarks) can be classified with a a multiclass classifier $\hat{R}$ as explained in Section 5.3. The generic feature set used in Equation 5.2 can now be defined in more detail using three different sorts. The first set of features are the local features of the word that we aim to classify,

$\phi_1(x) = \{\phi_{wf}(x), \phi_{pos}(x), \phi_{dprl}(x), \phi_{srl}(x), \phi_{sub}(x)\}$, the second includes the local features of the spatial indicator candidate of which the word may be an argument, $\phi_2(s) = \{\phi_{wf}(s), \phi_{sub}(s)\}$, and the third contains the relational features that relate the word to the sentence's candidate spatial indicator, $\phi_3(x, s) = \{\phi_{path}(x, s), \phi_{before}(x, s), \phi_{dis}(x, s)\}$. The detailed explanation of these features are provided in Chapter 4. Take the following sentence as an example,

The vase is on the ground on your left.

Here, the input features for classification of *vase* w.r.t. the first *on* are:

$$\underbrace{\text{vase, NN, SBJ, A0, NP-VP}}_{\phi_1(x)}\underbrace{\textbf{on}\text{, NP}}_{\phi_2(x)}\quad\underbrace{\text{NN} \uparrow \text{NP} \uparrow \text{S} \downarrow \text{VP} \downarrow \text{PP} \downarrow \text{IN, true, 3}}_{\phi_3(x,s)}$$

In a *multi-class classification* setting each word, represented by a feature vector, is separately classified, assuming that these classifications are independent. We use such a model in our initial experiments. In subsequent models, the class to which the words are assigned depends not only on their own feature values but also on the features of other words and relations among the various classes. The obtained class of a word may constrain the class of the next word. We therefore employ *conditional random field* (CRF) models. In these models, a sentence is a sequence of observations (i.e. words). Each observation can be described in terms of feature vectors, and the model outputs a label for each word in the sequence forming the nodes of a probabilistic graphical model.

After recognizing the trajector and landmark given a spatial indicator, we have all the relation elements. Relation extraction is performed in a straightforward way, by assembling all extracted roles and combining them into spatial triplets. Algorithm 3 shows the entire process, based on preposition disambiguation and trajector/landmark classification. The main purpose of this pipeline approach is to exploit a large external data source (TPP) for spatial sense disambiguation.

5.3.3 Learning Spatial Relations without a Priori Spatial Indicator Classification

Combining two steps of the aforementioned pipeline provides another option for learning spatial relations. We could omit the first step of using a dedicated classifier for spatial sense recognition, and learn to assign all spatial roles

Algorithm 3 Spatial-Relation-Extraction(S: sentence) **returns** relations SR

```
 1: {preposition disambiguation}
 2: for all $x \in S$ do
 3:     Use the trained binary classifier $\hat{sp}(x)$ and
 4:     construct the set $sp$ of all spatial indicators of the sentence $S$.
 5: for all $s \in sp$ do
 6:     {trajector and landmark classification}
 7:     for all $x \in S$ do
 8:         Use the trained multi-class classifier $\hat{R}$ and
 9:         construct the sets $tr_s$ and $lm_s$ according to the assigned labels.
10:     if $tr_s = \emptyset$ then $tr_s \leftarrow \{undefined\}$
11:     if $lm_s = \emptyset$ then $lm_s \leftarrow \{undefined\}$
12:     {relation extraction}
13:     $\mathrm{SR} \leftarrow \mathrm{SR} \bigcup \{\langle s, t, l\rangle \mid t \in tr_s, l \in lm_s\}$
14: return SR
```

jointly, i.e. tagging words with trajector, landmark, spatial_indicator or none, based on a training dataset.

As mentioned before, each sentence can contain several spatial prepositions and each word can carry different roles with respect to a different spatial preposition. Hence a simple classification setting can not solve the problem. The solution we use here is to, again, generate multiple examples from S, where each example contains a candidate pivot with specific features extracted for that word (e.g., path features from words to the pivot). For each example, the words are classified using these relational features. One must basically generate as many examples as there are words in S; in our practice, it suffices to do this procedure only for pivots that are prepositions. The main advantage of this setting is that the learning algorithm gets the freedom to classify trajectors, landmarks and indicators in the context of one another.

In the relation extraction step, we perform the same general steps as in Algorithm 3, differing primarily in that we take all prepositions as candidate spatial indicators in the preposition disambiguation phase (lines 1–4) and that the classifier $\hat{R}$ now uses all roles, including spatial_indicator.

5.3.4 Linear Chain Model

For classification of the words in the sentence with spatial roles we use a sequence tagging model based on the conditional random fields described in Chapter 2. Formally, each input sentence $X = (x_1, \ldots, x_T)$ is a sequence of words and output $Y = (y_1, \ldots, y_T)$ is the corresponding set of labels assigned

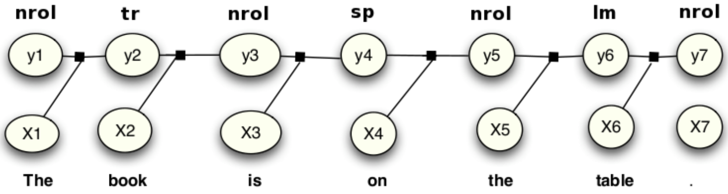


Figure 5.2: Instantiated graphical representation of linear-chain CRF, labeled given "on" as the pivot of the sequence.

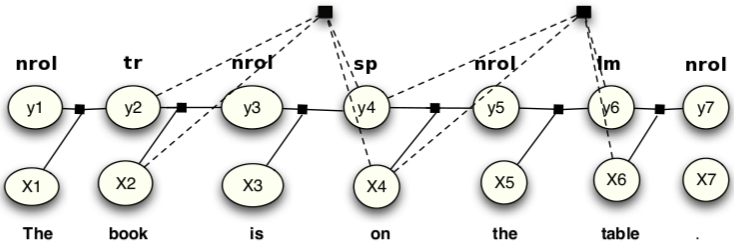


Figure 5.3: Graphical representation of CRF with preposition template. Prepositions are connected to the candidate trajectors and candidate landmarks i.e noun phrases. Factors occur as black squares. Labeled given "on" as the pivot of the sequence.

to X based on one spatial indicator candidate which is the pivot of the sequence. For example, in Figure 5.2, the sentence is annotated given the word "on" as the pivot of the sequence. In SpRL, Y ranges over the classes trajector (**tr**), landmark (**lm**), spatial indicator (**sp**) in the joint setting or none of these (**nrol**). The features of the words in the sequence are produced based on the features of each word itself and the relational features between each word and the pivot of the sequence.

In this setting, the spatial role label of a word in the sentence depends on the label of the word in the previous position. Considering sequential relationships can increase the learning model's accuracy. The conditional probability $p(y|x)$ [2] is calculated given one template, $\Psi_t(y_{t-1}, y_t, x_t)$ described in Chapter 2, in Formula 2.9.

For the CRF experiments we use Mallet[3]. The linear chain setting of Mallet uses a forward-backward algorithm to compute the marginal distributions and the Viterbi algorithm to compute the most probable sequence label

[2] An istantiation of X is denoted as x and an instantiation of Y as y.

[3] http://mallet.cs.umass.edu/download.php

assignment. For our task, allowing transitions unobserved in the training data during the inference and prediction phases adds more flexibility to the model, particularly when there are few training examples. This setting is called *fully-connected* in the Mallet tool, and we use it in our experiments. We refer to this setting as *linear-chain (FC)*.

5.3.5 Skip-chain Model with Preposition Template

In many relation extraction tasks, certain long-distance dependencies between entities play an important role. In our task, prepositions primarily play a spatial indicator role, while trajectors and landmarks are noun phrases. There could be many words in between the roles in the sentence that have no particular role and are assigned the *none* label. In light of this fact, we apply a version of a *skip-chain* CRF [142] to account for the probabilistic dependencies between distant labels. These dependencies are represented by augmenting the linear-chain CRF with factors dependent on the labels of the sentence's pivot preposition and noun phrases. The features on skip edges can incorporate information from the context of both endpoints, so the strong evidence of one endpoint can influence the label at the other endpoint. In our skip-chain CRF model, we exploit two clique templates: one is the normal sequential part (connecting neighboring words), $(\Psi_t(y_t, y_{t-1}, x_t))$, and the other connects pivot prepositions to candidate trajectors and landmarks, $(\Psi_{uv}(y_u, y_v, x_u, x_v))$. The u and the v are the positions which there are skip edges for them. In our model, the set of pairs of (u, v) includes all pairs of prepositions and nouns (see Figure 5.3). The probability of sequence label y given input x is defined based on the above mentioned templates (defined in Formula 2.14 and 2.15) and computed according to the Formula 2.13 for the skip-chain model described in Chapter 2.

We use *loopy belief propagation* as the approximate inference algorithm of GRMM,[4] in the implementations. This tool is developed as an add-on package to Mallet, and supports designing arbitrary factor graphs for general CRFs rather than basic linear chain models.

We compare the results of the CRFs defined in this section and Section 5.3.4 with two baseline approaches:

- **MaxEnt (baseline) model.** As a baseline learning model, we classify the words of a sentence independently using a standard maximum entropy classifier.
- **Simple baseline.** To encourage the use of machine learning, a simple baseline is employed: given a spatial preposition, the first head word

[4] http://mallet.cs.umass.edu/grmm/index.php

before the preposition is taken as the trajector and the *head word* after the preposition as the landmark. There is no learning from data in this setting, but the dependency tree is exploited to discover dependent headwords.

5.4 Experimental Study

In this section, we report on a set of experiments to evaluate various components of the spatial role labeling and relation extraction tasks. The research questions that we aim to answer empirically in this chapter are the following.

Q5.1. *How well can we detect the spatial sense of prepositions using available resources and methods?*

Q5.2. *If the spatial sense of a preposition is known or learned beforehand, how well can we learn its corresponding trajectors and landmarks from data?*

Q5.3. *What benefits lie in the sequential nature of finding the spatial sense of a preposition and then finding trajectors and landmarks (the so-called pipeline technique)?*

Q5.4. *What benefits lie in jointly recognizing spatial indicators, trajectors and landmarks, and how can long-distance dependencies help in this setting?*

Q5.5. *How do different pipelining methods affect the accuracy of the whole-relation extraction?*

Q5.6. *What is the effect of the used features on the extraction task?*

Q5.7. *What is the cross-domain performance of the approach on an unrestricted natural language text that contains both spatial and nonspatial information?*

Q5.8. *What are the main sources of errors in our approach?*

5.4.1 Datasets

For the experimental analysis in this chapter we use the corpora that are described in Chapter 3.

For the CLEF annotations the SemEval-0 version is used. This was the only available annotation for these experiments and the statistics are shown in Table 5.2. The other datasets in this chapter are GUM (Maptask), Fables and DCP with the aforementioned statistics where the number of their sequences for the sequence tagging setting in this chapter are 122, 864 and 809, respectively. Moreover, for the preposition disambiguation the standard

#Sentences	686
#Sequences	1430
#Spatial relations	869
#Trajectors	839
#Landmarks	741
#Spatial prepositions	735
#nonSpatial prepositions	695

Table 5.2: The statistics of the first version of CLEF, also referred as SemEval-0.

TPP data has been used. As mentioned above, we only consider prepositions as spatial indicators.

This restriction is natural in English texts and especially for our data. Ignoring lexical categories other than prepositions has a trivial influence on our experiments with this corpus. Three exceptional cases exist in CLEF, where the words *crossing*, *supporting* and *away* are tagged as spatial indicators and this is the case for seven sentences in GUM (Maptask) dataset. The other two corpora of DCP and Fables are used in cross-domain experiments of this chapter. The datasets are preprocessed as follows. We generate parse trees for the sentences using the Charniak parser[5] [21], and the LTH[6] tool [61] produces the semantic roles and several other features in CoNLL-2008 output format.[7]

5.4.2 Preposition Disambiguation

In this part we investigate the answer to the **Q5.1** experimental question and design a number of experiments to this aim.
In the **first experiment**, we investigate how well the semantic role labeling (SRL) recognizes the prepositions with spatial sense and tags them with LOC (location) or DIR (direction). This experiment is performed on the TPP corpus. We call this model SRL-LOC and compare it to preposition disambiguation over TPP. TPP contains 8,781 spatial prepositions and 14,681 nonspatial prepositions. We train multi-class classifiers using Naive Bayes (TPP-NB) and Maximum entropy (TPP-MaxEnt) techniques and classify the senses of the prepositions including their spatial sense. The results in table 5.3 show that the *precision* of SRL-LOC is fairly good. Whenever SRL recognizes

[5] http://www.cfilt.iitb.ac.in/ anupama/charniak.php
[6] http://barbar.cs.lth.se:8081/
[7] http://barcelona.research.yahoo.net/dokuwiki/doku.php?id=conll2008:format

the spatial sense, it is mainly correct; however, there are many cases in which SRL does not recognize spatial senses, rendering a lower recall and consequently a lower F1 and accuracy. For the preposition disambiguation, the 99% confidence interval for the accuracy and F1-measure of both TPP-MaxEnt and TPP-NB is (0.875 – 0.89) and (0.868 – 0.88), respectively. This report is over the spatial sense class and it clearly is more accurate compared to the results of SRL's recognition. This experiment provides an argument for the necessity of sense disambiguation even when recognizing only spatial prepositions. Table 5.4 gives results for some frequently used prepositions (e.g., *in*, *on*, *after*, *before*).

System	**Precision**	**Recall**	**F1**	**Accuracy**
SRL-LOC	0.83	0.49	0.53	0.59
TPP-NB	0.86	0.92	0.88	0.88
TPP-MaxEnt	0.88	0.91	0.88	0.88

Table 5.3: Detection of spatial or nonspatial preposition senses, relying on detected locatives by SRL compared to using a NB and a MaxEnt classifier for PP-disambiguation; 10-fold cross validation on TPP dataset.

PP	**TPP-NB**			**TPP-MaxEnt**			**SRL-LOC**		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
on	0.73	0.96	0.83	0.79	0.95	0.86	0.71	0.40	0.51
after	0.50	0.90	0.64	0.54	0.70	0.61	0.00	0.00	0.00
in	0.66	0.92	0.77	0.70	0.88	0.78	0.56	0.91	0.69
before	0.67	0.86	0.75	0.80	0.57	0.67	0.50	0.43	0.46

Table 5.4: Detection of spatial or nonspatial preposition senses for some frequently used prepositions in the TPP dataset.

In a **second experiment**, we evaluate our preposition disambiguation models in general. We tested their recognition of all *coarse-grained* senses on the preposition SemEval-2007 data [85]. Coarse-grained senses include 20 general classes of preposition senses, such as spatial, temporal, causal, and membership. Table 5.5 gives the accuracy of standard train/test split of the benchmark, using a maximum entropy classifier and a Naive Bayes classifier. This table shows the results of the best system in the SemEval-2007 challenge for this coarse-grained sense disambiguation, the accuracy of applying bag of words (BOW), using the most frequent (FreqSense) and first (FirstSense) senses as baselines. The difference between our system and the best system

System	Accuracy
Proposed-features (MaxEnt)	0.874
Proposed-features (NB)	0.86
MELB-YB (Best in SemEval-2007)	0.861
BOW (MaxEnt)	0.81
FreqSense	0.649
FirstSense	0.61

Table 5.5: Coarse grained sense disambiguation, TPP dataset.

from SemEval-2007 is statistically significant with a 95% confidence level (p < 0.05).

Although other work [153] on preposition sense disambiguation outperforms results of the SemEval-2007 challenge too, these authors report only on the results of *fine-grained* sense disambiguation, which was not required for spatial sense recognition in our setup. As the TPP data are a benchmark problem, we use a similar evaluation setting for comparison purposes and do not further experiment with different training regimes (in train/test splits). The current preposition disambiguation results are a promising start for spatial sense recognition and spatial relation extraction.

In the **third experiment** we move towards testing the preposition disambiguation models on our SpRL datasets. We built the final preposition sense classifiers using the whole TPP dataset. We implemented 34 binary classifiers for 34 prepositions and for classifying their spatial vs. non spatial sense. For some prepositions in CLEF, e.g., *opposite*, no classifier is trained-as there are no annotations in TPP for it. This issue occurred in 35 of the 1,430 cases. Table 5.6 shows the preposition disambiguation performance on GUM (Maptask) and CLEF. GUM (Maptask) is more domain-specific and contains more spatial prepositions (112/122), including a larger percentage (24/122) of prepositions that are not found in the TPP corpus and thus not recognized as spatial prepositions. This fact leads to a lower recall for spatial preposition recognition in this corpus in comparison to CLEF. We use the disambiguated prepositions obtained in this step in the pipeline of spatial role labeling.

5.4.3 Extraction of Trajector and Landmark

This part of the experiments is designed to answer the **Q5.2**, **Q5.3** and **Q5.4** in three sets of experiments. We note that, the classification of trajectors and landmarks is not an isolated classification of words, but a classification of

Corpus	Precision	Recall	F1	#Unrec. PPs
CLEF	0.858	0.818	0.84	35
GUM (Maptask)	0.97	0.71	0.82	24

Table 5.6: Preposition disambiguation trained on TPP and tested on CLEF and GUM (Maptask).

relations between a word and spatial pivot. This statement is the underlying assumption for relation extraction in the experiments described below. We show results for different settings: a) Using ground-truth spatial prepositions; b) Using a pipeline approach in which the preposition disambiguation is learned from external data; and c) Using a joint classification model in which spatial indicators, trajectors and landmarks are learned and classified together.

Using Ground Truth Spatial Prepositions

To extract the trajectors and landmarks related by a spatial pivot, we first use the disambiguated ground-truth pivots. We implemented two different classification settings. In one setting, **MaxEnt (baseline)**, we classify each word as trajector, landmark or none, based on its related extracted features described in section 5.3.2 and using a maximum entropy multiclass classifier. In the second setting, **Linear-chain**, we classify each word, using CRFs, considering its context (the sentence) and employing the same linguistic input features as the first setting. These results are compared to the **Linear-chain (FC)** and the **Simple baseline** described in Section 5.3. Tables 5.7 and 5.8 show the precision, recall and F1 measures for each tag using 10-fold cross-validation on the CLEF and GUM (Maptask) datasets.

Method	Trajector			Landmark		
	Pr	Rec	F1	Pr	Rec	F1
MaxEnt(baseline)	0.775	0.744	0.758	0.916	0.853	0.881
Linear-chain	0.870	0.744	0.801	0.950	0.869	0.907
Linear-chain (FC)	0.905	0.792	0.844	0.953	0.879	0.914
Simple baseline	0.269	0.413	0.326	0.456	0.784	0.576

Table 5.7: Extraction of trajector/landmark roles in the CLEF dataset relying on the ground-truth preposition sense; 10-fold cross-validation.

Method	**Trajector**			**Landmark**		
	Pr	Rec	F1	Pr	Rec	F1
MaxEnt(baseline)	0.862	0.931	0.891	0.776	0.762	0.750
Linear-chain	0.990	0.959	0.973	0.916	0.918	0.915
Linear-chain(FC)	1.000	0.969	0.983	0.947	1.000	0.971
Simple baseline	0.008	0.015	0.011	0.337	0.500	0.402

Table 5.8: Extraction of trajector/landmark roles in the GUM (Maptask) dataset relying on the ground-truth preposition sense; 10-fold cross-validation.

The results show that context-dependent classification models outperform the maximum entropy model and that the differences are statistically significant for $p < 0.05$, where the fully connected CRF model gives the best results. Using the fully connected setting of the simple tagger yields statistically significant and sharp improvements in trajector classification in CLEF and landmark classification in GUM (Maptask).

Pipeline Setting – Exploiting Preposition Disambiguation

In this experiment, we fully automate the tasks of recognizing spatial roles and the corresponding spatial relations. The preposition disambiguation and the extraction of trajector/landmark tasks are connected and followed by the whole-relation-extraction. The preposition classifier is trained on the TPP dataset. The landmark/trajector/none classifier is trained on the subset of GUM and also the CLEF dataset.

In this setting, various options are examined during the test phase. Each preposition in a sentence is given to the relevant classifier from the 34 TPP-classifiers. If it does not match a TPP preposition, it is an *unknown* preposition and treated in two distinct ways: a) *Nonspatial* and the model referred as **Pip1**; b) *Spatial* and the model referred as **Pip2**. If the preposition is recognized as spatial, the process of the trajector/landmark extraction is performed; otherwise, all words in the sentence are labeled as none with respect to that preposition. We compare these settings with two other settings: c) A setting in which every preposition is blindly assumed to be a spatial indicator referred as **Pip3**; d) The setting that uses the ground truth prepositions referred as **PipGt** and was reported as linear-chain in the previous section's experiments. These results, presented in Tables 5.9, 5.10, help to assess the effect of preposition disambiguation.

Method	**Trajector**			**Landmark**		
	Pr	Rec	F1	Pr	Rec	F1
Pip1	0.886	0.654	0.752	0.914	0.714	0.801
Pip2	0.889	0.685	0.773	0.916	0.741	0.819
Pip3	0.870	0.792	0.828	0.904	0.878	0.891
PipGt	0.905	0.792	0.844	0.953	0.879	0.914
JL	0.884	0.668	0.759	0.919	0.712	0.802
JLT	0.988	0.998	0.980	0.866	0.892	0.843

Table 5.9: Extraction of trajector/landmark on CLEF dataset, comparing pipeline, ground-truth and joint learning by 10-fold cross-validation.

The experimental results in Table 5.9 show that exploiting the linguistic features of the correct spatial preposition in the CLEF corpus (PipGt) improves the trajector and landmark extraction performance compared to pipelining, as expected. The difference is statistically significant ($p < 0.05$). In the complete extraction problem, i.e. with unknown spatial indicators, assuming all prepositions to be spatial (Pip3) yields the highest recall, as it allows the trajector/landmark classifier to find related arguments. The pipeline model assuming unrecognized prepositions as spatial (Pip2), receiving input from the preposition disambiguation module, improves precision but lowers recall compared to Pip3 and Pip2. Investigating the errors indicates that no trajectors and landmarks are generally extracted when nonspatial prepositions are recognized as spatial and the words are correctly classified as none. However, having a spatial preposition wrongly classified as nonspatial (in Pip1) prohibits trajector and landmark extraction, causing a drop in recall and also F1 compared to Pip2 and Pip3.

For GUM (Maptask) corpus, the results in Table 5.10 show that inputting the correct preposition (PipGt) does not make a difference compared to assuming all spatial in Pip3; moreover, pipelining (in Pip1 and Pip2) yields lower recall. GUM (Maptask)'s statistics show that more than 93% of the prepositions are spatial and errors in preposition disambiguation prohibit the extraction of related trajectors and landmarks, resulting in a sharp drop in recall with no significant variation in precision.

Joint Learning Setting

This model is described in section 5.3.3. Since spatial indicators are classified jointly with other spatial roles, here some of the errors caused by the pipelining

Method	**Trajector**			**Landmark**		
	Pr	Rec	F1	Pr	Rec	F1
Pip1	1.000	0.510	0.660	0.930	0.460	0.580
Pip2	1.000	0.701	0.801	0.937	0.660	0.752
Pip3	1.000	0.969	0.983	0.947	1.000	0.971
PipGt	1.000	0.969	0.983	0.947	1.000	0.971
JL	1.000	0.956	0.976	0.920	0.956	0.934
JLT	0.934	0.945	0.936	0.720	0.760	0.727

Table 5.10: Extraction of trajector/landmark on GUM (Maptask) dataset, comparing pipeline, ground-truth and joint learning by 10-fold cross-validation.

are corrected. However, as Table 5.9 shows on the CLEF dataset, the recall of the best pipeline system (Pip2), is slightly higher than jointly learning (JL) the trajector and landmark classification (the improvement is statistically significant only for ($p < 0.1$)), which implies the difficulty of the preposition disambiguation in the joint setting. Adding long distance dependencies to joint learning through the preposition template in JLT model greatly improves the performance on the CLEF dataset, particularly in trajector classification. Coupling the prepositions and the nouns via the defined template helps the joint setting for recognizing the spatial prepositions and consequently the roles. In contrast, a sharp decrease in landmark classification occurs on the GUM (Maptask) dataset when applying the JLT model. The difference in language characteristics in these datasets affects these results, which calls for further investigation. In Section 5.4.7, an error analysis categorizes the types of errors that can occur in the spatial role labeling task and the errors of two models (with and without a template) are compared using a test subsample. For GUM (Maptask), Table 5.10 shows that assuming all prepositions to be spatial outperforms other settings, including joint learning. The previous experiments show joint learning outperforming pipelining, though the pipeline setting uses the external resource TPP. Cross-domain differences and sentence types in TPP, CLEF, and GUM (Maptask) datasets account for this discrepancy. This issue will be discussed later in this chapter.

5.4.4 Whole Relation Extraction

This part of experiments is designed to answer the **Q5.5** about the whole relation extraction. The way that the spatial triplets are generated is explained in section 5.3 and the evaluation is detailed in Chapter 4. The results of

Method	**WR** (GUM)			**WR** (CLEF)		
	Pr	Rec	F1	Pr	Rec	F1
Pip1	0.874	0.534	0.663	0.653	0.605	0.628
Pip2	0.894	0.722	0.799	0.547	0.627	0.584
Pip3	0.870	0.948	0.907	0.391	0.722	0.507
PipGt	0.948	0.948	0.948	0.704	0.723	0.714
JL	0.888	0.904	0.896	0.704	0.737	0.720
JLT	0.672	0.703	0.684	0.830	0.830	0.830

Table 5.11: Extraction of whole relations (WRs) on GUM (Maptask) /CLEF, comparing pipeline, ground-truth and joint learning using 10-fold cross-validation.

all previously discussed models on the whole relation extraction from GUM and CLEF are represented in Table 5.11. The first noticeable result is that assuming all prepositions as spatial (Pip3) is generally impractical. The very low performance on CLEF indicates that the relation extraction by this assumption is not robust for unrestricted language, though this setting works well for trajector and landmark extraction on GUM (Maptask). Employing ground-truth prepositions (PipGt) provided the best results for GUM (Maptask), though we observed no significant difference compared to joint learning for relation extraction in CLEF. To explain how the joint learning setting can, in this particular case, perform as well as the ground-truth setting, we must examine the input and output features of the models. In the ground-truth setting, the (correct) spatial indicators function as input, and the classifier learns to label trajectors and landmarks. In the joint learning setting JL, the model learns to utilize the correlations between trajectors, landmarks *and* spatial indicators as outputs labels, so it considers the transitions between spatial indicators and other labels. In the two pipeline settings, Pip2 shows better results in GUM (Maptask) but worse results in CLEF. This finding is reasonable due to the prior distribution of spatial prepositions in GUM (Maptask) and CLEF, as discussed above. The JLT model gives the best results for whole relation extraction on CLEF. This setting is ideal for the SpRL problem when there are sufficient training examples, which is not always the case. The pipeline setting performs better in some trajector and landmark classifications, which signals the significance of exploiting the TPP resource. Our final experiments on texts from different domains in Section 5.4.6 highlight the importance of the TPP resource.

Data	**Features**	**Trajector**			**Landmark**			**WR**		
		Pr	Rec	F1	Pr	Rec	F1	Pr	Rec	F1
CLEF	All features	0.905	0.792	0.844	0.953	0.879	0.914	0.704	0.723	0.714
	-dis	0.889	0.792	0.836	0.956	0.879	0.915	0.697	0.717	0.707
	-SRL	0.893	0.795	0.840	0.961	0.876	0.916	0.701	0.717	0.709
	-dis-wordsubcat	0.883	0.770	0.822	0.954	0.871	0.911	0.680	0.693	0.687
GUM	All features	1.000	0.969	0.983	0.947	1.000	0.971	0.940	1.000	0.971
	-dis	1.000	0.969	0.983	0.920	0.987	0.951	0.932	0.947	0.940
	-SRL	0.987	0.956	0.969	0.917	0.924	0.916	0.874	0.894	0.884
	-dis-wordsubcat	0.983	0.969	0.975	0.920	0.946	0.929	0.906	0.921	0.913

Table 5.12: The effect of applying the distance (dis), word subcategorization (wordsubcat) and SRL feature for trajector and landmark extraction, using ground-truth preposition senses. The baseline uses all features.

5.4.5 Experimental Feature Analysis

This part of the experiments is to answer question **Q5.6** about the influence of the learning features. As mentioned in section 5.3, we exploit the linguistically motivated local and relational features which are described in Chapter 4. In the results reported above, we use all of the features described in that section. By investigating the features' impacts and omitting them one by one, we determined that almost all features contribute positively to the performance. The path feature contribution was marginal, especially for GUM (Maptask). Because GUM (Maptask) is a small corpus and the path feature has too many unique values in our dataset, its discriminative power is limited here. The complex path feature generally can produce some overfitting or inserts noise into the model, due to incorrect prepositional phrase (PP) attachments, for example. The distance between the preposition and its arguments is thus a valuable feature that helps determine whether a word is an argument of a preposition. The experiments with and without this feature show a positive impact on both datasets; an overall gain of approximately $1\% - 3\%$ for both GUM (Maptask) and CLEF is statistically significant only for $p < 0.1$. To understand the effect of our additional features, we use ground-truth preposition senses, and Table 5.12 shows the results.

Exploiting more discriminative structural features may compensate for the lack of lexical information, we therefore evaluate adding the subcategorization of a target word using the aforementioned definition. The last row in Table 5.12 for each dataset shows the performance using neither distance nor subcategorization. The quantitative effect of the SRL feature is represented in the same table. The table clearly shows the positive influence of this feature on GUM (Maptask), but it contributes less for CLEF. GUM (Maptask) contains directional instructions with few compound locative descriptions, so there are more direct relations between semantic roles, including AM-DIR, AM-LOC and being a landmark, as well as between a “patient" role and being a trajector.

5.4.6 Cross-domain Evaluation

This part of the experiments is to answer **Q5.7** about cross domain evaluation of the SpRL models. Although our methodology for extracting spatial semantics is domain independent, the results still depend on the observed lexical features. A model trained in one domain and later employed in another often performs poorly due to feature distribution changes. Other applications of machine learning methods share this problem [60], particularly the natural language processing area. As explained in previous sections, our main data set is CLEF, which contains many spatial descriptions but is still balanced with nonspatial information. The GUM (Maptask) corpus is much smaller, domain-specific and biased to spatial descriptions; learning from the same corpus in a cross-validation setting produces good results. We based our experiments on these two data sets to show that SpRL is feasible and that specific learning algorithms and representations are effective. In this section, we use our two additional annotated datasets from different domains and discuss experiments that explore domain portability of the learnt models and test the advantage of using external data resources (e.g., TPP) in that process.

Corpus	**Precision**	**Recall**	**F1**	**#Unrec. PP**
Fables (TPP-Maxent)	0.444	0.657	0.530	13
Fables (SRL-LOC)	0.495	0.420	0.454	-
DCP (TPP-MaxEnt)	0.584	0.687	0.631	29
DCP (SRL-LOC)	0.226	0.423	0.295	-

Table 5.13: Preposition disambiguation performance trained on TPP and tested on Fables and DCP.

Our first experiment concerns the intrinsic cross-domain nature of employing TPP data. As previously done for CLEF and GUM (Maptask) in Section 5.4.2, we evaluate preposition sense disambiguation performance on the new datasets Fables and DCP. This classifier is also used in the pipeline setting in subsequent experiments. The results in Table 5.13 indicate that the preposition spatial sense recognition (TPP-MaxEnt) is harder in these data sets than in CLEF and GUM (Maptask). However, for Fables and DCP datasets, the TPP-based model outperforms SRL in spatial preposition recognition. The results also show that the SRL system is more accurate for Fables than DCP. The more frequent use of compound verbs in Fables may account for this phenomenon, as the prepositions are mostly attached to verb phrases.

In a second set of experiments concerning trajector and landmark extraction, we applied the settings described in previous sections to Fables and DCP. The models were trained on CLEF and tested on these data sets. For the JL model, we applied the learned classifier on the unlabeled Fables/DCP data. For the pipeline settings (Pip1 and Pip2), the classifiers trained on TPP find the spatial indicators, after which we apply a classifier trained on CLEF (for trajectors and landmarks) to the unlabeled Fables/DCP data. Table 5.14 reports the results, to summarize the tables only the F1-measure is presented. Confidence intervals (90%) for the last column are (0.428–0.532) and (0.423–0.527), and all others have a lower variance. The table shows that Pip1 outperforms JL, demonstrating the benefits of the model trained on

Method	**Fables**			**DCP**		
	Traj.	Land.	Indica.	Traj.	Land.	Indica.
Pip1	0.080	0.397	0.530	0.208	0.435	0.631
Pip2	0.100	0.424	0.348	0.232	0.463	0.554
Pip3	0.181	0.342	0.245	0.293	0.447	0.431
PipGt	0.231	0.620	——	0.338	0.590	——
JL	0.113	0.378	0.45	0.223	0.432	0.614
JLT	0.101	0.292	0.333	0.163	0.319	0.409

Table 5.14: F1-measure of cross-domain evaluation; the classifiers learned on CLEF and tested on fable stories and DCP data.

TPP and the value of exploiting TPP in this experiment. The outperforming of the pipeline is statistically significant.

For trajectors and landmarks, the first unsurprising result is that given the ground-truth prepositions, trajectors and landmarks can be classified more accurately in both data sets. The whole relation extraction (not shown) proved more difficult here. Once more, in Section 5.4.7 we chose a sample of the errors to obtain a clearer analysis on cross-domain evaluation.

Dataset	**Trajector**	**Landmark**	**Indicator**	**Whole-rel**
Fable stories	0.544	0.569	0.638	0.481
Confluence	0.518	0.595	0.685	0.475

Table 5.15: F1-measure of 10-fold cross-validation; with JL model that has the best F1-measure averaged over all roles.

For completeness, we evaluate how well our techniques work on the additional datasets without training on CLEF using standard 10-fold cross validation in a third experiment (see Table 5.15). To summarize the tables, we only present the F1-measures of the most outperforming models, where we find that joint learning is the best setting for both datasets. Considering the previous cross-domain experiment, this result is reasonable. The joint learning setting shows higher performance with 10-fold cross validation because the training and test sets have similar (lexical) feature distribution. Overall, the evaluation of these two datasets performed worse than on our main datasets, CLEF and GUM (Maptask), because of the broad vocabulary range in these additional datasets and the lower proportion of spatial expressions. This situation requires more training examples to obtain acceptable accuracy. In Section 5.4.7, a brief discussion on the errors of this experiment is given too.

5.4.7 Error Analysis

In this part we look into the errors to answer the **Q5.8** about the type of difficulties that learning models should handle for SpRL. The experiments using the GUM (Maptask) and CLEF datasets clearly indicate that dependencies between observed nodes in the CRF model are advantageous for spatial role labeling. Most errors are classical information extraction errors. The lack of a huge training corpus with sufficient word occurrences results in invalid argument assignments concerning spatial semantics. Cross-domain experiments on Fables and DCP are most affected by this lack. In the pipeline setting, errors are primarily propagated from one phase to another. The more elaborate solution of jointly classifying prepositions and trajector/landmarks should, theoretically, provide a better solution. However, this setting suffers even more from the lack of lexical information, but shows promising results in general. This setting could be the best platform with the injection of the partially labeled external TPP resource. Many words in a sentence have ambiguous meanings, which also causes errors, as in other semantic annotation tasks. In particular, errors may occur more often in sentences with more than one relation due to the issues mentioned above. In the following subsections we consider three subsamples of sentences, two from test folds of CLEF and one from the Fables dataset, to investigate the error types and the ways that model characteristics and data characteristics cause certain errors.

Error Types

To understand the nature of the errors (i.e. other than those from pipelining), we manually inspected over 10% of the errors, 50 wrongly labeled sequences from the largest data set CLEF. We selected the setting with a given ground-truth preposition to analyze problematic issues in classifying trajector and landmark roles and relation extraction. Table 5.16 categorizes the errors based on their cause and gives the percentage of each category in the random sample.

Class	Description	Percentage
1	A role element is classified as none	48%
2	Nesting spatial relations	24%
3	Spatial focus shift	10%
4	Irregularity in the grammar	10%
5	Errors in the annotated data	8%

Table 5.16: Error classes assessed on CLEF.

- **Class 1**. One frequent error assigns none labels to words that play spatial roles. This error originates from two sources: the lack of lexical information and the high prior probability of the none class compared to role-holder words, leading to lower recall of both trajectors and landmarks. The latter generally causes errors in experiments on the CLEF dataset. In the sentence below, *"woman"* is wrongly classified as none, which the latter issue causes.

  Example: *A* $[woman]_{Tr}$ *holding a plastic bag* $[on]_{Ind}$ *the left.*

  However, the first cause (i.e. the lack of lexical information) generally affects errors in the cross-domain experiments in section 5.4.6.

- **Class 2**. These errors are caused when the sentence expresses spatial relations that are more complex. In these cases, multiple trajectors are assigned to a preposition. In nested relations, the spatial relation has the transitivity property, so the assigned roles are semantically correct. However, we avoid spatial reasoning in the hand-labeled data, and these relations have not been annotated. The transitivity property of the relation depends only on the context, type of relation and its trajector and landmark entities. Injecting these more complex inputs makes the learning more difficult for the machine learning model, particularly when it lacks training data. These additional role assignments are classification errors and cause lower precision particularly in trajector labeling in our dataset.

  Example: *A dark-haired girl in a white T-shirt is sitting at a* $[desk]_{Tr}$ $[in]_{Ind}$ *a* $[classroom]_{Lm}$.

  With respect to the second *"in"*, only *"desk"*is the annotated trajector, though the classifier also classifies *"girl"* as a trajector. This assignment is semantically correct, but as described above, it does not match the ground-truth annotations.

- **Class 3**. This type of error concerns cases in which the transitivity property does not hold. A preposition trajector cannot semantically be a trajector of the next preposition in the sentence, but the landmark of the first relation is often the trajector of the next. In other words, the spatial descriptions' focus changes from one trajector to another. In these cases, a wrong trajector is assigned to a preposition and is related to a wrong landmark.

  Example: *More kids sitting at their desks and a* $[blackboard]_{Tr}$ $[in]_{Ind}$ *the* $[background]_{Lm}$.

  Depending on the context, one can infer that only the *"blackboard"* is in the *"background"* and the desks are not. Hence, *"background"* is a wrong landmark for both *"kids"* and *"desks"*.

- **Class 4**. The sentences' grammar causes this type of error, primarily the phenomenon of *semantic ellipsis.*

  Example: *A king size bed with the night* $[table]_{Tr}$ $[on]_{Ind}$ *the* $[side]_{Lm}$. Here, *"bed"* is classified as the trajector of *"on"*, while "table "is actually the trajector. In fact, *"side"* should be labeled as the landmark that actually refers to the side of "bed".
- **Class 5**. The annotator, not the classifier, causes these errors. This fact implies that accuracy can, to some extent, vary.

Error Analysis Cross Folds and Models (CLEF and GUM)

Adding the preposition template had inconsistent impacts on the performance of CRF's on different datasets. Particularly, this impact was greatly positive on trajector classification in CLEF and negative on landmark classification in GUM. This inconsistency encouraged to take a small subset of testing examples and compare the errors of two models (with and without a template) to address the effects of adding the templates on each type of error.

In the CLEF dataset, several sentences contain nouns and prepositions between the pivot-preposition and its related trajector. The sequential joint learning makes errors due to assigning "none" to these long distance trajectors. The template performs the first correction to handle these long distance trajectors properly in the skip-chain CRF. To quantify, 65% (11 of 17) of the errors in the checked subsample (100 instances) are in this category, leading to lower recall in the linear-chain CRF. Those errors belong to class 1, and most are corrected by the skip-chain CRF model. The following sentence is an example:

Example: *A dark-skinned, dark-haired* $[boy]_{Tr}$ *with a gray shirt is standing in a room* $[in]_{Ind}$ *front of a* $[wall]_{Lm}$ *made of red bricks.*

The linear-chain model labels *"boy"* as "none" with respect to *"in" (front of)*, which the skip-chain model corrects it to "trajector".

The second type of error includes cases in which two trajector labels are assigned despite there being only one actual trajector. The previous subsection classifies and explains these errors as classes 2 and 3. In this subsample, we see that the long distance noun is the actual trajector in 3 of 17 such cases. In 3 other cases, the noun immediately before the preposition is the actual trajector. These errors, totaling 6 of 17 (36%), lead to a decrease in both recall and precision.

Example: *There is a wooden commode and a mirror on the left, a wooden bedside table with a table lamp next to the bed and a huge* $[fan]_{Tr}$ *on the wall* $[above]_{Ind}$ *the* $[bed]_{Lm}$.

The linear-chain tagger labels both *“wall"* and *“fan"* as trajectors with respect to *“above"*, while the general skip-chain CRF correctly tags only *“fan"* as the trajector.

The only error made by the skip-chain CRF concerning trajectors in our subsample is the example below, in which the trajector *“boy"* is assigned a “none" label with respect to the second *“in"*, in the sentence:

Example: *A dark-skinned, dark-haired* $[boy]_{Tr}$ *in a very colorful pullover is standing in between two desks* $[in]_{Ind}$ *the* $[classroom]_{Lm}$.

This error is not typical but merely arbitrary, as there are similar cases in the test data that the skip-chain CRF model correctly classifies.

Furthermore, the improved model outperforms using the ground-truth spatial indicator in trajector classification. This finding was unexpected but not contradictory. In the ground-truth and pipeline settings, the correlations between indicators and other role labels are not considered, while joint learning uses these additional correlations between output variables. The template clearly increases the probability of assigning role labels (i.e. trajector/landmark) instead of a none label, with the additional probabilistic factor connecting distant nouns to the pivot-preposition; this process corrects the long distance words labeled as none and increases the recall of both trajectors and landmarks. This feature removes one cause of class 1 errors. Because landmarks are usually in prepositional phrases and close to the pivot preposition, modeling long distance dependencies contributes less than for trajectors. However, it still increases recall of landmarks. It may, however, introduce additional false positive landmarks, as in the following example:

Example: $[Tourists]_{Tr}$ *are standing* $[in]_{Ind}$ *the* $[classroom]_{Lm}$ *of a school in front of the blackboard.*

Here, both *“school"* and *“classroom"* are labeled as landmarks of the first *“in"*. The F-measure, therefore, has less improvement in landmarks than in trajectors.

In contrast to CLEF, sentences are short in GUM (Maptask), and modeling long-distance dependencies does not improve recall. Some cases lack trajectors because sentences contain directional instructions in which *“you"* is the implicit trajector. The skip-chain CRF thus only does equally well or slightly worse in trajector classification. Fitting the more complex model to the small amount of data in GUM (Maptask) lowers both the recall and precision of landmarks. Additional investigation of one fold of the errors in the skip-chain CRF of GUM (Maptask) shows that many landmarks are annotated as none because both occurring a specific noun as a landmark in the training data and the combination of a landmark with a specific preposition are important to the model. However, the linear chain CRF is less strict and annotates them correctly. The additional probabilistic factor makes the model tend to overfit the data, strengthening the effects of the lack of training data and lexical information. The incorrect “none" labels here assigned are more primarily due to the lack of training data than to the frequently occurring “none" labels in sentences. The additional template can, therefore, also introduce class 1 errors, but for a different reason than mentioned above.

Error Analysis Cross Domains and Models (DCP and Fables)

The lower performance of cross-domain evaluation and also 10-fold cross validation on Fables and DCP encouraged an investigation on the incorrectly classified sentences in these datasets. A sample is selected from Fables's test errors because it shows more problematic than DCP.

Most of the errors belong to class 1. The high prior probability of the “none" labels in the sequence of words is the main cause. Adding the preposition template in the skip-chain CRF model increases the errors of this type. The increased complexity of this model and the limited training data typically cause *overfitting*, i.e. the model adapts to the training data characteristics too strongly and does not generalize properly. This type of error is more problematic for trajector classification, whereas the landmarks are frequently in prepositional phrases and close to the indicators. Syntactical information thus helps achieve higher recall for there. If the indicator has been identified correctly, landmarks are more easily recognized than trajectors.

For trajector classification, due to the variety of trajector syntactical features, lexical information is more discriminative and useful for the model. In the example sentence below, in which gold labels are indexed, the trajector is incorrectly classified as “none" because the word *eagle* does not occur as a trajector in the CLEF dataset:

Example: *An* $[Eagle]_{Tr}$ *sat perched* $[on]_{Ind}$ *a lofty* $[rock]_{Lm}$*, keeping a sharp look-out for prey.*

The next example is another case in which none of the roles are recalled and all are labeled as “none". Because the type and context of the texts differ from Fables to CLEF, contextual features are ineffective.

Example: *A* $[huntsman]_{Tr}$*, concealed* $[in]_{Ind}$ *a* $[cleft]_{Lm}$ *of the mountain and on the watch for game.*

Conversely, exploiting grammatical features introduces more false positives and decreases precision for landmarks. The following sentence is an example of this phenomenon:

Example: *One touch from you and I should be broken* in *pieces.*

The model wrongly classifies *in* as an indicator and *pieces* as a landmark while *in* has no spatial sense. For this example, the semantic role labeler labels *in* as AM-LOC, which is also incorrect. Despite dissimilarities in the sentences' vocabulary and context, there are several cases where all roles have been labeled correctly. Their similarity to typical spatial description grammatical structures in CLEF accounts for this and the below sentence is an example:

Example: *There were two* $[Cocks]_{Tr}$ $[in]_{Ind}$ *the same* $[farmyard]_{Lm}$*, and they fought to decide who should be master.*

We also briefly study the errors made by the system in 10-fold cross-validation inside these datasets. The trajector and landmark classification precision is nearly 100% for both datasets, but recall is very low, signifying that the major problem is again insufficient evidence for assigning the roles, i.e. a lack of training data and particularly a lack of positive examples. If we compare the overall number of prepositions to the number of spatial prepositions, there are many more non-spatial prepositions per sentence in the Fables and DCP compared to GUM and CLEF, which leads to a stronger bias toward assigning none labels. Having an unbalanced dataset (with respect to positive and negative examples) is a typical challenge for relation extraction with machine learning. Overall, the error analysis in these experiments indicates the main issues for successful transfer of models across different domains. It also suggests ways to improve spatial role labeling systems in the future.

Because labeling data to train a model in each domain of interest is inefficient, we have shown one way in which to use existing resources to alleviate the annotation labor. Experiments in different domains present difficulties in cross-domain transferability and indicate that learned classifiers become biased to the distribution of features and words in the training dataset. However, exploiting more general resources, such as TPP, can help reducing this bias.

5.5 Related Work

Our general investigation shows that in computational models for spatial information extraction, mostly geographical information and *toponym resolution* are considered [132, 80], hence the universal cognitive elements have been paid a minor attention. However, these elements find more importance when understanding *domain independent* and *unrestricted* natural language is targeted as in our research. Few research works exist that consider both the computational linguistic problem and the abstraction of spatial concepts in their systems [122, 66]. Moreover, most of these works noticeably consider visual information resources and that is why the linguistic structures and features have been paid less attention in the related works.

In the related spatial language research from a cognitive-linguistic point of view, spatial prepositions, their semantics' variation, and grounding their perceived meaning have been thoroughly investigated [53]. In the visual context, applying computational spatial preposition models to a visually situated dialog system is investigated [64]. Lockwood et al. [86, 87] describe a model for learning to classify visual scenes according to the spatial preposition depicted. They use SEQL, an existing model for analogical generalization, to construct relational descriptions from stimuli input, such as hand-drawn sketches, and their suggested model can distinguish between *in*, *on*, *above*, *below*, and *left* after being trained on simple sketches exemplifying each preposition. These efforts are valuable but remain too limited to ground unrestricted spatial natural language perception. In our work, having a holistic view on the spatial meaning the automatic mapping to the prepositions' meaning is performed by exploiting the first level of mapping the language to spatial roles. This plays an important role in the semantic representation of spatial expressions in a domain-independent way.

However, the *preposition disambiguation* that we use in this chapter, has been introduced as a formal benchmark task in SemEval2007 [85, 153]. The importance of prepositions in meaning conveyance has been extensively investigated [1], and prepositions' dominant role in language semantics has been experimentally proven. This fact also explains why prepositional sense disambiguation has recently received much attention in semantic text analysis [104, 30, 140]. Exploiting preposition disambiguation in this work shows the benefits of this computational linguistic task in spatial relation extraction. Some related research have noticed these primitive spatial elements in visual contexts and processing locative phrases [2]. But automatic extraction of these elements from language has been noticed in few applications containing multimodal environments or in tasks that are occupied with visual information and visualization. There are few works that focus on the linguistic aspect, with notable exceptions [84, 83] (for the Chinese language). Their focus on extracting similar trajector and landmark elements attempts to visualize fable stories. However, their approach is limited to a binary classification of the trajector role. The landmark is extracted using limited background knowledge instead of a machine learning approach. Kollar et al. in a recent work [66] presented a system for interaction between humans and robots. The robot follows natural language directions by extracting a sequence of spatial description clauses (SDCs) from the linguistic input and infers the most probable path through the environment given information only about environmental geometry and detected visible objects. Their spatial description clauses contain elements including figure (trajector), verb, spatial relation and landmark. Grounding the flexible spatial language of directions in perception is interesting, but it essentially assumes that directional instructions are given, which renders it to be domain-specific understanding which does not need disambiguation.

None of these works formalized the complete task of domain-independent spatial role labeling using machine learning nor, did they pay attention to linguistically motivated features. The idea of defining spatial role labeling is inspired by the more general task of semantic role labeling [90], but we consider different thematic roles related to spatial semantics and argue that these semantics deserve particular attention.

Researchers have applied skip-chain CRF's for named entity recognition [142]. In the above mentioned work, Kollar et al. [66], also used a CRF model (to extract SDCs) but with different settings and feature functions. However, they have not been used in spatial information extraction in the sense that we define. There are a number of related works that exploit machine learning models in restricted spatial settings. For example, Reinbergerr [117] presents an unsupervised method to extract spatial prepositional phrases from text corpora and use the output as preprocessed material to build a virtual environment. They use a shallow parser and select functional relations from which they can extract spatial information. That work manually evaluates the adequacy of the extracted relations. Another work transforms a textual description of a spatial scene in a sequence of prepositions into a graph with objects, annotating local reference systems as nodes and relations as arcs [23, 165]. Inference is realized by multiplying transformation matrices, constraint propagation and verification using machine learning techniques. By assigning values to the parameters and using heuristics for object placement, a visualization of the described spatial layout is generated from the graph. They also consider a limited set of predefined relations.

Spatial relations are also important in semantic image analysis. In one work, eight fuzzy directional relations, such as *right*, *left* and *above*, are supported [107]. All relations are evaluated for each pair of objects in the image. That work presents a learning approach, coupling support vector machines (SVMs) and a genetic algorithm (GA), for knowledge-assisted domain-specific semantic image analysis. There are also many challenges in video analysis, handling spatial and temporal relations and extracting those relations from video; an interesting work proposes a framework for learning object and event categories from video [139]. The work exploits graphical models, and spatio-temporal patterns in the video are represented using an activity graph.

To interpret spatial language for following navigational directions, a system is presented that does not use semantic annotation, but instead learns from human demonstration on the Maptask corpus [161]. In this work, a reinforcement learning setting derives the correspondence between the instruction language and path features. On the same corpus, an earlier work first manually maps the spatial language to conceptual NIUs (navigational information units) [82]. The combination of NIUs is then automatically interpreted as a spatial path using dynamic programming. The linguistic part of NIU extraction is ignored there. The same authors in a recent work start from natural language and map it to SDCs [66].

Moreover, several systems extract information directly from text and determine spatial relationships between objects in a 3D scene to generate such scenes from these textual descriptions. These systems consider the semantic models of spatial relations and their computational implementation. However, they are restricted to simple narratives, often invented by the authors, and do not consider a real corpus. For applying machine learning usually a limited number of relations is defined to keep the problem tractable. A more general overview of older vision and language systems can be found in [63].

According to our overview, in the related research in this domain, restricted languages extract very specific and application-dependent relations from text [63, 147, 84]. Previous research has not systematically covered spatial relation and role extraction from *unrestricted* natural language with machine learning methods, but we do so.

5.6 Conclusions

This chapter encompasses the first extensive computational investigation of the SpRL as a novel computational linguistic semantic task. We point to the unique characteristics and challenges of this task and design various learning models mostly based on probabilistic graphical models, namely CRFs. Performing experiments on various datasets and models we showed that a) Existing semantic role labeling models and also using dependency trees can perform very poorly for labeling the spatial roles and extraction of the spatial relations; b) Using machine learning for such a task can improve those baselines to a great extent; c) Context dependent models such as linear-chain CRFs improve the simple classification models; d) Exploiting long distance dependencies by modeling general CRFs improves the training from a larger dataset while showing over-fitting side effects on our smaller dataset; e) Cross domain evaluations showed a sharp drop in the performance of the models when tested on a different domain, which is due to the high influence of the lexical information in this semantic task; in this case, exploiting the external resources with the support of very large annotated data (TPP) for preposition disambiguation improved the SpRL results; f) The error analysis shows additional challenges to the critical lexical information, such as the difficulty of the relation extraction in the long sentences containing multiple spatial relations; g) The feature analysis shows the positive influence of the linguistically motivated and relational features, however the more complex features such as *path* were not useful particularly in the smaller datasets.

The relational nature of spatial role labeling and the difficulty in extraction of the relations compared to the roles motivated the use of more flexible relational learning models that can exploit background knowledge in a flexible setting for SpRL. These are the subject of the next chapter in this thesis.

Chapter 6

Relational Learning

The first experimental investigation of spatial role labeling in Chapter 5, using CRFs, highlighted the importance of the contextual features and the relational nature of this problem. This is the motivation of exploring the statistical relational learning models for SpRL. Many natural language processing problems, including SpRL, require one to deal with the underlying structure of the data, to employ knowledge about the domain such as ontologies, and to impose constraints on the output. Therefore, a relational formulation of SpRL is encouraged and many related problems can be treated as such [29].

SpRL is a well-fitting problem for relational learning models. We can treat the words in the natural language sentences as objects that have their own properties and also relationships to each other. Then SpRL is about extraction of the spatial relations between these objects. The goal of this chapter is to employ an expressive logical representation and relational model of SpRL. We utilize kLog [42], a novel framework with an expressive first order logical representation of the relational data and the background knowledge. It supports describing learning problems in a declarative, relational fashion [31]. Moreover, it applies a powerful and flexible graph kernel that has a high potential for capturing long distance dependencies in the data using extensive relational input features. We investigate a number of models represented in this framework, such as simple binary classification of the predicates with spatial role arguments, pipeline models and sequence tagging for SpRL. We formulate the learning model and our knowledge about the problem in a declarative way.

This chapter is structured as follows. In Section 6.1, the relational formulation of spatial role labeling in kLog and the features are defined formally, moreover the kLog framework itself is briefly introduced. In Section 6.2, we discuss two ways of modeling the learning problem: as triplet classification (i.e. hyperlink prediction) and sequence tagging. Section 6.3 reports on experimental results that show the promise of our approach, Section 6.4 points to the related work, and finally Section 6.5 concludes.

6.1 Relational Problem Statement

In Chapter 5, the SpRL problem was described and motivated, hence we directly state the relational formulation of the problem here. As before, the input is a natural language sentence S that is a sequence of T words $S = \langle x_1, x_2, \ldots, x_T \rangle$. In a relational learning setting, we assume each word is an entity with a number of properties and these entities can have relationships to each other. A *spatial relation* is presented as a predicate with three arguments `sr(SP, TR, LM)`. Three spatial role predicates are defined as `indicator(SP)`,`trajector(TR)`,`landmark(LM)` where `SP`$= x_i$, `TR`$= x_j$, `LM`$= x_k$, $i, j, k \in [1, T]$ and for each spatial triplet $i \neq j \neq k$. Each sentence contains $n \geq 0$ spatial relations. Each word is an argument of the spatial role predicates when it carries a specific spatial role. As in the previous chapter, for any spatial relation, when no word in the sentence S represents a trajector or a landmark, then the value of those arguments is “undefined”. In general, spatial indicators, trajectors and landmarks can be arbitrary segments that contain more than one word, but as in the previous chapter, we focus on individual words; namely the syntactical head word of a segment. We recall the example in Chapter 5:

A girl and a boy are sitting at the desk in the classroom.

Here we can roughly (i.e. using the actual word forms instead of using a unique identifier for each word which is discussed later) represent the spatial relations as: `sr(at,girl,desk), sr(at,boy,desk)`. Depending on the learning model these spatial roles can be predicted as intermediate steps,

```
trajector(girl) trajector(boy)
landmark(desk) landmark(classroom).
```

And the example with an implicit trajector is the following,

Go under the bridge.

In this sentence the target spatial relation is: `sr(under,undefined,bridge)` and the spatial roles are `indicator(under)` and `landmark(bridge)`.

Before describing the relational input features and modeling the problem, the kLog framework and language is discussed in the following subsection.

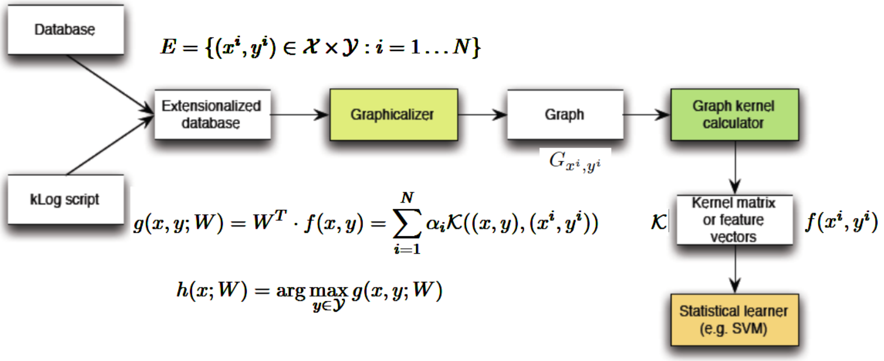

Database
$E = \{(x^i, y^i) \in \mathcal{X} \times \mathcal{Y} : i = 1 \dots N\}$
Extensionalized database
Graphicalizer
Graph
Graph kernel calculator
G_{x^i,y^i}
kLog script
$g(x, y; W) = W^T \cdot f(x, y) = \sum_{i=1}^{N} \alpha_i \mathcal{K}((x, y), (x^i, y^i))$
$\mathcal{K}$
Kernel matrix or feature vectors
$f(x^i, y^i)$
$h(x; W) = \arg\max_{y \in \mathcal{Y}} g(x, y; W)$
Statistical learner (e.g. SVM)

Figure 6.1: kLog flow.

6.1.1 Relational Representation and Learning in kLog

kLog [42][1] is a language designed for relational learning. Fig. 6.1 presents the workflow in kLog. kLog allows users to specify a relational database, background knowledge and the target problem in a declarative way. The main elements of the data model are *entities* and *relationships*. A kLog script contains kLog's *signatures* that represent the format of each table in a relational database. This is to some extent similar to the notion of *bias* in inductive logic programming. The data structure is naturally represented as an *entity-relationship* (E/R) diagram like the one shown in Fig. 6.2.

The purpose of kLog is to make it easy to define and maintain relational *features*. kLog is a domain-specific language embedded in Prolog, a language with both a declarative and a procedural semantics. Hence, feature definition in kLog lies somewhat between purely declarative (as in Markov logic [38]) and purely imperative (as in FACTORIE [92]) approaches.

kLog learns from *interpretations* [31], where an interpretation is essentially a set of ground atoms (or a set of tuples, since structured terms are not allowed in the language). Ground atoms can be either explicitly given as data (for *extensional* signatures) or deduced using Prolog's deduction mechanism (for *intensional* signatures). Intensional predicates are akin to Datalog rules [46] and tabling can also be used to avoid Prolog's procedural semantics, if desired. Under mild assumptions, grounding the E/R diagram (a process we call *graphicalization*) yields, for each interpretation, a bipartite graph whose nodes are ground atoms either of entity or relationship type, cf. Fig. 6.3. An edge from a relationship vertex to an entity vertex is created if the identifier

[1] http://www.dsi.unifi.it/~paolo/ps/klog.pdf

of the entity appears as an argument of the ground relationship atom. A graph kernel is subsequently used to compare interpretations by comparing the associated graphs. Unlike the approach in [162], kLog does not require a kernel on hypergraphs and any existing graph kernel can – in principle – be used. The current implementation uses a modified version of the *neighborhood subgraph pairwise distance kernel* (NSPDK) [28] which we describe briefly here.

NSPDK. It is a decomposition kernel, in which the similarity between graphs are calculated based on their subgraphs. The subgraphs are produced based on three given parameters namely, kernel points, radius and distance. A kernel point is the center of a subgraph, which can be any entity or relation in the graph. Given the radius r each entity or relation that is within a number of r edges away from the kernel point is considered as a node of the subgraph. Given the distance d, each subgraph around a kernel point that is within a distance d or less from the current kernel point will be considered. This is denoted by the relation $R_{r,d}(A_v, B_u, G)$ between two rooted subgraphs A_v, B_u and a graph G, which selects all pairs of neighborhood graphs of radius r whose roots are at distance d in a given graph G. The kernel $\kappa_{r,d}(G, G')$ between graphs G and G' on the relation $R_{r,d}$ is then defined as:

$$\kappa_{r,d}(G,G') = \sum_{\substack{A,B \in R^{-1}_{r,d}(G) \\ A',B' \in R^{-1}_{r,d}(G')}} \delta(A_v, A'_{v'})\delta(B_u, B'_{u'}), \tag{6.1}$$

where δ is a similarity function and has two versions of hard match (i.e. exact graph match) and soft match in kLog. For efficiency reasons, an upper bound is imposed on radius and distance (i.e. r^*, d^*) leading to the following kernel definition:

$$K_{r^*,d^*}(G,G') = \sum_{r=0}^{r^*}\sum_{d=0}^{d^*} \kappa_{r,d}(G,G'). \tag{6.2}$$

NSPKD is computed by creating *feature vectors* associated with the interpretations. Any model can then be trained using the final feature vectors that are produced by the NSPKD kernel.

Underlying classifiers. kLog is agnostic about the statistical procedure used to learn from the constructed feature vectors and several alternative models can be plugged-in. In this work, the underlying models we use are a standard binary support vector machine (SVM) implementation of LIBSVM [19] and a structured SVM implementation for sequence tagging of SVM-HMM.[2] The LIBSVM implementation provides various SVM

[2] http://www.cs.cornell.edu/people/tj/svm_light/svm_hmm.html

formulations that are all solved in the dual space. All these formulations are standard and well-known. The specific formulation that is used by default in our reported results is with the one called C-SVC in LIBSVM and we use it with a linear kernel. Other SVM implementations that work in the primal, such as stochastic gradient descent (SVM-SGD), are also integrated and can be called in kLog scripts. We avoid describing the standard binary classification formulations of SVM, but briefly point to the sequence tagging model of SVM-HMM here.

SVM-HMM. This model is used in conjunction with kLog in the current experiments. SVM-HMM is an extension of a SVM, for sequence tagging. It uses a similar linear model described in Chapter 2 as its decision function: $h(x) = \arg\max_y W^\top \cdot f(x, y)$, where $f(x, y)$ is the feature vector associated with each interpretation, (x, y), consisting of input (x) and output (y) ground atoms. A potential function equal to $W^\top \cdot f(x, y)$ is used to score the interpretation. Prediction, is the process of maximizing g with respect to y. Here, we directly exploit the sequence tagging implementation of SVM-HMM to solve the SpRL problem in kLog. SVM-HMM uses a *structural* (SVM) formulation and discriminatively trains models that are isomorphic to a kth-order hidden Markov model. In summary, given an observed input sequence $x = \langle x_1 \dots x_T \rangle$ of feature vectors over $x_1 \dots x_T$, the model predicts a tag sequence $y = \langle y_1 \dots y_T \rangle$ according to the following linear discriminant function,

$$h(x) = \arg\max_y w_{\text{emis}}^\top \cdot f_{\text{emis}}(x, y) + w_{\text{trans}}^\top \cdot f_{\text{trans}}(y), \tag{6.3}$$

where w_{emis} and w_{trans} are the emission and the transition weight vectors, respectively. The built-in NSPDK graph kernel in kLog produces the emission feature vectors $f_{\text{emis}}(x, y)$ and those are used by SVM-HMM.

6.1.2 Representing SpRL in kLog

The data model defines the entities and relationships using an E/R diagram. Fig. 6.2 shows the diagram used for SpRL. The main entity of our model is a word. Word properties are obtained using the same approach of Chapter 4. The difference here is that a first order formalism is used for representation of the data. Each word x_i is assigned an identifier `W_id` and the properties are represented as follows:

- The $\phi_{wf}(x_i)$, is represented as `word(W_id,Word_form)`.
- The $\phi_{pos}(x_i)$, is represented as `pos(W_id,Word_pos)`.
- The $\phi_{dprl}(x_i)$, is represented as `dprl(W_id,Word_dprl)`.

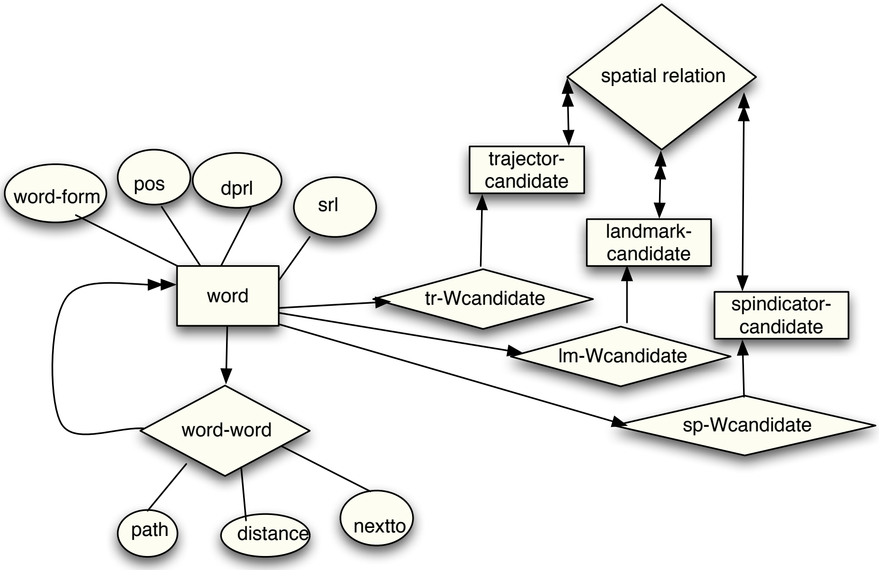


Figure 6.2: The E/R diagram.

- The $\phi_{srl}(x_i)$, is represented as `srl(Word_id,Word_srl)`.
- The $\phi_{sub}(x_i)$, is represented as `subcat(Word_id,Word_subcat)`.

In addition to the sequential relationships, a set of relational features between the words x_i and x_j with identifiers `w_i` and `w_j` in the sentence are exploited:

- The $\phi_{path}(x_i, x_j)$, is represented as `path(w_i,w_j,Path)`.
- The $\phi_{before}(x_i, x_j)$, is represented as `before(w_i,w_j)`.
- The $\phi_{dis}(x_i, x_j)$, is represented as `distance(w_i,w_j,Distance)`.

Each sentence is represented as an interpretation. This interpretation is equivalent to the unfolded (grounded) E/R diagram represented as a bipartite graph such as Fig. 6.3. This figure shows an example with kLog's graphical representation of the relational features for the sentence *"The kids are on the stairs."*. It shows a part of the facts that are stored in the relational tables. In addition to the stored database of facts which is called an extensional table, a kLog script is prepared that contains the signatures of the extensional tables as well as the logical rules that define how more facts can be deduced for the learning model. These rules deduce the intentional tables. We describe these two sources of data with examples here.

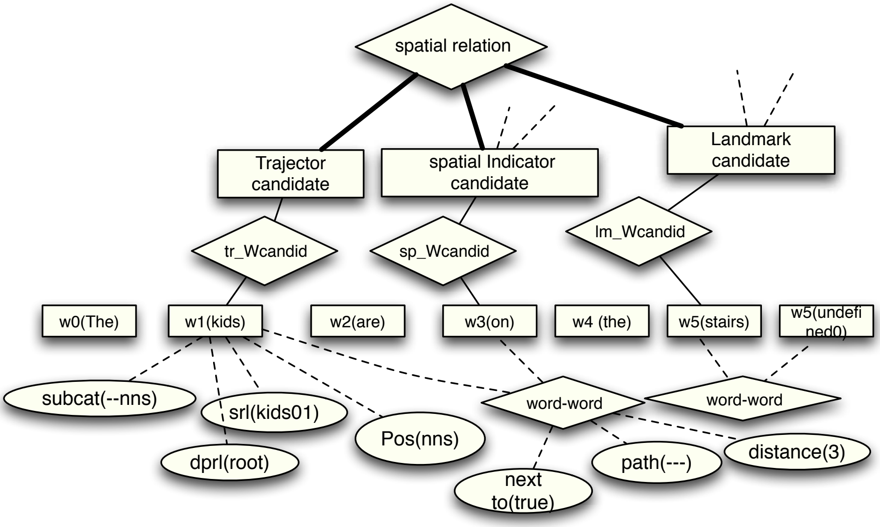


Figure 6.3: Grounded E/R for one interpretation.

Extensional tables. The given facts are stored directly in the database, in so-called extensional tables. In the above example some of these are:

```
word(w0,the). word(w1,kids). word(w2,are). pos(w2,vbp). pos(w3,in).
pos(w4,dt). dprl(w0,nmod). dprl(w1,sbj). dprl(w2,root). dprl(w3,prd).
dprl(w4,nmod). path(w1,w3,'NNS^NP^S_VP_PP_IN'). distance(w1,w3,2).
path(w5,w3,'NNS^NP^PP_NN'). distance(w5,w3,3).
```

Intentional tables. In kLog, logical rules can be used (as in Prolog) to define intensional relations and induce new features. For example, it is not needed to provide the *next* relation between adjacent words extensionally because it can be derived from the word identifiers using the following clause:

```
next(W1,W2):- word(W1,_),word(W2,_),W2 is W1+1.
```

This produces the following tuples in the data base, in so-called intensional tables:

```
next(w0,w1).next(w1,w2).next(w2,w3).next(w3,w4).next(w4,w5).
```

Background knowledge. In a kLog script, background knowledge is represented explicitly using logical rules in the problem specification and independent of the underlying machine learning model. In SpRL, in its binary classification formulation, to avoid an explosion in the number of negative examples, we limit the set of example spatial relations to a number of candidates. This is performed using a set of logical rules that we call **candidate selection rules**. In more details, trajectors and landmarks are mostly noun phrases and spatial indicators are often prepositions. To exploit this knowledge in kLog, for example, the following candidate selection rules for selecting trajector candidates are provided:

```
trajector_candidate(W):-
     word(W,_),pos(W,POS),
     member(POS,[nn,nns,prp,nnp,nnps]).
trajector_candidate(W):-
     word(W,'undefined').
```

Thus, a candidate trajector is either a noun or undefined. This is applied similarly in candidate selection for landmarks and spatial indicators. The rules produce new relational entities and intentional tables. For example applying the candidate selection rules on the above sentence yields:

```
indicator_candidate(w3).
trajector_candidate(w1).trajector_candidate(w5).
landmark_candidate(w1).landmark_candidate(w5).
```

Finally, we are able to generate the relational target of the learning problem using the logical language. This provides a flexible way to represent any structured output prediction in a relational form. The target formulation is discussed in more details in the next section. In the above example, the target roles include

```
indicator(w3). trajector(w1). landmark(w5).
```

and the target spatial relation is `sr(w3,w1,w5).`

This resembles a multi-predicate learning task [31] in which the predicates are nodes in the graphs (see 6.3). The graphs are turned into feature vectors using a graph kernel, which ultimately leads to a propositional learning problem in a high dimensional space.

6.2 Spatial Relation Extraction

In this section we discuss two different formulations of spatial relation extraction. The first formulation is a binary classification of ternary relations. That is, for every possible triplet of words in a sentence, we hypothesize a relation between them and classify it as being spatial or not. In the second formulation, we perform relational sequence tagging and then construct the target spatial relations using a set of logical rules.

6.2.1 Problem Formulation I: Triplet Classification

According to the problem statement, each sentence is associated with a set of true spatial relations in the form of `sr(SP,TR,LM)`. A direct formulation of extracting the `sr` predicates includes the following processes: a) Generate the possible triplets of words per sentence; b) Consider each produced true triplet as a positive and each false triplet as a negative example; c) Compute the features of each example; d) Classify each triplet as true or false based on its computed features. In kLog, the candidate triplets can be easily specified declaratively and generated. The triplets that are in the database are positives and the negatives are produced automatically based on the closed world assumption. Each triplet atom is a relational node in kLog's bipartite graph. The triplet's features contain the connected nodes in its neighborhood. The neighborhood means the scope of a given radius and distance parameters of kLog's graph kernel. In this formulation, the learning problem boils down to binary classification of these atoms using the relational features and the propositionalization process. The bottleneck of this approach is the vast number of possible negative examples as compared to the positive ones. We exploit the background knowledge and also a pipelining approach to deal with this problem. Due to the use of a logical representation in kLog, different settings are represented in the problem specification side, independent from the learning model.

Candidate Selection.

Instead of generating all combinations of arbitrary words, the candidate selection rules described in Section 6.1.2 are used to reduce the number of examples. Hence, the candidate triplets are produced by choosing their arguments from the previously defined candidate roles using the following rule:

```
sr_candidate1(I,T,L) :- indicator_candidate(I),
          trajector_candidate(T),landmark_candidate(L).
```

The predicate `sr_candidate1(I,T,L)` includes the positive as well as the negative examples; the positives are those for which `sr(I,T,L)` holds in the sentence and satisfy the candidate selection rules, e.g. POS tag restrictions. Since the coverage of candidate selection rules is not 100%, some actual positive examples may not be `sr_candidates1`.

Pipelining.

Another approach is to separately train three models to predict the spatial roles first and use the predicted roles to construct the examples for learning the sr predicates. In this case the positive and negative examples are produced using

```
sr_candidate2(I,T,L) :- indicator_predicted(I),
          trajector_predicted(T), landmark_predicted(L).
```

The newly introduced predicates such as indicator_predicted(I) are added to the database after the spatial role prediction. A high recall of this step is required to achieve a reasonable performance for sr prediction in the second phase. To this aim a threshold is adjusted using the precision-recall curve to increase the number of sr_candidates2. In this setting, the learning procedure becomes a stacked pipeline where the second stage of sr(I,T,L) prediction receives as examples the ground atoms for sr_candidate2 predicted in the first stage. kLog supports the implementation of such a layered learning approach using its scripting language.

6.2.2 Problem Formulation II: Relational Sequence Tagging

To consider the sequential correlations between spatial roles, the SVM-HMM model is used under kLog's representation. During the role prediction for each word the role of its adjacent words are considered. This model receives the input feature vectors created by kLog's graph kernel. As discussed in chapter 5, the sequence tagging formulation is not straightforward for SpRL because each word can play various roles with respect to various spatial indicators in a sentence. However, for one specific spatial indicator each word carries at most one role. This realistic assumption is used to set up relational sequence tagging. Each interpretation is presented as a sequence of word entities word(w_1,Wordform),...,word(w_n,Wordform) that are connected with the next(w_i,w_j) relation and one spatial-indicator candidate is chosen as the pivot of the interpretation. The word entities are assigned a role based on the selected pivot.

A sentence with k spatial indicator candidates generates k examples, each having a different pivot. In this setting a new predicate role(w_i, Role) is defined such that 1) role(w_j,spatial_indicator) is true for at most one w_j; 2) Each word w_i is assigned the property trajector or landmark, if it carries that role with respect to the spatial indicator w_j; and 3) the remaining words are assigned word(w_i,none). If the pivot is not a spatial indicator, all words in the example will have role(w_i,none).

Target generation rules. Assuming a fixed pivot implies that pairwise relations are obtained immediately after the tagging phase. We can declaratively program in kLog to produce sr relations using the below rules as a post-processing step to the learning and prediction. These rules we name *target generation rules.* These rules construct the ternary relations based on the predicted components in the sequence tagging.

```
sr_predicted(I,T,L):-pivot(I),indicator_predicted_seq(I),
   trajector_predicted_seq(T),landmark_predicted_seq(L).
sr_predicted(I,T,undefined):-pivot(I),indicator_predicted_seq(I),
   trajector_predicted_seq(T),word(W),
   \+landmark_predicted_seq(W).
sr_predicted(I,undefined,L):-pivot(I),indicator_predicted_seq(I),
   word(W),\+trajector_predicted_seq(W),
   landmark_predicted_seq(L).
sr_predicted(I,undefined,undefined):-pivot(I),
   indicator_predicted_seq(I),word(W)
   \+trajector_predicted_seq(W),\+landmark_predicted_seq(W).
```

The newly introduced predicates such as `indicator_predicted_seq(I)` are added to the database after the sequence tagging phase. If the role of trajector or landmark is not produced then the related argument in `sr` is 'undefined', cf. the example in Section 6.1.

In Section 6.3 the experimental results using SVM-HMM are presented and compared to their counterpart linear-chain CRF model.

6.3 Experiments

In this section, the experimental setup and the results for SpRL are presented. For the described problem formulations, the following experimental questions are investigated:

Q6.1. Can we get reasonable results using a simple binary predicate classification in kLog's relational learning for extraction of the true spatial roles and spatial relation predicates, given its kernel graph and the relational features?

Q6.2. What is the influence of using *background knowledge* in the form of candidate selection rules for classification of the spatial role and spatial relation predicates?

Q6.3. What is the effect of pipelining the classification of spatial role predicates and spatial relation predicates?

Q6.4. Does using SVM-HMM instead of binary SVM in classification of the role predicates improve the pipeline model?

Q6.5. Does using relational multiple sequence tagging improve over binary predicate classification and the pipeline model?

Q6.6. What are the benefits of relational representation and using sequence tagging in kLog compared to CRFs according to this case study?

Dataset. Our corpus in this chapter is a version of CLEF which has been used for the SemEval-2012 and described in Chapter 3.

Evaluation. The evaluation is based on 10-fold cross validation as explained in Chapter 4. For unary roles this is the classic evaluation based on true/false positive/negatives for each role. For the end-to-end evaluation of `sr(I,T,L)`, the final false/true positive/negative generated triplets are counted against the ground truth `sr`'s in the whole kLog's database.

6.3.1 Triplet Classification I

In these experiments, a kLog script produces sets of positive and negative triplets per each sentence to train a model for `sr` prediction. The experimental questions **Q6.1-Q6.4** are considered and to produce triplets the following options are examined: 1) producing all possible triplets; 2) using candidate selection rules; 3) predicting spatial roles and pipelining.

For predicting the spatial roles in the pipeline the use of SVM and SVM-HMM is examined. We describe the spatial role prediction first because it is used in other steps.

Predicting Spatial Role Predicates

In the first experiment, referred to as Model1, a model containing three independent binary SVM models is trained **using all words** as individual training examples. These form the baseline for unary predicates `trajector(W)`, `landmark(W)` and `indicator(W)`. In these models each word is an example. The total number of words in the data set is $21,308$. The number of positive roles trajectories, landmark and indicators are 1468, 1593 and 1462 respectively. The words with no role are negative examples. In spite of an imbalanced distribution of positives and negatives, the spatial role prediction is feasible with a reasonable accuracy, see Table 6.1. The values up to two decimal points are significant with 95% confidence interval.

In the second experiment, referred as Model2, the **candidate selection rules** (Section 6.1.2.) are used and this reduces the number of negative examples to a great deal, from 21308 to 7195 for trajector/landmark candidates (about 66%) and to 2903 for spatial indicator candidates (about 86.4%). However, by applying these rules, on average 11% of positive roles are not covered. In the Model2 setting also three independent binary SVMs are used. The results indicate a statistically significant improvement in the classification of candidate words compared to using all words ($p < 0.05$). In the third experiment referred as Model3, the binary classifiers in Model2 are replaced by SVM-HMM. SVM-HMM considers the

Model	Target predicate	F1	Precision	Recall	#examp.
Model1	`trajector(W)`	0.68	0.67	0.68	21308
	`landmark(W)`	0.68	0.64	0.71	21308
	`indicator(W)`	0.81	0.81	0.81	21308
Model2	`trajector(W)`	0.72	0.73	0.72	7195
	`landmark(W)`	0.74	0.73	0.76	7195
	`indicator(W)`	0.90	0.90	0.91	2903
Model3	`trajector(W)`	0.77	0.79	0.76	7195
	`landmark(W)`	0.86	0.88	0.85	7195
	`indicator(W)`	0.94	0.95	0.92	2903

Table 6.1: Unary spatial role prediction for word entities in kLog, 10-fold cross validation.

sequential relationships between words in the unary spatial role prediction (`trajector(W), landmark(W), indicator(W)` in separate models vs. none). Considering these correlations improved the results significantly, see Table 6.1.

Spatial Relation Prediction

If all triplets of words are considered as possible spatial relations, the number of examples produced for `sr` prediction will be equal to the cube of words per sentence $(= 12,346,353)$, while only 1,716 of these are positive relations. Hence, due to the huge number of negative examples, training a practical model is not feasible. By applying **candidate selection rules**, the number of triplets is reduced from 12,346,353 to 190,740. However, the disadvantage is that about 252 of positively annotated triplets are not covered and missed. This means 15% of the annotated relations that yield 1,464 positive triplets for this setting. The results of `sr` prediction in different experiments are presented in Table 6.2. The details of the settings are described in the following paragraphs.

sr_1 shows the results of triplet classification in one step by using our default binary SVM. The candidate selection rule (i.e. the predicate sr_candidate1 defined in Section 6.2.1) is used to produce examples. This is the baseline of triplet classification.

sr_1_1 exploits the predicted spatial role predicates in learning the `sr` prediction model. The role predicates are added to the database and consequently to the interpretations. Then those are used as new features attached to the words for the triplet classification. Once more the sr_candidate1 is used to generate examples yet with the difference that predicted roles such as the trajector_predicted are employed as new features.

Target predicate	F1	Precision	Recall
`sr_1(W,W',W")`	0.52	0.50	0.55
`sr_1_1(W,W',W")`	0.54	0.50	0.58
`sr_2(W,W',W")`	0.60	0.48	0.80
`sr_2_1(W,W',W")`	0.50	0.48	0.52
`sr_3(W,W',W")`	0.71	0.68	0.74
`sr_3_1(W,W',W")`	0.58	0.68	0.50

Table 6.2: Classification of *spatial relation* predicates (by one step, two pipeline steps of roles and relations, two pipeline steps of roles and relations when SVM-HMM is used for the roles), 10-fold cross validation.

The use of these features improved the results of sr_1_1 compared to sr_1 in Table 6.2 by 2% ± 1% with a 95% confidence interval for the F-measure.

sr_2 presents the performance of the relation classification in the pipeline model. In this experiment the first step of the pipeline uses the candidate selection rules and the binary SVM. Given the recall of spatial role prediction in Table 6.1, using these assignments for selecting candidates leads to a large miss about 40% of the true positives. However, we did stacking over the role classification and triplet classification steps. The sr_candidate2 predicate is used to generate examples for the second layer, see Section 6.2.1. The results of stacking are presented in sr_2 line in Table 6.2. These indicate a 6% improve in the F-measure compared to the one step relation classification. The missed positives are ignored in this line of reported results which implies a perfect system is needed in the first phase of pipelining to obtain this overall performance.

sr_2_1 evaluates the whole pipeline system starting from the input sentences. This end-to-end evaluation indicates a decrease in performance compared to sr_1. This is due to the propagated errors from the spatial role prediction step.

sr_3 aims to improve the performance of the pipeline by reducing the errors of the first step. It uses the stronger model SVM-HMM for spatial role prediction. The triplet classification is performed using the same binary SVM. Due to the improvement made in the first stage, the triplet classification in the second phase significantly improves compared to sr_2.

sr_3_1 examines the end-to-end evaluation of the pipeline model of sr_3. Compared to the counterpart of this experiment sr_2_1, an 8% ± 0 improvement is observed. This model performs 4% ± 0 better than sr_1_1 too (with a 95% confidence interval).

An overall comparison according to the results in Table 6.2 indicates that the pipeline system that uses the candidate selection rules along with SVM-HMM for the role prediction and then SVM for the second phase of triple classification is the best model in the above experiments. This answers the questions **Q6.1- Q6.4**.

6.3.2 Sequence Tagging for Relation Extraction II

The goal of this section is to investigate the experimental questions **Q6.5-Q6.6**. The main difference of these experiments compared to the last ones is that the direct target of the learning model is *relational sequence tagging*. This is performed using the sequence tagging technique of SVM-HMM [155] plugged into kLog. This type of modeling for SpRL is described in Section 6.2.2. The setting is without candidate selection unless for the spatial indicators. The number of examples in this experiment is equal to the number of candidate prepositions; that is about 2903 tagged sequences. The predicate `role(w_j,R)` is predicted by the sequence tagger. This predicate assigns a relational spatial role `R` to each word with respect to a predefined pivot in each sequence. Spatial triplets derived by a set of rules and the predicate `sr_predicted(I,T,L)` are produced directly based on the predicted relational roles in the sequence. Producing the interpretations in kLog for this experiment is described in Section 6.2.2.

kLog (SVM-HMM)				Linear-chain CRF			
Target pred	F1	Prec	Rec	Target pred	F1	Prec	Rec
`trajector(W)`	0.71	0.75	0.68	`trajector(W)`	0.79	0.83	0.76
`landmark(W)`	0.87	0.89	0.85	`landmark(W)`	0.88	0.92	0.84
`indicator(W)`	0.93	0.92	0.93	`indicator(W)`	0.94	0.92	0.96
`sr(W,W’,W")`	0.60	0.57	0.63	`sr(W,W’,W")`	0.65	0.65	0.65

Table 6.3: Results of CRF, SVM-HMM, 10-fold cross-validation.

Even by ignoring the missed positives in candidate selection of the triplet classifier, relational sequence tagging performs best for this task, see Table 6.3. In Table 6.3, we present the experimental results of our previous work [74] using CRFs too. The current results are not better but fairly comparative to the results of CRFs. Given the flexibility that is provided by kLog for declaratively representing the problem and considering the use of background knowledge these results are promising.

6.3.3 Experimental Analysis

Comparing the two main sets of experiments indicates the relational sequence tagging assuming a pivot is the best model which is molded using both CRFs and kLog. Although the results of applying CRFs are outperforming, kLog provides a declarative language to present the model, including generating examples for train/test from the database, performing relational feature engineering and generating the structure of the output, independent from the underlying learning model. These are promising answers to question **Q6.6**.

The performance of triplet classification was low due to the huge number of *negative examples*, even in the candidate selection setting. Pipelining and stacking have a low performance due to the *error propagation* between the spatial role prediction and the `sr` prediction phase.

In the following paragraphs we provide a brief analysis on the relational representation of the features and the flexible way that kLog deals with them in the graphicalization process and propositionalization. A more extensive analysis for the influence of the linguistic features and the errors is given in Chapter 5 within the propositional modeling.

Feature Analysis. We performed an experimental analysis of the features by gradually incorporating them in the training phase, the results were not significantly different than the feature analysis in Chapter 5. In the triplet classification, even after using candidates a huge number of negatives remained. Therefore, less *dense graphs* (i.e. with many edges) are used for the sake of efficiency. In these experiments the path and distance features are not used, but only the “next" relation connects the word entities in each interpretation. Path and distance are used in the sequence tagging approach.

As mentioned in Section 6.1.1, in the process of graphicalization and generating the propositionalized features in kLog, one can easily set two parameters called *radius* and *distance*. These guide the graph kernel in selecting the pairwise subgraphs for computing the kernel matrix. Intuitively this allows us to indicate the size of the relational features and an upper bound for the distance in the graph to be considered in feature generation. We observed that the “next" relation has a strong increasing influence in classification of the roles and consequently in classification of `sr` relations. This improvement was 6% for trajectors, 3% for indicators and 2% for landmarks. However increasing the distance and radius for generating relational features does not always increase the performance and obviously these parameters can be optimized by using a validation set. One other observation was that involving the predicted trajector roles improved the prediction of landmarks by 3% and the mutual influence of landmarks was also positive.

Error Analysis. We provide an extensive description of the error types for spatial role labeling in Chapter 5 with graphical models. Related to the kLog experiments, one main source of error in both triplet classification and sequence tagging is the relatively large number of negatives compared to the positives due to having a structured output i.e. a set of triplets. This leads to a bias towards no role assignment to the words in multi-class sequence tagging and also to classifying the triplets as negative in the triplet classification. However, this issue is more problematic in triplet classification and dramatically decreases the performance. We performed some experiments using a random selection of negatives to create a balanced data set. In these experiments a 10-fold-cross validation over the same data shows a very high precision and recall. However in a realistic setting in which *all* the *possible* negatives should be tested, even a 99% accuracy on the negatives introduces a large number of false positives and decreases the precision sharply. Moreover, the error analysis confirms again the importance of the lexical features and also the necessity of finding a solution to this well-known problem in linguistic semantic tasks.

6.4 Related Work

In this chapter we point to related works that apply statistical relational learning for natural language processing and general information extraction tasks. The most relevant relational learning models that are applied in different domains are relational Markov networks (RMN) used for relation extraction from biomedical texts [14], and also for link prediction in web data [149]. Markov logic networks also have become popular models for natural language processing and information extraction [38, 109, 96] and are used for semantic role labeling, information extraction and coreference resolution. All these works are based on learning and inference techniques in probabilistic graphical models. The advantage of using kLog compared to these models is its more flexible relational representation language based on the notions of relational and deductive databases exploiting Prolog for relational feature extraction and its ability of deductive analysis exploiting background knowledge. Moreover, using the powerful graph kernel it can capture the long distance dependencies by means of extensive relational input features instead of explicit modeling of the correlations between output variables as in the above mentioned models. kLog is successfully applied on some linguistic tasks such as hedge cue detection [159] and for identifying evidence based medicine categories [158].

6.5 Conclusion

We presented the SpRL problem using the relational learning framework of kLog motivated by its flexible first order logical representation of the data and knowledge compared to the propositional models. Moreover, kLog's powerful graph kernel can capture the long distance dependencies by using extensive input relational features. The experimental findings in this framework are: a) A binary classification of the spatial relation predicates with three argument performs poorly. This is due to the huge number of false predicates (i.e. triplets) compared to the number of true ones as training examples. In this setting, even the long distance relational features produced by the graph kernel can not help alleviating this problem; b) Using background knowledge in the form of *candidate selection rules* helped to reduce the number of negative triplets. This made the prediction of the spatial relation predicate feasible given the relational features though the performance was lower than some next settings; c) Various pipeline models which do role prediction in a first phase, for decreasing the negative examples, had the typical side effect of error propagation and a low performance; d) Considering the dependencies between the output entities by using SVM-HMM under kLog showed sharp improvements over the SpRL mentioned above models which is comparable to the experiments by using its CRF counterpart.

We find kLog a very flexible and suitable language for performing experiments in such relational domains. kLog easily allows querying from a database for retrieving various relational features and storing the new predictions as new tables in the database and hence facilitates building pipelines or various kinds of combinatory models over basic classification models. It provides the facility of deductive analysis over the declaratively described background knowledge, prior or posterior to the training by using the underlying Prolog. This is a great potential for integration of learning and logical reasoning models for our future goal of spatial understanding.

Given the remaining challenges such as dealing with a large number of negative examples in such relational domains with structured output, and given that the experimental results using SVM-HMM and CRFs confirm the advantage of considering the correlations between the output variables during training, we aim to integrate all these aspects in the last part of this thesis. Our next investigation over structured output prediction models in addition to considering background knowledge during training, applies not only to SpRL but also to the full *ontology population*. Mapping to the full spatial ontology is a richer problem for these kind of investigations and is the subject of Part III of this thesis of Chapters 7 and 8.

Part III

Structured Learning: from Language to Spatial Ontologies

Outline

Given the general targeted task defined in Part I, and the experimental investigation of Part II using graphical models and relational learning for SpRL, in this part we extend and investigate the problem from different angles. We explore the extended ontology population task, structured machine learning in a relational data domain and efficient approximate inference for global learning in the presence of global constraints.

We extend SpRL to a kind of general ontology population task. Ontology population is a relational learning problem in which the concepts and relations in a predefined ontology are assigned to arbitrary segments of the input text and the ontological constraints can be considered during training and prediction.

The main line of research and practice in this part of the thesis contains: a) Providing a unified structured learning framework for ontology population; b) Designing structured learning models that are able to exploit relational features, structural and ontological characteristics of the problem and constraints on the output; c) Designing efficient approximate inference models for global learning in the presence of global constraints; d) Experimental investigation and evaluation of the structured learning models on the spatial ontology population; e) Assessing the influence of the relational features and global constraints in the designed structured learning framework for spatial ontology population.

Our general learning framework for ontology population and the global inference during training and prediction are discussed in Chapter 7. The grounded model for mapping the natural language to our predefined spatial ontology and the experimental results for a variety of local and global models are the subject of Chapter 8. The following documents are related to this chapter.

Related Publication

Kordjamshidi, P., Moens, M. F. (2012). Spatial role labeling using structured support vector machines. *Proceedings of 21st Belgian-Dutch Conference on Machine Learning (BeneLearn): vol. 21.* Ghent, 24-25 May, pp. 71.

Kordjamshidi, P., Moens, M.F. (2013). Structured machine learning for mapping natural language to spatial ontologies. *In Proceedings of the International Workshop on Advances in Regularization, Optimization, Kernel Methods and Support Vector Machines.* (ROKS 2013).

Kordjamshidi, P., Moens, M.F. (2013). Structured Machine Learning for Spatial Ontology Population. *Special Issue of the Elsevier Journal of Web Semantics on Semantic Search* (Submitted).

Chapter 7

Structured Learning for Ontology Population

In this chapter, we introduce a general structured learning framework for mapping natural language to arbitrary ontologies or the so-called ontology population task. We view ontology population as a relational learning problem in which the input natural language sentences can be segmented and represented as a number of entities with their properties and relationships, and the output is an ontology of which the concepts are populated with the relevant input segments. We introduce a learning framework called Link-And-Label based on the notion of templates and a basic component-based loss in the context of constraint optimization for structured learning in relational domains. We propose an efficient inference approach called communicative inference in order to deal with the large number of output variables that should obey certain structural constraints. We discuss relevant decomposed inference approaches that can be applied in our learning framework. Although we do not propose a formal language for relational representation in this piece of work, we discuss the underlying structured learning model which can be connected to any formal relational representation language in the future.

In Section 7.1, we describe the Link-And-Label model. In Section 7.2, we discuss the global and decomposed inference algorithms to be applied during training and prediction. Section 7.3 provides an overview of the related research. In Section 7.4, we conclude.

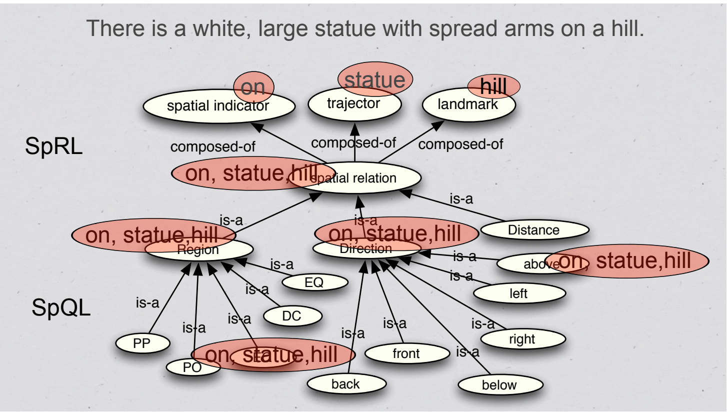


Figure 7.1: The spatial ontology populated by spatial roles and relations in one sentence.

7.1 Link-And-Label Model

Ontology population is the process of inserting concept and relation instances into an existing ontology. In a simplified view, an ontology can be thought of as a set of concepts, relations among the concepts and their instances. A concept instance is a realization of the concept in the domain, e.g. the instantiation of the concept as a phrase in a textual corpus [108]. Figure 7.1, shows an example of spatial ontology population. In this example, the segments of the input sentence, which are words such as *statue* and triplets such as *on,statue,hill*, populate the concepts and the relationships such as *trajector* and *Region* represented in the spatial ontology.

In the ontology population the components in the input space are grouped according to their types. There are relationships between the types such as *is-a* and *composed-of*, etc. The groups of the input components can be associated to various types and hence will have similar relationships according to their types. This means, learning to populate an ontology forms a relational learning problem.

To directly apply the general formulation of structured learning in Chapter 2, the structured inputs and outputs should be turned into a flat vector representation and a *loss function* and a solution to the *inference* problem should be provided to the learning algorithm. Designing these components for real world problems is very challenging. Having an expressive representation of the *learning model* is always useful besides having an expressive representation of the data model [101]. Particularly in relational domains, such a representation eases designing, assessing and improving the learning models.

In this section, we aim to provide a simple and useful abstraction for designing unified[1] structured learning models for ontology population from text. We describe the learning components (i.e. input/output/features/constraints/loss/inference) via a model which we name *Link-And-Label* (LAL). We explain the way we build the objective functions for the inference during training and during prediction according to the relational data and the knowledge for the ontology population task.

The Link-And-Label name, is inspired by the conceptualization process that a human performs in general. We usually group objects which are related to each other by having some commonalities and we label them as a new concept. In our case the various segments of the text are linked to each other and labeled as an instance or an indicator of a specific concept. The labels themselves are the new properties of the higher level concepts; therefore, by linking a number of labels (for example in the case of a composed-of relationship) we build more complex concepts and new labels (such as spatial relation which is built upon the concepts of trajector and landmark).

In general, concepts can have various relationships which are considered in designing ontologies [51]. The relationships between concepts describe the relationships between the instances of them. This feature stimulates treating ontology population as a relational learning problem exploiting the ontology as a first order representation of the output space. From the machine learning point of view, objects are grouped based on their known properties and relationships. Then their unknown properties and relationships are to be predicted. The goal of applying machine leaning models is to categorize the original objects in new groups with new types of relationships in an output space which represents a new semantic layer. According to the machine learning problem setting, we distinguish between the concepts and relationships in the *input* space and in the *output* space of the domain data. To show the Link-And-Label abstraction layer, first we describe the concepts and the terminology that we use based on the *input* and *output* distinction. Then we describe the form of the objective function of the training and the prediction for ontology population in this framework.

7.1.1 Input Space

We represent each input x by a set of components $\{x_1 \dots x_T\}$. The components can be of different *type*s. Each $x_k \in x$ is described by a vector of the feature

[1] By unified we mean a model that deals with relational input and outputs, considering the correlations between the outputs, e.g. ontological relationships and constraints.

values relevant for its type. The feature vector is denoted by ϕ_p where p is an arbitrary index that refers to a specific type. For instance, in semantic labeling of text an input type could be a word or a prepositional phrase composed of two noun phrases connected by a preposition, and each type is described by its own typical features (e.g., a single word by its part-of-speech, the prepositional phrase by the distance in words of its head nouns). We refer to each component in the input by an identifier. A component can be composed of a number of other input components in which case it is called a composed component and is identified by the identifiers of its parts. We refer to atomic components as single components. The features that describe a property of a single component are called local and the ones that describe the relation among more than one single component are called relational features (See Chapter 4).

7.1.2 Output Space

The output space y is represented by a set of *labels* $\mathbf{l} = \{l_1, \ldots, l_P\}$. Each label $l_p \in \mathbf{l}$ refers to specific semantics in the output. The labels are defined based on the nodes in the ontology and have ontological relationships to each other. To be able to represent complex output concepts in general for any arbitrary task, we distinguish between two types of labels, the *single labels* and *linked labels* that refer to an independent concept and to a configuration of a number of related single labels respectively. Linked labels can represent different types of semantic relationships among single labels. They can express composed-of, is-a and other semantics given in the ontology.
To show which labels are connected by a *linked label*, we represent the *linked labels* by a *label string*.

Definition A *label string* is the concatenation of a number of labels. In fact a label string shows the parts of the output that are linked to each other and construct a bigger semantic part of the whole structure.

For example as you see in figure 7.2 a *spatial relation* can be denoted by *sp.tr.lm* meaning that it is *composed of* these three single labels. Label strings can also imply is-a or other semantic relationships between concatenated labels.

Definition A label l is a sub-label of a label l' when all single labels that occur in the label string of l also occur in the label string of l', denoted by $l \prec l'$. In this case we call l' a super-label of l.

For example, in figure 7.2 *sp* is a sub-label of *sp.tr.lm*.

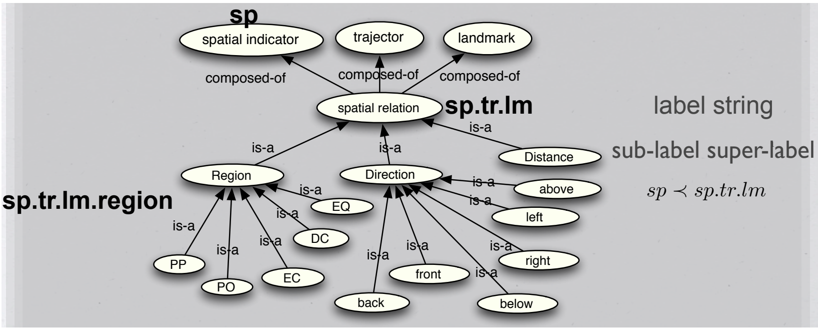


Figure 7.2: Syntactic relationships between labels represented via the label strings.

7.1.3 Connecting Input and Output Spaces

The objective function of a structured learning model, that is $g(x, y; W)$, described in Chapter 2, is defined over the combined feature representation of the inputs and outputs denoted by $f(x, y)$ (See Formula 2.19). In the LAL model, we treat labels as binary indicator functions that receive an input component and indicate whether the component has a certain label. In fact, the binary indicator function for each linked label is defined according to its semantics. For example a spatial relation label should be defined in a way to convey the *composed-of* semantics based on the labels of its components. We highlight two properties of our setting that deserve attention. Firstly, because we allow input components of different types, usually an output label only applies to a certain type of input component (e.g. a label can only be assigned to a word, another label only to a pair of syntactically connected words). Secondly, given a particular type, more than one input component of that type can be associated to a label. In fact, each label can refer to a set of components. We use both notations of $l_p(x_k)$ or l_{pk} to indicate the membership of the component x_k in the set of components with the label l_p.

To formally specify the connections between input components and output labels we use the notion of *template*. This notion has been used mostly in relational graphical models [26, 14] (see Chapter 2). The learning model is specified with a set of templates $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_P\}$, where P is the number of templates defined for the application. Each template $\mathcal{C}_p \in \mathcal{C}$ is specified by three main characteristics,

- *A subset of joint features* (i.e. a local joint feature function or a sub-mapping). Each local joint feature function is defined over a number of input type(s) and output label(s) which are associated to that template.

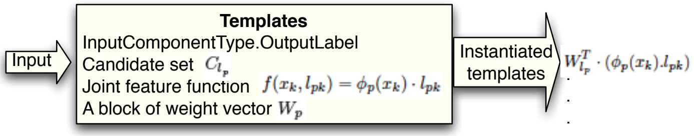

Input
Templates
InputComponentType.OutputLabel
Candidate set C_{l_p}
Joint feature function $f(x_k, l_{pk}) = \phi_p(x_k) \cdot l_{pk}$
A block of weight vector W_p
Instantiated templates
$W_{l_p}^T \cdot (\phi_p(x_k).l_{pk})$

Figure 7.3: The input is given to a set of predefined templates and the local joint feature functions are produced for the relevant components of the input.

The local joint feature function of a template $\mathcal{C}_p$ is denoted by $f_p(x_k, l_p)$, where x_k indicates a single or composed component of the input and l_p is a single or linked label that is entailed from the ontology labels $\mathbf{l}$.

- *Candidate generator.* It generates all candidate components upon which the specified subset of joint features is applicable, the set of candidates for each template is denoted as C_{l_p}.
- *A block of weights W_p.* This is a block of the weights in the main weight vector W of the model which is associated to that template and its sub-mapping.

The design of the learning model is represented by the specification of its templates which are designed based on the labels in the ontology and the links between them.

LAL objective function. In the main objective *discriminant function* $g = \langle W, f(x, y) \rangle$, we explicitly represent the weight vector W with its blocks of weights as $W = [W_1, W_2, \ldots, W_P]$, where each block W_p is associated with a template. Hence the objective can be written in terms of the instantiations of the templates and their related block of weights, W_p. In other words, g is a linear function in terms of the combined feature representation associated to each candidate input component and an output label according to the template specifications (see Figure 7.3),

$$g(x, y; W) = \sum_{l_p \in \mathbf{l}} \sum_{x_k \in C_{l_p}} \langle W_p, f_p(x_k, l_p) \rangle = \sum_{l_p \in \mathbf{l}} \sum_{x_k \in C_{l_p}} \langle W_p, \phi_p(x_k) \rangle l_{pk} =$$

$$\sum_{l_p \in \mathbf{l}} \langle W_p, \sum_{x_k \in C_{l_p}} (\phi_p(x_k) l_{pk}) \rangle, \tag{7.1}$$

where $f_p(x_k, l_p)$ is a joint feature vector which is an instantiation of the template $\mathcal{C}_{l_p}$ for its candidate x_k. This feature vector is computed by scalar multiplication of the input feature vector of x_k (i.e. $\phi_p(x_k)$), and the output label l_{pk}. This output label is the indicator function of label l_p for component x_k. C_{l_p} denotes the set of candidates for template label l_p. Each indicator function of a template linked label is applied on the relevant input component and its value is one when the intended semantics behind it holds for that component. For example if a template is an *and template*, it means the indicator function of the linked label is one if all included single label indicators are one when applied on the input parts. In this case we can represent the linked labels with the scalar product of the indicators of the sub-labels when forming the objective g. In this way, g is written in terms of the output labels. Hence, we can view the inference task as a *combinatorial constrained optimization* given the polynomial g which is represented in terms of labels, subject to the constraints that describe the relationships between the labels. For example, the *composed-of* relation between a linked label l denoted by the *label string* $l = l_i \dots l_j$, and its single sub-labels can be represented by the following constraint,

$$(l(x_c) = 1) \Rightarrow (l_i(x_1) = 1) \wedge \cdots \wedge (l_j(x_n) = 1),$$

where each label $l_{\_}$ applies only on a relevant type of component x_k and, $x_k \subseteq x_c$, $\forall k = 1, \dots, n$; and the *is-a* relationships can be defined as the following constraint,

$$(l(x_c) = 1) \Rightarrow (l'(x_c) = 1),$$

where l and l' are two distinct labels that are applicable on the type of component x_c. These are two commonly used ontological relationships also in our spatial ontology, but many other ontological relationships can be represented and directly exploited in a learning model.

In designing templates, the highly correlated labels should be linked to each other and be considered in one template (See specific examples in Chapter 8). However, there can be global correlations that are not considered inside templates due to complexity issues (e.g. a large number of candidates). The correlations between the variables of different templates can be modeled via a number of (hard) constraints. The constraints can hold between the instantiations of one template which implies the relations between the components of one type also referred to as *autocorrelations* according to the terminology used in *relational dependency networks* in [101]. These (hard) constraints are exploited during training in the loss-augmented inference and are imposed on the output structure during the prediction. Particularly, we treat the objective as a linear function in which the association between linked labels and single labels in addition to their global relationships are modeled via linear constraints. In other words, each linked label is considered as one new variable and its connection to other labels is reflected in the formulated constraints.

Each time, we need to do inference over an input example, we build a new instance of the main objective function and the related constraints. This process, at the conceptual level, is similar to knowledge-based model construction [164]. In contrast to KBMC models, we construct a non-probabilistic model which is a multinomial function with a number of first order[2] constraints that are also instantiated. The inference is performed by combinatorial optimization instead of probabilistic inference. Our LAL model is well-posed to be represented using a first order language. To shortly clarify this we note that the input components are typed so the types and the attributes of each type are easier represented using a first order language; the output labels can be seen as binary predicates that function on the arguments derived from the input space. The templates are defined based on the types of input and output, thus are a first order abstraction of the learning model structure. The constraints, as we showed in the two above examples, are inherently first order though in practice are compiled into a linear form.

We present how we apply this approach in Chapter 8, when modeling the relationships in our spatial ontology.

7.1.4 Component Based Loss

In the formulation of the structured learning in both structured SVM and structured perceptron frameworks any arbitrary loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ can be considered. However, we assume the loss function is decomposed in a similar way as the joint feature function. In this way we avoid increasing the complexity of the *loss-augmented* inference during training compared to the prediction time inference (see Foundations chapter, Section 2.1.3, Formula 2.22). Hence, we define a component based Hamming loss for various output components. Hamming loss counts the number of disagreements for two binary vectors λ and λ' with length n,

$$\Delta_H(\lambda, \lambda') = \mathbf{1} - (|\lambda \odot \lambda'| + |(\mathbf{1} - \lambda) \odot (\mathbf{1} - \lambda')|) = |\lambda| + |\lambda'| - 2|\lambda \odot \lambda'|, \tag{7.2}$$

where $\odot$ denotes the element-wise product and $|.|$ denotes the 1-norm of a vector which in the case for binary vectors is the number of ones. The Hamming loss is divided by the length of the vectors to get a normalized value between 0 and 1,

$$\Delta_{\tilde{H}}(\lambda, \lambda') = \frac{1}{n}(|\lambda| + |\lambda'| - 2|\lambda \odot \lambda'|). \tag{7.3}$$

[2] Because constraints can be over types of variables.

We define a component-based loss for each template label l_p by measuring the Hamming loss between the vector of predicted labels for all candidates (denoted by Λ_{l_p}) compared to the ground truth assignments (denoted by Λ'_{l_p}) and normalize by the length of the vectors (=number of candidates),

$$\Delta_{l_p}(\Lambda_{lp}, \Lambda'_{lp}) = \frac{1}{|C_{l_p}|} \sum_{k=1}^{|C_{l_p}|} \Delta_H(l_{pk}, l'_{pk}) \text{ where } \Delta_H(l_{pk}, l'_{pk}) = l_{pk}+l'_{pk}-2l_{pk}l'_{pk}.\text{[3]} \tag{7.4}$$

In addition to the above mentioned motivation and the complexity issues, using this type of loss function is very natural for ontology population because we basically perform a kind of collective classification over all input components with respect to the nodes in the ontology and a simple 0/1 loss over all (link) label assignments is jointly minimized at the end.

When some labels have priority for the application at hand, the Hamming loss between the labels can be simply weighted based on their priority and aggregated such as follows,

$$\Delta(Y, Y') = \sum_{p=1}^{P} \omega_{l_p} \Delta_{l_p}(\Lambda_{l_p}, \Lambda'_{l_p}), \tag{7.5}$$

where ω_{l_p} is the weight of each template label l_p and P is the number of all templates used in the modeling. Hereafter we refer to ω_{l_p}s as *preference*s which obtain their values from an expert. In this way, the loss also is yet expressed in terms of the labels, and the two inference problems for training in Formula 2.22 and prediction in Formula 2.18 are similar (in terms of variables and constraints) and finding a solution to one applies to the other. Therefore to represent and describe the inner inference at training time we only discuss the same objective g for the sake of brevity in the representation. Now that we formulated the main objectives, providing efficient solutions for inference over these objectives is the subject of the next section.

7.2 Global Training and Prediction

Punyakanok et al. [111] describe three fundamentally different solutions to learning structured output prediction at a high level,

- **Learning only (LO):** Local classifiers are trained and used to predict each output component separately.

[3] As if these binary labels are vectors with length one.

- **Learning plus inference (L+I):** Training is performed locally as in the LO models, but the global constraints/correlations among components are imposed during prediction. These models are formulated in a general framework referred to as constrained conditional models.
- **Inference based training (IBT):** Inference is used during training so that the constraints and dependencies among the variables are incorporated into the training process. In fact, these models are referred to as structured output learning models.

However for training, there is a spectrum of various model compositions between two extreme sides of strict local training as in LO and L+I schemes versus a strict global training in the IBT scheme. At the extreme side of strict local training, an independent binary classifier is trained for each output variable using local examples relevant for each part of the output. On the other extreme side of IBT training which we refer to as *global learning* [130], the inference during training is supposed to be solved over the entire output space for each training example. Global learning in its strict meaning implies performing exact inference considering all correlations and constraints among output variables which is not alway feasible and also not always required for the learning problems.

In an ideal IBT setting for the LAL model, a global combinatorial constraint optimization should consider all correlations and ontological relationships between the output labels in both training time and prediction time inferences. As described in Chapter 2, LP-relaxation is an efficient way to approximate a MAP solution for such complex problems. Though there are many off-the-shelf solvers that can provide such solutions, still this combinatorial optimization can become highly inefficient given our relational data domain. Because the objective function which is represented in terms of templates along with the first order constraints in practice can produce a large number of output labels and linear constraints, when instantiated for each example (depending on the overall candidate components of each example).

To solve this problem we make an additional layer of decomposition as a meta frame for applying off-the-shelf LP-solvers. *Decomposition* is the general approach to solve a problem by breaking it into smaller ones and solving each of the smaller ones independently [13]. This is an old approach which has been studied extensively in the optimization literature [11]. There are a number of general ideas and various techniques for decomposition through which one can approach solving large problems that are not solvable with one standard technique. Hereafter we use the term decomposition according to its technical definition rather than conceptual, provided by Samdani et al. [130] in the learning context as:

Definition Given a set of L binary output variables[4] indexed by $\{1, \dots, L\}$, a decomposition S is a set containing distinct and non-inclusive (possibly overlapping) index sets which are subsets of $\{1, \dots, L\}$:

$$\mathsf{S} = \{s_1, \dots, s_q | \forall i, s_i \subset \{1, \dots, L\}; \forall i, j, s_i \not\subseteq s_j\},$$

where q, the decomposition size, indicates the number of subspaces considered in the output space. Also given a set $s \subseteq \{1, \dots, L\}$, let $-s = \{1, \dots, L\} \backslash s$. $y_s \in \{0, 1\}^{|s|}$ denotes an assignment to the variables indexed by set s.

In this work by generalizing a formulation made in [130], we assume an arbitrary oracle can provide an assignment to y and we refer to it as y^{oracle}. Accordingly, we define (y_s, y_{-s}^{oracle}) as the output formed by replacing the labels in y^{oracle} indexed by s by the corresponding labels in y_s. For each training/test example a decomposition S^i is associated and an inference subproblem is defined as follows: Given an oracle assignment y^{oracle}, pick a set $s \in \mathsf{S}^i$ fix the labels in y_{-s}^{oracle} and find the best assignments to y_s, over all feasible selections of y and over all $s \in \mathsf{S}^i$ and return the labels. Before providing the details about the considered decomposition approaches based on this formulation, we discuss the loss-augmented inference and the role of violating examples in global learning both formally and intuitively.

7.2.1 Globality and Violating Examples

Solving the global loss-augmented inference in IBT models for the LAL's objective function implies selecting a globally violating output that updates all blocks of the target weight vector W jointly. In this way, the structured loss and the constraints over the output variables are exploited to achieve the optimum weight vector W^*. To make the role of the violating outputs more clear we first provide a relevant definition given in [59].

Definition The *standard confusion set* $CF_s(E)$ for training data E is the set of triplets (x, y, z) where z is a wrong output for input x. A triplet $TS = \langle E, f, CF \rangle$ is called a training scenario, where $CF = CF_s(E)$.

Definition A triplet (x, y, z) is said a violation in the training scenario given a weight vector W, if $(x, y, z) \in CF$ and the *learning constraint* 2.20 or 2.21 does not hold.

In global learning for a target model the most violated triplet denoted by $(x^i, y^i, \hat{y})$, is selected per training example (x^i, y^i) from the confusion set.

[4] Here we use capital L rather than l to refer to all output variable instances of an example rather than the type of the labels in the output space.

Thus the whole model weight W is updated jointly on the basis of one globally violating output per example. By contrast, when training local models in LO or L+I models, the blocks of the target weight vector are updated independently by considering the violating outputs related to each label which can contain all negative candidates for that label.

Approximating the global inference can result in finding less global violations and hence less ideal updates of the weight vector [59]. For structured SVMs the theoretical guarantees for the convergence of learning in the case of approximate inference have been investigated in [40]. The guarantees in case of structured perceptrons have been studied in [59]. The authors explain why in certain sub-optimal solutions to the loss-augmented inference, the convergence still is guaranteed.

If we approximate the global inference by decomposing the objective function, for instance in terms of individual labels, in its extreme case without considering any constraints, inference will be trivial. In this case, the most violated output with respect to each label will be the relevant input component with the opposite label. Hence, the training scenario contains all input components labeled with opposite output labels compared to the ground truth. Thus the training scenario becomes exactly similar to training local binary classifiers because the training scenario for a binary classifier for a particular label also uses a confusion set that contains all negative candidates of that label and all of them are potentially used for training. In other words, all candidates act as independent and identically distributed (i.i.d) examples for learning the weights related to each label as in local training of binary classifiers.

When decomposing a complex inference task into smaller ones, the number of violations that we then call local violations can be many more than the number of global violations and therefore they can redirect the training from learning the optimal W. If we use the decomposition in its conceptual sense rather than technical, the *learning-only* (LO) models are, in fact, a form of decomposed learning in which the correlations between the subproblems are totally ignored. Decomposing the inference in a way that each subproblem considers a number of labels together and thus selects more globally violating examples will help updating the model weights jointly. This will yield a solution that is closer to learning the optimal model. In the next section we discuss our proposed approach based on decomposing the output space for efficient global inference as well as the most recent relevant approaches that consider splitting the global inference and still exploit the global structure of the output.

Algorithm 4 Communicative inference

```
Given an example $(x, y)$ and the objective function $H(y)$
Given a decomposition over $y$ as $S^x = \{y_1, y_2\}$, $\{\mathrm{H(y)} \triangleq H(y_1, y_2)\}$
Given two disjoint subsets of constraints $c_1$ and $c_2$ over $y_1$ and $y_2$
$t \leftarrow 0$
Initialize $y_1^t, y_2^t$
repeat
    $t \leftarrow t + 1$
    $y_2^t \leftarrow \arg\max_{y_2} H(y_1^{t-1}, y_2)$
    $\{$A LP-relaxation subproblem subject to $c_2\}$
    $y_1^t \leftarrow \arg\max_{y_1} H(y_1, y_2^t)$
    $\{$A LP-relaxation subproblem subject to $c_1\}$
until $(y_1^{t-1} = y_1^t \wedge y_2^t = y_2^{t-1}) \vee\ t > Tmax$
$\hat{y} \leftarrow [y_1^t, y_2^t]$ $\{$ $\hat{y}$ is the MAP of $H$ over $y\}$
```

7.2.2 Communicative Inference

We propose an approach for decomposing the inference that can be applied at both training and prediction time and we call it *communicative inference.* The basic idea behind this approach is that given a *decomposition* by an expert, the inference *subproblems* are solved independently but communicate with each other by *passing messages*, that is, passing solutions. To implement this idea we use an approach which has a similar intuition as block coordinate descent (BCD) [154] methods also referred to as alternating optimization (AO) [12]. In these methods, given a general objective function H of multivariate y, to find the MAP of H we can divide the variables into a number of blocks assuming that each block has a local maximizer. Here, $H(y)$ is the objective of the loss-augmented inference problem at the training time and is equal to $g(x, y; W)$, in Formula 2.19, that is the trained objective function at the prediction time. Our suggested communicative inference is presented in algorithm 4 for a decomposition $\mathcal{S} = \{y1, y2\}$ containing two blocks of variables y_1 and y_2. At the starting point the oracle is a random initializer, assigning random values to the output variables in the decomposition set. Then at each step we optimize over one block of the target variables (i.e. one member of the decomposition set) while the other block is set with the last partial MAP assignment from the previous step, see figure 7.4. In fact, to solve a subproblem the oracle assignment is the solution from the previous iteration. In contrast to the standard setting of AO, in our setting the variables are discrete and each subproblem is solved approximately by a LP-relaxation technique with a relevant subset of constraints activated. As pointed in Section 2.1.7, the solution to LP-relaxation subproblems are optimal and the convergence is

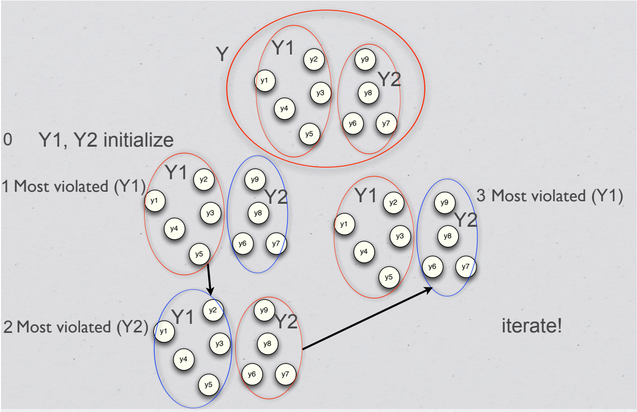


Figure 7.4: Finding the most violated y for a given input x using communicative inference (blue cycles show the fixed variables at each step).

guaranteed if the constraint matrix has the *unimodularity* property. It is easily proved that in this case the convergence of the communicative inference is also guaranteed. To shortly clarify, if we assume a maximization setting, then the value of the objective function at step i can not be less than its value at step $i-1$, otherwise the solution of step $i-1$ is selected as the optimal solution (because we know each subproblem is solved exactly). Given this monotonically increasing sequence of objective function values during the optimization iterations and given that we have a finite space (due to the given discrete space problem) the algorithm is alway convergent. However, in the worst case an exhaustive search can be performed.

At prediction time, we establish communicative inference between two arbitrary models, each of which can be trained jointly or independently. At training time, it allows a joint update over all blocks in the weight vector. Hence it can provide more globally violating examples. The main advantage of this approach is that the models can follow their own MAP methodology, based on any approximate or exact inference technique.

7.2.3 Decomposed Training (DecL)

In decomposing complex inferences during training, one can exploit some properties that are not available for simplifying inference during prediction. For example, the ground truth labels are available during training and the partial MAP solutions can be used at each iteration. In addition at training time the final goal of inference is to optimize W, therefore being globally optimal in y is marginal and not necessarily needed. These differences call for particular considerations for training time inference which have been taken into account in the DecL learning model in [130]. We look at DecL as an alternative approach to simplify the complex inference problem we need to deal with for the ontology population.

The general DecL algorithm is shown in Alg 5. This algorithm, given a decomposition, each time performs efficient learning by restricting the inference step to a limited part of the structured output spaces.

Formally, for a training instance $(x^i, y^i) \in E$, let $nbr(y^i) \subseteq \mathcal{Y}$ be a subset of the output space defining a neighborhood around y^i. The key idea behind decomposed learning (DecL) is to learn W by discriminating the supervised label y^i from only all $\tilde{y} \in nbr(y^i)$ instead of all $y \in \mathcal{Y}$. The $nbr(y^i)$ is generated by fixing a subset of the output labels to their ground truth in y^i, while allowing the rest of them to vary. This means, according to the general formulation of Section 7.2, the oracle here is the ground truth data. The confusion set each time contains only the wrong ys in the neighborhood of y^i. In the example shown in figure 7.5, the decomposition contains four subsets, each time three subsets are fixed with the values obtained from the ground-truth oracle (G-th) and one subset of variables can vary. The most violated example is found locally for the varying subset. The globally most violated output for each training example is chosen based on the MAP of all the solved subproblems.

Using the ground-truth y^i of each training example as the oracle (see above Section 7.2) leads to minimizing the following convex function over N training examples (cf. formula 2.23).

$$\mathrm{DecL}(W; E) = \sum_{i=1}^{N} {}_{s \in \mathsf{S}^i} \max_{y_s \in \{0,1\}^{|s|} : (y_s, y^i_{-s}) \in \mathcal{Y}} \big[g(x^i, (y_s, y^i_{-s}); W) - g(x^i, y^i; W)$$

$$+\Delta(y_s, y^i_s) \big] \,, \tag{7.6}$$

And accordingly the most violated constraint for each example is computed as it is shown in line 5 of Algorithm 5.

One label decomposition. By considering a decomposition S that contains s_i subsets with only one label, we will have an approach proposed in [138] called *Pseudo-Max* which allows one label to vary and sets the rest of them as ground truth. In other words $y_s \in \{0, 1\}$ in formula 7.6. In Pseudo-Max all blocks of the weight vector related to a single label are updated jointly. The violating examples are picked from the confusion set that vary only in one label compared to the ground truth. Hence no inference is needed and

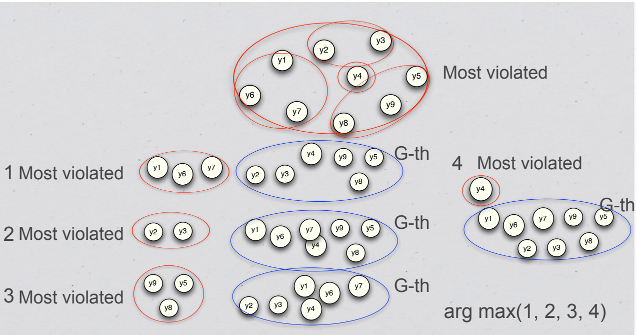


Figure 7.5: Finding the most violated y for a given input x given a decomposition S in DecL approach, Samdani et al. [130].

Algorithm 5 Sub-gradient-descent Alg. for DecL [130]

```
Given: training data: $E = (x^i, y^i)_{i=1}^N$; step sizes $\eta_t$; decompositions: $\mathsf{S} = (\mathsf{S}^1, \ldots, \mathsf{S}^N)$
$W \leftarrow \mathbf{0}$
for $t = 0$ to $\mathcal{T}$ do
    for $i = 1$ to $N$ do
        $\hat{y} \leftarrow \arg\max_{s \in \mathsf{S}^i \; y_s \in \{0,1\}^{|s|}:(y_s, y^i_{-s}) \in \mathcal{Y}} [g(x^i, (y_s, y_{-s}); W) + \Delta(y_s, y^i_s)]$
        $W \leftarrow W + \eta_t (f(x^i, y^i) - f(x^i, \hat{y}))$
```

the maximum number of violating examples will be as many as the number of output labels. Violating examples update the blocks of the weight vector related to a label each time. In the presence of large data with examples drawn from the right distribution this model can cover learning over all regions in the output space [138]. However, the theoretical proof about the Pseudo-Max does not consider the presence of constraints and only covers the case in which the correlations are modeled in the feature function as a pairwise Markov model, therefore the DecL extends the Pseudo-Max idea in various dimensions. This model is referred to as *DecL-1* in the experiments described in Section 8.3.

Pairwise decomposition. Each decomposition S contains all subsets of size 2 from the output space, so $y_s \in \{0,1\}^2$ in formula 7.6. The model allows two labels to vary while all other labels are fixed by the ground truth assignments. The violating examples are chosen by doing loss augmented inference in the space of only two variables each time, hence the pairwise correlations are directly considered. This model is referred to as *DecL-2* in the experiments described in Section 8.3.

7.2.4 Decomposition in Relational Domains

To fulfill the properties that theoretically guarantee the globality of the DecL, an exhaustive number of decompositions are needed. Though solving the inference in each subset of variables can be exact and efficient, producing all subproblems and adapting the constraints for each subset of variables is an extra overhead. However, in the relational domains where the variables can be grouped based on their type, the type of input components can guide the way we decompose the output space. In these problems considering the relational structure of the variables helps to find the decompositions that semantically are sensible. For example, in the Dec-2 setting, distinguishing a variable of type trajector against another variable with the same type is not sensible, but we need to be able to distinguish between the instantiations of trajectors and landmarks. Hence, we may decompose the output based on the type of the candidate components.

In the relational case, local learning (see Section 7.2.1) can be described as a decomposition in which the labels related to only one template can vary while the rest are fixed to the ground truth. Moreover, the fixed variables do not propagate any constraints to the free variables, and the candidate components are assumed to be independent by acting as i.i.d examples.

7.2.5 Decomposition in Pipeline Models

Traditionally, pipelines are defined as a cascade of models. At each stage one model has access to the initial input in addition to the predictions from models used in the previous stages [123]. Many natural language processing models include such pipelines of various subtasks of which performing them jointly can be beneficial. We can view the pipeline models as a kind of decomposition where in the decompositions of S each set s contains the output variables of one stage of the pipeline. In the first training iterations the output of the first stage of the pipeline is variable and the outputs of the rest are fixed. Afterwards the first stage variables obtain values from the ground truth and the second stage is learnt in the second set of iterations and so on. In this style the decomposition parts are ordered and training is done for each subset separately, in a sequential setting. Obviously, this type of decomposition does not have the properties that lead to the globally optimal solution for the learning. The main drawback of this type of decomposition is that the variables in different subsets s^i are highly correlated. There could be soft relationships between these variables or even one subset can impose hard constraints on the others. For example, in the two layered spatial semantic model, having a semantic label is meaningful for a triplet only if that triplet is a spatial one. In other words there is an *is-a* relationship between the variables of the two layers. Therefore, setting a variable in the second layer imposes hard constraints on the first layer variables. Therefore, one cascade direction of training is very natural in these situations. However, the order of the stages can be ignored by training in any arbitrary direction in the pipelines. The problem with this latter approach is that the constraints often can not be propagated when setting a succeeding stage as ground truth as in DecL because propagating the constraints in the succeeding stages that depend on the variable of the previous stages often does not give the freedom to do inference by restricting the feasible space. Moreover in the pipeline setting all violating examples from the subspaces are used for training (instead of the most violating among them). This latter property can be used in the DecL standard setting also to exploit directly from the partial inferences in the small subspaces of the output. Obviously, pipelines take less-global violating outputs compared to an ideal global model.

After all, to obtain an effective learning decomposition, sophisticated problem specific decompositions according to expert knowledge are needed and there are difficult trade-offs in selecting the decompositions as well. Having all correlated variables in one block often results in an inference task as difficult as the original one, while decomposing the relationships between the variables and ignoring correlations trades optimality.

7.3 Related Work

The related previous works on ontology population [127] mostly consider: a) An extensive preprocessing step applying NLP tools; b) External linguistic, web or relevant database resources; c) Learning in pipeline models for extraction of the terms, concepts and the relationships for classification or clustering over the extracted material; and d) Post processing for resolving the inconsistencies in the predictions. According to a comprehensive study in [166] the related works are at the level of term, concept and relation extraction, and a few works exist that implement more logic-based approaches for axiom extraction. To our knowledge there is no unified model proposed to extract these elements collectively in one unified framework of structured machine learning by considering the ontological relationships and background knowledge as global constraints, while we do in this work.

Our learning model is appropriate for structured learning in relational domains given that we deal with different types of components in the input and output. Our approach for producing the objective functions for inference during training and during prediction, is similar to knowledge-based model construction. To specify our model we use the notion of templates as in relational graphical models to produce the objective function for each example. In relational graphical models, clique templates [14] are grounded to produce the structure of the probabilistic inference over each relational input. However, by unrolling the templates, we construct a non-probabilistic model in the form of a multinomial function, instantiate a number of first order constraints and solve the inference with combinatorial approaches. To our knowledge, this connection has not been formalized before. Compared to relational learning frameworks discussed in Section 6.4, we only approach the relational structure at the conceptual level and no explicit relational language for relational representation is used. But the advantage of our model is the use of more efficient optimization techniques and LP-relaxation for inference-based-training, exploiting global constraints. To capture the global correlations in the probabilistic models, enough evidence in the data is required which is difficult to obtain while using a small training data set as it is the case for us. Compared to using kLog in chapter 6, the LAL model has the possibility of inference-based-training and exploiting global constraints during training and prediction. Exploiting global constraints in learning models but only during prediction is formally introduced in constrained conditional models [20]. Moreover compiling the propositional logical constraints for integer linear programming models is automated in a modeling language named learning based Java (LBJ) [120].

Another dimension of our investigation regards efficient inference techniques for global training and prediction. Recently in the field of natural language processing there is a tendency to combine structured models for various tasks that can promote each other. Different models are designed for inference in such joint settings [140, 129, 52]. Dual decomposition is a class of solutions that can theoretically cover a large variation of this type of solutions and it is investigated for natural language processing tasks in [128] and also for solving MAP problems in Markov random fields in [136]. This type of models have gained a high popularity recently because they are easy and efficient and can be solved by LP-relaxation and a subgradient algorithm. These techniques involve two steps of solving the separate local inferences and simple additive updates of Lagrange multipliers. Dual decomposition and LP-relaxation is used in [129] for the joint inference at prediction time only, to perform jointly lexicalized parsing and part of speech tagging. Each inference sub-problem is solved using dynamic programming. In fact the two local solutions are forced to agree by solving a linear programing relaxation. LP-relaxation is used also in [140] for joint learning and prediction for semantic role labeling and preposition disambiguation by imposing linear hard constraints to express

the correlations between the outputs of the two tasks. A totally different probabilistic solution for this type of problems is suggested in [52]. This work uses an EM-like approach in a probabilistic setting called expectation propagation for communication during training between three different models that target three types of annotations in parsing problems.
In contrast to the above mentioned joint settings for naturally separated problems, a body of research is about decomposing complex tasks to smaller subproblems. In very recent work Samdani and Roth [130] proposed the decomposed learning model particular for training which we also have applied in this work (see Section 7.2.3). Another relevant approach in [138] is the Pseudo-Max described in Section 7.2.3. Pseudo-Max alleviates the necessity of inference during training in the presence of large training data. In this direction when we decompose a complex task (compared to building joint settings), finding an appropriate decomposition is a problem by itself that often requires expert knowledge about the problem. Sontag et al. [137] choose clusters of variables based on a cluster effectiveness measure, in order to decompose the objective function. The MAP problem is solved for clusters of variables with integral constraints using LP-relaxation and then message passing is performed between the clusters. In this way tighter LP-relaxation upper bounds are obtained. This work is relevant to our approach though, the main goal of our communicative inference approach is scalability not optimality, nor the tightening of the LP-relaxation. Once more using LP-relaxation, in a different track for inference-based-training, Meshi et al. in [95], avoid explicit solving the loss-augmented inference in an independent step of LP-relaxation for each data point; instead they replace the LP with its dual and solve the dual of the structured prediction loss. The proposed communicative inference approach in this chapter can be considered as an approximation by message passing between two subproblems that are solved by LP-relaxation. In our problem a decomposition according to the semantic layers seems natural, therefore we do not investigate an automatic decomposition in contrast to the work in [137]. However we formally view the communicative inference as a kind of block coordinate descent approach [154, 48] or alternating optimization [12]. Alternating optimization provides a general framework for the EM [12] and similar approaches.

7.4 Conclusion

In this chapter, a novel unified framework of structured learning for mapping natural language to spatial ontologies is proposed. We provide a framework called the Link-And-Label model that is able to deal with relational data both in the input and in the output and is able to consider ontological relationships and background knowledge during training and prediction.

Using the notion of templates, as in relational graphical models, we formalize the relational structure of learning in the inference-based training models. Here, the objective function is produced by unrolling/grounding the templates and producing a multinomial function to be optimized, subject to the linearly grounded first order constraints.

This formalization provides a more clear illustration of the structure of the learning models and the tied parameters (via the blocks of weights). It also helps in designing decomposed inference algorithms exploiting the relational structure of the data by considering the types of the output variables when using constraint optimization techniques. We propose a novel and efficient decomposed inference approach for solving the inference for structured training and prediction in a global learning framework. The ontological relationships and background knowledge can be modeled in the form of linear constraints, and LP-relaxation techniques can be used to solve each subproblem in the decomposed space of output variables. The proposed framework will be used in Chapter 8 to design a learning model for mapping natural language to the predefined spatial ontology described in Chapter 4.

Chapter 8

Mapping Natural Language to Spatial Ontologies

In this chapter, we extend the task of spatial role labeling discussed in Part I to the task of full spatial ontology population described in Chapter 4. We refine this task using the described features and the structural characteristics of the problem in the unified structured learning framework described in chapter 7. In this framework, spatial roles and the composed spatial relations in the spatial role labeling (SpRL) layer and, the semantics of the relations in the spatial qualitative labeling (SpQL) layer are extracted in a global learning model. However given the large possible output spaces according to the spatial ontology, the global inference-based structured learning becomes intractable. To address this, we analyze various model compositions and decompositions in the framework of structured learning. To achieve a tractable model, we use the decomposed inference model during training and prediction. Using our proposed *communicative inference* approach, our global learning model outperforms the pipelining of the two layers as well as other state-of-the-art decomposed learning models when evaluated on the SemEval-1 version of our benchmark. The presented work in this chapter is a new step towards automatically describing text with semantic labels that form a structured ontological representation of its content.

This chapter is organized as follows. In Section 1, we specify the model formally in terms of its input and output, and the designed templates. In Section 2, we describe the objective, the loss function and the applied decompositions. Section 3 contains a detailed account of the experimental setup, the results of the SpRL and SpQL layers, their connection and discussions. Section 4, concludes with the most important findings.

8.1 Model Specification

In this section, we formulate the problem of mapping natural language to spatial ontologies. We represent the supervised structured learning model designed for solving this problem using the Link-And-Label model of Chapter 7 and specify: a) The input *components* and *types*; b) The output *single labels*, *linked labels* and *global constraints* over the output structure; c) The *joint feature templates*, *candidate generation* for the templates and the main *objective function.*

8.1.1 Input Space

The input part of each example, x, is originally a natural language sentence such as

"There is a white large statue with spread arms on a hill.",

and each sentence has a number of single components that are its contained *words*. The single components of x in the above example are identified as $x = \{x_1, \ldots, x_{14}\}$, where x_i is the identifier of the ith word in the sentence. Each word in the sentence is described by a vector of the local features denoted by $\phi_{word}(x_i)$, e.g., `(There,EX,SBJ,. . .)` describing the word form, the part of speech, etc. There are also components composed of *pair*s and *triplet*s of words and their descriptive vectors are referred to as $\phi_{pair}(x_i, x_j)$ and $\phi_{triplet}(x_i, x_j, x_k)$. We define a number of relational features describing the relationships between words (e.g., distance). A feature vector of a composed component such as a pair, $\phi_{pair}(x_1, x_2)$ is described by the local features of x_1, x_2 and the relational features between them, `(before, 1)`. The details of the linguistic meaning and values of the applied features is explained in chapter 4 and we refer back to it later in this section. A dummy word (x_{14} here) is added to the components of each sentence to be used for *undefined* roles.

8.1.2 Output Space

In the output space, an ontology $\mathcal{H}$ with Γ number of nodes is given. The nodes in the ontology are actually the target labels we tend to predict and we denote it as $l_{target} = \{l_i | l_i \in \mathcal{H}, i = 0 \ldots \Gamma\}$. The ontology is defined as a set of labels where $(\mathcal{H}, \prec)$ is a partial order. The symbol $\prec$ in our terminology represents the super-label relationship (see Section 7.1) thus

$$\forall \gamma, \gamma' \in \mathcal{H} : \gamma \prec \gamma' \text{ if and only if } \gamma \text{ is a super-label of } \gamma'.$$

The actual labels in the learning model are defined based on the nodes in the ontology, see chapter 4 figure 4.1.a. We define one single label *sp* which is an indicator function that receives a word x_i and indicates whether it is a spatial indicator, denoted as $sp(x_i)$ or briefly as sp_i. The roles of trajector and landmark are defined as linked labels. We present them with label strings *sp.tr* and *sp.lm* and their indicator functions act on a pair of words. The link label $sp.tr(x_i, x_j)$ $(sp.lm(x_i, x_j))$ receives two ordered words and indicates whether the first word is a spatial indicator and the second is a trajector (landmark) with respect to the first, this also is denoted briefly as $sp_i.tr_j$ $(sp_i.lm_j)$. We use two additional labels: *nsp* that indicates whether a word is not a spatial indicator, and *nrol* that indicates whether a pair is neither *sp.tr* nor *sp.lm*. These two labels are helpful to collect features of negative classes for distinguishing the spatial relations. The above mentioned labels are related to the SpRL semantic layer. The fine grained semantics of spatial triplets are indicated in lower nodes in the ontology related to the SpQL layer. All SpQL related nodes are linked labels related to triplets and the indicator function of each one identifies whether a spatial relation is of a certain spatial type such as *Region*, *Direction*, *EC*, and so on. We denote these linked labels by r_γ and the first one r_0 is the spatial relation label. These labels actually are linked labels that can be represented by label strings. For example, *sp.tr.lm.region* shows the single labels on which the *Region* node in the ontology directly depends.

Output structure. The structural properties of the output are described in Section 4.3.1. Here, given the introduced parameters for the representation of the input and output in our LAL model, those constraints are formalized in Formulas 8.1-8.12. The labels are represented as indicator functions for the candidate inputs. For example $sp(x_i)$ is equal to one if the candidate word x_i is a spatial indicator. The first two additional constraints 8.1-8.2 associate the labels in the learning model to the nodes in the ontology. Constraints 8.6-8.9 are the *horizontal* constraints including *multilabel*, *spatial reasoning* and *composed-of* constraints. The last three formulas show the *vertical* constraints including the *is-a* constraint in Formula 8.10, the *null-assignment* constraint in Formula 8.11 and the *mutual exclusivity* constraint in Formula 8.12.

$$nsp(x_i) \leftarrow 1 - sp(x_i) \tag{8.1}$$

$$nrol(x_i, x_j) \leftarrow (1 - sp.tr(x_i, x_j)) \wedge (1 - sp.lm(x_i, x_j)) \tag{8.2}$$

$$sp(x_i) \leftarrow sp.tr(x_i, x_j) \tag{8.3}$$

$$sp(x_i) \leftarrow sp.lm(x_i, x_j) \tag{8.4}$$

$$sp(x_i) \rightarrow \exists x_j, x_k \quad sp.tr(x_i, x_j) \wedge sp.lm(x_i, x_k) \tag{8.5}$$

$$x_k \neq x_j \leftarrow sp.tr(x_i, x_j) \wedge sp.lm(x_i, x_k) \tag{8.6}$$

$$sp.tr(x_i, x_j) \wedge sp.tr(x_k, x_j) \rightarrow x_i = x_k \tag{8.7}$$

$$sp.lm(x_i, x_j) \wedge sp.lm(x_k, x_j) \rightarrow x_i = x_k \tag{8.8}$$

$$r_0(x_i, x_j, x_k) \leftarrow sp(x_i) \wedge sp.tr(x_i, x_j) \wedge sp.lm(x_i, x_k) \tag{8.9}$$

$$r_\gamma(x_i, x_j, x_k) \leftarrow r_{\gamma'}(x_i, x_j, x_k) \quad \forall \gamma \prec \gamma' \quad \gamma, \gamma' \in \mathcal{H} \tag{8.10}$$

$$\sum_{\gamma \in \mathcal{H}_{leafs}} r_\gamma(x_i, x_j, x_k) \geq r_0(x_i, x_j, x_k) \tag{8.11}$$

$$\sum_{\gamma \in QSR_h} r_\gamma(x_i, x_j, x_k) <= 1, \quad \forall QSR_h \subset \mathcal{H}_{leafs}, \tag{8.12}$$

to clarify the notation in the last two constraints, as described in Section 4.2.2, in the lightweight ontology $\mathcal{H}$, we have three general types of spatial calculi models, *regional*, *directional* and *distal*. The leaf nodes in the ontology are constructed based on multiple spatial calculi. Here the set of leaf nodes defined as $\mathcal{H}_{leafs} = QSR_{regional} \cup QSR_{directional} \cup QSR_{distal}$. The *null-assignment* constraint 8.11 imposes at least one fine-grained semantic assignment in a leaf node when a spatial relation is predicted. In constraint 8.12, to express the mutual exclusivity we denote each group of leaf nodes that belong to a qualitative spatial representation model as QSR_h.

Output representation. At the prediction time, the model predicts the labels of the input components. The output is the spatial ontology that is populated by the input components (i.e. segments of the input sentence). The populated ontology can be represented as a set of the sets of components associated to each label in the ontology. For the above mentioned example in Section 8.1.1, to be more illustrative we represent the output example with the indicators with value one:

```
{ {sp(on)},{sp.tr(on,statue)},{sp.lm(on,hill)},
{sp.tr.lm(on,statue,hill)},
{sp.tr.lm.region(on,statue,hill)},
{sp.tr.lm.direction(on,statue,hill)},
{sp.tr.lm.region.EC(on,statue,hill)},
{sp.tr.lm.direction.above(on,statue,hill)} }
```

see also Fig 4.1.b.

8.1.3 Joint Feature Mapping and the Main Objective Function

Templates. To describe the structure of the joint feature functions, we define the templates (see Section 7.1) in our model. We use four main types of templates: *Role* templates, *Composed-of* templates, *Is-a* templates and *Negation* templates.

- A *Role* template connects an input component to a single label indicating the role of that component. We use a *Role* template, for instance, for spatial indicators denoted as *word.sp.* The input type of this template is a *single word.*
- A *Composed-of* template connects a composed input component to a linked label. A linked label in this type of template contains sub-labels that linking them constructs new complex parts of the output. We mainly use two main *Composed-of* templates to connect trajectors/landmarks and spatial indicators. These templates indicate whether a pair of words have the *trajector-of* or *landmark-of* relationship and compose a part of a *spatial relation.* We denote them as *pair.tr*, *pair.lm.* We define an additional *Composed-of* template which is more complex and connects the three labels trajector, landmark and spatial indicator. This template indicates whether three words compose a spatial relation and it is denoted as $triplet.r_0$.
- An *Is-a* template connects a single or a composed component to a linked label. The linked label contains sub-labels that have is-a relationships to each other.

  For all semantic types of spatial relations we use such a template that connects triplets to their spatial relationship semantics, such as regional, directional, etc. We show them as $triplet.r_\gamma$ indicating the type of input that is *triplet*, and the semantic label r_γ linked label. It connects the *spatial relation* type to the more fine grained spatial semantics.
- A *Negation* template indicates when a single or linked label (referring to a single or composed concept in the output) is not assigned as one. We use two *Negation* templates. The first template is with a single label

for non-spatial indicators denoted as *word.nsp*. The second contains a negative linked label that indicates when a word is not a trajector nor a landmark with respect to a spatial indicator candidate. This *Negation* template is denoted as *pair.nrol*.

Now we specify the candidate generators and features of these templates.

Candidate generators. The spatial indicators are prepositions in our model which are mostly tagged as IN and TO by parsers. Hence, we prune their candidates based on the Pos-tags. Since prepositions belong to a closed lexical category, we collect a lexicon for prepositions according to our corpus. For the roles of trajector and landmark also a subset of words is selected. We define three basic sets of useful words in our problem as,

$$C_1 = \{x_i | Pos(x_i) \in \{IN, TO\} \vee x_i \in PrepositionLexicon\}. \tag{8.13}$$

$$C_2 = \{x_i | Pos(x_i) \in \{NN, NNS\} \vee Dprl(x_i) = SBJ \vee x_i = undefined\},$$

$$C_3 = \{x_i | Pos(x_i) \in \{NN, NNS, PRN\} \vee x_i = undefined\},$$

and choose the candidates for the labels based on these as follows:

$$C_{nsp} = C_{sp} = C_1, \quad C_{sp.tr} = C_{sp} \otimes C_2, \quad C_{sp.lm} = C_{sp} \otimes C_3,$$

$$C_{nrol} = C_2 \cup C_3, \quad C_{r_\gamma} = C_{sp} \otimes C_2 \otimes C_3,$$

where each C_{label} denotes the set of candidates for a *label*, $Pos(x_i)$ is a function that returns the Pos-tag of a word x_i and $Dprl(x_i)$ returns the label assigned by the dependency parser to a word x_i. *PrepositionLexicon* is the collected list of possible prepositions according to the available corpus. For the trajectors, the roles are assigned to singular (NNS) or plural nouns (NN) or the words that are labeled as subject (SBJ) in the dependency tree. For the landmarks the roles are assigned to singular, plural or proper nouns (PRN). Moreover, *undefined* can be a candidate for both roles. These linguistic features are extracted by the syntactic and the dependency parser.

Input feature functions. The input part of each template is characterized by a binary input feature vector, which is produced based on the local and relational features of the input components. We denote this feature vector using the symbol ϕ indexed by the relevant label,

$\phi_{sp}(x_i) \triangleq \phi_{nsp}(x_i) \triangleq$ *local features of the word* x_i.

$\phi_{sp.tr}(x_i, x_j) \triangleq \phi_{sp.lm}(x_i, x_j) \triangleq \phi_{sp.nrol}(x_i, x_j) \triangleq$ *local features of the word* x_j,

and relational features of the pair x_i *and* x_j.

$\phi_{triplet_{r_\gamma}}(x_i, x_j, x_k) \triangleq$ *local features of* x_i, x_j, x_k *and,*

relational features of the pair x_i, x_j *and, the pair* x_i, x_k $\forall r_\gamma \in \mathcal{H}$.

The local and relational features are described in Chapter 4.

Link-And-Label objective. Each instantiation of a template represents a joint feature sub-mapping. It is calculated by the product of a vector of the input features and an output label which is a single valued binary variable. For example, $\phi_{sp_i}.sp_i$ refers to the input features of the ith spatial indicator candidate multiplied by the value of sp_i, and $\phi_{sp_i tr_j}.sp_i tr_j$ refers to the features of ith spatial indicator candidate with respect to jth trajector candidate multiplied by the label $sp_i.tr_j$. We capsulate these two parts in a joint feature function f_p, associated to each template p with a label and its relevant input. We represent these, for example, as $f_{sp}(sp_i)$ and $f_{sptr}(sp_i tr_j)$. In this joint feature function, the label name and the index make the necessary connection to the input candidate. There is no need to show the x component explicitly. This function will be a zero vector if the label of the candidate is zero and will be equal to the input features of the candidate if the label is one. We have a block of weights for each template in the target model as,

$$W = [W_{sp}, W_{nsp}, W_{sptr}, W_{splm}, W_{spnrol}, W_{r_0}, \dots, W_{r_\Gamma}].$$

To construct the objective function $g = \langle W, f(x, y) \rangle$, each candidate for each label that is generated according to a template specification, is mapped to a joint feature vector (referred to as local joint feature). The local joint feature function is associated to a block W_p of weights for that template. In fact, the parameters of the variables related to one template are tied. The objective function is a linear function of feature values implying that we should sum over all the produced feature vectors multiplied by their weights,

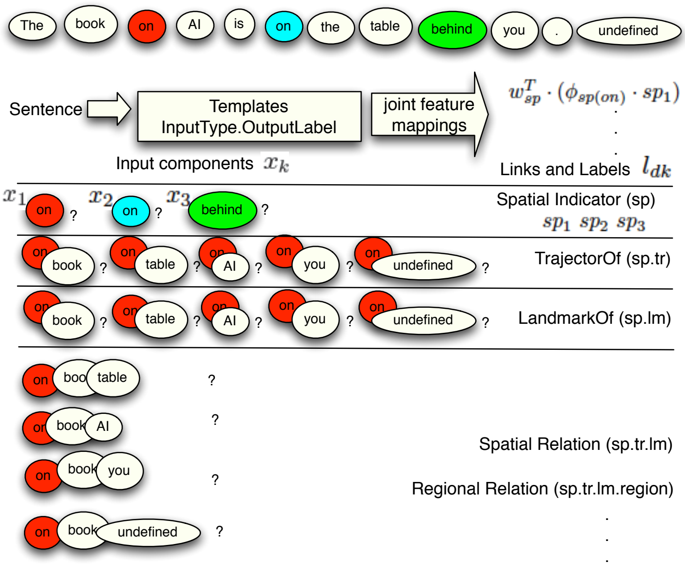
$$
\begin{aligned}
&\langle W_{sp}, f_{sp}(sp_1)\rangle + \langle W_{sp}, f_{sp}(sp_2)\rangle + \quad \cdots + \langle W_{sp}\ f_{sp}(sp_{_{SP}})\rangle + \qquad (8.14)\\
&\langle W_{nsp}, f_{nsp}(nsp_1)\rangle + \langle W_{nsp}, f_{nsp}(nsp_2)\rangle + \cdots + \langle W_{nsp} f_{nsp}(nsp_{_{SP}})\rangle +\\
&\langle W_{sptr}, f_{sptr}(sp_1 tr_1)\rangle + \langle W_{sptr}, f_{sptr}(sp_1 tr_2)\rangle \cdots + \langle W_{sptr}, f_{sptr}(sp_1 tr_{TR})\rangle +\\
&\qquad\qquad \ddots\\
&\langle W_{sptr}, f_{sptr}(sp_{SP} tr_1)\rangle + \cdots + \langle W_{sptr}, f_{sptr}(sp_{SP} tr_{TR})\rangle +\\
&\langle W_{splm}, f_{splm}(sp_1 lm_1)\rangle + \cdots + \langle W_{splm}, f_{splm}(sp_1 lm_{LM})\rangle +\\
&\qquad\qquad \ddots
\end{aligned}
$$

The above expression shows how we can compute the score of g for each input sentence and a proposed output. During prediction when finding the best assignments, we rewrite the f_p local joint feature functions as the product of input feature functions ϕ and the unknown output labels, as it is shown in figure 8.1.

We obtain a function g in terms of the labels. We can rewrite and represent the instances of a same template, which are associated to a same block of the weight vector, compactly as,

$$
\begin{aligned}
&\sum_{i\in C_{sp}} \langle W_{sp}, \phi_{sp_i}\rangle \cdot sp_i + \sum_{i\in C_{sp}} \langle W_{nsp}, \phi_{nsp_i}\rangle \cdot nsp_i +\\
&\sum_{i\in C_{tr}} \sum_{j\in C_{sp}} \langle W_{sptr}, \phi_{sp_i tr_j}\rangle \cdot sp_i tr_j + \sum_{i\in C_{lm}} \sum_{j\in C_{sp}} \langle W_{splm}, \phi_{sp_i lm_j}\rangle \cdot sp_i lm_j +\\
&\sum_{i\in C_{nrol}} \sum_{j\in C_{sp}} \langle W_{nrol}, \phi_{sp_i nrol_j}\rangle \cdot sp_i nrol_j +\\
&\sum_{\gamma=1}^{\Gamma} \sum_{i\in C_{sp}} \sum_{i\in C_{tr}} \sum_{i\in C_{lm}} \langle W_{\gamma}, \phi_{sp_i tr_j lm_k r_\gamma}\rangle \cdot sp_i tr_j lm_k r_\gamma. \qquad (8.15)
\end{aligned}
$$

This is the objective function that we need to maximize in the prediction time in order to find the best label assignments considering the global constraints mentioned in Section 8.1.2. We refer to the first three lines in 8.15 as the F_{SpRL} and the last line as the F_{SpQL}. The constraints can be between the variables related to one template, for example the counting constraints over



Each '?' is a binary variable indicating a label (single/linked)to be predicted for a candidate component of the input.

Figure 8.1: The input sentence is segmented according to the templates' generators, and the produced features over the input components are joined to unknown labels.

spatial indicators; or can be formulated more globally across templates, for instance by the *spatial reasoning* constraints.

8.1.4 Component Based Loss

As described in Section 2.1 , in structured training models we need to find the most violated output in Formula 2.22 for each training example per training iteration. Moreover, as mentioned in Chapter 7, to have a loss which factorizes similar to the feature function, we define a component-based loss for each label l according to Formula 7.5. We write a loss by considering all the instantiated labels according to the templates as follows,

$$\Delta_{sp}(\Lambda_{sp},\Lambda'_{sp}) = \sum_{i\in C_{sp}} (1-2sp'_i)sp_i + \sum_{i\in C_{sp}} sp'_i$$

$$\Delta_{nsp}(\Lambda_{nsp},\Lambda'_{nsp}) = \sum_{i\in C_{sp}} (1-2nsp'_i)nsp_i + \sum_{i\in C_{sp}} nsp'_i$$

$$\Delta_{sptr}(\Lambda_{sptr},\Lambda'_{sptr}) = \sum_{i\in C_{sp}}\sum_{j\in C_{tr}} (1-2sp'_itr'_j)sp_itr_j + \sum_{i\in C_{sp}}\sum_{j\in C_{tr}} sp'_itr'_j$$

$$\Delta_{splm}(\Lambda_{splm},\Lambda'_{splm}) = \sum_{i\in C_{sp}}\sum_{j\in C_{lm}} (1-2sp'_ilm'_j)sp_ilm_j + \sum_{i\in C_{sp}}\sum_{j\in C_{lm}} sp'_ilm'_j$$

$$\Delta_{spnrol}(\Lambda_{spnrol},\Lambda'_{spnrol}) = \sum_{i\in C_{sp}}\sum_{j\in C_{nrol}} (1-2sp'_inrol'_j)sp_inrol_j +$$

$$\sum_{i\in C_{sp}}\sum_{j\in C_{nrol}} sp'_inrol'_j$$

$$\Delta_{r_\gamma}(\Lambda_{r_\gamma},\Lambda'_{r_\gamma})) = \sum_{\gamma=1}^{\Gamma}\sum_{i\in C_{sp}}\sum_{i\in C_{tr}}\sum_{i\in C_{lm}} \omega_\gamma(1-2r'_\gamma sp'_itr'_jlm'_k)sp_itr_jlm_kr_\gamma +$$

$$\sum_{\gamma=1}^{\Gamma}\sum_{i\in C_{sp}}\sum_{i\in C_{tr}}\sum_{i\in C_{lm}} \omega_\gamma sp'_itr'_jlm'_kr'_\gamma. \tag{8.16}$$

In fact, we count all the wrong label assignments which have been made for all the template instantiations (see Hamming loss Formula 7.2). The first five lines in 8.16 are related to the SpRL labels which are averaged and aggregated with the loss of spatial semantic labels related to the SpQL layer. In the SpQL part, the *preferences* ω_γ of the labels r_γ inversely depend upon the distance of the label node from the r_0 node in the spatial ontology. The nodes closer to the leaves are assigned a lower value. This implies that we give a higher priority to the classification of more general semantics than the fine-grained spatial semantics in the leaf nodes of the tree. This is very straightforward and corresponds to the semantics of our problem. More specifically, we set the preferences such that firstly all the siblings with a common parent have a similar preference, secondly the preference of the parent is two times larger than the preference of its children, and thirdly the sum of all the preferences in the ontology is equal to one, so that we have a loss value between 0 and 1. Given these conditions we have a unique way to assign preferences to labels in the ontology.

8.2 Local-Global Training and Prediction Models

In this section we collect the required pieces from the last sections and chapters and discuss the model variations belonging to the spectrum of totally local and totally global training and prediction models that we design.

The global loss augmented objective is the following function,

$$\sum_{i \in C_{sp}} [\langle W_{sp}, \phi_{sp_i} \rangle + \frac{(1-2sp'_i)}{|C_{sp}|}] \cdot sp_i + \sum_{i \in C_{sp}} [\langle W_{nsp}, \phi_{sp_i} \rangle + \frac{(1-2nsp'_i)}{|C_{sp}|}] \cdot nsp_i +$$

$$\sum_{i \in C_{sp}} \sum_{j \in C_{tr}} [\langle W_{sptr}, \phi_{sp_i tr_j} \rangle + \frac{(1-2sp'_i tr'_j)}{|C_{sp}||C_{tr}|}] \cdot sp_i tr_j +$$

$$\sum_{i \in C_{sp}} \sum_{j \in C_{lm}} [\langle W_{splm}, \phi_{sp_i lm_j} \rangle + \frac{(1-2sp'_i lm'_j)}{|C_{sp}||C_{lm}|}] \cdot sp_i lm_j +$$

$$\sum_{i \in C_{sp}} \sum_{j \in C_{nrol}} [\langle W_{spnrol}, \phi_{sp_i nrol_j} \rangle + \frac{(1-2sp'_i nrol'_j)}{|C_{sp}||C_{nrol}|}] \cdot sp_i nrol_j +$$

$$\sum_{\gamma=1}^{\Gamma} \sum_{i \in C_{sp}} \sum_{j \in C_{tr}} \sum_{k \in C_{lm}} \left[\langle W_\gamma, \phi_{sp_i tr_j lm_k r_\gamma} \rangle + \omega_\gamma (1 - 2sp'_i tr'_j lm'_k r'_\gamma)\right] \cdot sp_i tr_j lm_k r_\gamma +$$

$$\frac{1}{|C_{sp}|} \sum_{i \in C_{sp}} [nsp'_i + sp'_i + \frac{1}{|C_{tr}|} \sum_{j \in C_{tr}} sp'_i tr'_j + \frac{1}{|C_{lm}|} \sum_{j \in C_{lm}} sp'_i lm'_j +$$

$$\frac{1}{|C_{nrol}|} \sum_{j \in C_{nrol}} sp'_i nrol'_j] + \sum_{\gamma=1}^{\Gamma} \sum_{i \in C_{sp}} \sum_{j \in C_{tr}} \sum_{k \in C_{lm}} \omega_\gamma sp'_i tr'_j lm'_k r'_\gamma. \tag{8.17}$$

We train the parameters W of the function g in the framework of discriminative inference-based structured prediction models such as structured SVM, structured perceptron and average perceptron algorithms described in Section 2.1.3. To train the parameters W including all W_p blocks of weights jointly, we need to maximize this objective function globally. This yields the most violated outputs for each training example per training iteration. This MAP problem over loss-augmented g containing both semantic layers of SpRL and SpQL is computationally highly complex. Moreover, the solution to this inference should fulfill the structural constraints discussed in Section 8.1.2, which makes it even harder. The feasible output space contains the space of all possible triplets multiplied by all possible ontological semantic assignments for each triplet, that is $O(n^3 \times 2^\Gamma)$ where n is the number of candidate words per sentence, which is assumed to be in the order of the length of the sentence n

and Γ is the number of nodes in the ontology. Even the above formulation for the two layers in 8.17 is computationally complex, because of the large number of $sp_itr_jlm_kr_\gamma$ variables, which is $O(n^3 * \Gamma)$ in this formulation, given that for each i, j, k all the $r_1..r_\gamma$ variables are also correlated and should fulfill the ontological constraints. To solve this as a combinatorial optimization problem using off-the-shelf solvers is still challenging. We consider this objective as a linear function and provide a linear formulation of the constraints as formulas 8.18-8.26 and solve it with LP-relaxation techniques. The global constraints 8.1-8.9 involve the variables of the SpRL layer and help to exploit the internal structure of the triplets and their global correlations in the sentence. The constraints are described in section 8.1.2 and here their linear formulation is provided. The last three additional constraints 8.23 are to limit the number of roles per sentence.

$$sp_i + nsp_i = 1, \quad sp_itr_j + sp_ilm_j + sp_inrol_j = 1 \tag{8.18}$$

$$sp_itr_j - sp_i \leq 0, \quad sp_ilm_j - sp_i \leq 0 \tag{8.19}$$

$$sp_i - \sum_j (sp_itr_j) \leq 0, \quad sp_i - \sum_j (sp_ilm_j) \leq 0 \tag{8.20}$$

$$sp_itr_j + sp_ilm_j \leq 1 \tag{8.21}$$

$$\sum_i (sp_itr_j) \leq 1, \quad \sum_i (sp_ilm_j) \leq 1 \tag{8.22}$$

$$\sum_i sp_i \leq MaxSp, \sum_i sp_itr_j \leq MaxTr, \sum_i sp_ilm_j \leq MaxLm. \tag{8.23}$$

The ontological constraints over the semantic assignments represented in the constraints 8.10-8.12 involve the variables of the SpQL layer. Here, the linear formulation of these constraints (that are described in section 8.1.2) is provided,

$$sp_itr_jlm_kr_{\gamma'} - sp_itr_jlm_kr_\gamma \leq 0, \quad \forall \gamma \prec \gamma' \quad \gamma, \gamma' \in \mathcal{H} \tag{8.24}$$

$$\sum_{\gamma \in QSR_h} sp_itr_jlm_kr_\gamma \leq 1, \quad \forall h, \quad \forall QSR_h \subset \mathcal{H}_{leafs} \tag{8.25}$$

$$\sum_{\gamma \in \mathcal{H}_{leafs}} sp_itr_jlm_kr_\gamma \geq sp_itr_jlm_kr_0. \tag{8.26}$$

Now that we have the global objectives 8.15 and 8.17 along with the features and all the constraints, we can easily formulate different LO, L+I and IBT learning models described in Section 7.2. We design and experiment with various models per semantic layer. Afterwards, we build a global model for both layers which necessitates going beyond off-the-shelf solvers and using our proposed communicative inference approach. In the models listed below we increase the level of *globality* in the training and prediction gradually. By the level of globality we literally mean the number of output variables that are considered collectively/jointly during training and prediction.

- **LO setting per layer.** A basic model which can be still practical, depending on the application, is the LO model. In the LO setting, independent binary classifiers are built for each single/linked label in the SpRL and SpQL layers referred to as **LOSpRL** and **LOSpQL** respectively. As discussed before, this setting can be seen as a kind of decomposed learning in which the cross templates constraints as well as autocorrelations are ignored and the individual input components are treated as i.i.d. examples for the relevant binary classifiers. These local models make independent binary predictions per label.
- **(L+I) setting per layer.** In this setting, locally trained models are used, but the joint prediction is performed by constrained optimization of objective 8.15 subject to the constraints 8.18-8.26. The main prediction time objective function is split into two parts, each of which relates to one layer with its own independent constraints. We refer to these models as **LISpRL** and **LISpQL** for the SpRL and SpQL layers.
- **IBT setting per layer.** In this setting, the objective 8.17 is solved during training per layer. In other words, it is split into two parts each part containing its own loss function and considering its own independent constraints. These models are referred to as **IBTSpRL** and **IBTSpQL**, for the SpLR and SpQL respectively.

Since the second layer is an extension of the first, all the unknown labels of the first layer are also unknown in the second, making the learning in the second layer as complex as the global learning over both layers. Therefore in all above mentioned models when modeling the learning and prediction in the second layer we assume the first layer of spatial relations are available and the focus of the training prediction is on the semantics of the relations. This assumption is useful for analyzing the difficulties of the two layers independently in the experimental section. However, we need to connect the two layers and make global training and prediction models encompassing both layers. Therefore the following models are designed:

- **L+I setting joining two layers.** In this model, we use the above mentioned IBT models per layer but make a global optimization jointly for both layers during prediction time. We use *communicative inference* Algorithm 4 during the prediction while solving each sub-problem using constraint optimization and LP-relaxation. We call this model **EtoE-IBTCP**.

- **IBT setting joining two layers.** To train a model jointly for both layers, we use the *communicative inference* (algorithm 4), which connects the two LP-relaxation subproblems in the training time for solving the global loss augmented inference. This globally trained model makes a joint prediction for the two layers too. We call this model **EtoE-IBTCTCP**.

In Section 8.3, the communicative approach is compared to the instantiations of the decomposed learning **DecL-1**, **DecL-2** described in Section 7.2.3. Moreover, we use another instantiation based on decomposing the two layers referred to as **DecL-SpRL-SpQL**. We compare all mentioned models to pipelining the IBT models of the two layers in the **EtoE-pipe** model.

8.3 Experimental Results and Analysis

We provide an empirical investigation of the efficiency and the performance of the designed structured learning models for mapping natural language to spatial ontologies. The experiments are organized based on the evaluation and comparison of LO, L+I and IBT learning schemes. In the end, our global model based on communicative inference is compared to some DecL variations and the pipeline model.

The applied base learning techniques are the structured SVM using the SVM-struct Matlab wrapper [157] (coded as **SSVM**) based on Algorithm 1 in Chapter 2 and our implementation of structured perceptron (coded as **SPerc**) and averaged structured perceptron (coded as **AvgSPerc**) based on Algorithm 2 in Chapter 2. For local learning settings, a binary SVM (coded as **BSVM**) is used. For the LP-solver, we use the Matlab optimization tool (*bintprog*) which employs branch and bound techniques.

The experiments of this chapter are based on the SemEval-1 edition of the annotated data described in Chapter 3. Table 8.1 shows statistics about the spatial roles in this edition and the influence of the candidate pruning, which leads to some missed positive roles in these experiments. The evaluation methodology is according to the metrics described in Chapter 2.[1] For the statistical significance of the differences we use the t-test over ten folds. Due to the computational complexity and the motivations described in section 8.2, the empirical investigations are also structured in three main parts, first we

[1] The evaluations are over the positive class of each label and all are denoted with a similar symbol to the related label but in bold e.g. (sp=1 as **sp), etc.**

Component	Annotated	Pos.candidates
sp	1466	1437
tr	1588	1555
lm	1184	1152
r0	1706	1619

Num. of sentences	1213
Num. of all words	20,095

Table 8.1: Spatial roles statistics, positives after candidate selection.

investigate the SpRL layer, then the SpQL layer given the SPRL ground-truth triplets, and finally the connection between the two layers using communicative inference is discussed.

8.3.1 SpRL

The goal of the experiments described in this section is to answer the following questions about learning the SpRL layer in the three LO, L+I and IBT settings pointed out in Section 8.2,

Q8.1. What is the performance of local models which make local predictions for the output variables in this layer *(i.e. LOSpRL evaluation)*?

Q8.2. Does global prediction in the L+I setting improve the results *(i.e. LISpRL vs. LOSpRL)*?

Q8.3. Does considering correlations between output variables in the IBT model improve the results *(i.e. IBTSpRL vs. LISpRL)*?

Q8.4. What is the influence of the local and relational features in the IBT model? What is the influence of the applied constraints in the IBT model *(i.e. IBTSpRL features and constraints)*?

LO and L+I Models

A variety of local models can be trained for prediction of the single and linked labels in the SpRL layer. In this section we implement a variety of LOSpRL and LISpRL models and present our experimental results and analysis.

LOSpRL-1. This is a basic LO model with one BSVM classifier. The training examples are the positive and the negative candidate triplets of *sp.tr.lm* linked label. In fact these examples are produced with the same basis as the triplet *Composed-of* template described in Section 8.1.3 with the same features. Each candidate triplet acts as an i.i.d positive or negative example. The binary *output* of the classifier indicates whether a triplet is *spatial* or *not spatial.* The classified triplets then imply which words play which single spatial roles.

The problem of this setting is the enormous amount of negative examples compared to the positive ones. There are 10,809,000 possible combinations of three words in the corpus without pruning the candidates. This amount will be even more if the undefined roles are considered. Even by pruning the set of possible candidate words, 193,890 triplets are generated while there are only 1706 positive ones in the corpus. Moreover, 87 positive triplets are missed due to the candidate pruning. This means only 1619 positive examples will remain at the end.

The LOSpRL-1 given our base features, failed to yield meaningful results. However, this setting is useful if more combinatory relational features based on the training data are used in the presence of some stronger heuristics for candidate pruning. The work in [121] uses such a model and uses automatic feature selection methods to obtain a large number of combinatory features. In addition, it exploits some linguistic resources such as Penn TreeBank for pruning the candidates. In this way they could achieve reasonable results for this task compared to other more sophisticated settings. In our model in Chapter 6 also the long distance contextual features used by exploiting a graph kernel yielding reasonable results with binary classification of the triplets. However, our cross domain experiments in chapter 5 show SpRL is very sensitive to the domain of the data and achieving a domain independent model for it is difficult. According to our analysis, using models that are trained on more domain specific and complex patterns that are automatically drawn from the training data gives a higher chance to a failure over the cross domain experiments. To our knowledge the spatio-linguistically motivated features and background knowledge that are independent from a specific training set are more reliable for achieving a domain independent model. Therefore, it is necessary to design models that are able to exploit more abstract and higher level structural and relational features in addition to the background knowledge, in a more sophisticated and structured way.

LOSpRL-2. This is an LO model with three BSVM classifiers. The training examples for each classifier are positive and negative candidate single words associated with the *sp*, *tr* and *lm* single labels. In fact these examples are produced with the same basis as the single *Role* templates described in Section 8.1.3 with the same features. Each candidate word acts as an i.i.d positive or negative example. A word is a positive example for a role classifier if it plays that role in at least one spatial relation in the sentence. The binary output of each classifier indicates whether a word has a role or not. In this model, the triplets are predicted by a naive use of the predicted single roles. We produce all possible combinations of the positive predicted roles. This means simply calculating the Cartesian product of the three predicted sets of roles per sentence, $sp \times tr \times lm$. The undefined roles are simply produced when a specific role set is empty. If the *sp* set is empty, this is an indication that there is no spatial relation in the sentence even if a trajector or a landmark are predicted.

The total number of words in the corpus is about 20,095 words (ignoring punctuation). More than 85% of the examples are negatives. We pruned the set of examples by using only the candidate roles described in Section 8.1.3 in the training phase. In this way, we reduced the number of negatives dramatically and this lead to a reasonable accuracy for classification of each individual component.

The first points in all graphs of figure 8.2 show the results of this experiment that is using **LOSpRL-2** for each label. It is reasonable that the naive way of producing the pairs and triplets yields a low value for F1 for these linked labels since the relational features that should help the prediction of the links, have not been used in this experiment.

LISpRL-1. This is an L+I model based on the trained LOSpRL-1 model. This model uses SpRL constraints 8.3-8.4 for the *joint prediction* of the roles and triplets. During the joint optimization if there are any trajectors and landmarks then there should be a spatial indicator in the sentence detected. Adding these two constraints improves the classification of trajectors and landmarks compared to the LOSpRL-2 model. However, to fulfill the constraints, it recalls every possible spatial indicator leading to a dramatic decrease in the precision of this label. The drop in the precision of spatial indicators causes a clear decrease mostly in the precision of the pair and triplet predictions and consequently a decrease in F1, e.g. this decrease is about 10% in F1 for the triplets. All LISpRL-1 points of the graphs in figure 8.2 show the results of this experiment in precision, recall and F1 and for all the labels (i.e. roles, pairs and triplets).

LISpRL-2. We gradually add more constraints at prediction time. We add *multi-label Constraint* 8.6 which allows maximum one label for each word with respect to a specific spatial indicator. Apparently the system distinguishes between the trajectors and landmarks based on their local features and does not assign the two roles simultaneously to one word. Hence, adding this constraint in this setting does not make any improvements in the predictions compared to LSpRL-1.

The next two constraints that we add to LISpRL-2 at prediction time, are the *spatial reasoning constraints* 8.7-8.8. These do not allow a trajector or a landmark to be connected to more than one spatial indicator. By adding these constraints, the predictions about some single roles are corrected and an increase about 2% for trajectors and 3% for landmarks in F1 is achieved. Even with such an improvement for single roles we observe a drop in the performance of the prediction of pairs and triplets. Adding the additional composed-of constraint 8.5 which imposes the prediction of trajector and landmark when there is an indicator, increases the precision of all these roles compromising the previous constraints 8.3-8.4.

Although the results of LISpRL-2 outperform the results of LISpRL-1, still the graphs show a lower F1 for the pairs and triplets in LISpRL-2 compared to the LOSpRL-2. This observation is reasonable. Since no relational features are used to give priority to certain pairs compared to others, the connections are chosen almost randomly because of the lack of distinguishing features. The constraints drop the recall of the connections and after all we get a drop in both recall and precision in prediction of the pairs and consequently the triplets. Adding the relational features is the subject of the next experiment. The results of the final LISpRL-2 using all constraints is presented in the graphs of figure 8.2.

LISpRL-3. This is an L+I model which is trained with more complex local models compared to the above mentioned L+I models. It uses two BSVM pair classifiers instead of single role classifiers. The training examples for each classifier are, positive and negative candidate pairs of words associated with the *sp.tr* and *sp.lm* linked labels. In fact these examples are produced with the same basis as the *Composed-of* templates for the pairs described in Section 8.1.3 with the same features. Each candidate pair acts as an i.i.d positive or negative example. The binary output of each classifier indicates whether two words are spatially linked to each other or not. The advantage of using these binary classifiers is that the relational features between words are considered and this can help the prediction of the links between the roles.

In this setting the link between triplets can be produced based on the link between pairs. Before pruning there are 424500 pairs. For *sp.tr*, 21773 is the number of the training pairs after pruning. For *sp.lm*, this number is 25284 after pruning. The experiments without pruning the candidates did not yield any meaningful results, pruning the candidates using the same rules for individual candidates but using only local features did not provide any significant improvement over a random classifier. However, after pruning the candidates and then adding relational features of 'before' and 'distance', a noticeable difference is made in the performance of the L+I model, this increase is about 24% in F1 for triplets. Since the evaluation done after pruning, there is a drop in the recall given the missed positive candidate pairs (about 2% – 3%). A triplet is produced when two predicted pairs share a spatial indicator. We simply introduce the undefined roles when a pair of *sp.tr* is predicted and a pair of *sp.lm* for the same *sp* is not predicted and vice versa. To see the effect of the constraints in this model, we experiment by gradually adding the constraints 8.6-8.9. The constraints effect the precision and recall in different ways, and no improvement achieved for F1 of the triplets which is of the most interest. The third point of the graphs in figure 8.2 show the results of this setting. By comparing these results with the previous one for

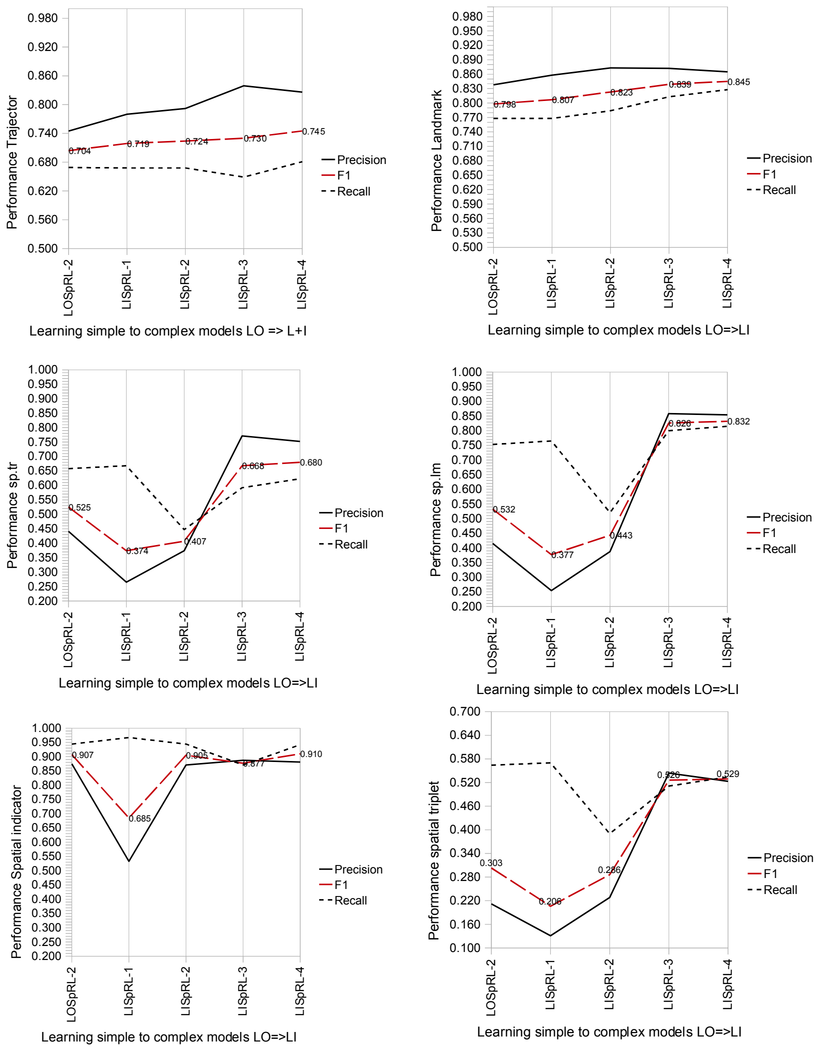


Figure 8.2: Performance of labeling spatial relations and their components based on training local models using BSVMs, 10-fold cross validation on SemEval-1.

the prediction of the individual basic components, it can be seen that the main problem in this setting is the lower recall of *sp* compared to all previous settings. In contrast to the previous models, the LISpRL-3 model does not train for this label independently. In general prediction of *sp* seems easier than prediction of trajectors and landmarks but the influence of the errors in *sp* prediction is higher in the final performance since it will multiply the errors of the triplet prediction. This is the motivation of the next experiment. In the next step we try to exploit the high performance of the spatial indicator's binary classifier in a joint optimization with the pair classifiers.

LISpRL-4. This is an L+I model by training three independent BSVM binary classifiers including one for single *sp* role and two for the pairs of *sp.tr* and *sp.lm*. The horizontal constraints 8.6-8.9 are applied during prediction. Here also the examples for each classifier are produced based on *sp Role* template and the *Composed-of* pair templates. A joint optimization is performed by combining the binary model for the classification of spatial indicators and the pairwise setting for trajectors and landmarks. We impose the *composed-of* constraints that imply if a pair is predicted as positive then the spatial indicator also should have been labeled as positive; These constraints do not improve the results, since the recall of *sp* is already as high as its singular classification in this setting. In the graphs of figure 8.2 LISpRL-4 shows the maximum performance we could obtain from training these three model separately and performing a joint optimization by imposing the horizontal constraints.

Overall, these experiments show the essential importance of considering relational features in the LO and L+I models. The constraints mostly were not significantly improving the F1 measure of the spatial triplets, which is in our most interest. In the next section we experiment with joint training of the components.

IBT Model

In this section we experiment with the main global model which considers local and relational features as well as the constraints during both training and prediction. We perform loss-augmented inference using LP-relaxation (see formula 8.17) to train a global model for the SpRL layer. We empirically investigate how individual features and constraints influence such a model, starting with the SSVM framework. As mentioned before, to generate the joint feature maps we perform a candidate pruning step which is computationally trivial but to some extent essential for achieving a tractable model. The candidates are selected based on the provided definitions in 8.14. This pruning inevitably leads to a number of missed positives. The number of missed positives are about 2%-3% for each role. For spatial indicators we could cover all positives by building a lexicon. Performing experiments on the ground truth data shows that the candidate pruning leads to about 5% drop in the final recall of spatial relations.

Feature analysis in IBT. In this part of the experiments, the local and relational features described in Chapter 4 are gradually added to the model. As mentioned above the evaluations are over all singular, pairwise and triplet components in the output. The graphs in figure 8.3 represent the trend of the changes in the performance in terms of precision, recall and F1 for all aforementioned components. Each point in the horizontal axis is labeled with the features used in the evaluated model. The vertical axis shows the performance in terms of the corresponding evaluation metrics. In the evaluated models in these graphs the constraints 8.1-8.9 are used in both training and prediction. Naturally, the binary constraint of 8.1 is used only if the *nsp* label is employed in the model.

The first three graphs show the performance over the singular components. For trajectors, adding the local and relational features smoothly increases the precision, recall and F1. However, the two features of *spc* and *und* (see Chapter 4) are less relevant for trajectors since the first is related to spatial indicators and the second mostly to the landmarks. Therefore, a slightly negative impact on the classification of trajectors is observed when adding those two features. For the landmarks the impact of the first three local features is sharply increasing. The remaining features have a smoother impact on improving the classification of this role. For spatial indicators the most influencing feature is the word form and adding all other features causes a 2% improve in total, for this role. However, the accuracy of spatial indicator has more impact on the final classification of the spatial triplets because each missed or wrongly recalled spatial indictor can lead to several mistakes in the prediction of the pairs and triplets.

The second row of figure 8.3, shows the performance over the pairwise components. The expected observation in these two graphs is the high impact of the relational features, as it is theoretically expected. The local features have a very smooth increasing effect on classification of pairwise roles. A similar trend is observed for the influence of the features on the classification of the spatial triplets in the last row of the figure. Relational features have a stronger influence in classification of pairs and triplets than local features.

Constraint analysis in IBT. In the integrated structured learning model, the constraints 8.1-8.4 are essential to establish a meaningful relationship between the labels in the inference objective function. Looking into the learnt parameters in the experiments of section 8.3.1 showed that the weight of the negative features of the spatial indicator (related to the *nsp* label) are nearly

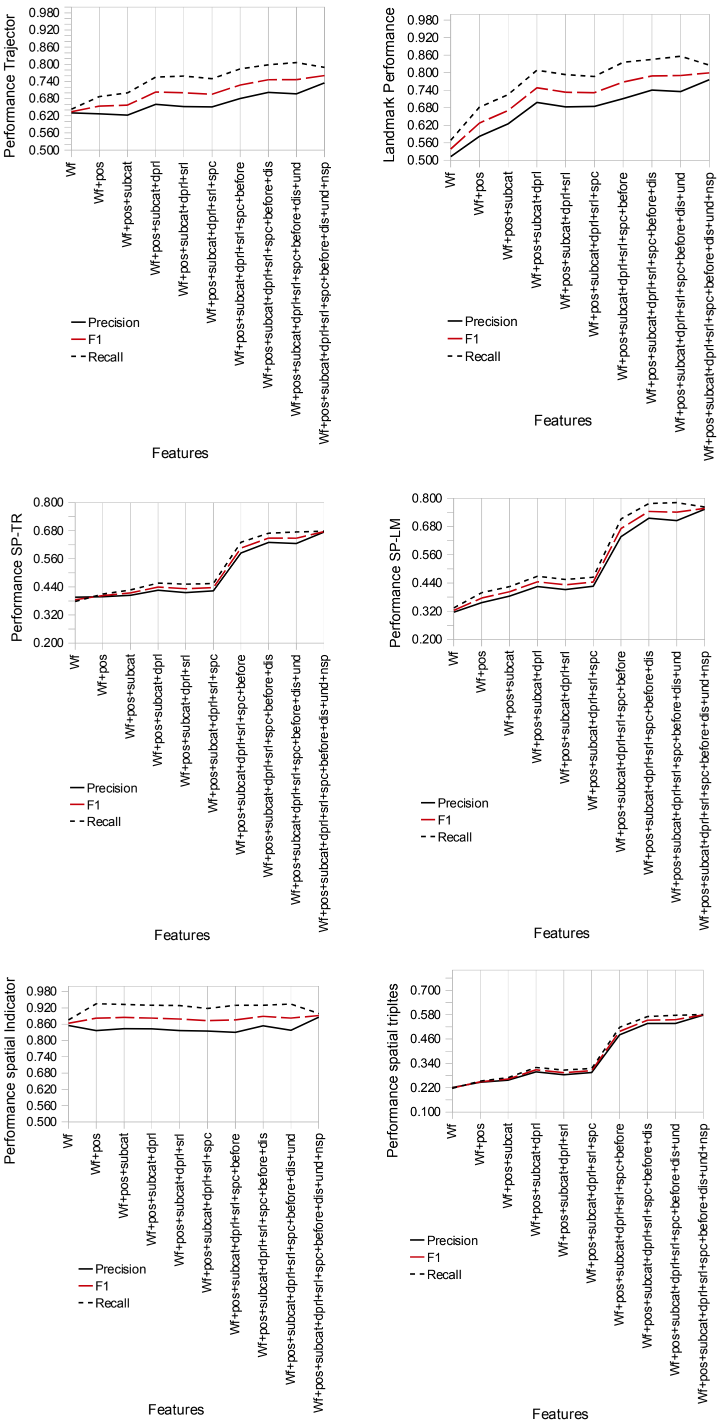


Figure 8.3: Performance of the spatial triplets and their components when adding local and relational features and *nsp* to the IBT model (SSVM), 10 fold-cross validation, SemEval-1.

the same as the weights of its positive features (related to *sp*) but with an opposite sign. Therefore, this component has a minor effect on the overall performance of the model. Obviously, the constraint over these two opposite labels in Formulas 8.1 is needed only if *nsp* is included in the model. In the final results we preserved this component for its positive, though minor final impact. On the contrary, keeping the none roles against being trajector and landmark and imposing constraint 8.2 was essential to obtain meaningful results. Similarly making the connections between pairwise variables and the singular *sp*s via *composed-of* constraints in 8.3-8.4 is essential. Without these constraints, no acceptable results are achieved due to the critical inference task of finding the most violated constraints.

The *spatial reasoning* constraints in 8.7-8.8 and *counting* constraints (with the linear Formulation 8.23) aim to provide some background knowledge related to the spatial language. When using *spatial reasoning* constraints, there is a drop of about 0.01 in F1 of the triplet prediction, but the difference is not statistically significant. However, by investigating the errors, we observe that by imposing the hard *spatial reasoning* constraints, the model is less robust to the noise present in the annotated data. Therefore, it decreases the recall in some cases that the sentences are annotated different from our spatial reasoning concerned guidelines. The *counting constraints* also decrease the recall of the model without any improvement in the precision. Therefore, we did not use them in our final experiments.

To conclude the experiment on the SpRL layer, we show the detailed results of the three main models that use the same local and relational features but in the three final LOSpRL, LISpRL-4 and the final IBTSpRL model using SSVM in Tables 8.2, 8.3, 8.4. Moreover, we implemented our best model, which is the IBT, with structured perceptron and also structured average perceptron. The averaged perceptron results IBTSpRL-AvgPerc shown in Table 8.5 are about 3% better than the basic structured perceptron model (F1 (SpRL)=0.574). It also outperforms SSVM for the SpRL layer, the differences are statistically significant (p=0.05). We remind that the instances of the spatial relation node are produced according to the (r_0) using the property in Formula 8.9.

Overall. The important findings related to experimental questions **Q8.1-Q8.4**, are as follows; Exploiting the structure of the output via a global constraint optimization during prediction increases the performance of labeling spatial relations in the SpRL layer compared to local predictions. Moreover the IBT model is the best performing one compared to the other two settings. The relational features were essential for obtaining reasonable results over pair and triplet predictions. Constraints were often useful but sometimes lead to a drop in recall, for example when using spatial reasoning and counting constraints.

Target	Precision	Recall	F1	Annotated	Pos. cand	Neg. cand
sp	0.875	0,944	0,907	1466	1437	1992
sp.tr	0.776	0.590	0.668	1693	1640	20133
sp.lm	0.868	0.793	0.827	1196	1161	24123
r0	0.498	0.510	0.503	1703	1619	–

Table 8.2: LO: Local training-Local prediction for single label *sp*, linked labels *sp.tr*, *sp.lm* and producing r_0 using rule 8.9, BSVM, LOSpRL.

Target	Precision	Recall	F1	Annotated	Pos. cand	Neg. cand
sp	0.881	0,942	0,909	1466	1437	1992
sp.tr	0.752	0.622	0.678	1693	1640	20133
sp.lm	0.853	0.815	0.832	1196	1161	24123
r0	0.526	0.533	0.529	1703	1619	–

Table 8.3: L+I: Local training-Global prediction for single label *sp*, linked labels *sp.tr*, *sp.lm* and producing r_0 using rule 8.9, BSVM, LISpRL-4.

Target	Precision	Recall	F1
sp	0.886	0.899	0.892
sp.tr	0.674	0.678	0.676
sp.lm	0.753	0.763	0.757
r0	0.578	0.581	0.579

Table 8.4: IBT: Global training-Global prediction over *sp*, *sp.tr*, *sp.lm* (using *nsp*, *sp.nrol*) building r_0 using rule 8.9 SSVM, IBTSpRL-1.

Target	Precision	Recall	F1
sp	0.905	0.8416	0.871
sp-tr	0.728	0.610	0.662
sp-lm	0.828	0.766	0.794
r0	0.663	0.554	0.602

Table 8.5: IBT: Global training-Global prediction over *sp*, *sp.tr*, *sp.lm* (using *nsp*, *sp.nrol*) building r_0 using rule 8.9 IBTSpRL-2.

8.3.2 SpQL Given SpRL

The goal of the experiments in this section is to answer the following questions about the spatial qualitative labeling (SpQL) given ground truth spatial triplets.

Q8.5. What is the performance of an LO model for SpQL layer given the spatial triplets (*i.e. LOSpQL evaluation*)?

Q8.6. Does the global prediction in the L+I setting improve the results of this layer *(i.e. LOSpQL vs. LISpQL)* ?

Q8.7. Does considering correlation between output variables in the IBT model improve the results of this layer *(i.e. LISpQL vs. IBTSpQL)*?

Q8.8. What is the influence of lexical and syntactical features in the IBT model? What is the effect of the applied constraints such as is-a and mutual exclusivity constraints *(i.e IBTSpQL features and constraints)*?

LO and L+I

LOSpQL. This is an LO model which contains a BSVM binary model for classification of each semantics type in the ontology. The training examples are ground truth spatial triplets which can be positive or negative with respect to each semantic type. Given the spatial triplets, the linked labels of the SpQL layer contain only one single semantic variable each, which reduces the number of candidates to a great extent and makes training local models feasible. Table 8.6 shows that LOSpQL model works fairly well in classification of course-grained semantics and particularly for the labels with a larger number of positive examples.

LISpQL. This is an L+I model which uses locally trained binary classifiers for each semantics type but performs global prediction and imposes the ontological constraints 8.10-8.12 for the SpQL layer. Table 8.7 shows the results of this experiment. For the single labels of the ontology (when using ground truth triplets) as expected imposing the constraints decreases the number of false positives leading to an increase in the overall weighted average of precision and a drop in recall. However the overall F1 measure increases by 0.007 when compared to the local predictions in table 8.6. In the nodes EC, PP, PO, BELOW, RIGHT, BEHIND, FRONT and ABOVE a dramatic increase is visible. Only in two nodes of LEFT and DC there is a drop in F1. In these two nodes though the precision increases, the decrease in the recall is comparatively more. Distinguishing between LEFT and RIGHT is difficult for our model since their features are often similar except for the lexical form. In general, these results indicate that imposing the *constraints*

Class	Precision	Recall	F
Region	0.9428	0.8927	0.916
Direction	0.8421	0.9189	0.8761
Distance	0.1143	0.8354	0.1985
EQ	0.3	0.7	0.1
DC	0.4073	0.6249	0.4495
EC	0.5328	0.8415	0.631
PO	0.0137	0.6071	0.0255
PP	0.5678	0.8405	0.6713
BELOW	0.68	0.76	0.5973
LEFT	0.3703	0.9606	0.5038
RIGHT	0.1075	0.9923	0.1872
BEHIND	0.492	0.9762	0.6357
FRONT	0.2338	0.9638	0.3632
ABOVE	0.6796	0.7706	0.6961
W.Avg.	0.69178	0.87708	0.73472

Table 8.6: LO, Given G-truth triplet, Local training-Local prediction for labels $r_{1..O}$, BSVM, LOSpQL, 10-fold cross validation, SemEval-1.

dramatically increases the performance of the lower level nodes in the ontology *compensating the lack of examples* for those nodes.

IBT

IBTSpQL-1. In this model we use LP-relaxation to solve the loss-augmented inference during training for the SpQL layer subject to the vertical constraints in the ontology. The employed features are the triplet features described in Section 8.1.3. We observed that using the SpQL constraints 8.10-8.12 for finding the most violated label assignments, provides a sharp improvement in the results compared to training local models particularly in terms of precision. The results of the IBTSpQL-1 model are shown in table 8.8. However, adding the mutual exclusivity constraints during training improves the results very slightly, but the convergence of the training is faster compared to not using these constraints. Our analysis shows that this is due to the selection of more elegant violating examples from the confusion set in the loss-augmented inference, that respect the structure of the output in the initial training iterations, from the confusion set. Imposing the constraints makes the LP-solver slower but the number of iterations for the cutting-plane algorithm of the SSVM to find a stable working set for Max-Margin optimization decreases

Class	Precision	Recall	F
Region	0.9387	0.9282	0.9325
Direction	0.693	0.9393	0.7854
Distance	0.1141	0.8354	0.1982
EQ	0.3	0.7	0.1
DC	0.6076	0.3555	0.3858
EC	0.6564	0.7515	0.6923
PO	0.7	0.4786	0.3698
PP	0.7713	0.7473	0.7536
BELOW	0.8667	0.81	0.7217
LEFT	0.561	0.2974	0.3302
RIGHT	0.2044	0.8305	0.3099
BEHIND	0.7476	0.9179	0.8108
FRONT	0.789	0.9199	0.8328
ABOVE	0.7247	0.7706	0.7261
W.Avg.	0.73909	0.81496	0.74172

Table 8.7: (L+I), Given G-truth triplet, Local training-Global prediction, constraints 8.9- 8.11 with LP-relaxation for labels $r_{1..O}$, BSVM, LISpQL, 10-fold cross validation, SemEval-1.

and consequently overall training time is to some extent faster then when using constraints in this specific case.

IBTSpQL-2. This is the same IBT model as IBTSpQL-1, but it receives only the **lexical features** of the ground truth triplets. This experiment indicates whether the lexical information and the words (i.e. lexical form of the trajector, landmark and spatial indicator) are sufficient to distinguish the semantics of the spatial relation between the spatial entities. The results in table 8.9 show that by removing the syntactical features such as dependency labels, sub-categorization and other features, the total performance of the model in terms of weighted macro average over F-measure decreases by 3%. This observation implies that different spatial semantics are expressed by using different linguistic constructs. For example, in directional relations, the probability of having the subject of the sentence as a trajector is higher, particularly in dynamic contexts. In topological relations both trajector and landmark are mentioned explicitly in the sentence so that having an undefined argument is less possible in this case. The drop in performance is higher for the directional relations which means that these follow more regular structural patterns compared to topological or distal relations. However, the performance of the model which relies only on the lexical information is still comparatively high. This result proofs the dominancy of this feature in

Class	Precision	Recall	F
Region	0.9359	0.9508	0.9427
Direction	0.8906	0.9179	0.9029
Distal	0.8196	0.7854	0.7896
EQ	0.9	0.7	0.6
DC	0.5962	0.6028	0.5816
EC	0.7243	0.7804	0.7466
PO	1	0.5286	0.5444
PP	0.7807	0.793	0.7833
BELOW	0.8167	0.76	0.6717
LEFT	0.5181	0.7551	0.5529
RIGHT	0.5174	0.3332	0.3488
BEHIND	0.9203	0.9024	0.9029
FRONT	0.8383	0.897	0.8593
ABOVE	0.8465	0.8212	0.8128
W.Avg.	0.8223	0.8442	0.82134

Table 8.8: (IBT) G-truth triplet, Global training-Global prediction for SpQL, SSVM, IBTSpQL-1, 10-fold cross validation, SemEval-1.

distinguishing the spatial semantics.

Constraints only during training. One finding in the experiments is that when the constraints are applied during training in the IBT model, applying them during prediction is required for obtaining an accurate prediction. This is expected because when using constraints during training, the trained model relies on the structure of the output in addition to the weight vectors to distinguish between labels. Consequently the absence of the constraints during prediction time leads to an inaccurate prediction (F1=0.264).

IBTSpQL-3. The last experiment in the IBT framework for the SpQL layer, is a model similar to IBTSpQL-1 but it makes use of the averaged structured perceptron instead of SSVM. Table 8.10 shows the results. Although the AvGSPerc yields the best results for the SpRL layer, for the SpQL layer the SSVM yields results with a higher F1 measure (3%).

Overall. Answering the main experimental questions **Q8.5-Q8.8** for this layer, we consistently observe that IBT outperforms L+I and both of these models outperform the LO model. Since the textual corpus contains descriptions about images, the frequency of the topological relations is high. Therefore, in the results of various settings for the SpQL layer, the performance of the models is higher in recognizing the topological type of relationships compared to the directional and distal categories. Particularly for the distal relations there are only 82 annotated examples in the data set, therefore the models have a lower

Class	**Precision**	**Recall**	**F**
Region	0.9223	0.9097	0.9155
Direction	0.8063	0.8841	0.8415
Distance	0.7391	0.7688	0.738
EQ	0.8	0.7	0.5
DC	0.6259	0.6568	0.6284
EC	0.747	0.7214	0.7295
PO	0.85	0.5786	0.6031
PP	0.7941	0.7917	0.7907
BELOW	0.88	0.69	0.6127
LEFT	0.4544	0.6644	0.4986
RIGHT	0.4027	0.4403	0.395
BEHIND	0.9578	0.944	0.9458
FRONT	0.6479	0.7742	0.6992
ABOVE	0.8715	0.8212	0.8028
W. AVG	0.799	0.81733	0.79928

Table 8.9: G-truth triplet, Global training-Global prediction for SpQL, SSVM, IBTSpQL-2 using only lexical features, 10-fold cross validation, SemEval-1.

Class	**Precision**	**Recall**	**F**
Region	0.9212	0.9479	0.933
Direction	0.9012	0.8676	0.883
Distal	0.6757	0.7672	0.6999
EQ	0.6	0.7	0.3
DC	0.4905	0.7141	0.5658
EC	0.7179	0.6466	0.67
PO	0.9	0.5786	0.5445
PP	0.709	0.7321	0.7162
BELOW	0.3313	0.76	0.3775
LEFT	0.6793	0.5286	0.5792
RIGHT	0.4155	0.564	0.4246
BEHIND	0.8821	0.8403	0.8489
FRONT	0.8406	0.8039	0.8151
ABOVE	0.9846	0.7201	0.7828
W.Avg.	0.80065	0.80468	0.79109

Table 8.10: (IBT) G-truth triplet, Global training-Global prediction for SpQL, IBTSpQL-3, AvGSPerc, 10-fold cross validation, SemEval-1.

performance for this category. The distal relations are often signaled by specific words present in the sentence. The words such as *far*, *close* can decrease the ambiguity about the distal semantics if the relation has been extracted correctly from the sentence. We suppose by enriching the corpus with additional training examples of distal and directional relations, our learning models will perform more accurately in these two categories too.

8.3.3 End-to-End SpRL-SpQL

Although LP-solvers are very efficient, yet having the target global model for each layer does not scale up for a global SpRL-SpQL model due to the large number of candidate triplets and the large number of constraints. Hence in the following sections we experiment on different solutions for building the end-to-end SpRL and SpQL. Particularly we aim to answer the following research questions empirically,

Q8.9. What is the performance of connecting the two IBT models trained independently for the two layers and making the prediction in a pipeline *(i.e. EtoE-pipe evaluation)*?

Q8.10. Can we use the above mentioned model, but practically make a global prediction over both layers? Can the communicative inference algorithm help to make this global prediction *(i.e. EtoE-pipe vs. EtoE-IBTCP)*?

Q8.11. Can we practically train a global model having a global loss-augmented inference and jointly train for the two layers? Can the communicative loss-augmented inference help to achieve such a global model over the two layers *(i.e. EtoE-pipe vs. EtoE-IBTCTCP)*?

Q8.12. How can DecL variations help to have a globally trained model? What are the difficulties given the empirical results *(i.e. DecL vs. EtoE-IBTCTCP)*?

Pipeline. A straightforward approach is to use the separately trained models of IBTSpRL and IBTSpQL in a pipeline for prediction of all the nodes in the ontology. We refer to this models as EtoE-pipe.
EtoE-pipe-1. In this model the IBTSpRL-1 predicts the first layer and then the prediction is piped to the IBTSpQL-1 while both use SSVM training. The results of the two layers are shown in Tables 8.4 and 8.11.
EtoE-pipe-2. In this model the IBTSpRL-2 predicts the first layer and then the prediction is piped to the IBTSpQL-3 model while both use AvGPerc training. The results of the two layers are shown in Tables 8.5 and 8.12.

These tables clarify the answer to the question **Q8.9**, and build a baseline for the connection between the two layers upon which we investigate the possibility of any improvement by joint inference over the two layers.

Class	**Precision**	**Recall**	**F**
Region	0.5856	0.6027	0.5924
Direction	0.4941	0.5072	0.4986
Distal	0.4055	0.3671	0.368
EQ	0.9	0.7	0.6
DC	0.2789	0.3025	0.2815
EC	0.4649	0.5385	0.4898
PO	0.7	0.3143	0.325
PP	0.5653	0.5514	0.5564
BELOW	0.6619	0.525	0.4317
LEFT	0.2539	0.4322	0.2764
RIGHT	0.1511	0.1763	0.0985
BEHIND	0.5245	0.534	0.5213
FRONT	0.4421	0.4865	0.4548
ABOVE	0.5757	0.5555	0.5579
W.Avg.	0.498	0.522	0.4982

Table 8.11: Pipeline SpRL and SpQL, SSVM, 10-fold cross validation, SemEval-1 (EtoE-pipe-1).

Class	**Precision**	**Recall**	**F**
Region	0.6668	0.541	0.5937
Direction	0.6024	0.5442	0.57
Distal	0.633	0.4087	0.477
EQ	0.9	0.7	0.6
DC	0.3828	0.3044	0.3304
EC	0.5711	0.4419	0.4862
PO	0.85	0.4643	0.4584
PP	0.5774	0.48	0.5214
BELOW	0.6	0.55	0.49
LEFT	0.4486	0.292	0.331
RIGHT	0.372	0.537	0.359
BEHIND	0.602	0.563	0.573
FRONT	0.558	0.508	0.525
ABOVE	0.654	0.485	0.513
W.Avg	0.593	0.4934	0.5266

Table 8.12: Pipeline SpRL and SpQL AvGSPerc, 10-fold cross validation, SemEval-1 (EtoE-pipe-2).

It is worth to notice that in this experiment, the EtoE-pipe-1 using AvGPerc outperforms the EtoE-pipe-2 using SSVM.

Communicative inference during prediction. Here the same independently trained models are used for the two layers and the suggested algorithm of communicative inference, Algorithm 4, is applied only during prediction.

EtoE-IBT-CP-1. The trained models are same as EtoE-pipe-1 based on SSVM. The results over the two layers after using communicative inference during prediction are reported in Tables 8.13 and 8.15.

EtoE-IBT-CP-2. The trained models are same as EtoE-pipe-2 based on AvgSperc. The results over the two layers after using communicative inference during prediction are reported in Tables 8.14 and 8.16.

The results of EtoE-IBT-CP-1, show about 0.001 improvement for SpQL and about 0.01 (significant only for $p = 0.1$) improvement over the SpRL when it receives feedback from SpQL during prediction compared to EtoE-pipe-1 model. The results on EtoE-IBT-CP-2 for the SpRL and SpQL are consistently outperforming (~ 0.003) compared to EtoE-pipe-2; the improvement for SpRL is not significant and for SpQL is significant for $p = 0.1$. This experiment provides a rather positive answer to the question **Q8.10**.

Target	Precision	Recall	F1
sp	0.888	0.861	0.874
sp-tr	0.675	0.648	0.661
sp-lm	0.770	0.7442	0.7566
r0	0.595	0.570	0.582

Table 8.13: SpRL by communicative inference during prediction, SSVM, 10-fold cross validation, SemEval-1, (EtoE-IBTCP-1).

Communicative inference during training. In this experiments a global model is trained using communicative inference during training for finding the most violated examples.

Target	Precision	Recall	F1
sp	0.907	0.838	0.870
sp-tr	0.732	0.610	0.663
sp-lm	0.831	0.764	0.795
r0	0.669	0.556	0.605

Table 8.14: SpRL by communicative inference during prediction, AvGSPer, 10-fold cross validation, SemEval-1, (EtoE-IBTCP-2).

Class	Precision	Recall	F
Region	0.6059	0.6002	0.6018
Direction	0.4902	0.471	0.4793
Distal	0.4464	0.3504	0.373
EQ	1	0.7	0.7
DC	0.3252	0.2918	0.2923
EC	0.484	0.5329	0.4987
PO	0.7	0.3143	0.325
PP	0.5662	0.5472	0.5551
BELOW	0.64	0.545	0.473
LEFT	0.2474	0.381	0.2608
RIGHT	0.1775	0.2307	0.1328
BEHIND	0.5064	0.4994	0.4976
FRONT	0.463	0.4519	0.4535
ABOVE	0.5224	0.5389	0.5177
W.Avg.	0.50967	0.50983	0.49913

Table 8.15: SpQL by communicative inference during prediction SSVM, 10-fold cross validation, SemEval-1, (EtoE-IBT-CP-1).

Class	Precision	Recall	F
Region	0.6672	0.5415	0.5945
Direction	0.6159	0.5472	0.5781
Distal	0.668	0.4087	0.4877
EQ	0.9	0.7	0.6
DC	0.3781	0.3044	0.3286
EC	0.571	0.4407	0.4849
PO	0.85	0.4643	0.4584
PP	0.5818	0.4822	0.5248
BELOW	0.6	0.55	0.49
LEFT	0.4522	0.292	0.3321
RIGHT	0.3767	0.5442	0.3649
BEHIND	0.6106	0.5627	0.5761
FRONT	0.565	0.5075	0.531
ABOVE	0.6544	0.4851	0.513
W.Avg.	0.5979	0.49456	0.52924

Table 8.16: SpQL by communicative inference during prediction, AvGSPerc, 10-fold cross validation, SemEval-1 (EtoE-IBT-CP-2).

Target	Precision	Recall	F1
sp	0.9046	0.84	0.8693
sp.tr	0.7332	0.6246	0.6726
sp.lm	0.831	0.7687	0.7969
r0	0.6731	0.5728	0.6171

Table 8.17: SpRL by communicative inference during training and prediction, AvGSPerc, 10-fold cross validation, Semval-1, (EtoE-IBT-CTCP).

EtoE-IBT-CTCP. This model is trained based on AvgSperc which gives the best results in the above experiments. The global training-time and prediction-time inferences are both based on the communicative inference which connects the solutions of the two IBT models over the two layers. The results on this model are presented in Tables 8.17 and 8.18. These results show that the communicative inference during training improves IBTSpRL-2 in table 8.14 (significant for $p = 0.1$). This improvement is due to the provided feedback by the second layer about the type of the spatial relations to the first layer for recognizing the spatial roles. However the performance of the second layer dropped in this setting compared to EtoE-IBT-CP-2 (see table 8.16) which does communicative inference only during prediction. This behavior can be due to the high performance of the semantic type labeling in general (see tables 8.10, 8.8) compared to the role labeling, therefore the feedback from the semantic types can promote the role labeling while learning from the noisy role labels in the presence of a small data set is more tricky for recognizing the semantic types. This result is an answer to the question **Q8.11**.

Decomposed learning (DecL). In another set of experiments some variations of the DecL algorithm in Section 7.2.3 are implemented to obtain a global training model.

DecL-1. In this basic setting, each decomposition member contains one label of an arbitrary type similar to the *Pseudo-max* approach.

DecL-2. In this setting, each decomposition member contains a pair of labels each of which has an arbitrary type.

DecL-SpQL-SpRL-1. In this setting we use a relational decomposition based on the types of the variables in each SpRL/SpQL layer defined as $\mathcal{S} = \{\mathcal{S}^1, \mathcal{S}^2\}$ where $\mathcal{S}^1 = \{sp, sp.tr, sp.lm, nsp, sp.nrol\}$ and $\mathcal{S}^2 = \{r_\gamma\}$, $\gamma > 0$.

DecL-SpRL-SpQL-2. This is a different variation of DecL in which a heuristic that we call piping violation is applied but it uses the same decomposition as the DecL-SpRL-SpQL-1 setting. In this model the most violated SpRL is found and then by fixing this MAP assignment the most violated SpQL is found.

Class	Precision	Recall	F
Region	0.6269	0.5454	0.581
Direction	0.6177	0.5738	0.5916
Distal	0.5127	0.3232	0.3451
EQ	0.9	0.7	0.7
DC	0.2379	0.2382	0.2183
EC	0.5061	0.3926	0.433
PO	0.85	0.4143	0.3917
PP	0.4947	0.5024	0.4954
BELWO	0.5583	0.575	0.38
LEFT	0.4674	0.255	0.1878
RIGHT	0.3389	0.5903	0.3729
BEHIND	0.607	0.5678	0.582
FRONT	0.5621	0.533	0.5377
ABOVE	0.5436	0.5033	0.518
W.Avg.	0.55317	0.49138	0.500

Table 8.18: SpQL by communicative inference during training and prediction, AvGSPer, 10-fold cross validation, SemEval-1, (EtoE-IBT-CTCP).

None of the first three settings provide comparable results to the aforementioned models. This outcome is due to: a) Relational nature of the problem; 2) Pipeline nature of the two semantic layers.

In DecL-1, DecL-2 where the relational nature of the problem and the type of the output labels is ignored, there will be a large number of sets in each decomposition and we need to generate all possible local combinations (for DecL-x and x>1 cases) and update the constraints based on the active variables for each decomposition. This generation overhead is highly inefficient and resembles a greedy generative search approach for inference which is both inefficient and inaccurate for our relational problem. Even in the case of using LP-relaxation for each sub-problem, updating the target objective and the global constraints for each sub-problem is an overhead that can be even less efficient than doing a global training using LP-relaxation.

In DecL-SpRL-SpQL-1 training, though we consider a relational decomposition based on the type of labels and their semantics, the pipeline nature of the two layers introduces new difficulties. If SpQL is set by the ground-truth to find the most violated SpRL, then in the obtained violating example the most violating relations are connected to null semantic labels and not to the true SpQL labels. These false negative semantic labels result in extensive weight updating for the SpQL variables. This provides a poor training for those blocks of weights which feedbacks badly in finding the most violated SpRL in later iterations. Moreover,

Pipe(SSVM)	Pipe(AvGPer)	Comm(AvGPer)	DecL(AvGPer)
8h53m3s	1h16m9s	1h24m24s	2h9m7s

Table 8.19: Training time per fold AvGPer(20 Iter), SSVM(80 Iter); averaged over 10 folds.

Communicative	Pipeline
28.4s	15s

Table 8.20: Prediction time per fold; averaged over 10 folds.

this leads to finding an inappropriate most violated SpQL when setting SpRL to ground-truth too. Consequently there is no improvement in learning the blocks of the weights of the two layers.

The DecL-SpRL-SpQL-2, on the contrary, provides reasonable results (F1(SpRL)= 0.61, F1(SpQL)=0.50) compared to other DecL settings. In this model, using the above mentioned heuristic, we alleviated the problem of DecL-SpRL-SpQL-1 setting caused by the pipeline nature of the two layers. The result is slightly better than the pipeline model (significant for $p = 0.1$) for SpRL but it shows a drop of about 0.02 in F1 measure for SpQL.

After all, obviously the DecL algorithm is appropriate for the cases where we can naturally partition the variables ignoring some correlations. In our case the labels in the second layer are all tightly correlated to the variables of the first layer, therefore in one hand the decompositions that take this correlations into account lead to the same complexity of the original problem and on the other hand the split based on the two layers can not achieve better results compared to the pipelining. Moreover, the basic local training models based on the templates were more efficient and effective than performing all the local decompositions of instantiated labels. These experiments imply the necessity of a more sophisticated modeling for the decompositions in the case of such a relational problem containing various types of components in the input and various types of labels in the output. This last analysis clarifies our answer to the above mentioned question, **Q8.12**.

Efficiency analysis. The AvGSPerc is highly efficient compared to SSVM. Although the cutting plane algorithm in SSVM reduces the duration of each training iteration, according to our experiments the number of iterations required for SSVM are many more than AvGSPerc. We achieve the best models in 10-20 iterations for AvGSperc while for SSVM by setting the training error to less than 0.1 at least 80 iterations are needed to converge. In table 8.19 the training duration of the pipeline model, which is the sum of the training time of the two layers, is reported per fold by averaging over 10 folds (iterations: SSVM=80, AvGSPrec=20).

The time tables show that using the communicative inference for training global SpRL-SpQL is highly efficient. In fact the communicative inference is converged in a few i.e. often less than 10 iterations. Therefore the efficiency of communicative training is comparable to the pipeline model. In the decomposed models of DecL-1 and DecL-2 the training is highly inefficient being about 10 times slower than the pipeline model due to generating the decompositions and propagating the constraints. However the DecL reported in the table 8.19 is the DecL-SpRL-SpQL case and has a reasonable training time. The reported prediction durations of the communicative inference and the pipelining in table 8.20 indicates the efficiency of the communicative inference. The numbers are obtained by averaging over 10 folds.[2]

8.4 Related Work

Structured support vector machines and structured perceptrons are among the most well-known discriminative structured learning approaches. These techniques have been successfully applied on different structured output prediction tasks in the natural language processing domains such as question answering [16], natural language statistical parsing [25] and also in other domains such as computer vision [103]. Our applied structured learning formulation is very generic and similar to the works in [150, 130, 155]. There are other max-margin based formulations of structured output prediction which are problem specific or formulated for a certain type of loss function [79, 114]. To our knowledge, we are the first to consider such techniques for a semantic task and considering ontological constraints during *both training and prediction.* However, considering global constraints *during prediction* in constrained conditional models have been used in other tasks such as semantic role labeling [110, 140], text compression and summarization [22], information extraction [126] and coreference resolution [34]. Exploiting constraints during training in an efficient way is investigated in [130], when proposing the DecL. This model has been applied on multi-class classification and named entity recognition only, while we pinpoint to the challenges of applying such training models in more complex relational tasks and for semantic extraction.

[2] The reported times are meant to give a fairly comparative assessment of the efficiency of the approaches. They are not accurately measured based on the CPU running time, but based on the time stamps of the saved results in the files.

8.5 Conclusion

In this chapter, we designed a global learning model for populating our proposed spatial ontology. The proposed model is able to flexibly exploit the local, relational features and global constraints. The training is based on structured SVMs and structured perceptrons. To perform inference during the training and the prediction, we use an ILP formulation and LP-relaxation techniques. Due to the complexity of populating the ontology in a joint setting for both SpRL and SpQL layers, the experiments are first performed based on the independent layers, and then the two layers are connected using the communicative inference algorithm for a global loss-augmented optimization jointly over the two layers. The experiments indicate that the global IBT learning models outperform the LO and L+I models for both layers. Moreover, using relational features is essential to achieve a reasonable performance for extraction of the spatial relations, and global constraints can help improving the models.

Particularly for the second layer the *is-a* and *mutual exclusivity* constraints are useful for both faster convergence in optimization and the accuracy of the structured model. Our proposed communicative inference was a solution to connect the two layers efficiently because learning an integrated end-to-end model was not feasible using off-the-shelf solvers. By using the communicative algorithm the second layer provides feedback to the first layer to adjust the weights of the features given the predicted spatial semantics of the triplets. The evaluation of the final end-to-end model using this algorithm shows an improvement compared to a baseline pipelining when the communication is used during *prediction*. Not only the final output, but also the performance of the first layer improved, particularly in recognition of the landmarks after the communicative inference.

According to our experiments the results of the communicative inference during *training* are also promising. The improvement is small but consistent and efficient. It seems that the communicative inference during learning can be effective if the two communicating models are sufficiently accurate (e.g. about 0.80), otherwise the noisy feedback might drop the accuracy compared to training independent models. In our case, the feedback of SpQL during training improves the model for the SpRL layer, but not vice versa. However the negative side effects of communicative inference can happen even in the case of communication during prediction, which was not the case in our experiments. Our additional experiments indicate that providing effective DecL decomposition models for such a relational domain necessitates a particular attention to be paid to the relational decompositions for inference. For example the types of the variables in such problems can guide automatic decompositions that are more sensible according to the semantics of the problem.

Epilogue

Chapter 9

Conclusion and Future Work

In this chapter the main conclusions of this thesis are presented. We point to the main future directions and the potential extensions of this research.

9.1 Conclusion

The outcome of this thesis on "Structured Machine Learning for Mapping Natural Language to Spatial Ontologies", is relevant to three areas when building machine learning models to connect natural language to spatial ontologies. These areas are spatial cognition, computational linguistics and machine learning.

- **Spatial cognition.** From the spatial cognition point of view, our computational models establish a long-needed bridge between earlier formalizations of space and spatial language. The models are independent from complex cognitively based linguistic principles, but are supported by actual usage of natural language and based on intelligent computational models that can learn from experience. This approach is language independent and it is robust in dealing with the ambiguity and polysemy in the spatial language.
- **Computational linguistics.** From the computational linguistic point of view, this work has a number of contributions in this field. We provide an annotation scheme, corpora and a well-formulated task for a particular semantic extraction that is useful for many applications, which received the attention as a standalone computational linguistic challenge. We also investigate the linguistic properties and structural characteristics of this problem that are useful for performing this task in

a machine learning framework -given that machine learning is the most dominant approach for most of the computational linguistic tasks.

- **Machine learning.** From the machine learning point of view the task that we tackle goes beyond the classical applications of machine learning and involves state-of-the-art challenges in this field. These challenges are dealing with the rich internal structure of the language and extraction of the semantics which possess rich structural characteristics. We design a number of models in the well-known frameworks for such complex tasks such as probabilistic graphical models, statistical relational learning and non-probabilistic structured output prediction models. In the last framework, we formalize a general model for extraction of the semantics in text represented in any arbitrary ontology. We formalize a structured learning model from relational data. We integrate efficient techniques based on constraint optimization for inference during training and prediction. In our developed framework we can consider relational features, global ontological constraints and background knowledge efficiently.

Now, we describe the conclusions and contributions of this thesis in a more fine-grained way and align with the **research questions** and **challenges** of Chapter 1.

The first phase of this research was to answer the questions **Q 1** and **Q 2** about *spatial primitives* and *formal representations*. These are the basis of our further computational models. In this direction, we proposed a spatial annotation scheme. The proposed scheme is modulated in two cognitive and formal representation layers through which a bridge between the language and multiple spatial calculi is established. Compared to the previous work, we use generic and domain independent spatial notions and can cover dynamic as well as static spatial information in our scheme. The spatial primitives are connected to *multiple* calculi models. This utility aids to cover the various aspects of spatial semantics in spatial expressions and to alleviate the problem of the gap between natural language and formal spatial representation and reasoning models. Mapping to formal models provides the possibility of spatial reasoning over the extracted information which is not possible to do on natural language directly. Due to the compatibility of the applied spatial notions with spatial visual perception, this scheme has a high potential to be used in the applications with multimodal settings. We started with a limited set of elements in this first practice to be able to proceed with constructing feasible computational models. However, the abstraction layers provide a modular view on the target spatial ontology which is useful for further extensions at each layer. Providing this scheme is the **first major contribution** of this thesis.

The second phase of this research was to answer the questions **Q 3** and **Q 4** about the *task* and *corpora*. Given the ambiguity and polysemy in the language, we assume machine learning models are best situated to deal with mapping natural language to a spatial semantic representation by learning from experience. To set up a supervised machine learning practice, we annotated rich corpora according to the proposed scheme. These corpora have been used as the first benchmark for spatial information extraction in SemEval2012. In this direction we formulate a computational linguistic task in two layers of *spatial role labeling* (SpRL) and *spatial qualitative labeling* (SpQL). We view and formalize this task in the framework of ontology population. Formulating a new computational linguistic task and corpora for statistical machine learning practices, is the **second major contribution** of this thesis.

The third phase of this research was to investigate the necessity of defining a standalone CL task and to answer *empirically* yet again question **Q 3** and question **Q 5** about problem features and solution models. Our first experiments confirm the importance of this task. We empirically show that a semantic role labeler and a dependency parser, which can be helpful in this respect, mostly fail to extract the basic spatial primitives and the spatial dependencies. We hypothesize a set of features and properties according to the studies over spatial language as well as similar semantic tasks such as spatial role labeling. These properties are applied in two distinct classes: features and constraints. A variety of state-of-the-art machine learning techniques are applied to investigate this problem initially in the first SpRL layer.

For the experiments over the SpRL layer, first we designed learning models based on *probabilistic graphical models*, particularly conditional random fields. The experimental results show the importance of considering context dependent features and long distance dependencies. The cross domain evaluations indicate the importance of the lexical features which is expected. Using the additional resources such as the preposition project data can improve the disambiguation of the prepositions and increase the domain portability of the models.

The relational nature of the problem calls for considering *statistical relational learning* approaches, particularly kLog. We provide a relational representation of the data using an ER diagram. For the learning problem, we program the learning problem declaratively using a logical language and exploit the background knowledge in a very flexible setting. Instead of considering the correlations between outputs a graph kernel can compensate for capturing the global structure of the model by making an extensive use of contextual features. This experiment shows fairly successful, though adding the sequential correlation between the outputs using SVM-HMM in kLog still shows improvements. The extensive usage of the contextual features can lead to over-fitting and motivates exploiting more domain independent structural characteristics of the problem via global constraints.

Our empirical investigation provides, for the first time, a qualitative as well as quantitative error analysis and pinpoints to the practical challenges of the spatial role labeling task in its various aspects. These experimental results obtained in two above mentioned main machine learning frameworks are the **third major contribution** of this thesis.

In the fourth phase of this research, we extend the task and the machine learning investigation of the previous stage by working on a full ontology population task encompassing both the SpRL and SpQL layers. We explore the answer to question **Q 5** yet, but in a more complex setting. In this direction, we generalize the problem and develop a unified structured prediction model for ontology population.

We propose a model called *Link-And-Label* (LAL) model in which the objective function is defined based on the notion of templates. The objective is produced by unrolling a set of templates and producing a multinomial function to be optimized subject to the linearly grounded first order constraints. The templates provide a clear view on the structure of the learning models and the ways their parameters are tied in the relational domains. The defined loss function also is defined based on the Hamming distance between the components of each type which is easy to integrate in the loss-augmented inference. Our model is the first unified structured learning framework which is proposed for ontology population. In previous works the ontology components are mostly dealt with locally or in pipelines, and its most elaborated case can only consider hierarchical relationships. The proposed global learning framework is instantiated for the case of spatial ontology population. In this model we deal with ontological relationships such as *is-a* and *composed-of*. We consider background linguistic and common sense knowledge in the form of constraints. Formalizing this global framework for supervised ontology population is the **fourth and the most important contribution** of this work.

In the last phase of this research, while working in the global ontology population framework, we need to answer the last research question **Q 6**, regarding the efficiency issues of global inference. Though the inference in the proposed LAL model is performed based on the combinatorial techniques and LP-relaxation, it is still a challenge to solve this problem using off-the-shelf solvers. In this direction we propose an approach for decomposing the inference to simpler subproblems each of which can be solved using existing solvers but those subproblems can communicate to each other. The approach we call *communicative inference* uses the ideas of *alternation optimization.* The decomposition is made by an expert. However, the LAL representation of the learning model can direct the decomposition approaches to act more effectively by considering the type of the variables particularly for further extensions of this work and doing automatic decomposition. In practice, the global (joint) learning model for all components of the ontology, was only feasible when we decomposed the two SpRL and SpQL layers and used the proposed communicative inference. In our best results applying the proposed communicative inference during training improves the results of the SpRL but not of the SpQL. Applying it during prediction improved the results of both slightly compared to pipelining the two tasks. The proposed inference technique in the context of ontology population and the empirical study applied on the spatial ontology is the **fifth major contribution** of this thesis.

9.1.1 Additional Note on Various Learning Frameworks

Here we provide an overview and conclusion of the experimental results obtained from various learning models for extraction of the elements in the spatial ontology.

The extraction of the spatial roles and triplets in the first layer are examined with various models in this thesis and in the works of participants in our proposed SemEval-2012 [69] and SemEval-2013 [67] shared tasks.

In general, in structured output problems, the most simple learning setting is a binary classification of the whole target structure. However, the experiments confirm that, it is not easy to obtain reasonable results from this setting. The main commonly known problem of such a setting is the large space of possible output structures that can be paired with the input part of a training/test example and relatively few true structures compared to the very large space of possible wrong structures. This produces a very large number of negative examples compared to the few positive ones for training a binary classifier. Only an extensive preprocessing before training a model can make such a basic approach functioning. This preprocessing can include, pruning the negatives by some background knowledge, automatic feature extraction [121] and using kernels as in Chapter 6 and in [4] to produce a higher dimensional feature space. At a more technical level biasing the learning models towards giving more importance to the errors in classification of positives compared to the errors in classification of negatives in parameter settings of the models when possible, is also a usual effort.

In this thesis, this type of models firstly are examined for the SpRL layer in our experiments within kLog. In those experiments the pruning step is performed based on some linguistic background knowledge about POS-tags of the roles and then the features are produced by the kLog graph kernel. However, none of these models were outperforming other settings. The best results in this binary setting achieved by Roberts et al. [121], the first participant of the SpRL shared task using the binary SVM classifier of LibLinear implementation. They could outperform the previous CRF models according to the official train-test one split of the shared task of SemEval-2012. However, using 10-fold cross validation our template based CRF model of Chapter 5 outperformed their model.

In the model of Roberts, et al., using the high recall heuristic for pruning the negatives is important (as in kLog) and the automatic feature selection approaches find the best set of features by analyzing the training data. Due to the extreme use of various combinatorial features, the risk of such an approach is over-fitting the model on the training data. This can happen using extensive contextual features in kLog too. Another solution for dealing with negative examples in the binary classification setting is a pipeline model. After classification of the roles, they are piped to another step of relation classification. In this way the training model of the second phase does not suffer form the huge space of negatives but still from the errors made by the first step. This approach also is experimented in kLog and also in the work of the participant of SemEval-2013 [4]. In SemEval-2013 participant system, for classification of the relations in the second step of the pipeline a tree kernel is used to produce contextual features. The results of this work is not comparable to other results due to the difference in the data sets and the evaluation setting, but in the experiments performed in this thesis, the pipeline model within kLog was performing poorly compared to other settings.

A different approach from the binary classification setting is to consider the parts of the output structure and model the correlations among output parts explicitly. This is in fact the structured learning approach. In this direction, our conditional random field model considers the sequential relationships between the spatial roles in their classification and constructs the relations based on the extracted roles. In kLog a similar setting is used by calling SVM-HMM for tagging the words in a sequence. However, the kLog language can not represent these type of output correlations for the learning model hence the sequential relations are read from the file format according to the applied SVM-HMM implementation. The SVM-HMM is used by the second SpRL participant later in SemEval-2013, in a first step of role classification in the same SpRL problem and also an extended problem in which the motion indicator and path elements are additionally classified. The results of the models are not comparable due to the difference in the data and the evaluation settings. But within the kLog framework the SVM-HMM setting was the best model compared to the binary classification and pipeline settings. However, our CRF model is still outperforming compared to the experiments with SVM-HMM on the same data.

All these experiments indicated the importance of considering the correlations between the parts of the output explicitly. Hence, we targeted this approach in a more flexible setting with non-probabilistic structured output prediction models. Using structured SVMs and structured perceptrons, we modeled the correlations between the roles that construct spatial relations. We worked in the primal formulation using the original feature space with SVM-struct.

The results show again that, with a slightly modified annotated corpus, the CRF models perform better on the SpRL layer. However in our structured learning setting the sequential relations are not modeled in contrast to the CRF modes. The flexibility of this setting facilitates considering these correlations in an extended model. For example, the Viterbi inference can be formulated in a linear program [125] and integrated to improve the results obtained from structured support vector machines and structured perceptrons.

The second layer of the ontology with all its nodes are considered only in this thesis and the data has not been completely released yet. Hence the comparisons and the results are towards investigation of the influence of increasing globality of the learning model in the sense of considering output correlations compared to training local models for the nodes and parts of the ontology. The results show that the global constraints often help improving the learning models.

After all, each learning model evaluates some aspects of this problem. All models indicate the importance of considering the contextual and relational features. kLog and other kernel-based models are more flexible in automatically producing and considering these features in the input level. The extensive contextual features might capture global corrections between the parts of the output. However, in the cases where there exist background knowledge about the form of these global correlations then exploiting these is an advantage (e.g. ontological constraints). The CRF models are vey efficient and capable as far as the correlations are sequential or about very limited long distance dependencies. Applying the very global correlations and constraints is technically difficult in the graphical models. Hence, the structured output prediction models based on constraint optimization techniques are best situated to consider these global corrections.

We assume, the latter models are the most flexible frameworks to consider all above mentioned elements. They are extendable for using kernels, for considering sequential dependency as in linear chain models and in considering arbitrary long distance dependencies. Decomposition approaches are well situated to deal with the complexity of inference in such learning models. However automatic decomposition and having a sufficiently general inference approach for all kinds of problems is a very important and difficult research question to be worked out by researchers in the future. Our Link-And-Label abstraction layer for learning in relational data domains and the proposed component-based loss function provide a framework that is sufficiently general for designing efficient global learning and inference models for ontology population in various data domains.

9.2 Future Directions

This thesis in its various dimensions opens up many potential future research directions to which we point in the following sections.

9.2.1 Domain Portability

From a computational linguistic point of view, the problem faces the typical challenges of the linguistic semantic tasks which is the lack of lexical information in the training data. Although accounting for the structure of the output helps improving the learning models, this shortage leaves always a challenge. In this direction, semi-supervised learning and feature expansions using latent word language models [35, 58, 36] are useful future directions. In the context of spatial semantics, using the linguistic resources to make general abstractions over the spatial entities can be useful, for example, to replace words with a higher level of abstraction over them such as being a *physical object*.

9.2.2 Relational Learning and Efficient Inference

Although we use relational data including various types and components in both input and output, we encode the propositionalized relations and linear constraints directly into the model. Hence an automatic propositionalization of the relational features and constructing the objective of the inference, starting from a relational language remain as a future work. For example, ontology representation languages such as OWL can be used in our context of ontology population and then the relational data and the knowledge can be decoded automatically into the learning model. Another future direction is to investigate more sophisticated inference algorithms for such relational domains. Particularly, an automatic decomposition by analyzing *parts* of the large ontologies and their relevant *subsets* of constraints in the efficient framework of LP-relaxation and dual decomposition techniques, seems as an optimistic research direction. A relational decomposition using the first order abstraction that the ontologies provide can guide a relational decomposition that is based on the types of groups of variables rather than individual variables. In the same direction, formalizing the usual pipelines of the computational linguistic tasks in a general decomposition framework in which each layer can provide feedback to the other layers in a principled way, seems a challenging research direction which is closely related to the models and decomposition ideas in this thesis.

9.2.3 Extended Applications

Our proposed ontology population model, can be applied on similar tasks where the segments in the text are labeled with semantics that belong to a predefined ontology. This applies to, for example, populating the concepts in the semantic web, or to information extraction from biomedical texts considering biological ontologies. Since our spatial annotation is compatible with the spatial perception from visual data, a well-fitting application is to use our model in a setting where the spatial information from images or videos are extracted using the same representation. These sources of information can be integrated and complement each other, for example, to help understanding a spatial configuration or a scene.

9.2.4 Spatial Ontology

The proposed annotation scheme includes a number of elements that are not considered in our computational models yet. These elements are mostly about dynamic spatial information including motion, path and frame of reference. In the current models, the classification of the roles and relations are word based. These models can be extended to labeling and linking the phrase constituents of a sentence such as in semantic role labeling.

The annotation scheme itself can be extended to cover the *spatial adjectives* which are useful for dealing with more complex spatial notions such as *size* and *shape*. Using this information expressed in the sentences can improve the semantic assignments to the spatial relationships. More sophisticated treatment of the distal relations is another dimension for extending the ontology which necessitates providing additional data. Currently some extensions on the *spatial role labeling* task regards *motion indicators* and *path*. They are the subject of the second workshop hold in SemEval2013[1] based on our proposed task. Moreover the resources that we pointed out in Section 4 are extended and annotated with these elements. Finally mapping to other types of spatial ontologies such as General Upper Model is beneficial given that it is well formalized in OWL and using Description logic. The obstacle of this line of research is the lack of annotated data to perform such a mapping in a supervised framework and for evaluation of the learning models.

9.2.5 Spatial Reasoning

One goal of the proposed annotation scheme was to connect natural language to qualitative spatial reasoning models. The recent research trend in this

[1] http://www.cs.york.ac.uk/semeval-2013/task3/

area aims to provide models and tools that are able to reason over multiple qualitative models such as SparQ [163]. Integrating our models with these tools is an immediate step in order to ground the idea of spatial reasoning over natural language which has not been possible before. Moreover, there are very recent research works that model reasoning over probabilistic qualitative spatial relations. In this direction the probabilistic label distributions can be used directly if probabilistic graphical models are applied for the ontology population task. In the case of using structured output prediction models there are approaches to produce preference scores and use them as additional information to the predicted output [94]. We assume that integrating spatial reasoning in our learning models can provide additional feedback for checking the inconsistencies in the predicted relations and improving the learning predictions. This is a hypothesis that empirically can be investigated. This setting can be compared to a setting in which we allow spatial reasoning with annotated examples during training of the models. It can be investigated whether these kinds of reasoning can be captured and learned by structured learning models. Although, this latter setting seems rather complex than being feasible, it is, worth an exploration.

Bibliography

[1] T. Baldwin, V. Kordoni, and A. Villavicencio. Prepositions in applications: A survey and introduction to the special issue. *Computational Linguistics*, 35(No. 2):119–149, 2009.

[2] M. Barclay and A. Galton. An influence model for reference object selection in spatially locative phrases. In C. Freksa, N. S. Newcombe, P. Gärdenfors, and S. Wölfl, editors, *Spatial Cognition VI: Learning, Reasoning and Talking about Space*, number 5241 in Lecture Notes in Artificial Intelligence, pages 216–232. Springer, 2008.

[3] M. Barclay and A. Galton. A scene corpus for training and testing spatial communications. In *Proceedings of the AISB Convention (Communication, Interaction, and Social Intelligence)*, 2008.

[4] Emanuele Bastianelli, Danilo Croce, Roberto Basili, and Daniele Nardi. UNITOR-HMM-TK: Structured kernel-based learning for spatial role labeling. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 573–579, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[5] J. Bateman and S. Farrar. Towards a generic foundation for spatial ontology. In Achille C. Varzi and Laure Vieu, editors, *Formal Ontology in Information Systems: Proceedings of the Third Conference (FOIS-2004)*. IOS Press, 2004.

[6] J. Bateman, T. Tenbrink, and S. Farrar. The role of conceptual and linguistic ontologies in discourse. *Discourse Processes*, 44(3):175–213, 2007.

[7] J. A. Bateman. Language and space: A two-level semantic approach based on principles of ontological engineering. *International Journal of Speech Technology*, 13(1):29–48, 2010.

[8] J. A. Bateman, J. Hois, R. Ross, and T. Tenbrink. A linguistic ontology of space for natural language processing. *Artificial Intelligence*, 174(14):1027–1071, 2010.

[9] John Bateman and Scott Farrar. Spatial ontology baseline. SFB/TR8 internal report I1-[OntoSpace]: D2, Collaborative Research Center for Spatial Cognition, University of Bremen, Germany, 2004.

[10] John A. Bateman. Ontological diversity: The case from space. In *Proceedings of the 2010 Conference on Formal Ontology in Information Systems: Proceedings of the Sixth International Conference (FOIS)*, pages 5–16. IOS Press, 2010.

[11] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, second edition, 1999.

[12] J. C. Bezdek and R. Hathaway. Some notes on alternating optimization. In Nikhil R. Pal and Michio Sugeno, editors, *Advances in Soft Computing*, volume 2275 of *Lecture Notes in Computer Science*, pages 288–300. Springer Berlin Heidelberg, 2002.

[13] S. Boyd, L. Xiao, A. Mutapic, and J. Mattingley. Notes on decomposition methods, notes for EE364B. Online, www.stanford.edu/class/ee364b/lectures.html, 2007.

[14] Razvan Bunescu and Raymond J. Mooney. Statistical relational learning for natural language information extraction. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*, pages 535–552. MIT Press, 2007.

[15] T. Butko, C. Nadeu, and A. Moreno. A multilingual corpus for rich audio-visual scene description in a meeting-room environment. In *ICMI Workshop on Multimodal Corpora for Machine Learning: Taking Stock and Roadmapping the Future*, pages 1–6, 2011.

[16] Yunbo Cao, Wen Yun Yang, Chin Yew Lin, and Yong Yu. A structural support vector method for extracting contexts and answers of questions from online forums. *Information Processing and Management*, 47(6):886–898, 2011.

[17] J. Carletta. Assessing agreement on classification tasks: the Kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

[18] L. A. Carlson and S. R. Van Deman. The space in spatial language. *Journal of Memory and Language*, 51:418–436, 2004.

[19] C. C. Chang and C. J. Lin. Libsvm: a library for support vector machines, 2001.

[20] Ming Wei Chang, Lev Arie Ratinov, and Dan Roth. Structured learning with constrained conditional models. *Machine Learning*, 88(3):399–431, 2012.

[21] E. Charniak and M. Johnson. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, ACL '05, pages 173–180. ACL, 2005.

[22] James Clarke and Mirella Lapata. Global inference for sentence compression an integer linear programming approach. *The Journal of Artificial Intelligence Research*, 31(1):399–429, 2008.

[23] B. Claus, K. Eyferth, C. Gips, R. Hörnig, U. Schmid, S. Wiebrock, and F. Wysotzki. Reference frames for spatial inference in text understanding. In *Spatial Cognition, An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*, pages 241–266. Springer-Verlag, 1998.

[24] A. G. Cohn and J. Renz. Qualitative spatial representation and reasoning. In F. van Harmelen, V. Lifschitz, and B. Porter, editors, *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*, pages 55–596. Elsevier, 2008.

[25] Michael Collins. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *Proceedings of the Association for Computational Linguistics-02 Conference on Empirical Methods in Natural Language Processing*, volume 10 of *EMNLP '02, ACL*, 2002.

[26] Michael Collins. Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, ACL '02, pages 489–496. ACL, 2002.

[27] Michael Collins. Parameter estimation for statistical parsing models: theory and practice of distribution-free methods. In *New Developments in Parsing Technology*, pages 19–55. Kluwer, 2004.

[28] F. Costa and K. De Grave. Fast neighborhood subgraph pairwise distance kernel. In J. Fürnkranz and T. Joachims, editors, *Proceedings of the 26th International Conference on Machine Learning ICML*, pages 255–262. Omnipress, 2010.

[29] J. Cussens. Issues in learning language in logic. In A. Kakas and F. Sadri, editors, *Computational Logic: Logic Programming and Beyond*, volume 2408 of *LNCS*, pages 171–177. 2002.

[30] D. Dahlmeier, H. T. Ng, and T. Schultz. Joint learning of preposition senses and semantic roles of prepositional phrases. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, volume 1 of *EMNLP '09, ACL*, pages 450–458, 2009.

[31] L. De Raedt, P. Frasconi, K. Kersting, and S. Muggleton, editors. *Probabilistic Inductive Logic Programming: Theory and Applications*, volume 4911 of *LNCS*. Springer, 2008.

[32] Luc De Raedt. *Logical and relational learning*. Cognitive Technologies. Springer, 2008.

[33] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: a probabilistic Prolog and its application in link discovery. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2468–2473. AAAI Press, 2007.

[34] Pascal Denis and Jason Baldridge. Joint determination of anaphoricity and coreference resolution using integer programming. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT'07)*, pages 236–243, 2007.

[35] K. Deschacht and M. F. Moens. Semi-supervised semantic role labeling using the latent words language model. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP '09)*, pages 21–29. ACL, 2009.

[36] Koen Deschacht, J. De belder, and Marie Francine Moens. The latent words language model. *Computer Speech and Language*, 26(5):384–409, 2012.

[37] Thomas G Dietterich, Pedro Domingos, Lise Getoor, Stephen Muggleton, and Prasad Tadepalli. Structured machine learning: the next ten years. *Machine Learning*, 73(1):3–23, 2008.

[38] P. Domingos and M. Richardson. Markov logic: A unifying framework for statistical relational learning. In *ICML'04 Workshop on Statistical Relational Learning and its Connections to Other Fields*, pages 49–54, 2004.

[39] Max Egenhofer. Reasoning about binary topological relations. In Oliver Gunther and Hans Jorg Schek, editors, *Advances in Spatial Databases*, volume 525 of *Lecture Notes in Computer Science*, pages 141–160. Springer Berlin Heidelberg, 1991.

[40] T. Finley and T. Joachims. Training structural SVMs when exact inference is intractable. In *International Conference on Machine Learning (ICML)*, pages 304–311. ACM, 2008.

[41] Thierry Fontenelle. FrameNet and frame semantics: a special issue. *International Journal of Lexicography*, 16(3), 2003.

[42] Paolo Frasconi, Fabrizio Costa, Luc De Raedt, and Kurt De Grave. kLog: a language for logical and relational learning with kernels. *CoRR*, abs/1205.3981, 2012.

[43] C. Freksa. Qualitative spatial reasoning. In D. M. Mark and A. U. Frank, editors, *Cognitive and Linguistic Aspects of Geographic Space*, pages 361–372. Kluwer Academic Publishers, 1991.

[44] Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, volume 2, 1999.

[45] A. Galton. Spatial and temporal knowledge representation. *Journal of Earth Science Informatics*, 2(3):169–187, 2009.

[46] H. Garcia-Molina, J. D. Ullman, and J. Widom. *Database Systems: The Complete Book*. Prentice Hall Press, 2 edition, 2008.

[47] Fausto Giunchiglia and Ilya Zaihrayeu. Lightweight ontologies. In *Encyclopedia of Database Systems*, pages 1613–1619. 2009.

[48] L. Grippo and M. Sciandrone. On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Operations Research Letters*, 26(3):127–136, 2000.

[49] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.

[50] M. Grubinger, P. Clough, H. Müller, and T. Deselaers. The IAPR benchmark: a new evaluation resource for visual information systems. In *Proceedings of International Conference on Language Resources and Evaluation (LREC)*, pages 13–23, 2006.

[51] N. Guarino and P. Giaretta. Ontologies and knowledge bases: towards a terminological clarification. *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, pages 25–32, 1995.

[52] David Hall and Dan Klein. Training factored PCFGs with expectation propagation. In *Proceedings of of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP '12)*, pages 1146–1156, 2012.

[53] A. Herskovits. *Language and Spatial Cognition: An Interdisciplinary Study of the Prepositions in English*. Cambridge University Press, 1986.

[54] J. Hois and O. Kutz. Counterparts in language and space: Similarity and [Sscr]-connection. In *Proceedings of the 2008 Conference on Formal Ontology in Information Systems: Proceedings of the Fifth International Conference (FOIS)*, pages 266–279, 2008.

[55] J. Hois and O. Kutz. Natural language meets spatial calculi. In C. Freksa, N. S. Newcombe, P. Gärdenfors, and S. Wölfl, editors, *Spatial Cognition VI. Learning, Reasoning, and Talking about Space*, volume 5248 of *LNCS*, pages 266–282. Springer, 2008.

[56] J. Hois and O. Kutz. Towards linguistically-grounded spatial logics. In J. A. Bateman, A. G. Cohn, and J.s Pustejovsky, editors, *Spatial Representation and Reasoning in Language: Ontologies and Logics of Space*, number 10131 in Dagstuhl Seminar Proceedings. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2011.

[57] Blake Howald and E. Katz. On the explicit and implicit spatiotemporal architecture of narratives of personal experience. In Max Egenhofer, Nicholas Giudice, Reinhard Moratz, and Michael Worboys, editors, *Spatial Information Theory*, volume 6899 of *Lecture Notes in Computer Science*, pages 434–454. Springer Berlin / Heidelberg, 2011.

[58] Fei Huang, Arun Ahuja, Doug Downey, Yi Yang, Yuhong Guo, and Alexander Yates. Learning Representations for Weakly Supervised Natural Language Processing Tasks. *Computational Linguistics*, 2013.

[59] Liang Huang, Suphan Fayong, and Yang Guo. Structured perceptron with inexact search. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '12, pages 142–151. ACL, 2012.

[60] W. Jiang, E. Zavesky, S. F. Chang, and A. C. Loui. Cross-domain learning methods for high-level visual concept classification. In *15th IEEE International Conference on Image Processing (ICIP)*, pages 161–164, 2008.

[61] R. Johansson and P. Nugues. LTH: semantic structure extraction using nonprojective dependency trees. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 227–230, 2007.

[62] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition.* Prentice Hall, second edition, 2008.

[63] J. D. Kelleher. *A Perceptually Based Computational Framework for the Interpretation of Spatial Language.* PhD thesis, School of Computing Dublin City University, 2003.

[64] J. D. Kelleher and F. J. Costello. Applying computational models of spatial prepositions to visually situated dialog. *Computational Linguistics*, 35(2):271–306, 2009.

[65] Alexander Klippel and Rui Li. The endpoint hypothesis: a topological-cognitive assessment of geographic scale movement patterns. In *Spatial Information Theory, COSIT'09*, pages 177–194, 2009.

[66] T. Kollar, S. Tellex, D. Roy, and N. Roy. Toward understanding natural language directions. In *Proceeding of the 5th ACM/IEEE International*

Conference on Human-Robot Interaction, HRI '10, pages 259–266. ACM, 2010.

[67] Oleksandr Kolomiyets, Parisa Kordjamshidi, Marie-Francine Moens, and Steven Bethard. Semeval-2013 task 3: Spatial role labeling. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 255–262, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[68] Oleksandr Kolomiyets and Marie Francine Moens. Machine learning approaches for temporal information extraction: a comparative study. In *Proceedings AST 2009: Applications of Semantic Technologies (Lecture Notes of Informatics),*, pages 3150–3161, 2009.

[69] P. Kordjamshidi, S. Bethard, and M. F. Moens. SemEval-2012 task 3: Spatial role labeling. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval)*, volume 2, pages 365–373. ACL, 2012.

[70] P. Kordjamshidi, J. Hois, M. van Otterlo, and M. F. Moens. Machine learning for interpretation of spatial natural language in terms of QSR. pages 1–5, 2011.

[71] P. Kordjamshidi, J. Hois, M. van Otterlo, and M. F. Moens. Learning to interpret spatial natural language in terms of qualitative spatial relations. Oxford University Press, 2013.

[72] P. Kordjamshidi, M. van Otterlo, and M. F. Moens. From language towards formal spatial calculi. In Robert J. Ross, Joana Hois, and John Kelleher, editors, *Workshop on Computational Models of Spatial Language Interpretation (CoSLI'10, at Spatial Cognition)*, pages 17–24, 2010.

[73] P. Kordjamshidi, M. van Otterlo, and M. F. Moens. Spatial role labeling: task definition and annotation scheme. In Nicoletta Calzolari, Choukri Khalid, and Maegaard Bente, editors, *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*, pages 413–420, 2010.

[74] P. Kordjamshidi, M. van Otterlo, and M. F. Moens. Spatial role labeling: towards extraction of spatial relations from natural language. *ACM - Transactions on Speech and Language Processing*, 8:1–36, 2011.

[75] Stefan Kramer, Nada Lavrač, and Peter Flach. Relational Data Mining. chapter Propositionalization approaches to relational data mining, pages 262–286. Springer-Verlag New York, Inc., 2000.

[76] Frank R. Kschischang, Brendan J. Frey, and Hans Andrea Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 1998.

[77] Kow Kuroda, Masao Utiyama, and Hitoshi Isahara. Getting deeper semantics than Berkeley FrameNet with MSFA. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, 2006.

[78] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, ICML '01, pages 282–289, 2001.

[79] C. H. Lampert. Maximum margin multi-label structured prediction. In J. Shawe-Taylor, R. S. Zemel, P. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 289–297. 2011.

[80] Jochen L Leidner. *Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding of Place Names.* PhD thesis, School of Informatics, University of Edinburgh, 2008.

[81] S. C. Levinson. *Space in Language and Cognition: Explorations in Cognitive Diversity.* Cambridge University Press, 2003.

[82] M. Levit and D. Roy. Interpretation of spatial language in a map navigation task. *IEEE Transactions on Systems, Man and Cybernetics*, 37(3):667–679, 2006.

[83] H. Li, T. Zhao, S. Li, and Y. Han. The extraction of spatial relationships from text based on hybrid method. *International Conference on Information Acquisition*, pages 284–289, 2006.

[84] H. Li, T. Zhao, S. Li, and J. Zhao. The extraction of trajectories from real texts based on linear classification. In *Proceedings of the NODALIDA 2007 Conference*, pages 121–127, 2007.

[85] K. Litkowski and O. Hargraves. SemEval-2007 Task 06: Word-sense disambiguation of prepositions. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 24–29. ACL, 2007.

[86] K. Lockwood, K. Forbus, D. T. Halstead, and J. Usher. Automatic categorization of spatial prepositions. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society. Stressa*, pages 1705–1710, 2006.

[87] K. Lockwood, A. Lovett, and K. Forbus. Automatic classification of containment and support spatial relations in English and Dutch. In C. Freksa, N. S. Newcombe, P. Gärdenfors, and S. Wölfl, editors, *Spatial*

Cognition VI: Learning, Reasoning and Talking about Space, number 5241 in LNCS, pages 283–294. Springer, 2008.

[88] I. Mani. SpatialML: annotation scheme for marking spatial expression in natural language. Technical Report Version 3.0, The MITRE Corporation, 2009.

[89] I. Mani, J. Hitzeman, J. Richer, D. Harris, R. Quimby, and B. Wellner. SpatialML: annotation scheme, corpora, and tools. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, and D. Tapias, editors, *Proceedings of the Sixth International Language Resources and Evaluation (LREC)*. European Language Resources Association (ELRA), 2008.

[90] L. Màrquez, X. Carreras, K. C. Litkowski, and S. Stevenson. Semantic role labeling: an introduction to the special issue. *Computational Linguistics*, 34(2):145–159, 2008.

[91] Andre Filipe Torres Martins. *The geometry of constrained structured prediction: applications to inference and learning of natural language syntax.* PhD thesis, Universidade Tecnica de Lisboa (Instituto Superior Tecnico) and Carnegie Mellon University, 2012.

[92] A. McCallum, K. Schultz, and S. Singh. FACTORIE: probabilistic programming via imperatively defined factor graphs. In *Advances in Neural Information Processing Systems 22, NIPS*, pages 1249–1257, 2009.

[93] N. McIntyre and M. Lapata. Learning to tell tales: a data-driven approach to story generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 217–225. ACL, 2009.

[94] Avihai Mejer and Koby Crammer. Confidence in structured-prediction using confidence-weighted models. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 971–981. ACL, 2010.

[95] Ofer Meshi, David Sontag, Tommi Jaakkola, and Amir Globerson. Learning efficiently with approximate inference via dual losses. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 783–790. Omnipress, 2010.

[96] Ivan Meza-Ruiz and Sebastian Riedel. Jointly identifying predicates, arguments and senses using Markov logic. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, pages 155–163. ACL, 2009.

[97] Tom Mitchell. *Machine Learning.* McGraw Hill, 1997.

[98] Marie Francine Moens. *Information Extraction: Algorithms and Prospects in a Retrieval Context.* The Information Retrieval Series 21. New York: Springer, 2006.

[99] R. J. Mooney. Learning to connect language and perception. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 1598–1601, 2008.

[100] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[101] Jennifer Neville and David Jensen. Relational dependency networks. *Journal of Machine Learning Research*, 8:653–692, 2007.

[102] Joakim Nivre. *Inductive Dependency Parsing.* Springer-Verlag, New York, 2006.

[103] S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3-4):185–365, 2011.

[104] T. O'Hara and J. Wiebe. Exploiting semantic role resources for preposition disambiguation. *Computational Linguistics*, 35(2):151–184, 2009.

[105] B. Onyshkevych and S. Nirenburg. Lexicon, ontology, and text meaning. In J. Pustejovsky and S. Bergler, editors, *Lexical Semantics and Knowledge Representation*, pages 289–303. Springer-Verlag, Berlin, 1992.

[106] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: an annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

[107] G. T. Papadopoulos, V. Mezaris, S. Dasiopoulou, and I. Kompatsiaris. Semantic image analysis using a learning approach and spatial context. In Y. S. Avrithis, Y. Kompatsiaris, S. Staab, and N. E. O'Connor, editors, *Semantic Multimedia, First International Conference on Semantics and Digital Media Technologies, SAMT*, volume 4306 of *LNCS*, pages 199–211. Springer, 2006.

[108] Georgios Petasis, Vangelis Karkaletsis, Georgios Paliouras, Anastasia Krithara, and Elias Zavitsanos. Knowledge-driven multimedia information extraction and ontology evolution. chapter Ontology population and enrichment: state of the art, pages 134–166. Springer-Verlag, Berlin/Heidelberg, 2011.

[109] Hoifung Poon and Pedro Domingos. Joint inference in information extraction. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, volume 1 of *AAAI'07*, 2007.

[110] Vasin Punyakanok, Dan Roth, Wen Tau Yih, and Dav Zimak. Semantic role labeling via integer linear programming inference. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, pages 1346–1352, 2004.

[111] Vasin Punyakanok, Dan Roth, Wen Tau Yih, and Dav Zimak. Learning and inference over constrained output. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 1124–1129. Morgan Kaufmann Publishers Inc., 2005.

[112] J. Pustejovsky and J. L. Moszkowicz. Integrating motion predicate classes with spatial and temporal annotations. In Donia Scott and Hans Uszkoreit, editors, *COLING 2008: Companion volume D, Posters and Demonstrations*, pages 95–98, 2008.

[113] J. Pustejovsky and J. L. Moszkowicz. The role of model testing in standards development: The case of iso-space. In *Proceedings of LREC'12*, pages 3060–3063. European Language Resources Association (ELRA), 2012.

[114] Xipeng Qiu, Wenjun Gao, and Xuanjing Huang. Hierarchical multi-class text categorization with global margin maximization. In *Proceedings of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing*, pages 165–168, 2009.

[115] David A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In *Proceedings of the 3rd International Conferance on the Principles of Knowledge Representation and Reasoning, KR'92*, pages 165–176, 1992.

[116] T. Regier and M. Zheng. Attention to endpoints: a cross-linguistic constraint on spatial meaning. *Cognitive Science*, 31(4):705–719, 2007.

[117] M. L. Reinbergerr. Automatic extraction of spatial relations. In *Proceedings of the Portuguese Conference on Artificial Intelligence*, pages 331–337, 2005.

[118] J. Renz and B. Nebel. Qualitative spatial reasoning using constraint calculi. In M. Aiello, I. Pratt-Hartmann, and J. van Benthem, editors, *Handbook of Spatial Logics*, pages 161–215. Springer, 2007.

[119] Jochen Renz, Reinhold Rauh, and Markus Knauff. Towards cognitive adequacy of topological spatial relations. In *Spatial Cognition II*, pages 184–197, 2000.

[120] N. Rizzolo and D. Roth. Learning based Java for rapid development of NLP systems. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation*, 2010.

[121] K. Roberts and S.M. Harabagiu. UTD-SpRL: a joint approach to spatial role labeling. In **SEM 2012: The First Joint Conference on*

Lexical and Computational Semantics, Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval'12), pages 419–424, 2012.

[122] R. Ross, H. Shi, T. Vierhuff, B. Krieg-Brückner, and J. Bateman. Towards dialogue based shared control of navigating robots. In C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, and T. Barkowsky, editors, *Proceedings of Spatial Cognition IV: Reasoning, Action, Interaction*, pages 478–499. Springer, Berlin/ Heidelberg, 2005.

[123] D. Roth and K. Small. Active learning for pipeline models. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 683–688, 2008.

[124] D. Roth and W. Yih. A linear programming formulation for global inference in natural language tasks. In Hwee Tou Ng and Ellen Riloff, editors, *Proceedings of the 2004 Conference on Computational Natural Language Learning (CoNLL)*, pages 1–8. ACL, 2004.

[125] Dan Roth and Wen Tau Yih. Integer linear programming inference for conditional random fields. In *Proceedings of the International Conference on Machine Learning*, pages 736–743, 2005.

[126] Dan Roth and Wen tau Yih. Global inference for entity and relation identification via a linear programming formulation. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning.* MIT Press, 2007.

[127] J. M. Ruiz-Martìnez, J. A. Minarro-Gimènez, D. Castellanos-Nieves, F. Garcìa-Sànchez, and R. Valencia-Garcia. Ontology population: an application for the E-tourism domain. *International Journal of Innovative Computing, Information and Control (IJICIC)*, 7(11):6115–6134, 2011.

[128] Alexander M. Rush and Michael Collins. A tutorial on dual decomposition and Lagrangian relaxation for inference in natural language processing. *The Journal of Artificial Intelligence Research (JAIR)*, 45:305–362, 2012.

[129] Alexander M. Rush, David Sontag, Michael Collins, and Tommi Jaakkola. On dual decomposition and linear programming relaxations for natural language processing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, 2010.

[130] Rajhans Samdani and Dan Roth. Efficient decomposed learning for structured prediction. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[131] Taisuke Sato and Yoshitaka Kameya. PRISM: a language for symbolic-statistical modeling. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI7)*, pages 1330–1335, 1997.

[132] Frank Schilder, Yannick Versley, and Christopher Habel. Extracting spatial information: grounding, classifying and linking spatial expressions. In *Proceedings of the Workshop on Geographic Information Retrieval at SIGIR 2004*, 2004.

[133] Stephanie Schuldes, Michael Roth, Anette Frank, and Michael Strube. Creating an annotated corpus for generating walking directions. In *Proceeding of the ACL-IJCNLP 2009 Workshop: Language Generation and Summarization.*, pages 72–76, 2009.

[134] Q. Shen, X. Zhang, and W. Jiang. Annotation of spatial relations in natural language. In *Proceedings of the International Conference on Environmental Science and Information Application Technology*, volume 3, pages 418 – 421, 2009.

[135] H. Shi and T. Tenbrink. Telling Rolland where to go: HRI dialogues on route navigation. In K. Coventry, T. Tenbrink, and J. Bateman, editors, *Spatial Language and Dialogue*, pages 177–189. Oxford Univ. Press, 2009.

[136] David Sontag, Amir Globerson, and Tommi Jaakkola. Introduction to dual decomposition for inference. In Suvrit Sra, Sebastian Nowozin, and Stephen J. Wright, editors, *Optimization for Machine Learning.* MIT Press, 2011.

[137] David Sontag, Talya Meltzer, Amir Globerson, Tommi Jaakkola, and Yair Weiss. Tightening LP relaxations for MAP using message passing. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 503–510. AUAI Press, 2008.

[138] David Sontag, Ofer Meshi, Tommi Jaakkola, and Amir Globerson. More data means less inference: a Pseudo-Max approach to structured learning. In *Advances in Neural Information Processing Systems 23*, pages 2181–2189, 2010.

[139] M. Sridhar, A. G. Cohn, and D. C. Hogg. Learning functional object-categories from a relational spatio-temporal representation. In M. Ghallab, C. D. Spyropoulos, N. Fakotakis, and N. M. Avouris, editors, *18th European Conference on Artificial Intelligence*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 606–610. IOS Press, 2008.

[140] Vivek Srikumar and Dan Roth. A joint model for extended semantic role labeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 129–139. ACL, 2011.

[141] O. Stock, editor. *Spatial and Temporal Reasoning.* Kluwer Academic Publishers, 1997.

[142] C. Sutton and A. MacCallum. Introduction to conditional random fields for relational learning. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*, pages 108–143. MIT Press, 2006.

[143] Charles Sutton and Andrew McCallum. Collective segmentation and labeling of distant entities in information extraction. In *ICML Workshop on Statistical Relational Learning and Its Connections*, 2004.

[144] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012.

[145] Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. Dynamic conditional random fields: factorized probabilistic models for labeling and segmenting sequence data. *Journal of Machine Learning Research*, 8:693–723, 2007.

[146] L. Talmy. The fundamental system of spatial schemas in language. In B. Hampe, editor, *From Perception to Meaning: Image Schemas in Cognitive Linguistics*, pages 37–47. Mouton de Gruyter, 2006.

[147] D. A. Tappan. *Knowledge-Based Spatial Reasoning for Automated Scene Generation from Text Descriptions.* PhD thesis, New Mexico State Univ. Las Cruces, 2004.

[148] Ben Taskar, Pieter Abbeel, and Daphne Koller. Discriminative probabilistic models for relational data. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, UAI'02, pages 485–492. Morgan Kaufmann Publishers Inc., 2002.

[149] Ben Taskar, Ming Fai Wong, Pieter Abbeel, and Daphne Koller. Link prediction in relational data. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16.* MIT Press, 2004.

[150] Benjamin Taskar, Carlos Guestrin, and Daphne Koller. Max-margin Markov networks. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems.* MIT Press, 2004.

[151] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2011.

[152] T. Tenbrink and W. Kuhn. A model of spatial reference frames in language. In Max Egenhofer, Nicholas Giudice, Reinhard Moratz, and

Mike Worboys, editors, *Conference on Spatial Information Theory (COSIT'11)*, pages 371–390. Springer, 2011.

[153] S. Tratz and D. Hovy. Disambiguation of preposition sense using linguistically motivated features. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion*, volume Student Research Workshop and Doctoral Consortium of *NAACL '09*, pages 96–100. ACL, 2009.

[154] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109:475–494, 2001.

[155] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(2):1453–1484, 2006.

[156] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag New York, Inc., 1995.

[157] A. Vedaldi. A MATLAB wrapper of SVM$^{\text{struct}}$. `http://www.vlfeat.org/~vedaldi/code/svm-struct-matlab.html`, 2011.

[158] M. Verbeke, Vincent Van Asch, Roser Morante, P. Frasconi, Walter Daelemans, and Luc De Raedt. A statistical relational learning approach to identifying evidence based medicine categories. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 579–589. ACL, 2012.

[159] Mathias Verbeke, Paolo Frasconi, Vincent Van Asch, Roser Morante, Walter Daelemans, and Luc De Raedt. Kernel-based logical and relational learning with kLog for hedge cue detection. In Stephen H. Muggleton, Alireza Tamaddoni Nezhad, and Francesca A. Lisi, editors, *Inductive Logic Programming*, volume 7207 of *Lecture Notes in Computer Science*, pages 347–357. Springer Berlin/Heidelberg, 2012.

[160] Laure Vieu. Spatial representation and reasoning in artificial intelligence. In Oliviero Stock, editor, *Spatial and Temporal Reasoning*, pages 5–41. Springer Netherlands, 1997.

[161] A. Vogel and D. Jurafsky. Learning to follow navigational directions. In *ACL '10: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 806–814. ACL, 2010.

[162] G. Wachman and R. Khardon. Learning from interpretations: a rooted kernel for ordered hypergraphs. In *Proceedings of the 24th International Conference on Machine Learning ICML*, volume 227, pages 943–950, 2007.

[163] J. Wallgrün, L. Frommberger, D. Wolter, F. Dylla, and C. Freksa. Qualitative spatial representation and reasoning in the SparQ-Toolbox. In Thomas Barkowsky, Markus Knauff, Gérard Ligozat, and Daniel R. Montello, editors, *Spatial Cognition V Reasoning, Action, Interaction*, volume 4387, chapter 3, pages 39–58. Springer Berlin/Heidelberg, 2007.

[164] M. P. Wellman, J. S. Breese, and R. P. Goldman. From knowledge bases to decision models. *Knowledge Engineering Review*, 7(1):35–53, 1992.

[165] S. Wiebrock, L. Wittenburg, U. Schmid, and F. Wysotzki. Inference and visualization of spatial relations. In C. Freksa, C. Habel, W. Brauer, and K. Wender, editors, *Spatial Cognition II, Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications*, volume 1849, pages 212–224. Springer Berlin / Heidelberg, 2000.

[166] Wilson Wong, Wei Liu, and Mohammed Bennamoun. Ontology learning from text: a look back and into the future. *ACM Computing Surveys*, 44(4):20:1–20:36, 2012.

[167] J. Zlatev. Holistic spatial semantics of Thai. *Cognitive Linguistics and Non-Indo-European Languages*, pages 305–336, 2003.

[168] J. Zlatev. Spatial semantics. In D. Geeraerts and H. Cuyckens, editors, *The Oxford Handbook of Cognitive Linguistics*, pages 318–350. Oxford Univ. Press, 2007.

Curriculum Vitae

Parisa Kordjamshidi was born on September 22, 1977 in Mehran, Iran. She received her diploma in Mathematics and Physics from Nasibeh high school, Ilam, in June 1994. In the same year, she entered Isfahan university, Iran, and followed a bachelor program in Software Engineering and graduated in January 1999. She started her Master in Software Engineering in 2000 at Tarbiat Modares university, Tehran, Iran and graduated in Dec 2002. Afterwards she was lecturing for computer science in Shahid Chamran university and at 2004 she became a faculty member at Ilam university in Iran. In 2007, she moved to Belgium and started her PhD research voluntarily in Ghent university, working on fuzzy approaches for machine learning and uncertainty modeling. In September 2008, she obtained a PhD grant on a different subject from a joint project of DTAI and LIIR research groups. During her PhD, she has explored the structured and relational learning models, natural language processing issues and spatial information representation. She will defend her thesis on " Structured Machine Learning for Mapping Natural Language to Spatial Ontologies", with supervisory of Marie-Francine Moens in July 2013 at Katholieke Universiteit Leuven.

Publication List

Journal Articles

Kordjamshidi, P., van Otterlo, M., Moens, M. F. (2011). Spatial role labeling: Towards extraction of spatial relations from natural language. *ACM Transactions on Speech and Language Processing, 8 (3), article 4, 36 p.*

Kordjamshidi, P., Moens, M. F. (2013). Structured machine learning for spatial ontology population. *Special Issue of the Elsevier Journal of Web Semantics on Semantic Search* (submitted).

Book Chapters

Kordjamshidi, P., Hois, J., van Otterlo, M., Moens, M. F. (2013). Learning to interpret spatial natural language in terms of qualitative spatial relations. In: Tenbrink T. (Ed.), Wiener J. (Ed.), Claramunt C. (Ed.), *Representing Space in Cognition: Interrelations of Behavior, Language, and Formal Models* Oxford University Press.

Conference/Workshop full Papers

Kordjamshidi, P., van Otterlo, M., Moens, M. F. (2010). Spatial role labeling: Task definition and annotation scheme. In Calzolari, N. (Ed.), Khalid, C. (Ed.), Bente, M. (Ed.), *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10).* Malta, 19-21 May 2010 (pp. 413-420) European Language Resources Association (ELRA).

Kordjamshidi, P., van Otterlo, M., Moens, M. F. (2010). From language towards formal spatial calculi. In Ross, R. (Ed.), Hois, J. (Ed.), Kelleher,

J. (Ed.), Proc. of 1st Workshop COSLI'10. *Computational Models of Spatial Language Interpretation (COSLI).* Mt.Hood/Portland, OR, USA, 15-August 2010 (pp. 17-24).

Kordjamshidi, P., Frasconi, P., van Otterlo, M., Moens, M. F., De Raedt, L. (2011). Spatial relation extraction using relational learning. *Latest Advances in Inductive Logic Programming. ILP.* Windsor Great Park, United Kingdom, 31 July-3rd August (pp. 1-6).

Kordjamshidi, P., Frasconi, P., van Otterlo, M., Moens, M. F., De Raedt, L. (2012). Relational learning for spatial relation extraction from natural language. In Muggleton, S. H. (Ed.), Tamaddoni-Nezhad, A. (Ed.), Lisi, F. (Ed.), *Inductive Logic Programming: LNCS, Vol. 7207. ILP. Windsor, 31st July-3rd August (pp. 204-220).* Berlin Heidelberg: Springer.

Kordjamshidi, P., Bethard, S., Moens, M. F. (2012). SemEval-2012 Task 3: Spatial role labeling. **SEM 2012: Proceedings of the First Joint Conference on Lexical and Computational Semantics – Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation* (SemEval 2012), pp. 365-373, Montreal, Canada, ACL.

Kolomiyets, O., Kordjamshidi, P., Bethard, S., Moens, M.F. (2013). SemEval-2013 Task 3: Spatial Role Labeling. **SEM 2013: Proceedings of the Second Joint Conference on Lexical and Computational Semantics – Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation* (SemEval 2013), pp. 255-262, Atlanta, Georgia, USA, ACL.

Extended abstracts

Kordjamshidi, P., Moens, M.F. (2013). Structured machine learning for mapping natural language to spatial ontologies. *In Proceedings of the International Workshop on Advances in Regularization, Optimization, Kernel Methods and Support Vector Machines.* (ROKS 2013).

Kordjamshidi, P., Moens, M. F. (2012). Spatial role labeling using structured support vector machines. *Proceedings of 21st Belgian-Dutch conference on machine learning (BeneLearn): vol. 21.* Ghent, 24-25 May, pp. 71.

Kordjamshidi, P., Hois, J., van Otterlo, M., Moens, M. F. (2011). Machine learning for interpretation of spatial natural language in terms of QSR. *International Conference on Spatial Information Theory (COSIT1). Extended abstract, Vol. Spatial Information Theory (technical report). COSIT.* Belfast, 12-16 September (pp. 1-5).

O thou who art above all imaginations, conjectures, opinions and ideas,
Above anything people have said or we have heard or read,
The assembly is finished and life has reached its term
And we have, as at first, remained powerless in describing thee.
(Saadi, 13th centry)