KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science

Representation Learning for Weakly-Supervised Natural Language Processing Tasks

Geert Heyman

Supervisors:
Prof. dr. Marie-Francine Moens
Dr. Ivan Vulić

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Computer Science

December 2018

Representation Learning for Weakly-Supervised Natural Language Processing Tasks

Geert HEYMAN

Examination committee:
Prof. dr. Adhemar Bultheel, chair
Prof. dr. Marie-Francine Moens, supervisor
Dr. Ivan Vulić, supervisor
Prof. dr. ir. Hendrik Blockeel
Prof. dr. ir. Patrick Wambacq
Dr. Vincent Vandeghinste
Prof. dr. Eneko Agirre
(University of the Basque Country)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Computer Science

December 2018

Uitgegeven in eigen beheer, Geert Heyman, Celestijnenlaan 200A box 2402, B-3001 Leuven (Belgium)

Acknowledgements

Looking back on the past years, my journey as a PhD student has been exciting but also very challenging. This thesis would have never been possible without the support of many different people.

First of all, I would like to thank my promotor Prof. dr. Marie-Francine Moens for granting me the opportunity to pursue a PhD, giving me the freedom to learn new things and investigate my ideas while offering me advice and encouragement. Sien, over the course of my PhD I have never known a period were you did not have a very busy schedule, nevertheless you always made time to give me advice, feedback and encouragement. Many thanks.

Also a very heartfelt thanks to my co-promotor, dr. Ivan Vulić. Ivan has also always been very generous with his advice and support. Furthermore, he was quick to point out new interesting publications that could help me with my research and was a big help by showing me how to write convincing scientific papers. Hvala ti, Ivane!

I would also like to thank Prof. dr. Adhemar Bultheel, Prof. dr. ir. Hendrik Blockeel, Prof. dr. ir. Patrick Wambacq, dr. Vincent Vandeghinste, and Prof. dr. Eneko Agirre for accepting to be part of my PhD examination committee. Their careful reviews and feedback helped me to improve this dissertation.

The majority of my research was carried out in the context of the SCATE project. I would like to thank all partners for the fruitful collaboration, especially my fellow researchers on work package 3: Ayla Rigouts, Els Lefever, and Iulianna van der Lek-Ciudin. On a similar note, I would like to thank 4C and the other partners of the PAPUD project, particularly Veerle Liébaut, Evelyn Caris, and Stijn Jacques. I also thank IWT and ITEA3 for funding the SCATE and PAPUD projects respectively.

I was also fortunate to be surrounded by great labmates: Aparna, Bregt, Doaa, Elias, Katrien, Golnoosh, Graham, Guillem, Jan, Juan Carlos, Niraj, Ruiqi, Shurong, Sumam, Susana, Thomas, Tuur, and Quynh. Thank you *Liirians* for the brainstorms, collaborations, and, most importantly, the lunches at Alma. A big thank you to Quynh

and Tuur who were very supportive during the final stage of my PhD. Also thank you Tuur for letting me ride along with the *TuurExpress* many times and to endure my silly jokes about the Netherlands!

A PhD can be quite stressful at times, I thank everyone who provided me with *fun distractions*. Thank you Jasper, Tim, An, Vincent, Jessa, Chris, Wim, Liirians, chess friends, and football players in the department for the dinners, pancakes, chess games, runs, football matches, and friendship.

Finally, a very big thank you to my family, especially my parents and brother. Thank you for your unconditional and invaluable support!

Abstract

In recent years, representation learning has obtained impressive results across a wide range of machine learning tasks in domains such as natural language processing, computer vision and speech recognition. Rather than relying on hand-crafted data representations, representation learning aims to acquire representations automatically from data. Its successes have been achieved on problems with a large amount of annotated data – datasets that comprise millions of training examples are no exception. For many important problems, there is no abundance of labeled data, however. This is particularly the case within the domain of natural language processing where, especially for languages other than English, such large-scale datasets are often lacking.

In this thesis, we investigate and propose representation learning models in settings where the amount of annotated training data is limited. This thesis has four main contributions which each shed a different light on how representation learning can be used in weakly-supervised settings.

First, we design a new cross-lingual probabilistic topic model that can infer cross-lingual representations for words and documents after being trained on a collection of document pairs that are similar, but not necessarily identical in content. With this, we provide a means to obtain cross-lingual representations for words and documents that are interpretable and valuable features for tasks such as cross-lingual document classification, and this without the need for parallel data.

Second, we design methods to construct multilingual embedding spaces without using bilingual dictionaries, parallel corpora or any other type of multilingual supervision and study the effectiveness of these spaces on down-stream natural language processing tasks (i.e., bilingual lexicon induction, multilingual document classification, and multilingual dependency parsing). In contrast to previous research, our most effective method combines the following desirable properties: the method incorporates dependencies between all targeted languages; the method still works when targeting languages with very different characteristics (e.g., projecting English in the same vector space as Finnish and/or Hungarian); and empirical evidence indicates that the method

is stable as it never produced degenerate solutions in our experiments.

Third, we propose a deep learning model to tackle the correction of context-dependent dt-errors, one of the most prominent spelling errors in Dutch, without using labeled examples of dt-mistakes. The model is designed to predict the correct suffixes of verbs given their stem and the context in which they occur. Hence the data requirements are limited to high-quality Dutch text, which is available in abundance. In comparative tests with other systems including the spell checker that comes with Microsoft Word, the proposed model obtains the best results by a large margin.

Fourth, we present a new approach for obtaining bilingual dictionaries that combines character-level and word-level information to extract translations from non-parallel texts. Different from the majority of prior work, we frame this task as a classification problem rather than as a retrieval problem. This enables combining unsupervised and weakly-supervised representation learning techniques to seamlessly integrate word-level and character-level information. In particular, from a set of seed translations, the model learns character-level representations rather than relying on hand-crafted feature extraction techniques and learns how to fuse it with word-level representations that encode corpus statistics. The major findings are a) that the incorporation of character-level information is particularly useful in the biomedical domain, where many terms have their origin in Greek and Latin or are acronyms or abbreviations, and b) that learning character-level representations is superior to the hand-crafted representations which were used in prior work. Although we evaluate primarily on biomedical terms, the method is domain-agnostic and holds promise to support translation mining in other domains.

The main conclusion of this dissertation is that representation learning is very much applicable to weakly-supervised natural language processing problems both as a means to inject data-driven prior knowledge into tasks by inducing textual input representations from unlabeled texts and as a paradigm for obtaining abstractions from labeled text data that are not uncovered with classical feature engineering.

Beknopte samenvatting

In de afgelopen jaren heeft het leren van representaties tot indrukwekkende resultaten geleid voor een breed gamma van taken in de domeinen van natuurlijke taalverwerking, computervisie en spraakherkenning. In plaats van te steunen op manueel gebouwde representaties, stelt het leren van representaties als doel om automatisch representaties te extraheren uit data. De successen van het leren van representaties zijn behaald op problemen waarvoor een grote hoeveelheid geannoteerde data beschikbaar is - datasets die bestaan uit miljoenen voorbeelden zijn geen uitzondering. Voor veel belangrijke problemen is er echter geen overvloed aan gelabelde data. Voor natuurlijke taalverwerking in het bijzonder zijn zulke datasets vaak niet beschikbaar, vooral wanneer het andere talen dan Engels betreft.

In deze thesis onderzoeken en bouwen we modellen die representaties leren in omgevingen waar de hoeveelheid geannoteerde training data beperkt is. Deze thesis heeft vier belangrijke contributies die elk een nieuw licht werpen op hoe het leren van representaties gebruikt kan worden in zwak gesuperviseerde omgevingen.

Ten eerste ontwikkelen we een nieuw cross-linguaal probabilistisch topic model dat cross-linguale representaties voor woorden en documenten kan infereren na getraind te zijn op een collectie van documentparen waarvan de onderwerpen gelijkaardig, maar niet noodzakelijk identiek zijn. Hiermee bieden we een middel aan om cross-linguale representaties voor woorden en documenten te verkrijgen die interpreteerbaar zijn en waardevolle invoervoorstellingen bieden voor taken zoals cross-linguale document classificatie, en dit alles zonder parallelle training data nodig te hebben.

Ten tweede ontwerpen we methodes om multilinguale vector ruimtes te construeren zonder bilinguale woordenboeken, parallelle corpora of andere vormen van multilinguale supervisie te gebruiken en bestuderen we de effectiviteit van deze ruimtes op relevante natuurlijke taalverwerking taken (i.e., bilinguale lexicon inductie, multilinguale document classificatie en het multilinguaal detecteren van syntactische afhankelijkheden). In tegenstelling tot eerder onderzoek, combineert onze meest effectieve methode de volgende voordelen: de methode incorporeert afhankelijkheden

tussen alle gemodelleerde talen; de methode werkt nog altijd wanneer we talen beschouwen met karakteristieken die onderling zeer verschillen (bijv. Engels projecteren in dezelfde ruimte als Fins en/of Hongaars); en empirische resultaten tonen aan dat de methode stabiel is omdat er nooit gedegenereerde oplossingen geproduceerd werden.

Ten derde stellen we een model voor dat diepe representaties leert om dt-fouten, een belangrijk spellingsprobleem in de Nederlandse taal, automatisch te detecteren en te corrigeren zonder gebruik te maken van gelabelde voorbeelden. Het model is ontworpen om de correcte uitgangen van werkwoorden te voorspellen gegeven hun stam en de context waarin ze voorkomen. Bijgevolg worden de datavereisten gereduceerd tot goedgeschreven Nederlandse teksten, dewelke in grote hoeveelheden beschikbaar zijn. In een vergelijkende studie met andere systemen inclusief de spell-checker van Microsoft Word behaalt ons model met voorsprong de beste resultaten.

Ten vierde stellen we een nieuwe aanpak voor om bilinguale woordenboeken te induceren die informatie op karakter- en woordniveau combineert om vertalingen te extraheren uit teksten die niet parallel zijn. In tegenstelling tot de meeste voorgaande onderzoeken kaderen we deze taak als een classificatieprobleem i.p.v. een ophaalprobleem. Dit maakt het mogelijk om het leren van ongesuperviseerde en zwak gesuperviseerde voorstellingen met elkaar te combineren om woord- en karakterniveau representaties naadloos met elkaar te integreren. Meer specifiek leert het model uit een verzameling van initieel gekende vertalingen hoe woordparen op karakterniveau kunnen worden voorgesteld i.p.v. zulke voorstelling manueel te bepalen en leert het hoe deze karaktervoorstelling gecombineerd kan worden met woordniveau representaties die de corpusstatistieken van een woord samenvatten. De hoofdbevindingen zijn dat a) het incorporeren van karakterniveau representaties zeer bruikbaar is in het biomedische domein, waar vele termen hun oorsprong vinden in het Grieks of Latijn ofwel acroniemen of afkortingen zijn, en b) dat het leren van karakterniveau representaties superieur is aan de manueel gedefinieerde voorstellingen die gebruikt werden in eerder onderzoek.

De algemene conclusie van deze dissertatie is dat het leren van representaties ook een geschikte techniek is voor zwak gesuperviseerde natuurlijke taalverwerkingsproblemen, enerzijds als een manier om voorkennis uit tekstuele data te halen en anderzijds als paradigma om abstracties te ontdekken uit gelabelde data die niet uitgedrukt kunnen worden met klassieke, manueel gedefinieerde representaties.

List of Abbreviations

BiLDA bilingual LDA. 34

BLI bilingual lexicon induction. 5

BWE bilingual word embedding. 58

BWESG bilingual word embedding skip-gram. 28

C-BiLDA comparable bilingual LDA. 34

CL-KCCA cross-lingual kernel canonical correlation analysis. 35

CL-LSI cross-lingual latent semantic indexing. 35

CLDC cross-lingual document classification. 46

ED edit distance. 117

IHS Incremental Hub Space. 66

LDA latent Dirichlet allocation. 26, 32

LSTM long short-term memory. 24

MAP maximum a posteriori. 17

ML maximum likelihood. 17

MWE multilingual word embedding. 58

NLP natural language processing. 1

pLSA probabilistic latent semantic analysis. 32

PTM probabilistic topic model. 32

RL representation learning. 2

RNN recurrent neural network. 23

SGD stochastic gradient descent. 18

SGNS continuous skip-gram with negative sampling. 28

SHS Single Hub Space. 64

SVM support vector machine. 50

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural language processing (NLP) is a subdomain of artificial intelligence that researches how computers can automatically process and analyze natural language. NLP is engaged in building models that automatically classify texts (e.g., sentences or documents), analyze the syntactic structure of texts (e.g., part-of-speech tagging, constituency parsing, or dependency parsing), translate lexicons or documents (e.g., bilingual lexicon induction, machine translation), detect and correct spelling mistakes, etc. In the early days, the majority of NLP systems were rule-based, meaning that language experts were hand-coding rules to analyze text. The past few decades, however, NLP models have become predominantly statistical. The research focus shifted from finding appropriate rules or grammars to proposing algorithms that can *learn* how to process text from experience. The field that studies such learning algorithms is known as **machine learning**. Experience often comes in the form of examples of the desired output of the system (e.g., a category label) that corresponds to a given input (e.g., a sentence).

In settings with an abundance of such training examples, natural language processing, and related fields such as speech recognition and computer vision, have obtained impressive results with machine learning models. Unfortunately, training examples are not readily available for many important problems and creating them is typically a labor-intensive process where human annotators manually label example inputs with the desired system output. Especially for non-English languages annotated training data is hard to find. Therefore, there is a pressing need to find NLP models/algorithms that operate well in **weakly-supervised** settings, i.e., with a small number of training examples.

Any model aimed at computing a set of outputs from a set of input variables requires some mathematical representation of the inputs, typically using one or more vectors

that represent the most important properties of the input. The performance of machine learning models very much depends on the quality of the input representation. In classical machine learning such feature vectors are hand-crafted. That is, a human expert defines which properties are most relevant for computing/predicting the outputs and writes a deterministic program from extracting these features from the raw data. For many important problems proposing features that lead to optimal classification performance is a non-trivial task.

Representation learning (RL) is a branch of machine learning aimed at *learning* representations from data. Its key idea is to find a function that maps the raw inputs to a *meaningful* representation. This function is automatically selected from a predefined function family (e.g., a neural network) based on a criterion in terms of the training data (e.g., the function should map inputs to representations that maximally explain the variability in the training data). The training criteria is hence a formal specification of what representations are considered meaningful.

The representations, or equivalently the function that computes them, can either be learned directly on the training examples for the task at hand, which is known as supervised representation learning; or they can be learned from unlabeled data, known as unsupervised representation learning. In the former case, the representations are optimized jointly with the rest of the machine learning model. In the latter case, the representations are optimized to explain some unlabeled dataset, and in a second step used as input to the machine learning model that is trained on the task at hand. Whereas in feature engineering prior knowledge is mostly injected by hand-crafting an input representation, RL uses structural priors. That is, prior knowledge is brought into a model by specifying the family of functions that map raw inputs to meaningful representations and by expressing preferences to certain functions in the family (e.g., by putting prior distributions on function parameters). One such prior that has proven to be extremely useful for many practical problems is that functions for learning representations can be decomposed in many simple functions that are subsequently applied (e.g., $f(x) = f_1(f_2(f_3(x)))$). The number of subsequential functions in a model referred to as the *depth* of the RL model, and the set of techniques that uses this prior is known as **deep learning**.

When fuelled by large amounts of annotated training data, supervised representation learning and deep learning in particular have advanced the state of the art in various natural language processing problems. However, the success of RL in weakly-supervised settings has been limited. In this thesis, we aim to bridge this gap by exploring the use of representation learning techniques for natural language processing problems in four distinct scenarios with a limited amount of supervision.

In the remainder of this introductory chapter, we describe the context in which the research of this thesis was performed, we state the main goals of the thesis, and we lay out the structure of this text.

1.1 Research Context

The majority of this work has been executed in the context of research projects with a strong focus on multilingual techniques: the Smart Computer-Aided Translation Environment (SCATE) project (IWT-SBO 130047), which investigated translation technology and how it could improve the translator's workflow; and the Profiling and Analysis Platform Using Deep Learning (PAPUD) project (EU ITEA3 16037 and VLAIO HBC.2017.0498), which aims to develop a data analytics platform that can process text in multiple languages. Consequently, three of the four main contributions of this dissertation are situated in a multilingual setting.

Additional financial support from the ACCUMULATE (ACquiring CrUcial Medical information Using LAnguage TEchnology; IWT-SBO 150056) and MARS (MAchine Reading of patient recordS; C22/15/16) projects co-motivated the decision to evaluate the work on bilingual lexicon induction in the biomedical domain.

An inquiry from the *Instituut voor Levende Talen (ILT), KU Leuven* about the potential progress of NLP techniques in correcting dt-mistakes, a common spelling mistake in Dutch, was the catalyst for the fourth contribution.

1.2 Problem Statement

The primary goal of this dissertation is to study and propose representation learning techniques for natural language processing problems with a limited amount of supervision. To this end, and considering the afore-mentioned research context, we define four main research topics.

I) Learning bilingual representations for documents and words from comparable corpora The available amount of training data for a given NLP task is typically language-dependent. As data annotation is expensive, large-scale datasets are nearly exclusively found for common languages such as English. When designing a system aimed at processing text in a resource-poor language,[1] it would be desirable to be able to transfer knowledge from training data in resource-rich languages to the system in the resource-poor language. A promising strategy to that end is to learn bilingual text representations. These are representations that map texts (e.g., words or documents) to a bilingual space such that semantically related texts have similar representations, regardless of their language. For instance, in a bilingual space of English and Dutch documents, an English document about tennis should be more similar to a Dutch

[1] With the term *resource-poor languages* we refer to languages for which the amount of annotated training data is very limited.

document about cycling compared to an English document about politics. High-quality bilingual representations hence allow processing text in a resource-poor target language after training a classifier on annotated examples in a resource-rich source language.

To make bilingual representation learning practical, it is important that the bilingual representations can be easily obtained. Whereas most prior works focused on RL models that learn bilingual representations from parallel data (i.e., parallel corpora, which are a collection of translated sentence pairs; or bilingual dictionaries), we aim to learn representations from non-parallel corpora. In particular, the main research question for this topic is the following.

> RQ 1 Can a probabilistic topic model learn high-quality, interpretable bilingual representations from a collection of bilingual, subject-aligned document pairs without using any additional bilingual supervision?

II) Multilingual word representations from monolingual corpora In the second main research topic, we again investigate transfer learning potential of cross-lingual text representations, but we move to a problem setting that is more general than the one in topic I) in two respects. First, the goal is to research word representations from truly monolingual corpora. Removing the requirement for bilingual documents enables training representations on massive web crawl corpora. This implies that we have to design RL models that are very efficient to train. Second, instead of learning *bilingual* spaces that align two languages, we aim to learn *multilingual* spaces that align a variable number of languages, which enables the simultaneous knowledge transfer from multiple source languages. For this topic, we address two main research questions.

> RQ 2 Can we learn multilingual spaces without supervision of which the representations are valuable for cross-lingual transfer learning? Can we do this for sets of languages that have widely different characteristics?

> RQ 3 When constructing the multilingual representations, is it beneficial to incorporate dependencies between all languages by incrementally growing a multilingual space?

III) Dt-correction using representation learning For our third topic, we explore weakly-supervised RL for a verb spelling correction problem in Dutch that has no annotated training examples. We illustrate how the lack of annotated training examples can be bypassed by recasting the problem to verb suffix prediction, which only assumes correctly-written text for supervised training.

From an application point of view, the aim is to design a system that can accurately detect and correct context-dependent dt-errors, which are one of the most frequent

spelling errors in Dutch. Unfortunately, current systems such as Microsoft Word's spelling checker have a hard time detecting such verb errors because, by definition, inferring the correct spelling of such verbs requires an understanding of the sentence in which they occur. We pose the three research questions.

RQ 4 Can we design an accurate RL system that tackles verb spelling correction without annotated training examples?

RQ 5 What structural priors are important for predicting verb suffixes, a classification task that requires a syntactic understanding of sentences?

RQ 6 Is there a way to provide insight into the model predictions of the proposed RL model? In particular, is it possible to give feedback to the user when the system proposes spelling corrections?

IV) A representation learning framework for bilingual lexicon induction In the fourth research topic, we look at combining supervised and unsupervised RL in a single framework to tackle bilingual lexicon induction. Bilingual lexicon induction (BLI) is the task of inducing bilingual dictionaries from monolingual corpora. An interesting BLI use-case is the cross-lingual alignment of terminology as a crucial step in the automatic acquisition of bilingual terminology dictionaries (i.e., dictionaries which translators can rely on to correctly and consistently translate specialized texts). We therefore study BLI from domain-specific corpora in the (terminology-heavy) medical domain.

Two important characteristics of this setting are that 1) we are not only interested in translating the most frequent words, a setup that is typically maintained in BLI research, but want to induce translations across the entire frequency spectrum, and 2) many of the terms exhibit patterns on the morphology level that may facilitate identifying translations. We therefore aim to study how to integrate character-level representations into BLI models and investigate whether it is beneficial to take a full RL approach by learning the character-level representations, rather than using hand-crafted features used in prior work. More specifically, we pose the following research questions.

RQ 7 Can we use RL to integrate cross-lingual word-level and character-level representations?

RQ 8 Is it beneficial to learn cross-lingual character-level representations for weakly-supervised tasks instead of manually extracting these representations?

These four topics and their corresponding research questions all seek to investigate if and how representation learning can advance natural language processing in settings where the amount of annotated training data is limited. In sum, we study how to

automatically acquire text representations with weak supervision and/or without any supervision, and verify the quality of the representations on important NLP problems.

1.3 Outline

The remainder of this thesis is structured as follows. Chapter 2 provides an overview of the theoretical foundations on which the research in this thesis rests. It covers all concepts in linear algebra, probability theory, and representation learning that are required for a profound understanding of this thesis. Chapters 3-6 present the main contributions. Chapters 3 and 4 both study unsupervised representation learning techniques, whereas Chapter 5 applies representation learning in a supervised setting, and Chapter 6 presents a system that combines unsupervised and supervised RL techniques. More specifically, the outline of the thesis is as follows.

Chapter 3 introduces a new probabilistic topic model for learning cross-lingual representations for words and documents from a collection of bilingual, subject-aligned document pairs. The model relaxes the assumption made in prior research on bilingual topic models that there should be an exact match between the themes in paired documents. The effectiveness of the model and the representations it produces are tested by training on a collection of topic-aligned Wikipedia documents and two cross-lingual document classification datasets.

Chapter 4 presents two methods for mapping word representations trained on monolingual data for an arbitrary number of languages to one coherent multilingual vector space where representations of similar words and translations are represented by similar vectors. The effectiveness of the approaches is evaluated on different benchmarks and different tasks.

Chapter 5 tackles an important Dutch spelling problem with an advanced RL model. Although there exists little data with annotated spelling errors, we are able to train the model on large amounts of unannotated data by casting the problem to verb suffix prediction. We evaluate the model on multiple datasets and against different systems including the commercial spell-checker that comes with Microsoft Word.

Chapter 6 proposes a representation learning paradigm to bilingual lexicon induction that combines word-level representations for words or phrases that are learned without supervision with character-level representations that are trained from a seed lexicon. To evaluate our approach, we constructed a challenging but realistic BLI dataset consisting of a fairly limited number of documents in a specialized domain and a bilingual dictionary containing the translations of words and phrases in these documents. With this setup, we mimic a scenario were translations are sought for domain-specific terms.

The content of each contribution chapter is based on one or more scientific articles that either have been published (Chapters 3 and 6), or have been accepted for publication (Chapter 5), or will be submitted (Chapter 4) in/to well-known peer-reviewed journals or international conferences. The references to the publications can be found at the end of each chapter. In the final chapter, Chapter 7, we summarize the contributions and conclusions of the thesis and discuss interesting directions for future research.

Chapter 2

Fundamentals

In this chapter, we give an overview of fundamental concepts that are important for a complete and in-depth understanding of the contributions of this thesis.

2.1 Linear Algebra

We assume the reader is familiar with basic linear algebra concepts such as scalars, vectors and matrices, and the basic operations between them. In this section, we first enumerate the notation conventions, definitions, and properties of vectors and matrices that are used in this dissertation. Next, we give a brief summary of the singular value decomposition theorem and of two of its relevant applications.

2.1.1 Conventions and Definitions

Variables that represent a scalar will be written in non-bold, italic characters, e.g., $x = 1$.

Vector variables will be written in bold italic, e.g., $\boldsymbol{x} = (x_1, x_2, ..., x_n)$.

Intuitively, a vector norm measures the size of a vector. The Euclidean norm $||\cdot||_2$ is used most frequently, it measures the Euclidean distance of a vector w.r.t. the origin and is calculated as follows.

$$||\boldsymbol{x}||_2 = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}$$

The similarity between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is often measured with cosine similarity:

$$cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^T \boldsymbol{y}}{||\boldsymbol{x}||_2 ||\boldsymbol{y}||_2}$$

Matrix variables will be written in bold italic upper case.

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix}$$

The above matrix is $m \times n$-dimensional, where m is the number of rows and n the number of columns.

To denote a submatrix of $\boldsymbol{X}$, we use slicing notation:

$$\boldsymbol{X}_{i:j,k:l} = \begin{pmatrix} x_{ik} & \dots & x_{il} \\ \vdots & \ddots & \vdots \\ x_{jk} & \dots & x_{jl} \end{pmatrix}$$

The n-dimensional identity matrix is denoted with $\boldsymbol{I_n}$. When the dimension is clear from the context, we drop the subscript.

$$\boldsymbol{I_3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The trace of a matrix $\boldsymbol{X}$ is the product of the elements on its main diagonal. We denote it by tr:

$tr(\boldsymbol{X}) = x_{11} x_{22} \dots x_{nn}$, where n is the smallest of the two dimensions of $\boldsymbol{X}$

An orthogonal matrix is any matrix $\boldsymbol{O}$ for which holds that $\boldsymbol{O}\boldsymbol{O}^\intercal = \boldsymbol{I}$. The definition implies that $\boldsymbol{O}^\intercal \boldsymbol{O} = I$ and $\boldsymbol{O}^\intercal = \boldsymbol{O}^{-1}$ also hold. An important property of orthogonal matrices is that they do not change the Euclidean norm of the vectors they transform: ($||\boldsymbol{x}||_2 = ||\boldsymbol{O}\boldsymbol{x}||_2$). They can be described as a combination of a rotation and reflection.

2.1.2 Singular Value Decomposition

The singular value decomposition theorem states that a real-valued $m \times n$-matrix $\boldsymbol{M}$ can be decomposed into a multiplication of three matrices $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\intercal$ s.t. $\boldsymbol{U}$

is an $m \times m$ orthogonal matrix, $\boldsymbol{\Sigma}$ is an $m \times n$ diagonal matrix with non-negative real numbers on the diagonal, and $\boldsymbol{V}$ in an $n \times n$ orthogonal matrix. The diagonal entries $\sigma_1, \sigma_2, ..., \sigma_{min(m,n)}$ of $\boldsymbol{\Sigma}$ are known as the singular values of $\boldsymbol{M}$. In fact, there are multiple such decompositions, by convention the singular values are sorted in descending order $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{min(m,n)}$. It is possible that one or more of the singular values will be zero.

One way to interpret this theorem is that any transformation $\boldsymbol{Mx}$ of a vector $\boldsymbol{x}$ with a real-valued matrix $\boldsymbol{M}$, can be decomposed in three steps: 1) first, $\boldsymbol{x}$ is rotated with the orthogonal matrix $\boldsymbol{V}^\intercal$; 2) then, the dimensions of the rotated vector $\boldsymbol{V}^\intercal\boldsymbol{x}$ are scaled with the corresponding singular values $\sigma_1, \sigma_2, ..., \sigma_{min(m,n)}$; 3) after scaling there follows another rotation, this time with the orthogonal matrix $\boldsymbol{U}$.

SVD has two applications that are relevant in the context of this thesis: dimensionality reduction and the orthogonal Procrustes problem. SVD-based dimensionality reduction can be interpreted as a basic form of representation learning. The idea is that by representing the raw input data in a lower dimensional space with features that still explain most of the variance in the original data, it should be easier to make the correct generalizations over a (limited) set of annotated training examples. Let $\boldsymbol{X}$ be a data matrix, where the rows represent data points and the columns the *raw* features with which a data point is represented. Then SVD provides a means to compress $\boldsymbol{X}$ to a new matrix $\boldsymbol{R} = \boldsymbol{\Sigma}_{1:m,1:r}\boldsymbol{V}^\intercal_{1:r,1:n}$ which preserves as much of the information of $\boldsymbol{X}$ as possible, meaning that we can approximately reconstruct the original matrix $\boldsymbol{X}$ from $\boldsymbol{R}$ as $\tilde{\boldsymbol{X}} = \boldsymbol{U}_{1:m,1:r}\boldsymbol{R}$.

The orthogonal Procrustes problem is an important component in Chapter 4. It is an optimization problem with the goal to find the orthogonal transformation $\boldsymbol{W}^{(*)}$ that maps a matrix $\boldsymbol{X}$ as close as possible to a matrix $\boldsymbol{Z}$, where the distance between the matrices is measured with the Frobenius norm:

$$\boldsymbol{W}^{(*)} = \arg\min_{\boldsymbol{W}} ||\boldsymbol{XW} - \boldsymbol{Z}||_{\boldsymbol{F}} \text{ , with } \boldsymbol{WW}^\intercal = \boldsymbol{I}$$

The orthogonal Procrustes problem can be solved based on the SVD decomposition of $\boldsymbol{XZ}^\intercal$ (Schönemann, 1966):

$$\boldsymbol{W}^{(*)} = \boldsymbol{UV}^\intercal \text{ , with } \boldsymbol{U\Sigma V}^\intercal = SVD(\boldsymbol{XZ}^\intercal)$$

2.2 Probability Theory

Probability theory is a framework for dealing with uncertainty in systems. Goodfellow et al. (2016) define three sources of uncertainty: 1) inherent stochasticity in the system that is being modeled; 2) incomplete observability, even for deterministic processes

it can be useful to model them in a probabilistic framework when not all variables are directly observed; 3) incomplete modeling, sometimes we do not have a complete understanding of a process. To deal with a lack of knowledge a probabilistic framework can be used. Incomplete modeling could also be caused by practical considerations, e.g., continuous location measurements of an object can be discretized to make computations tractable, which results in uncertainty about the exact location of the object.

In this section, we revise fundamental concepts in probability theory that have immediate relevance in the context of this dissertation.

2.2.1 Basic Concepts

The **sample space** or **universe** is the set of all possible worlds in the system that is being modeled. For instance, to model two successive coin flips the sample space consists of four possible worlds: $\{(Heads, Heads), (Heads, Tails), (Tails, Heads), (Tails, Tails)\}$.

A **probability model** associates a **probability** (a numerical value between zero and one) to each possible world, such that the sum of the probabilities of the possible worlds equals one.

Probabilities can also be assigned to unions of multiple possible worlds, called **events**. Formally, an event is a subset of the sample space to which a probability is assigned. For our coin example we could consider the event that the two successive coin flips have the same face: $\{(Heads, Heads), (Tails, Tails)\}$.

To describe a possible world one typically uses one or more **random variables**. A random variable is a variable whose domain describes possible outcomes of a random phenomenon. For the coin example we could define two random variables X_1, X_2 both with domain $\{Heads, Tails\}$ to describe the outcome of the first and second coin flip respectively.

The interpretation of probabilities has been a source of debate (Russell and Norvig, 2010): from the **frequentist's** point of view probabilities can only come from experiments, they view the probability of an event as the fraction of experiments that it would be observed in after performing an infinite number of experiments. From the **Bayesian** viewpoint, probabilities describe the belief in the occurrence of an event. In this view, a model can have some prior assumption about the probability of an event, which can be updated as evidence in the form of experiments is acquired.

2.2.2 Probability Distributions

A **probability distribution** describes the likelihood of each possible value in the domain of a random variable. For discrete random variables such as the ones in our coin example, the (discrete) probability distribution $P(X)$ is described by enumerating all values and their corresponding probability (e.g., $P(X_1) = <P(X_1 = Heads) = 0.5, P(X_1 = Tails) = 0.5>$). For continuous random variables there are infinitely many values all of which have zero probability. For such variables the probability distribution is described with a **probability density function** $p(X)$ [1], which defines the probability that the outcome of the random variable falls within a given range $X \in (x_1...x_2)$ as follows:

$$P(x_1 \leq X \leq x_2) = \int_{x_1}^{x_2} p(X)dX$$

The concept of a distribution can also be extended to multiple variables: the distribution that describes the joint occurrence of N random variables X_1, ..., X_N is called the **joint probability distribution** $P(X_1,...,X_N)$ of X_1, ..., X_N. $P(X)$ is also called the **marginal distribution** of X. In the remainder of this subsection, we enlist commonly-used probability distributions that are important for this dissertation, particularly for Chapter 3.

The *Bernoulli distribution* is a two-dimensional distribution which defines the probability of success in a single trial (e.g., the chance to get *heads* in a single coin toss) and has a single parameter $\delta \in (0...1)$. We denote a Bernoulli distributed random variable with $X \sim Bernoulli(\delta)$. The distribution is calculated as follows:

$$P(X;p) = \begin{cases} p & \text{, if } X = success \\ 1-p & \text{, if } X = failure \end{cases}$$

The *Binomial distribution* generalizes the Bernoulli distribution to multiple independent trials (e.g., n consecutive coin flips). We denote a Binomial distributed random variable with $X \sim Binomial(n,p)$. Given the probability of success p and the number of trials n, the probability distribution is computed as follows:

$$P(X = x;n,p) = \binom{n}{x} p^x (1-p)^{(n-x)}$$

where x is the number of trials that succeeded and $\binom{n}{x} = \frac{n!}{x!(n-x)!}$

[1] Lowercase p is typically used to describe continuous distributions, whereas upper case P is used for discrete distributions. For notational convenience we will sometimes use P for properties/definitions that hold for both discrete and continuous distributions.

The *Multinomial distribution* generalizes the Binomial distribution to experiments with more than two outcome values (e.g., rolling a die n consecutive times). It is parametrized by an N-dimensional vector $\phi = (\phi_1, \phi_2, ..., \phi_N)$ for which each element ϕ_i lies between zero and one and for which the sum equals one;[2] and the number of trials n. We denote a set of multinomial variables as $X_1, X_2, ..., X_n \sim Multinomial(n, p)$. It is computed as follows:

$$P(X =< x_1, x_2, ..., x_N >; n, \phi) = \frac{n!}{\prod_{i=1}^{N} x_i!} \prod_{i=1}^{n} \phi_i^{x_i}$$

where each x_i denotes the number of times that i^{th} outcome

value/category has been selected.

The special case of the Multinomial distribution where $n = 1$ is sometimes referred to as a Multinoulli or categorical distribution (Goodfellow et al., 2016; Murphy, 2012).

The *Beta distribution* is a one-dimensional, continuous distribution with two parameters χ^S, χ^T [3] with a domain $(0...1)$. The distribution is typically used to express prior knowledge about the success probability parameter p of the Bernoulli or Binomial distributions, in which case it can be interpreted a distribution over distributions. The distribution is defined as:

$$p(X = p; \chi^S, \chi^T) = \frac{1}{B(\ (\chi^S, \chi^T\ ))} p^{(\chi^S - 1)} (1 - p)^{(\chi^T - 1)}$$

with B the Beta function and Γ the Gamma function:

$$B(\ (x_1, x_2, ..., x_N)\ ) = \frac{\prod_{i=1}^{N} \Gamma(x_i)}{\Gamma(\sum_{i=1}^{N} x_i)}$$

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$$

The distribution parameters χ^S, χ^T are called concentration parameters because their ratio determines the mode of the distribution[4] and their size determines how concentrated/peaked the distribution is at the mode. When used as a prior on the success probability parameter p, these parameters can be interpreted as pseudo-counts

[2] Note that due to this summation constraint there are only N-1 degrees of freedom when choosing these parameters.

[3] These parameters are usually referred to as α and β, but we use χ^S, χ^T to avoid confusion with the parameters of the Dirichlet distribution.

[4] The mode of a distribution of a random variable X is the value for X at which the distribution function takes it maximum value.

for the number of failures and successes respectively: the higher their values the stronger the prior and the more experiments are needed to significantly change the prior.

The *Dirichlet distribution* is the generalization of the Beta distribution to N dimensions and hence can be used to represent prior knowledge on the ϕ parameters of a Multinomial distribution. It is parametrized by an N-dimensional vector of concentration parameters $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_N)$. The distribution is defined as:

$$p(X = \phi; \alpha) = \frac{1}{B(\phi)} \prod_{i=1}^{N} \phi_i^{\alpha_i - 1}$$

For all the distribution functions we introduced so far the parameters have an intuitive interpretation, e.g., the parameter of a Bernoulli distribution is the probability of success. It is also possible to define **black-box distribution** functions, where the parameters do not have a clear interpretation. Distributions parametrized by neural networks, which are introduced in Section 2.3.1, are good examples.

2.2.3 Calculating with Joint Distributions

The **full joint distribution** of a probability model is the joint distribution of all random variables that are modeled. Given the full joint distribution it is possible to assign a probability to any assertion about the possible worlds. For instance for the coin example, given the distribution $P(X_1, X_2)$ we can calculate the probability of flipping $Heads$ the first time $P(X_1 = Heads)$, or flipping $Tails$ the second time given the first outcome was $Heads$ $P(X_2 = Tails | X_1 = Heads)$, etc. The following equations allow to compute these probabilities:

$$P(X_1) = \sum_{x_2 \in X_2} P(X_1, X_2)$$

$$P(X_2 | X_1) = \frac{P(X_1, X_2)}{P(X_1)}$$

where $\sum_{x_2 \in X_2}$ means summing over all values in the domain of X_2. From the latter equation we can derive **Bayes' rule** which is the theoretical justification of many machine learning methods:

$$P(X_1 | X_2) P(X_2) = P(X_1, X_2) = P(X_2 | X_1) P(X_1)$$

$$P(X_2 | X_1) = \frac{P(X_1 | X_2) P(X_2)}{P(X_1)}$$

These equations hold in general for a model with N random variables:

$$P(X_1, X_2, X_3, ..., X_{N-1}) = \sum_{x_N \in X_N} P(X_1, X_2, ..., X_N)$$

$$P(X_1 | X_2, X_3, ..., X_N) = \frac{P(X_1, X_2, ..., X_N)}{P(X_2, X_3, ..., X_N)}$$

These equations also hold for continuous random variables where the summations are replaced by integrals over the domain of the continuous random variables.

The most general way to specify the joint distribution is by enumerating all possible worlds (i.e., all the value combinations of all the random variables) and their corresponding probabilities. For large models, with a large number of variables, this becomes intractable as this requires storing as many entries as the product of the domain sizes of the random variables. For example, if we want to model two successive coin tosses we need $2 \cdot 2 = 4$ entries, whereas for ten successive coin tosses we already need $2^{10} = 1024$ entries. A way to deal with this problem is to make independence assumptions. If two or more random variables are mutually independent their joint distribution factors into their marginals:

$$P(X_1, X_2, ..., X_N) = P(X_1)P(X_2)...P(X_N)$$

Under this assumption, the full joint distribution of ten successive coin tosses can be specified in terms of the distributions of the marginals so only $2 \cdot 10 = 20$ entries are required. A weaker type of independence assumption is conditional independence, where a set of random variables $\mathscr{X}$ is independent of a set of random variables $\mathscr{Y}$ under the observation of a third set of random variables $\mathscr{Z}$:

$$P(\mathscr{X}, \mathscr{Y} | \mathscr{Z}) = P(\mathscr{X} | \mathscr{Z}) P(\mathscr{Y} | \mathscr{Z})$$

2.2.4 Estimating Distributions

Once the probability model and its independence assumptions have been specified, the goal is to obtain meaningful probabilities based on the prior knowledge (if applicable) and the data/evidence we possess. The evidence or data comes in the form of observations of the random variables in previous experiments. The term *experiments* is used in a broad sense here. For successive coin tosses, experiments are previous successive coin tosses; for language modeling, where the goal is to estimate the probability of a sentence, experiments are sentences that are observed in a collection of texts. In the context of machine learning, the data is split into multiple portions: a **training dataset**, which is used to estimate the parameters, a **validation dataset** to

tune hyper-parameters[5], and a **test dataset** to evaluate how good the final model (i.e., with set values for the hyper-parameters and model parameters) performs. It is possible that some random variables are not observed in the data but are nevertheless important to describe the process that is being modeled. These are called **latent variables**.

Estimating a distribution $P(X;\boldsymbol{\theta})$ comes down to estimating the parameters $\boldsymbol{\theta}$ of the distribution. Two important estimation criteria are **maximum likelihood (ML)** and **maximum a posteriori (MAP)**.

Maximum likelihood estimation aims at finding the vector of model parameters $\boldsymbol{\theta}$ that best explains the data. This criterion is linked to the frequentist interpretation of probabilities and assumes that there is one correct value for each parameter and the goal is to estimate this value from experiments:

$$\boldsymbol{\theta}_{ML} = \arg\max_{\boldsymbol{\theta}} \sum_{z \in Z} P(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{Z} = \boldsymbol{z}; \boldsymbol{\theta})$$

Where $\boldsymbol{X}$ are the observed variables and $\boldsymbol{Z}$ are the latent variables which have to be summed/integrated out. Maximum a posteriori estimation takes a different viewpoint and considers the parameters $\boldsymbol{\Theta}$ itself random variables[6] which follow a certain prior distribution $P(\boldsymbol{\theta})$. This is in line with the Bayesian interpretation of probabilities. The optimal parameter value is obtained as follows:

$$\boldsymbol{\Theta}_{MAP} = \arg\max_{\boldsymbol{\Theta}} \sum_{z \in Z} P(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{Z} = \boldsymbol{z} \mid \boldsymbol{\Theta}) P(\boldsymbol{\Theta})$$

It is not always necessary to obtain values for the parameters $\boldsymbol{\theta}$, namely when the goal is to make predictions/probabilistic assertions about the random variables. In this case, we can keep using the distribution of $\boldsymbol{\theta}$ to do inference: $P(\boldsymbol{X} = \boldsymbol{x}) = \int_{\boldsymbol{\theta}} P(\boldsymbol{X} = \boldsymbol{x}|\boldsymbol{\theta}) P(\boldsymbol{\theta})$.

Now that we have specified the criteria for obtaining the distribution parameters, the question remains how to compute the optimum for the chosen criteria. When the joint probability distribution is simple enough, it could be possible to obtain the optimal value for $\boldsymbol{\theta}$ analytically by setting the gradient of the objective (ML or MAP) w.r.t. $\boldsymbol{\theta}$ equal to zero and solving for $\boldsymbol{\theta}$. However, it may happen that there is no closed-form solution for $\boldsymbol{\theta}$ or that calculating the solution would be intractable because there are too many latent variables.[7] In this case, one needs to resort to approximate optimization techniques. There are many such techniques but we will limit our explanation to two techniques that are used in this dissertation: Gibbs sampling and stochastic gradient descent.

Gibbs sampling is a Monte Carlo Markov chain (MCMC) estimation technique used to obtain a sequence of observations that approximately come from a given distribution

[5] Hyper-parameters are parameters that are fixed before estimating the model parameters.

[6] To stress that the parameters are now random variables we denote them with capital $\boldsymbol{\Theta}$ instead of $\boldsymbol{\theta}$.

[7] Recall that every latent variable has to be summed or integrated out.

for which direct sampling is difficult. With this sequence of observations the full joint distribution, distributions on a subset of the variables, and the values of latent variables can then be approximated. Let $X_1, X_2, \ldots, X_N$ be the random variables that need to be sampled[8] then we obtain a sequence of samples as follows:

- Start with an initial sample $x_0^{(0)}, x_1^{(0)}, \ldots, x_N^{(0)}$.
- Then iteratively obtain new samples for each random variable X_i by sampling from the conditional distribution below until some convergence criterium is met (e.g., after a fixed number of steps is reached or based on the ML/MAP objective):

$$X_i^{(t+1)} \sim P(X_i^{(t+1)} | X_0^{(t+1)}, X_1^{(t+1)}, \ldots, X_{i-1}^{(t+1)}, X_{i+1}^{(t)}, \ldots, X_N^{(t)})$$

  where the superscripts denote the time step/iteration

It is not always necessary to sample all random variables of the probability model. It may be possible to sum/integrate variables out and calculate them after convergence based on other random variables. This computationally more efficient variant is called **collapsed Gibbs sampling**.

Another common optimization technique is **stochastic gradient descent** (SGD). SGD assumes the training objective is written as a loss function (i.e., as a quantity we want to minimize) that can be derived w.r.t. every parameter that is being estimated. The idea of SGD is to iteratively select samples from the training dataset and make slight local changes to each of the parameters such that the loss function becomes smaller when evaluated on the example. Specifically, the new value of the parameter vector $\boldsymbol{\theta}$ is calculated by subtracting a fraction λ of the gradient of the loss function L w.r.t. $\boldsymbol{\theta}$ from its old value:

$$\boldsymbol{\theta}^{new} \leftarrow \boldsymbol{\theta}^{old} - \lambda \nabla_{\boldsymbol{\theta}} L(\boldsymbol{X_i}, \boldsymbol{\theta})$$

λ is the **learning rate**, a parameter that controls the degree to which the parameters are updated. In the most basic version of SGD, the learning rate is a hyper-parameter that is fixed at the beginning of training. For most systems, the gradients are often calculated on a batch of examples rather than on a single example. This concept is known as **mini-batching** and both leads to a more stable estimate of the gradient as well as a higher degree of parallelism in the computations, which can be exploited by specialized hardware such as Graphics Processing Units (GPUs).

There exist several other modifications to SGD that result in better convergence properties such as using an adaptive learning rate and taking into account the gradients

[8] Note that when taking a Bayesian perspective this set may include the distribution parameters.

of previous batches to help gradients accelerate in the right direction. These result in a plethora of SGD-based optimization algorithms such as Adaptive Gradient Algorithm (AdaGrad, Duchi et al., 2011), Root Mean Square Propagation (RMSProp, Hinton et al., 2012), and Adaptive Moment Estimation (Adam, Kingma and Ba, 2015). Throughout this dissertation, we utilize Adam, which adapts the learning rates for each parameter separately, because its recommended hyper-parameters (e.g., the initial learning rate) are robust across different problems and hence require little to no tuning.

2.3 Representation Learning

Representation learning is a branch of machine learning concerned with the study of techniques that automatically learn good feature representations for the raw input data. First, we introduce neural networks, a set of representation learning techniques that play a prominent role throughout this thesis. Following that, we discuss two important representation learning techniques for text: latent Dirichlet allocation, a well-known representation learning model for documents and words that will be the foundation of the work in Chapter 3; and word embeddings, a paradigm to represent words that is used in Chapters 4-6.

2.3.1 Artificial Neural Networks

Artificial neural networks (further simply *neural networks*) are a prominent class of models within representation learning. They can be seen as a heavily simplified version of biological neural networks in human/animal brains. Neural networks constitute of a collection of interconnected neurons. Each neuron computes its output value y_j by calculating a weighted sum of values of its input neurons x_1, ..., x_N, optionally adding a bias term b_j to the sum, followed by applying an activation function g to the result:

$$y_j = g(\sum_{i=1}^{N} w_{ji} x_i + b_j) \text{ , where each } w_{ji} \text{ is a weight}$$

Neurons can be grouped in layers: neurons in the same layer receive inputs from the same set of neurons, use the same activation function, and send their outputs to the same set of neurons. When every neuron in a layer is connected to all its input neurons, the layer is said to be **fully connected**. Figure 2.1 shows an example of a simple neural network consisting of three layers. The first and last layers are called the input and output layers respectively, and the layers in between are called hidden layers.

It is possible to write the output computation of an M-dimensional layer with N input neurons succinctly with matrix-vector operations. Let $\boldsymbol{W}$ be an $N \times M$-dimensional

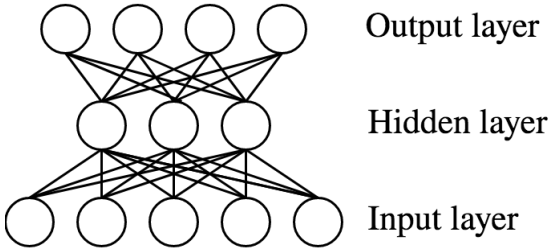


Figure 2.1: Illustration of a simple feed forward neural network with a 5-dimensional input layer, a 3-dimensional hidden layer, and a 4-dimensional output layer.

weight matrix, b an M-dimensional bias vector, and $\boldsymbol{x}$ the N-dimensional vector with the outputs of the input neurons, then we calculate the M-dimensional output of the layer $\boldsymbol{y}$ as:

$$\boldsymbol{y} = g(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$$

There are several commonly-used activation functions:

The **sigmoid** function (see Figure 2.2a) takes a scalar as input and squashes it to a range of $(0,1)$. This makes it suitable to predict the probability of success for a Bernoulli experiment and makes it a common choice at the output layer of a binary classification problem.

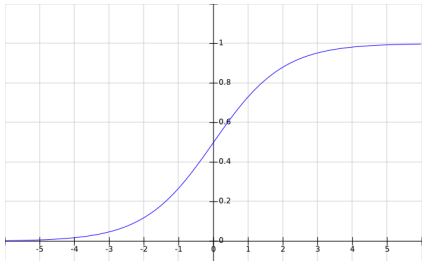
$$sigmoid(x) = \frac{1}{1+e^{-x}}$$

The **hyperbolic tangent** or tanh function (see Figure 2.2b) is similar to the sigmoid function but squashes its input to a range of $(-1,1)$:

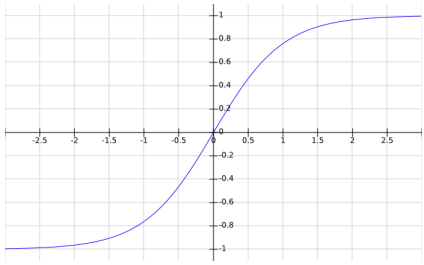
$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

The **ReLU function** (see Figure 2.2c) converts a scalar to a range between zero and infinity $(0,\infty)$ and results in more sparse feature activations. It was proposed as an activation function for neural networks fairly recently by Jarrett et al. (2009); Nair and Hinton (2010).

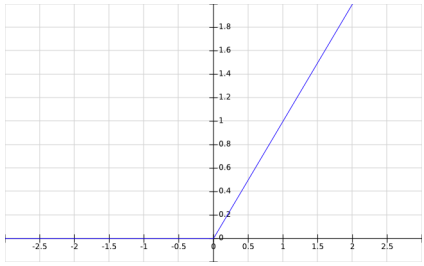
$$ReLU(x) = \begin{cases} 0 \text{ , if } x < 0 \\ x \text{ , if } x \geq 0 \end{cases}$$



(a) $sigmoid(x)$



(b) $tanh(x)$



(c) $ReLU(x)$

Figure 2.2: Visualization of commonly-used non-linear activation functions (notice the different axis scales).

Although, the above activations functions operate on scalars they can be generalized to vector inputs by applying the function to each element of the vector. This is typically denoted with the same symbol, e.g., $tanh(\boldsymbol{x}) = (tanh(x_1)\ ,\ tanh(x_2),\ \ldots\ ,\ tanh(x_N))$

The **softmax function** converts a vector to a probability distribution. This makes it a common choice at the output layer of a multilabel classification problem.[9]

$$softmax(\boldsymbol{x}) = \frac{e^{\boldsymbol{x}}}{\sum_{x_i \in \boldsymbol{x}} e^{x_i}}$$

Two interesting properties of these functions are that 1) their derivatives are defined for their entire domain, making them suitable for gradient-based parameter training; and 2) that they are non-linear. While it is possible to use linear layers, i.e., use the identity function as the activation function, it should be noted that the computation of multiple subsequent linear layers can always be expressed with a single linear layer. Therefore in practice, linear layers are only seen at the input and/or output layers.

Networks with two or more hidden layers are called **deep neural networks**. Although it has been shown that even with a single layer neural networks are universal function approximators (Cybenko, 1989), the assumption that functions which describe real-life phenomena are decomposable in many simple functions appears to work well in practice. Put differently, adding more layers to a neural network is not merely aimed at increasing the model capacity - this can be easily achieved by adding more neurons to a single hidden layer - but is a structural prior that often yields a better approximation of the target function.

Typically, the output neurons of a neural network can be interpreted as the parameters of a conditional probability distribution $P(Y|X)$. This implies that ML/MAP can be used as the criteria to estimate network parameters (i.e., the weight matrices and bias vectors). For the neural network models in this dissertation (see Chapters 5 and 6) we have used the ML objective. For most neural networks it is not possible to obtain a closed-form solution for the training objective. Instead, the optimization of neural networks is typically done with (some variation of) stochastic gradient descent. Because of the compositional nature of neural network functions, and using the chain rule, the gradients of the loss function w.r.t. the parameters and outputs of layer i can be calculated efficiently using the gradients of layer $i+1$. The algorithm that efficiently computes gradients for neural networks in a top-down fashion (i.e., from the output layer to the input layer) is known as backpropagation.

[9] When $\boldsymbol{x}$ is a vector, $e^{\boldsymbol{x}}$ denotes the element-wise exponential function $e^{\boldsymbol{x}} = (e^{x_1}, e^{x_2}, ..., e^{x_N})$

2.3.2 Neural Network Building Blocks

In this section, we introduce important building blocks for designing neural networks that are relevant in the context of this thesis.

First, we need a way to represent the input and output variables. In the context of NLP, we typically work with categorical variables which can be represented using **one-hot vectors**. One-hot vectors are vectors for which all dimensions but one are zero. A categorical variable that can take N possible values can be encoded with an N-dimensional one-hot vector, where the non-zero entry signals which of the N categories the vector represents.

Next, we introduce building blocks for computing the outputs from the inputs. The most basic computation is a **fully connected** or **dense** layer, which was already introduced in Section 2.3.1. Figure 2.1 displays two fully connected layers: 1) the hidden layer is fully connected to the input layer, and 2) the output layer is fully connected to the hidden layer. A neural network like the one in Figure 2.1 for which the neuron connections do not form any cycles is known as a **feed forward** neural network. When a fully connected layer projects/embeds high-dimensional one-hot inputs into a low-dimensional space using the identity function as the activation function it is called an **embedding layer**. Conceptually an embedding layer comes down to associating a vector to every dimension of the one-hot vector.

Another important architecture is the recurrent neural network (RNN). RNNs are a means to map a variable length vector sequence $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$ to a single vector $\boldsymbol{h}_t$. The core assumption is that a good representation $\boldsymbol{h}_t$ for the sequence $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$ can be computed based on $\boldsymbol{x}_t$ and the representation $\boldsymbol{h}_{t-1}$ of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}$. $\boldsymbol{h}_t$ is called the **state** of time step n. To calculate the first hidden state an initial hidden state vector $\boldsymbol{h}_0$ has to be defined. This is either a vector with all entries set to zero or a vector that is trained jointly with the rest of the network parameters. From the state, the output $\boldsymbol{y}_t$ for time step t is calculated.[10] Figure 2.3 is a high-level illustration of the computations of an RNN unrolled across four time steps.

A **simple/vanilla RNN** or **Elman network** (Elman, 1990) is specified by the following recursive equations, where g_h and g_y are non-linear activation functions:

$$\boldsymbol{h}_n = g_h(\boldsymbol{W}_x\boldsymbol{x} + \boldsymbol{W}_h\boldsymbol{h}_{n-1} + \boldsymbol{b}_h)$$

$$\boldsymbol{y}_n = g_y(\boldsymbol{W}_y\boldsymbol{h}_n + \boldsymbol{b}_y)$$

While Elman networks could theoretically learn to incorporate dependencies of arbitrary length, finding the parameter values for $\boldsymbol{W}_x$, $\boldsymbol{W}_h$, $\boldsymbol{b}_h$ that do so is infeasible due to the vanishing gradient problem. The vanishing gradient problem refers to the phenomenon

[10] It is also possible to use the state directly as the output $\boldsymbol{y}_t = \boldsymbol{h}_t$.

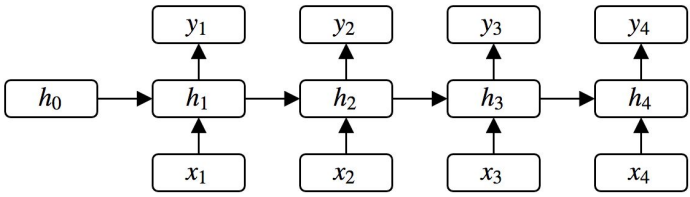


Figure 2.3: Illustration of the computations of an RNN unrolled across four time steps. $\boldsymbol{h}_0$ is the initial state of the RNN. Each other state h_t is computed based on the previous state $\boldsymbol{h}_{t-1}$ and the current input $\boldsymbol{x}_t$.

where the gradients can become so small that they have no significant effect on the parameter update. It manifests itself in vanilla RNNs and deep neural networks when training with gradient-based optimization techniques such as SGD. This is a consequence of the fact that computing the gradients of parameters/activations in bottom layers/early time steps requires repeatedly applying the chain rule. As a result, these gradients are computed as a product of a large number of factors. If many of these factors are small the gradient vanishes. Similarly, the gradient may explode when many of these factors are > 1.

A way to counteract the vanishing gradient problem is to introduce short-cuts in the network that copy the values of neurons in the previous layer or state to the next layer. Specifically, to add a short-cut to a layer $\boldsymbol{y} = g(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$, the neurons of a subsequent layer $\boldsymbol{y}$ could be calculated as a weighted sum of 1) the neurons of the previous layer $\boldsymbol{x}$; and 2) the *standard* transformation on the previous layer:

$$\boldsymbol{y} = \boldsymbol{f} \odot \boldsymbol{x} + \boldsymbol{i} \odot g(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$$

where the $\boldsymbol{i}$ and $\boldsymbol{f}$ are weight vectors also called gates in this context

and $\odot$ denotes elementwise multiplication.

Long short-term memory (LSTM) is a type of RNN that applies this principle. It uses a gate $\boldsymbol{f}_t$ to control what information is propagated from the previous to the next state, a gate $\boldsymbol{i}_t$ to control what information is extracted from the new input $\boldsymbol{x}_t$; and a gate $\boldsymbol{o}_t$ to control what information of the state is exposed to the output neurons. An LSTM network is defined with the following recursive equations:

$$\boldsymbol{f}_t = sigmoid(\boldsymbol{W}_{f,x}\boldsymbol{x}_t + \boldsymbol{W}_{f,h}\boldsymbol{h}_{t-1} + \boldsymbol{b}_f)$$

$$\boldsymbol{i}_t = sigmoid(\boldsymbol{W}_{i,x}\boldsymbol{x}_t + \boldsymbol{W}_{i,h}\boldsymbol{h}_{t-1} + \boldsymbol{b}_i)$$

$$\boldsymbol{o}_t = sigmoid(\boldsymbol{W}_{o,x}\boldsymbol{x}_t + \boldsymbol{W}_{o,h}\boldsymbol{h}_{t-1} + \boldsymbol{b}_o)$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot tanh(\boldsymbol{W}_{c,x}\boldsymbol{x}_t + \boldsymbol{W}_{c,h}\boldsymbol{h}_{t-1} + \boldsymbol{b}_c)$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot tanh(\boldsymbol{c}_t)$$

Note that the gate variables are calculated from the current input $\boldsymbol{x}_t$ and previous hidden state $\boldsymbol{h}_{t-1}$, allowing the network to learn when to forgot and when to remember information from previous time steps. Different variations on the LSTM gating are possible, e.g., the Gated Recurrent Unit (GRU) (Cho et al., 2014) is another commonly-used alternative. LSTMs and GRUs are widely used to model language as it requires representing variable length sequences with long-range dependencies (e.g., sentences, documents).

It is a common misconception that LSTMs also solve the exploding gradient problem. Introducing short-cuts does not inhibit the gradients from becoming large. Fortunately, exploding gradients turn out to be less of a problem in practice. It can be prevented by **clipping gradient norms** to a given threshold.

Attention mechanisms are another important building block. Their intent is to filter out relevant information from a set of input vectors $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, ... , $\boldsymbol{x}_n$ (also called the *values*) based on a query vector $\boldsymbol{q}$ that encodes what type of information should be retrieved from the inputs. Formally, an attention mechanism calculates an attention vector as a weighted average of the input vectors, where the weights are computed based on a comparison of the inputs with the query such that input vectors that are more similar to the query vector get higher attention weights. The similarity function between the query vector and one of the inputs is called the **attention function**. As RNNs and attention mechanisms both compute a single vector from multiple input vectors they may look similar at first glance but they are fundamentally different in the fact that an attention mechanism does not take into account the sequence order of its inputs and aggregates its inputs as a simple linear combination. In this respect, the intent of the attention mechanism is to *focus* on a limited number of elements from a *set* of inputs, whereas an RNN creates a *summary* of a *sequence* that is biased towards the last elements. Attention is often used to further enhance the memory of RNNs. Although RNNs such as the LSTM can model long-range dependencies their memory is limited because their state is a fixed-length vector. By allowing RNNs to query previous states with attention their memory capacity is extended.

An important application of this concept is found in neural machine translation: when generating the next target word y_t in a translation of a source sentence x_1, x_2, ... , x_n, attention filters the source sentence by attending to the word representations that are most relevant for generating y_t (typically the representation(s) of the word(s) which has/have to be translated next). Attention is explained in more detail in Chapter 5.

A final aspect of neural network architectures we need to discuss is regularization. Neural networks are powerful function approximators, so it is important to not let them overfit on random artifacts of the training data that do not generalize over the

true distribution. To this end, there exist several regularization techniques which limit the capacity of the model during training. Classical techniques such as L_1 and L_2 regularization have been used, but in recent year dropout regularization (Srivastava et al., 2014) has become more popular.[11] **Dropout** is a technique that randomly (i.e., according to a Bernoulli distribution with probability p) drops neurons and their connections during training. This forces each neuron to represent a feature that is meaningful independent of the other neurons. In this thesis, we consistently use dropout for regularizing neural networks.

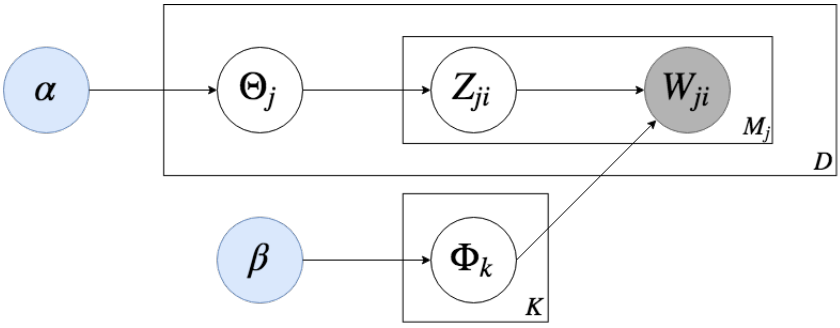


Figure 2.4: Graphical representation of the latent Dirichlet allocation model. Blue circles denote parameters of prior values, which are set in advance; gray circles denote observed random variables; and the white circles denote latent random variables.

Algorithm 1: LDA: GENERATIVE STORY

initialize: (1) the total number of topics: K;
(2) the values for Dirichlet priors parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$;
sample K times $\boldsymbol{\Phi}_k \sim Dirichlet(\boldsymbol{\beta})$;
for $j \leftarrow 1$ **to** D **do**
  sample $\boldsymbol{\Theta}_j \sim Dirichlet(\boldsymbol{\alpha})$
  for $i \leftarrow 1$ **to** M_j **do**
    sample $Z_{ji} \sim Multinomial(1, \boldsymbol{\Theta}_j)$
    sample $W_{ji} \sim Multinomial(1, \boldsymbol{\Phi}_k)$, with $Z_{ji} = z_k$

2.3.3 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is a generative graphical model for documents. Algorithm 1 shows LDA's generative story and Figure 2.4 shows its equivalent

[11] Note that is also feasible to combine multiple regularization methods, but it comes at the cost of more hyper-parameters that need tuning.

graphical representation using plate notation. The model assumes that a document D_j is constructed by sampling M_j random variables $Z_{j1},...,Z_{jM_j}$ from a K-dimensional multinomial distribution $Multinomial(1,\boldsymbol{\Theta_j})$. Each Z_{ji} is a latent random variable with domain $\{z_1,z_2,...,z_K\}$ for which every element z_k is associated with a multinomial distribution $Multinomial(1,\boldsymbol{\Phi_k})$ over words. After sampling the value z_k for Z_{ji}, a word W_{ij} is sampled from the word distribution $Multinomial(1,\boldsymbol{\Phi_k})$ that is associated with z_k. M_j hence corresponds to the number of words in document D_j. Both the parameters of the word distributions $\boldsymbol{\Phi_k}$ and the parameters of the topic distributions $\boldsymbol{\Theta_j}$ are considered random variables and are drawn from Dirichlet prior distributions, $Dirichlet(\boldsymbol{\alpha})$ and $Dirichlet(\boldsymbol{\beta})$ respectively.

The key strength of LDA is not its ability to assign probabilities to documents, but rather the fact that the parameters can be used to infer valuable and interpretable representations for documents and words, i.e., documents can be represented by their topic distribution $\boldsymbol{\Theta}_j$ and word representations can be derived from the per-topic word distributions $\boldsymbol{\Phi}_1,...,\boldsymbol{\Phi_K}$(Vulić et al., 2015).

LDA is an example of a probability model for which it is impossible to express a solution to the ML/MAP objective in closed form. To estimate the parameter values $\boldsymbol{\Theta}_j$ and $\boldsymbol{\Phi}_k$ collapsed Gibbs sampling is a common choice.

2.3.4 Word embeddings

In Section 2.3.2, we introduced the concept of an embedding layer: a linear, fully connected layer that projects one-hot input vectors to a low-dimensional space. The majority of neural network-based NLP systems uses an embedding layer to project words, represented as one-hot vectors, to a low-dimensional space. Hence, every word will be represented by a unique low-dimensional vector[12] known as a **word embedding**. Word embeddings can be trained with supervision (i.e., on annotated training examples of the task at hand) or without supervision (i.e., on unannotated text corpora). An example of the latter which is repeatedly used in this dissertation is the continuous Skip-gram model (also commonly referred to as *word2vec*, which is one of the software packages that implements this model).

The continuous Skip-gram (Mikolov et al., 2013b) is a log-linear two-layer neural network, in its most basic variant (Mikolov et al., 2013b) it predicts the probability that a word x_i occurs in the local context of word x. The network consists of an embedding layer that projects the one-hot representation $\boldsymbol{x}$ of word x to its word embedding $\boldsymbol{h}$, followed by a fully connected layer with the softmax activation function.

$$\boldsymbol{h} = \boldsymbol{W}\boldsymbol{x}_{one\text{-}hot}$$

[12] Word embedding dimensions typically range between 25 and 1000, which is small compared to typical vocabulary sizes.

$$\boldsymbol{o} = softmax(\boldsymbol{V}\boldsymbol{h})$$

$$P(x_i|x) = o_i$$

Formally, the local context of a word occurrence in a text corpus is comprised of the first n words to its left and right respectively, where n is known as the window size. Suppose, for example, that *the cat sat on the mat* is a sentence that is part of the preprocessed text corpus, then the model will update the network weights such that the likelihood of predicting the local context from *the* increases: $P(cat|the)$, $P(sat|the)$, $P(on|the)$, $P(the|the)$, $P(mat|the)$. Analogously, the likelihoods of the local contexts of the other words in the sentence are optimized. Each neuron in the output layer corresponds to the probability of a word in the vocabulary. The weights of the input connections of these neurons are collectively called the context embeddings, whereas the weights of the input layer are called the input word embeddings. Similar to LDA, the intent of the continuous Skip-gram model is to extract good word representations from its parameters (the word embedding weights in particular), the probability estimates it learns will not be used after training.

It is important that word embedding models are computationally efficient such that they can be trained on large amounts of text data. When the vocabulary (i.e., the number of unique words in the corpus) is large, the computations in the last layer of continuous Skip-gram become very expensive. Therefore, in practice, the objective is approximated with *negative sampling* (Mikolov et al., 2013a). For continuous Skip-gram with negative sampling (SGNS) the softmax activation function is replaced by the element-wise sigmoid and the training objective is to increase the value of the output neuron that corresponds to the context word while decreasing the values of N_s other randomly sampled neurons. Instead of needing to calculate a large matrix-vector multiplication and softmax activation $\boldsymbol{o} = softmax(\boldsymbol{V}\boldsymbol{h})$, the negative sampling objective can be implemented very efficiently with a few vector dot products.

There are many other word embedding models (Mikolov et al., 2013a; Pennington et al., 2014; Bojanowski et al., 2016, *inter alia*), for instance, continuous bag-of-words uses the inverse architecture of continuous Skip-gram (i.e., predicting words given their local context). Bilingual word embedding Skip-gram (BWESG; Vulić and Moens (2016a)) is another variant that extends SGNS to learn bilingual word embeddings from subject-aligned document pairs. It uses a shuffling scheme to convert aligned document pairs to single *multilingual* documents and then trains the SGNS model with a larger window size on these pseudo-multilingual documents.

2.4 Conclusion

In this chapter, we provided a dense overview of the mathematical foundations and key concepts on which the research presented in this thesis is built. Chapter 3 rests on probability theory and the latent Dirichlet allocation model; Chapter 4 uses the linear algebra fundamentals, including the solution of the Procrustes problem, to project word embeddings of different languages to the same vector space; Chapters 5 and 6 build on probability theory and the neural network fundamentals; and word embeddings are an important component throughout Chapters 4-6.

Chapter 3

C-BiLDA: Extracting Cross-lingual Topics from Non-Parallel Texts by Distinguishing Shared from Unshared Content

In this chapter, we introduce a new RL model for learning bilingual text representations without supervision on non-parallel training corpora. Specifically, we propose C-BiLDA, a bilingual extension to the latent Dirichlet allocation model (see Section 2.3.3 of the fundamentals chapter). Because the word and document representations induced by LDA are easily interpretable yet still result in good classification performance when used as input features for NLP tasks, LDA is one of the most widely used models for learning text representations without supervision.

The bilingual extension we propose learns representations of words/documents such that words/documents with similar meaning get similar representations regardless of their language. Such representations are very useful in weakly-supervised settings as they allow knowledge transfer from resource-rich languages to languages with a limited amount of annotated training data. For example, a classifier that uses multilingual input representations can be trained on annotated data in English and later applied to any of the other languages for which multilingual representations had been induced. Unlike related research efforts, we extend LDA to two languages without assuming the

availability of a parallel corpus to train the representations. It requires a collection of bilingual document pairs which are linked by their main subject only (e.g., a collection of document pairs where each pair consists of an English and Dutch news article on the same event).

We empirically validate our new RL model by training the model on a multilingual collection of Wikipedia articles and evaluating the representations on two cross-lingual document classification datasets.

3.1 Introduction

Cross-lingual text mining aims to induce and transfer knowledge across different languages to help applications such as cross-lingual information retrieval (Levow et al., 2005; Ganguly et al., 2012; Vulić et al., 2013), document classification (Prettenhofer and Stein, 2010; Ni et al., 2011; Guo and Xiao, 2012a), or cross-lingual annotation projection (Zhao et al., 2009; Das and Petrov, 2011; van der Plas et al., 2011; Kim et al., 2012; Täckström et al., 2013; Ganchev and Das, 2013) in cases where translation and class-labeled resources are scarce or missing. In this chapter, we utilize probabilistic topic models to perform cross-lingual text mining. Probabilistic topic models (PTMs) are unsupervised generative models for representing document content in large document collections. Probabilistic topic models assume that every document is associated with a set of hidden variables, called topics, which determine how the words of the document were generated. Formally, a topic is a probability distribution over terms in a vocabulary. Informally, a topic represents an underlying semantic theme (Blei and McAuliffe, 2007). A representation of a document by such semantic themes has the advantage of being independent of both word-choice and language. Fitting a probabilistic topic model on a text collection is done by assigning the values to the hidden variables that best explain the data (see Section 2.2.4).

In monolingual settings the majority of text mining research using topic models is based on the probabilistic latent semantic analysis (pLSA) (Hofmann, 1999) or latent Dirichlet allocation (LDA) (Blei et al., 2003) models and its variants. Both are probabilistic models that take into account that word occurrences in the same document often belong to the same topic. This is done by associating a topic distribution to every document, rather than having a single topic distribution for the whole corpus. The models thus consist of two types of probability distributions: (1) distributions of topics over documents (further *per-document topic distributions*) and (2) distributions of words over topics (further *per-topic word distributions*). After learning the topic model on a training corpus, the obtained per-topic word distributions can be used to infer per-document topic distributions on unseen documents. The important difference between pLSA and LDA is that the latter takes the Bayesian approach for modeling

the per-document topic distributions, i.e., the per-document topic distributions come from a Dirichlet-shaped prior distribution. pLSA in contrast uses point-estimates for the topic probabilities of documents, which makes it more vulnerable to overfitting. pLSA and LDA have found applications in document clustering, text categorization and ad-hoc information retrieval, but are not suited for cross-lingual text-mining since they were designed to work with monolingual data.

In multilingual settings, knowledge is mined from text by relying on machine-readable multilingual dictionaries or by using multilingual data. Since machine-readable dictionaries are not available for all languages pairs, the latter approach is more flexible. *Multilingual data* either refers to *parallel corpora* or *comparable corpora*. A parallel corpus is a collection of documents in different languages, where each document has a direct translation in the other languages. Hence, a parallel corpus is data-aligned at the sentence level. Parallel corpora are high-quality multilingual data resources, but they are not widely available for all language pairs and they are limited to a few narrow domains (e.g., the parliamentary proceedings of the Europarl corpus (Koehn, 2005)). Therefore, text mining from comparable corpora has gained interest over the last few years. A comparable corpus is a collection of documents with similar content which discusses similar themes in different languages, where documents in general are not exact translations of each other and are not strictly aligned at the sentence level. Unlike parallel corpora, comparable corpora by default comprise both shared and non-shared content.

A corpus built from Wikipedia using inter-wiki links to align content at the document level is a straightforward example of a comparable corpus, since the aligned article pairs may range from being almost completely parallel to containing non-parallel sentences. There are several other ways to acquire comparable corpora however. In the past years researchers have shown that comparable corpora can be automatically compiled from the Web. Utsuro et al. (2002) construct comparable corpora with document alignments from English and Japanese news websites. To obtain a collection of similar documents they look at the dates of the articles and they rely on a machine translation tool to find document alignments. Talvensaari et al. (2008) leverage the process of focussed crawling to obtain domain-specific comparable corpora with paragraph alignments. The method was applied to gather comparable corpora in the genomics domain, and it was shown to be superior to a (general) parallel corpus in finding genomics-related term translations. Apart from the resources we can find on the Web, organizations often possess domain-specific corpora which allow to construct comparable corpora. In recent work for example, English and Chinese discharge summaries were used to create a comparable corpus in the healthcare sector (Xu et al., 2015). For even more approaches towards constructing document-aligned comparable data, we refer the interested reader to the relevant literature (Utiyama and Isahara, 2003; Tao and Zhai, 2005; Vu et al., 2009). While comparable corpora are typically cheaper, more abundant, more easily obtainable and more versatile than parallel corpora, they also

constitute noisier and more challenging cross-lingual text mining environments.

Multilingual topic models such as bilingual LDA (BiLDA) (De Smet and Moens, 2009; Mimno et al., 2009) or Collaborative PLSA (C-PLSA) (Jiang et al., 2012) exploit the fact that the linked documents in multilingual corpora share content. These models assume that while the shared content is expressed with words from different vocabularies, the content can be represented in the same space of latent cross-lingual topics. Put differently, multilingual topic models learn cross-lingual topics which serve as a bridge between the different languages. The per-document word distributions constitute a language-independent document representation, while the language-specific information is modeled in per-topic word distributions. Topic models in this framework do not rely on sentence alignments, which makes them more robust to noisy data. However, the models assume that the topic distributions of linked documents are identical, which is not the case for comparable corpora.[1]

Contributions. The main contribution of this chapter is a novel multilingual topic model specifically tailored to deal with non-parallel data. This model called *comparable bilingual LDA (C-BiLDA)* may be observed as an extension of the BiLDA model. However, unlike BiLDA, which assumes that two documents in an aligned document pair (e.g., a pair of aligned Wikipedia articles) share their topics completely (i.e., by modeling only one shared topic distribution), our new C-BiLDA model allows a document to elaborate more on certain topics than the document in the other language to which it is linked.

As another contribution, we show how to utilize our C-BiLDA model in the task of *cross-lingual knowledge transfer for multi-class document classification* for three language pairs. We show results on two datasets for a C-BiLDA-based transfer model which outscores LDA- and BiLDA-based transfer models previously reported in the literature (De Smet et al., 2011; Ni et al., 2011).

3.2 Related Work

One line of work in multilingual topic modeling explores multilingual topic models that are based on the premise of using readily available machine-readable multilingual dictionaries to establish links between content given in different languages which are in turn necessary to extract these latent cross-lingual topics (Boyd-Graber and Blei, 2009; Jagarlamudi and Daumé III, 2010; Zhang et al., 2010; Boyd-Graber and Resnik, 2010; Hu et al., 2014). However, bilingual dictionaries are typically incomplete - if these are

[1] For instance, Wikipedia articles about Madrid in English and Spanish address many common topics such as “demographics”, “geography and location” or “climate”, while at the same time, only the Spanish article contains text (i.e., a non-shared topic) about “the emblems of the city”, and only the English article elaborates on “business schools” or “Bohemian culture” in Madrid.

available at all - as they often lack translations for domain-specific words. In contrast, a more flexible multilingual topic modeling framework attempts to extract these latent topics solely on the basis of given multilingual data without any external resources at all. Because of its higher flexibility and scalability, our model is situated within this modeling framework. The standard multilingual model within this framework is called *bilingual LDA* (BiLDA) (De Smet and Moens, 2009; Ni et al., 2009; Platt et al., 2010; Zhang et al., 2013) or, by its extension to more than two languages, *polylingual LDA* (Mimno et al., 2009; Krstovski and Smith, 2013).

All these models neglect one quite obvious fact - although dealing with comparable datasets which are inherently non-parallel and typically exhibit a degree of variance in their thematic/topical focuses, these models presuppose a perfect (or parallel) correspondence on extracted cross-lingual topics. More concretely, the models assume that the topic distributions of aligned documents are identical.

Aside from multilingual topic models, there are other approaches to mine cross-lingual word representations from multilingual corpora. Low-rank methods and neural net models are two other commonly used approaches. Low-rank methods use decompositions of co-occurrence matrices to find cross-lingual representations of words and/or documents. In multilingual text mining, cross-lingual latent semantic indexing (CL-LSI) and cross-lingual kernel canonical correlation analysis (CL-KCCA) are two established low-rank methods. Given a parallel corpus, CL-LSI (Littman et al., 1998) concatenates the aligned document pairs and then applies LSI to find cross-lingual representations. CL-KCCA was proposed as an alternative to CL-LSI by Vinokourov et al. (2002). After applying KCCA between the documents of source and target language respectively, semantic vectors for source and target language are constructed by projecting the respective document sets onto the k first correlation vectors. Each semantic vector corresponds to a cross-lingual topic. Documents can then be mapped to a cross-lingual representation by projecting their vector representation on the semantic vectors. Depending on its language, a document is projected on the semantic vectors of the source or target language. In the experiments of Vinokourov et al. (2002), CL-KCCA with a linear kernel outperformed CL-LSI in both cross-lingual information retrieval and document classification.

The main focus of the neural net models lies on learning distributed word representations (dense real-valued vectors), which are shared across languages, by optimizing some criteria as a function of the data and the output of a neural network for which the words serve as input. Klementiev et al. (2012) jointly train neural language models for two languages to induce shared cross-lingual distributed word representations. The neural language model learns distributed representations of words so that they can be used to predict the representation of the next word given the $n-1$ previous words. To jointly learn the language models the multi-task learning setup of Cavallanti et al. (2010) is used. Learning each vocabulary word in each language is considered a different task. To determine the degree of relatedness between two

corresponding tasks, the approach requires the availability of *hard word alignments*, that is, links between words in parallel documents, where linked words are (part of) each others translations. Kočiský et al. (2014) take a different approach and learn word representations that predict the representation of a word in the target language given $n-1$ words in a parallel sentence in the source language. Both approaches build document representations simply as (weighted) averages of word representations. Instead of predicting a single word, Chandar et al. (2014) learn to predict the bag-of-words representation of a target language sentence given the source language sentence.

Gouws et al. (2015) have proposed a multilingual extension of the well-known word2vec models (Mikolov et al., 2013a). Hermann and Blunsom (2014b,a) use a compositional vector model (CVM) to derive distributed representations for sentences and documents from distributed representations of words. The distributed representations are learned by minimizing the energy between the distributed representation of parallel sentences.[2] Soyer et al. (2015) also use a composition function to compose words to phrases and sentences. They optimize both a bilingual objective and a monolingual objective. The bilingual objective is to minimize the energy between aligned sentence pairs. The monolingual objective aims to enforce that the energy between a sentence and a sub-phrase of the sentence is smaller than the energy between a sentence and a randomly sampled sub-phrase.

All these neural network based approaches actually need a strong bilingual signal given by (at least) a parallel corpus of a significant size (typically Europarl) in order to mine the knowledge from comparable datasets. In this work, we significantly alleviate the requirements, as we explicitly model both the shared and non-shared content in a document pair without the need for parallel data. In other words, unlike all previous work, our new model aims to extract *cross-lingual topics directly from non-parallel data by distinguishing between shared and unshared content, without any additional resources such as readily available bilingual lexicons or parallel data*.

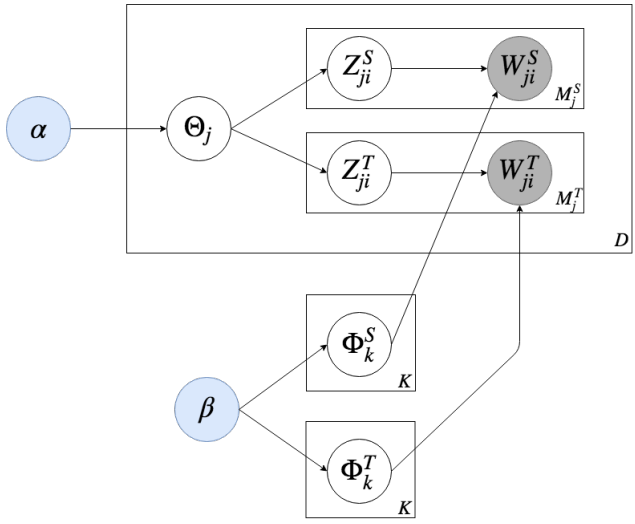
3.3 Comparable Bilingual LDA

This section provides a full description of the newly designed C-BiLDA model. First, we define the standard BiLDA model, detect its limitations, and then introduce our new model which is able to handle comparable data. We present its core modeling premises, its relation to BiLDA, its generative story, and its training procedure by Gibbs sampling. For a brief introduction to the monolingual LDA model, we refer the reader to Section 2.3.3 of the fundamentals chapter. In Table 3.1 we summarize the notation used throughout this chapter.

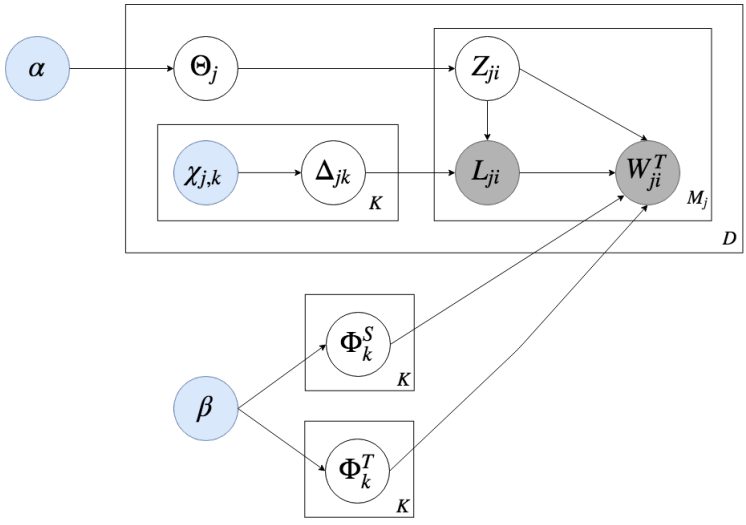
[2]The energy between two vectors X and Y is defined as $||X-Y||^2$.

	Documents, words and topics
D	number of aligned document pairs
$d_j = (d_j^S, d_j^T)$	j-th pair of aligned documents
M_j and M_j^S	number of words in document pair d_j and source language document d_j^S respectively.
V_S	vocabulary of the source language
$\|V\|^S$	size of the vocabulary of the source language
w_l^S	l-th word of the source language vocabulary
W_{ji}^S	i-th word token of d_j^S
$\mathbf{w}$	vector with all word tokens in the corpus
L_{ji}	language of the i-th word token of document pair d_j
$\mathbf{l}$	vector for which the i-th element is the language (L_S or L_T) of the i-th element in $\mathbf{w}$
$\mathscr{Z}$	set of latent cross-lingual latent topics
K	number of topics
z_k	k-th latent cross-lingual topic in $\mathscr{Z}$
Z_{ji}	topic assigned to the i-th word token of d_j
Z_{ji}^S	topic assigned to the i-th word token of d_j^S
$\mathbf{z}$	vector with all topic assignments in the corpus
	Distribution parameters
$\mathbf{\Theta}_j$	topic distribution of the document pair d_j
$\mathbf{\Theta}_j^S$	topic distribution of the source document d_j^S
Δ_{jk}	probability that an occurrence of topic z_k in document pair d_j is assigned to a word in the source document
Θ_{jk} and Θ_{jk}^S	probability that a word token in document pair d_j and document d_j^S respectively is assigned to topic z_k
$\mathbf{\Phi}_k^S$	distribution of the words in the source language for topic z_k
	Hyper-parameters
α	parameter value of the symmetric Dirichlet prior on all θ_j
β	parameter value of the symmetric Dirichlet prior on all ϕ_k
χ_{jk}^S, χ_{jk}^T	parameter values for the Beta prior on all Δ_{jk}
$\boldsymbol{\chi}_{jk}$	2-dimensional vector $(\chi_{jk}^S, \chi_{jk}^T)$
$\mathbf{\Omega}$	set of all hyper-parameters
	Gibbs counters
$n_{j,k}$	number of word tokens assigned to topic z_k in document pair d_j
$n_{j,k}^S$	number of word tokens assigned to topic z_k in document d_j^S
$n_{j,k,\neg i}$ or $n_{j,k,\neg i}^S$	number of word tokens assigned to topic z_k in document pair d_j or document d_j^S, excluding the word token at position i
$v_{k,l}^S$	number of times that word w_l^S is assigned to topic z_k.
$v_{k,\cdot,\neg ji}^S$	number of times that word w_l^S is assigned to topic z_k, not counting the word token at position i in document d_j^S.
$n_{j,\cdot}^S$ or $v_{\cdot,l}^S$	replacing a subscript variable with a dot means summing over all possible values of that variable, e.g. $n_{j,\cdot}^S = \sum_{k=1}^K n_{j,k}^S$

Table 3.1: A summary of the notation used throughout this chapter. For the language-specific notation we only show the notation for the source language (with the S superscript), while their counterpart in the target language is always obtained by replacing the S superscript with the T superscript.



(a) BiLDA



(b) C-BiLDA

Figure 3.1: Graphical representations of (a) BiLDA vs. (b) C-BiLDA in plate notation. BiLDA assumes that documents in an aligned document pair share all of their topics. Because of this assumption there is no need to represent the language l_{ji} of a topic occurrence. C-BiLDA on the other hand, allows the topic distributions of aligned documents to be different by assigning a language l_{ji} to every topic occurrence $z_{ji} = z_k$ depending on z_k: the source language is assigned to z_{ji} with probability Δ_{jk} and the target language with probability $1 - \Delta_{jk}$. M_j^S and M_j^T are the respective lengths of the source language document and the target language document in the j-th aligned document pair. M_j is the length of the document pair as a whole.

3.3.1 Bilingual Topic Modeling

Assume that we possess an *aligned bilingual document corpus*, which is defined as $\mathscr{C} = \{d_1, d_2, \ldots, d_D\} = \{(d_1^S, d_1^T), (d_2^S, d_2^T), \ldots, (d_D^S, d_D^T)\}$, where $d_j = (d_j^S, d_j^T)$ denotes a pair of aligned documents in the source language L_S and the target language L_T, respectively. D is the number of aligned document pairs in the bilingual corpus. The goal of bilingual probabilistic topic modeling is to learn for the bilingual corpus a set of K latent cross-lingual topics $\mathscr{Z} = \{z_1, \ldots, z_K\}$, each of which defines an associated set of words in both L_S and L_T (further denoted with superscripts S and T). A *bilingual probabilistic topic model* of a bilingual corpus $\mathscr{C}$ is a set of multinomial distributions of words with values $P(w_i^S|z_k)$ and $P(w_i^T|z_k)$, $w_i^S \in V_S$, $w_i^T \in V_T$, where V_S and V_T are vocabularies associated with languages L_S and L_T. The aligned documents in a document pair need not be the exact translation of each other, that is, the corpus may be comparable and consist of documents which are only loosely equivalent to each other (e.g., Wikipedia articles in two different languages, news stories discussing the same event).

Each document, regardless of its language, may be uniformly represented as a mixture over induced latent cross-lingual topics using the probability scores $P(z_k|d_j)$ from per-document topic-distributions. This topic model-based representation allows for representing documents written in different languages in the same shared "topical" cross-lingual space. Topic modeling also enables learning the same cross-lingual representation for unseen data by utilizing the per-topic word distributions from the trained model to infer per-document topic distributions on the new data.

The per-topic word and per-document topic distributions are learned in such a way so that they optimally explain the observed data. The exact calculation for this maximum likelihood criterion is intractable. Therefore, several approximate techniques have been proposed: Expectation-Maximization, variational Bayes, Gibbs sampling, etc. In this chapter we opt for the Gibbs sampling training technique, because of its popularity in literature and its ease of implementation. In its most general form, Gibbs sampling is a method to generate approximate samples from a joint distribution when directly sampling from the distribution is difficult or impossible (see also Section 2.2.4). Starting from a random initial state, the Gibbs sampling algorithm generates a sample from the distribution of each variable in turn, conditioned on the values of all other variables in the current state (Bishop, 2006). Because the initialization of the sampling chain is done randomly, the samples in the beginning of the process are not representative. Therefore we start collecting samples when the chain reaches a stationary state (after the so-called *burn-in* period). Since successive samples are highly dependent, we only collect a sample for the variables every I-th value (e.g., every 20-th value).

3.3.2 Bilingual LDA

Bilingual LDA (Ni et al., 2009; De Smet and Moens, 2009; Mimno et al., 2009; Platt et al., 2010; Zhang et al., 2013) assumes that aligned documents have exactly the same per-document topic distributions. The graphical representation of BiLDA is given in Figure 3.1a. The model uses the same $\mathbf{\Theta}_j$ to model per-document topic distributions of documents in a pair. For each document pair d_j, a shared per-document topic distribution $\mathbf{\Theta}_j$ is sampled from a (symmetric) conjugate Dirichlet prior with K parameters $\alpha_1, \ldots, \alpha_K$. Then, for each word position i in the source document of the current document pair d_j a cross-lingual topic z_k is sampled from $\mathbf{\Theta}_j$ (we denote this assignment by $z^S_{ji} = z_k$). Following that, a word is generated for every position i in document d^S_j by sampling from the multinomial distribution $\mathbf{\Phi}^S_k$ that corresponds to the topic z_k assigned to this position. Each word token W^T_{ji} from the target language side is also sampled following the same procedure, the only difference being that words are now sampled from the $\mathbf{\Phi}^T_k$ distributions. Note that words at the same positions in source and target documents in a document pair not need be sampled from the same latent cross-lingual topic. The only constraint imposed by the model is that the overall distributions of topics over documents in a document pair modeled by $\mathbf{\Theta}_j$ have to be the same. The validity of this assumption/constraint is dependent on the actual degree of thematic alignment between two coupled documents, and it perfectly fits only parallel documents which share all their topics. β is the parameter value of the symmetric Dirichlet prior on language-specific per-topic word distributions.

3.3.3 C-BiLDA: Extracting Shared and Non-Shared Topics

Modeling Assumptions. When one has to deal with a true comparable corpus, the assumption of "parallelism" exploited by BiLDA in its modeling premises is no longer valid, and it introduces several points of noise in the final output. As the same topics with the same proportions are used in both documents of a pair, there exists a clear discrepancy between learned topics and the actual content. In order to deal with the added difficulties caused by the "comparability" of the corpus and given document pairs, we extend the basic bilingual LDA model from sect. 3.3.2.

C-BiLDA allows a document to focus more on some topics than its counterpart in the other language by modeling the probability that a topic occurrence in a document pair belongs to the source document. To this end we explicitly model the language L_{ji} for every word occurrence W_{ji} as an observed random variable and for each document introduce K parameters Δ_{jk} describing the probability that a topic occurrence $z_{ji} = z_k$ in document pair d_j generates a word in the source language.

Algorithm 2: C-BILDA: GENERATIVE STORY

initialize: (1) the total number of topics: K;
(2) the values for Dirichlet priors parameters α and β;
(3) the values of all χ^S_{jk} and χ^T_{jk} (in Figure 3.1b we use χ_{jk} as an abbreviation for $< \chi^S_{jk}, \chi^T_{jk} >$)

sample K times $\Phi^S_k \sim Dirichlet(\beta)$;
sample K times $\Phi^T_k \sim Dirichlet(\beta)$;
for $j \leftarrow 1$ **to** D **do**
  sample $\Theta_j \sim Dirichlet(\alpha)$
  sample K times $\Delta_{jk} \sim Beta(\chi^S_{jk}, \chi^T_{jk})$
  for $i \leftarrow 1$ **to** M_j **do**
    sample $Z_{ji} \sim Multinomial(1, \Theta_j)$
    sample $L_{ji} \sim Bernoulli(\Delta_{jk})$, with $Z_{ji} = z_k$
    if $L_{ji} = 1$ **then**
      sample $W_{ji} \sim Multinomial(1, \Phi^S_k)$, with $Z_{ji} = z_k$
    else
      sample $W_{ji} \sim Multinomial(1, \Phi^T_k)$, with $Z_{ji} = z_k$

Generating the Data. Figure 3.1b shows the plate representation of C-BiLDA. As in the BiLDA generative process, all topics of a document pair are drawn from the same distribution Θ_j , but source and target documents can have a preference to certain topics. After generating a topic $Z_{ji} = z_k$ from Θ_j, we sample the language l_{ji} associated with this topic occurrence from a Bernoulli distribution $Bernoulli(\Delta_{jk})$, where Δ_{jk} is the probability that topic z_k will generate a word in the source language for document pair j. We place a Beta prior with parameter values χ^S_{jk} and χ^T_{jk} on all Δ_{jk}. These values can be interpreted as pseudo-counts for observing topic z_k in the source/target document of document pair d_j respectively. After sampling a topic-language pair, a word is generated in the same way as in the BiLDA model, that is, by sampling from the word distribution of the sampled topic in the sampled language. The distributions Θ_j, Φ^S_k, Φ^T_k and corresponding hyper-parameters α and β are the same as in BiLDA (see sect. 3.3.2). Alg. 2 summarizes the generative story of C-BiLDA.

Relation with BiLDA. In its original formulation BiLDA looks quite different from C-BiLDA. This is because with the BiLDA assumptions, it is not necessary to model the language of a word as a random variable. However, we can represent BiLDA exactly like C-BiLDA with the exception of using a single Δ_j (representing the probability that any topic will generate a word in the source document) per document, instead of using K Δ_{jk} variables per document (one for each topic)[3]. Therefore, C-BiLDA allows

[3] By writing out the joint probability conditioned on all language assignments L_{ji}, one can check that these formulations are indeed equivalent.

a document to focus more on a particular topic than its counterpart or, in the extreme case, to contain topics that do not occur in its counterpart. The added flexibility also has a downside since it increases the risk of overfitting the data. By setting an appropriate prior on all Δ_{jk} variables, we can avoid that C-BiLDA learns models that are too complex. By setting the prior values of $\chi^S_{j1},...,\chi^S_{jK}$ to the same value and similarly for the values of $\chi^T_{j1},...,\chi^T_{jK}$, we make the a priori assumption that the topic distributions for source and target document are identical (like in BiLDA). In our experiments we set $\chi^S_{jk} = \frac{1}{2}\chi_m M^S_j$ and $\chi^T_{jk} = \frac{1}{2}\chi_m M^T_j$. The document sizes M^S_j and M^T_j are observed, so only the value of χ_m must be set manually. The higher the value of χ_m, the more weight we give to the prior assumption that the source and target document topic distributions are the same, and the closer the C-BiLDA relates to BiLDA.

Training. To infer the values of the unobserved variables, we utilize Gibbs sampling (Geman and Geman, 1984; Bishop, 2006). Note that from the vector of all topic assignments: $\mathbf{z}$ together with the observed word and language variables, all other latent variables can be derived. The values of all $\mathbf{\Theta}_j$, Δ_{jk}, $\mathbf{\Phi}^S_k$ and $\mathbf{\Phi}^T_k$ can be integrated out of the formulas and calculated afterwards. All other variables are observed (the word tokens in the bilingual corpus and their corresponding languages) or are hyper-parameters that have to be set in advance (α, all χ_{jk} and β). We can therefore use a collapsed Gibbs sampler that in each iteration samples the topic assignments for each word in turn from their probability distribution conditioned on all other variables. The high-level Gibbs sampling procedure for C-BiLDA is shown in alg. 3, below we derive the necessary update formulas for Z_{ji}.

$$
\begin{aligned}
&P(Z_{ji} = z_k | W_{ji} = w_l, l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{w}_{\neg ji}, \mathbf{l}_{\neg ji}, \Omega) \\
&= \frac{P(Z_{ji} = z_k, W_{ji} = w_l, l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{w}_{\neg ji}, \mathbf{l}_{\neg ji}, \Omega)}{P(W_{ji} = w_l, l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{w}_{\neg ji}, \mathbf{l}_{\neg ji}, \Omega)} \\
&= \frac{P(W_{ji} = w_l, \mathbf{w}_{\neg ji} | Z_{ji} = z_k, l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{l}_{\neg ji}, \Omega) \cdot P(Z_{ji} = z_k, l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{l}_{\neg ji} | \Omega)}{P(\mathbf{w}_{\neg ji} | l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{l}_{\neg ji}, \Omega) \cdot P(w_{ji} | l_{ji}, \Omega) \cdot P(l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{l}_{\neg ji} | \Omega)} \\
&\propto P(W_{ji} = w_l | Z_{ji} = z_k, l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{w}_{\neg ji}, \mathbf{l}_{\neg ji}, \Omega) \cdot P(Z_{ji} = z_k | l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{l}_{\neg ji}, \Omega) \\
&\propto P(W_{ji} = w_l | Z_{ji} = z_k, l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{w}_{\neg ji}, \mathbf{l}_{\neg ji}, \Omega) \cdot \frac{P(Z_{ji} = z_k, l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{l}_{\neg ji} | \Omega)}{P(l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{l}_{\neg ji}, \Omega)} \\
&\propto P(W_{ji} = w_l | Z_{ji} = z_k, l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{w}_{\neg ji}, \mathbf{l}_{\neg ji}, \Omega) \cdot \frac{P(l_{ji}, \mathbf{l}_{\neg ji} | Z_{ji} = z_k, \mathbf{z}_{\neg ji}, \Omega) \cdot P(Z_{ji} = z_k, \mathbf{z}_{\neg ji} | \Omega)}{P(\mathbf{l}_{\neg ji} | \mathbf{z}_{\neg ji}, \Omega) \cdot P(\mathbf{z}_{\neg ji} | \Omega)} \\
&\propto P(W_{ji} = w_l | Z_{ji} = z_k, l_{ji}, \mathbf{z}_{\neg ji}, \mathbf{w}_{\neg ji}, \mathbf{l}_{\neg ji}, \Omega) \cdot P(l_{ji} | z_{ji} = z_k, \mathbf{z}_{\neg ji}, \mathbf{l}_{\neg ji}, \Omega) \cdot P(z_{ji} = z_k | \mathbf{z}_{\neg ji}, \Omega)
\end{aligned}
$$

Algorithm 3: GIBBS SAMPLING FOR C-BILDA: AN OVERVIEW

Algorithm `gibbsSampler()`
  repeat
    `sampleTopics();`
  until *burn-in criterion satisfied*
  repeat
    for $i \leftarrow 1$ **to** I **do**
      `sampleTopics();`
    end
    collect a sample: estimate Θ_{jk}, Δ_{jk}, Φ^S_{kl}, Φ^T_{kl} *from the current topic assigments using Equation (3.6)-(3.9)*;
  until *enough samples collected*
  estimate the posteriors of Θ_{jk}, Δ_{jk}, Φ^S_{kl}, Φ^T_{kl} *by averaging over the collected samples*;
Procedure `sampleTopics()`
  foreach *word token in the corpus* **do**
    update/estimate *the probability to assign the word token to one of the cross-lingual topics conditioned on all other variables (for C-BiLDA apply Equation (3.1))*;
    sample *a new topic assignment for the word token*;
  end

$$\propto \begin{cases} E[\Theta_{jk}|\mathbf{z}_{\neg ji},\alpha]\cdot E[\Delta_{jk}|\mathbf{z}_{\neg ji},\mathbf{l}_{\neg ji},\chi^S_{jk},\chi^T_{jk}]\cdot E[\Phi^S_{kl}|\mathbf{z}_{\neg ji},\mathbf{l}_{\neg ji},\mathbf{w}_{\neg ji},\beta] & \text{if } l_{ji}=S \\ E[\Theta_{jk}|\mathbf{z}_{\neg ji},\alpha]\cdot (1-E[\Delta_{jk}|\mathbf{z}_{\neg ji},\mathbf{l}_{\neg ji},\chi^S_{jk},\chi^T_{jk}])\cdot E[\Phi^T_{kl}|\mathbf{z}_{\neg ji},\mathbf{l}_{\neg ji},\mathbf{w}_{\neg ji},\beta] & \text{if } l_{ji}=T \end{cases} \tag{3.1}$$

$$\text{with } E[\Theta_{jk}|\mathbf{z}_{\neg ji},\alpha] = \frac{n_{j,k,\neg i}+\alpha}{n_{j,\cdot,\neg i}+K\alpha} \tag{3.2}$$

$$\text{and } E[\Delta_{jk}|\mathbf{z}_{\neg ji},\mathbf{l}_{\neg ji},\chi^S_{jk},\chi^T_{jk}] = \frac{n^S_{j,k,\neg i}+\chi^S_{jk}}{n_{j,k,\neg i}+\chi^S_{jk}+\chi^T_{jk}} \tag{3.3}$$

$$\text{and } E[\Phi^S_{kl}|\mathbf{z}_{\neg ji},\mathbf{l}_{\neg ji},\mathbf{w}_{\neg ji},\beta] = \frac{v^S_{k,\cdot,\neg ji}+\beta}{v^S_{k,\cdot,\neg ji}+|V|^S\cdot\beta} \tag{3.4}$$

$$\text{and } E[\Phi^T_{kl}|\mathbf{z}_{\neg ji},\mathbf{l}_{\neg ji},\mathbf{w}_{\neg ji},\beta] = \frac{v^T_{k,\cdot,\neg ji}+\beta}{v^T_{k,\cdot,\neg ji}+|V|^T\cdot\beta} \tag{3.5}$$

The final estimates of the posteriors of Θ_{jk}, Δ_{jk}, Φ^S_{kl} and Φ^T_{kl} are calculated by estimating their posteriors for every sample that is taken using equations (3.6)-(3.9) and then taking the average of these estimates over all samples.

$$E[\Theta_{jk}|\mathbf{z},\alpha] = \frac{n_{j,k}+\alpha}{n_{j,\cdot}+K\alpha} \tag{3.6}$$

$$E[\Delta_{jk}|\mathbf{z},\mathbf{l},\chi^S_{jk},\chi^T_{jk}] = \frac{n^S_{j,k}+\chi^S_{jk}}{n_{j,k}+\chi^S_{jk}+\chi^T_{jk}} \tag{3.7}$$

$$E[\Phi^S_{kl}|\mathbf{z},\mathbf{l},\mathbf{w},\beta] = \frac{v^S_{k,l}+\beta}{v^S_{k,\cdot}+|V|^S\cdot\beta} \tag{3.8}$$

$$E[\Phi^T_{kl}|\mathbf{z},\mathbf{l},\mathbf{w},\beta] = \frac{v^T_{k,l}+\beta}{v^T_{k,\cdot}+|V|^T\cdot\beta} \tag{3.9}$$

Inferring topic distributions. For certain tasks (e.g., information retrieval) it is necessary to infer a topic model on unseen data. Inferring the model actually denotes calculating per-document topic distributions on unseen documents based on the output of the trained model. Again, we use Gibbs sampling to approximate the distribution, but now we use the per-topic word distributions that were learned from the training dataset. Therefore, we only update the n counters. Furthermore, the inference is done monolingually, that is one language at a time. The updating formula for the source language L_S is:

$$\begin{aligned} &P(z^S_{ji}=k|w^S_{ji}=w_l,\mathbf{z}^S_{\neg ji},\mathbf{w}^S_{\neg ji},\alpha,\beta) \propto E[\Theta_{jk}|\mathbf{z}^S_{\neg ji}]\cdot E[\Phi^S_{kl}|\text{training data}] \\ &\text{with } E[\Theta_{jk}|\mathbf{z}^S_{\neg ji}] = \frac{n^S_{j,k,\neg i}+\alpha}{n^S_{j,\cdot,\neg i}+K\alpha} \end{aligned} \tag{3.10}$$

Where the n counters count topic assignments for *unseen documents* and $E[\Phi^S_{kl}|\text{training data}]$ is the estimate of Φ^S_{kl} on the training data.

3.4 Knowledge Transfer via Cross-Lingual Topics for Document Classification

The per-topic word distributions of multilingual topic models can be used for a variety of tasks. One application is to map the distributions to per-word distributions, e.g., $P(z_k|w_i)$ or $P(w_i,w_j)$. This results in a type of distributed word representation for w_i, which in turn can be used to find word associations and/or extract translation pairs, etc. (Vulić et al., 2011). In this work, we demonstrate the utility of our new

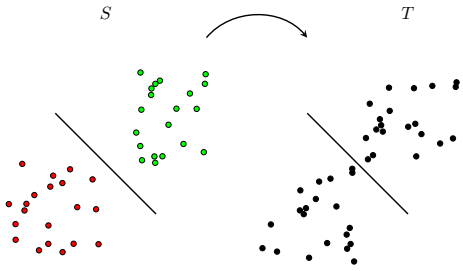


Figure 3.2: An intuition behind cross-lingual knowledge transfer for document classification. Green and red circles denote labeled examples, while black circles denote unlabeled examples.

C-BiLDA model on yet another task: cross-lingual document classification, as it is a well-established cross-lingual task that gives insight into cross-lingual text mining models and their ability to learn semantically-aware document representations.

Problem Definition. Cross-lingual document classification (CLDC) is the task of assigning class labels to documents written in the target language given the knowledge of the labels in the source language (Bel et al., 2003; Gliozzo and Strapparava, 2006). It starts from a set of labeled documents in the (resource-rich) source language, and unlabeled documents in the target language. The objective is to learn a classification model from the labeled documents of the source language and then *transfer this knowledge* to the target language and apply it in the classification model for the target language documents (see Figure 3.2 for a more intuitive presentation).

Previous Work. Early approaches to the problem of CLDC tried to utilize automatic machine translation tools to translate all the data from S to T, which effectively reduced the problem to monolingual classification (Bel et al., 2003; Fortuna and Shawe-Taylor, 2005; Olsson et al., 2005; Rigutini et al., 2005; Ling et al., 2008; Wei and Pal, 2010; Duh et al., 2011; Wan et al., 2011). Other approaches rely on machine translation tools along with multi-view learning (Amini et al., 2009; Guo and Xiao, 2012a) or co-training techniques (Wan, 2009; Amini and Goutte, 2010; Lu et al., 2011). However, machine translation tools may not be freely available for many language pairs, which limits the portability of these models. In addition, translating all the text data is often time-consuming and expensive.

Another line of prior work aims to induce cross-lingual representations for documents given in different languages, which enables the knowledge transfer for CLDC using the shared language-independent feature spaces. A plethora of CLDC models have been proposed (Gliozzo and Strapparava, 2006; Prettenhofer and Stein, 2010; Pan et al.,

Table 3.2: Statistics of the Wikipedia and Europarl training sets.

	Wikipedia-dataset			Europarl-dataset		
	EN-ES	EN-FR	EN-IT	EN-ES	EN-FR	EN-IT
$\|V^S\|$	29,201	27,033	23.346	33,444	33,574	33,552
$\|V^T\|$	27,745	20,860	31.388	36,839	34,538	36,092
#Doc-pairs	18.672	18.911	18.898	9,415	9,428	9,461

2011; Wang et al., 2011; Klementiev et al., 2012; Guo and Xiao, 2012b; Xiao and Guo, 2013a,b; Hermann and Blunsom, 2014a; Chandar et al., 2014), but all these models again assume that parallel corpora or external translation resources are readily available to induce these cross-lingual shared representations.

Finally, in order to overcome these issues, another line of recent work (De Smet et al., 2011; Ni et al., 2011) operates in a minimalist setting; it aims to learn these *shared cross-lingual representations directly from non-parallel data* without any other external resources such as high-quality parallel data or machine-readable bilingual lexicons. These approaches train a multilingual topic model (e.g., BiLDA) on *comparable data* to induce topical representations of documents, and use per-document topic distributions as classification features. In this thesis, we show that for this setup the application of C-BiLDA instead of BiLDA leads to a better performance.

Knowledge Transfer via Latent Topics. The idea is to take advantage of the cross-lingual representations by means of latent cross-lingual topics. First, a topic model (e.g., BiLDA or C-BiLDA) is trained on a bilingual training corpus (e.g., Wikipedia). Following that, given a CLDC task, with a labeled set of documents in the source language and an unlabeled document collection in the target language, one uses the trained topic model to infer the cross-lingual representations by means of per-document topic distributions for each (previously unseen) document. Each document is then taken as a data instance in the classification model and the features are defined as probabilities coming from per-document topic distributions. The value of each feature of an instance (e.g., a document d_j^S) is the probability of the corresponding topic z_k in the document: $P(z_k|d_j^S)$ (see Section 3.3.1). The assumption of the cross-lingual knowledge transfer is that when we express the documents in the cross-lingual space, the training examples in the source language are representative for the test data in the target language. Finally, one is free to choose any classifier (e.g., Maximum Entropy, Naive Bayes, Support Vector Machine) to perform classification.

3.5 Experimental Setup

Training Datasets. To train the topic models on a comparable corpus, we use the training dataset of De Smet et al. (2011) for the same CLDC task (while the dataset from (Ni et al., 2011) is not publicly available). It consists of three bilingual corpora with aligned Wikipedia articles in three language pairs: English-Spanish (EN-ES), English-French (EN-FR), and English-Italian (EN-IT). The datasets were collected from Wikipedia *dumps*, and the alignment between articles in a pair was obtained by following the inter-lingual Wikipedia links. Stop words were removed using the stopword lists of the Snowball project,[4] and only words that occur at least 5 times were retained. To show the influence of the degree of parallelism in the training data, we also train C-BiLDA and BiLDA on a parallel corpus extracted from Europarl.[5] The resulting dataset uses the same language pairs as the Wikipedia dataset and the processing was done in the same way. Table 3.2 lists statistics of the training datasets.

CLDC Datasets. We test our models by performing CLDC on two different datasets. We run the trained topic models on these test datasets, that is, we infer the per-document topic distributions, which are then used for training and testing a classifier. In all experiments, we regard English as the resource-rich language and learn class labels for test documents in the other 3 target languages (ES/FR/IT) with labels removed from their documents.

The first dataset again comes from De Smet et al. (2011). It was constructed using Wikipedia. The dataset for each language pair contains up to 1,000 Wikipedia articles (which are not present in the training sets) annotated with 5 high-level labels/classes: *book* (books), *film* (films), *prog* (computer programming), *sport* (sports) and *video* (video games). Since not every Wikipedia in every language contains the same number of articles, sometimes less than 1,000 articles for each class was crawled from Wikipedia *dumps*. For more details about the dataset construction, we refer the interested reader to (De Smet et al., 2011).

To compare the BiLDA and C-BiLDA models on a larger corpus we constructed a second dataset from the Reuters corpora RCV1/RCV2 (Lewis et al., 2004). The dataset contains up to 30,000 documents per language. Since our training dataset does not include the English-German language pair that was used by Klementiev et al. (2012), we could not reuse their dataset. We constructed the dataset with the procedure from Klementiev et al. for the three language pairs in our training dataset: we use the top-level category labels that are assigned to the documents: *CCAT* (Corporate/Industrial), *ECAT* (Economics), *GCAT* (Government/Social), *MCAT* (Markets); and only consider

[4] http://snowball.tartarus.org/

[5] For each language pair (EN-ES, EN-FR, EN-IT) we used all parallel document pairs present in the Europarl v7 source release. The .txt files can be aligned across languages based on their filename.

Table 3.3: Number of documents in the CLDC datasets.

	Wikipedia-dataset					RCV1/RCV2-dataset			
	book	*film*	*prog*	*sport*	*video*	*MCAT*	*CCAT*	*GCAT*	*ECAT*
EN	1,000	1,000	1,000	1,000	1,000	7,441	12,934	7,216	2,409
ES	1,000	1,000	263	1,000	1,000	9,694	30	1,997	1279
FR	1,000	1,000	592	1,000	1,000	5,878	65	20,987	3,070
IT	1,000	1,000	290	1,000	764	7,553	263	1,520	3,664

documents with a single top-level topic. Similar to Klementiev et al. (2012), we sample randomly from the original RCV1/RCV2 corpora, but for the language pairs in our training dataset. The documents from both datasets were preprocessed in the same manner as in the training datasets. Table 3.3 displays the size of the CLDC datasets.

Models in Comparison. We test the ability of our new C-BiLDA model to transfer the knowledge needed for cross-lingual document classification, and compare it to other topic modeling approaches for knowledge transfer previously reported in the literature. The models in comparison are:

1. *CL-LSI-TR*. A CLDC model based on CL-LSI (Littman et al., 1998). In order to come up with uniform cross-lingual representations, it combines each aligned pair of documents into an artificial "merged document", keeping no language-specific information. On the merged documents (monolingual) LSI is applied. The rank reduced term-document matrix (where the new rank is equal to the number of topics) is then used to project the documents in the cross-lingual space in which we train the classifier.
2. *CL-KCCA-TR*. This model is based on the CL-KCCA model of Vinokourov et al. (2002). The semantic vectors of the source/target language are used to project documents of the source/target respectively in the cross-lingual space in which we train the classifier. Like Vinokourov et al. (2002) we use a linear kernel.
3. *LDA-TR*. This was the baseline model in (De Smet et al., 2011). Similar to CLLSI-TR it combines each aligned pair of documents into an artificial "merged document". The merged documents are then used to train a monolingual LDA (Blei et al., 2003) model, which is then inferred on the test documents. Per-document topic distributions are then used as features for classification.
4. *BiLDA-TR*. This is the best scoring model in De Smet et al. (2011) and Ni et al. (2011). It also significantly outperformed models relying on machine translation tools and bilingual lexicons (Ni et al., 2011). BiLDA is trained on aligned documents and then inferred on test data. Per-document topic distributions are again used as features for classification (see Section 3.4).
5. *C-BiLDA-TR-*χ_m, with $\chi_m \in \{0.125, 0.25, 0.5, 1, 2\}$. As for BiLDA, we train C-BiLDA on aligned document pairs to obtain per-document topic distributions. We use different values of χ_m (recall from sect. 3.3.3 that χ_m determines the

values of the prior parameters χ_{jk}^{S} and χ_{jk}^{T}).

Parameters. Following prior work, we use a Support Vector Machine (SVM) for classification with all transfer models. For SVM, we employ the SVM-Light package[6] (Joachims, 1999) with default parameter settings. Investigating other choices for classifiers, as well as different classifier settings is beyond the scope of this chapter. All models are trained for different number of topics K, ranging from 20 to 300 in steps of 20. CL-LSI was implemented using the *truncated SVD* module of scikit-learn[7] (Pedregosa et al., 2011). For CL-KCCA we used KCCA package by Hardoon et al. (2004). The regularization parameter κ was set using the method proposed in Hardoon et al. (2004).

Hyper-parameters α and β in LDA and BiLDA are set to the standard values according to (Steyvers and Griffiths, 2007): $\alpha = 50/K$ and $\beta = 0.01$. In the case of C-BiLDA, we show the results for different values of χ_m: $\{0.125, 0.25, 0.50, 1, 2\}$. The higher the χ_m value, the higher the influence of the priors on δ_{jk}. The topic models have been trained by Gibbs sampling. As the burn-in criterion, we check if the relative difference of the perplexity between two iterations is smaller than a predefined small threshold value (we use 0.0001 in all training procedures). After the burn-in period, we gather samples every $I = 20$ iterations. The total number of iterations (including the burn-in period) is set to 1000. Perplexity is a measure for the likelihood of the data for a given statistical model. The perplexity on a corpus $\mathscr{C}$ for a statistical model $\mathscr{M}$ which like our topic models assumes documents can be represented as a bag of words is defined as:

$$perplexity(\mathscr{C}|\mathscr{M}) = exp\Big(-\frac{\sum_{j=1}^{D}\sum_{i=1}^{M_j} log\big(p(w_{ji}|\mathscr{M})\big)}{\sum_{j=1}^{D} M_j}\Big)$$

Evaluation Metrics. For each category, precision is calculated as the number of correctly labeled documents divided by the total number of documents that have been labeled this way. Recall is defined as the number of correctly labeled documents divided by the actual number of documents with that label given by the ground truth. Precision and recall are then combined into balanced *F-1 scores*. We calculate macro F-1 scores by taking the average of the F-1 scores over all categories and all Ks. For BiLDA and C-BiLDA, we also report the perplexities on the training datasets. Perplexity measures how well a statistical model fits the data.

[6] http://svmlight.joachims.org/
[7] http://scikit-learn.org/

Table 3.4: Perplexity scores of the BiLDA and C-BiLDA models and their difference (the perplexity score of BiLDA minus the perplexity score of C-BiLDA) on the Wikipedia training datasets averaged across the number of topics and χ_m values. From the perplexity scores and the difference in perplexity scores of C-BiLDA and BiLDA we can rank the training datasets according to their level of parallelism.

	Wikipedia			Europarl		
	EN-ES	EN-FR	EN-IT	EN-ES	EN-FR	EN-IT
perpl. BiLDA	2827	2544	3042	1564	1391	1600
perpl. C-BiLDA	2787	2504	2839	1581	1402	1615
perpl. BiLDA - perpl. C-BiLDA	40	40	203	-17	-11	-15

3.6 Results and Discussion

Perplexity and Comparability. In this paragraph we analyze the perplexity of C-BiLDA and BiLDA on the different training datasets. Table 3.4 shows average perplexity scores of C-BiLDA and BiLDA models trained on the parallel Europarl corpora and the comparable Wikipedia corpora. The perplexity scores confirm our hypothesis that BiLDA is more suited for modeling parallel data, while C-BiLDA is tailored for more divergent, comparable data.

In Table 3.4 we also show the difference in perplexity between the two models: perplexity BiLDA – perplexity C-BiLDA. We expect this difference to be an indicator of the degree of comparability of a multilingual corpus. The larger the difference between the perplexity of BiLDA and the perplexity of C-BiLDA models, the less parallelism we expect to find in the data because we expect C-BiLDA to model non-parallelism in a better way. The results in Table 3.4 confirm this hypothesis since on the comparable Wikipedia dataset the difference in perplexity values is higher than for the parallel Europarl datasets. The results also indicate that the EN-IT Wikipedia dataset is less parallel than the EN-FR and EN-ES Wikipedia datasets since the difference in perplexity is larger. For the Wikipedia datasets the overall perplexity is higher for EN-ES than for EN-FR. This is an indication that the latter is the Wikipedia dataset with the most parallelism.

CLDC Task. Table 3.5 summarizes the performance in the CLDC task of the transfer models (TRs) with representations trained on Wikipedia. F-1 scores are macro-averaged over different category labels and averaged over different Ks. Table 3.5 also ranks the training datasets in their degree of comparability, based on the perplexity analysis in the previous paragraph. Figure 3.3 shows how F-1 scores fluctuate on the Reuters test dataset across different K values for BiLDA and C-BiLDA with $\chi_m = 2$. From these results we may observe several interesting phenomena:

(i) The difference between LDA on one side and BiLDA and C-BiLDA is very profound. While all these transfer models are based on the same principle and use per-document topic distributions to provide language-independent document representations, separating the vocabularies and training a true bilingual topic model on individual documents from aligned pairs (instead of removing all language information from the corpus) is clearly more beneficial for the CLDC task. Similar findings have been reported for cross-lingual information retrieval (Jagarlamudi and Daumé III, 2010; Vulić et al., 2013) and word translation identification (Vulić et al., 2011, 2015).

(ii) Also the difference between the low-rank approximation methods (CL-LSI, CL-KCCA) on one side and C-BiLDA and BiLDA is profound. An explanation for this may be that the use of priors in the probabilistic framework is a robust way to deal with the non-parallelism in comparable corpora.

(iii) When comparing BiLDA with the C-BiLDA transfer models we see that the C-BiLDA models generally perform better. For the CLDC task on the Wikipedia test set, both the C-BiLDA transfer models and the BiLDA transfer model have good F-1 scores, indicating that the models learn representations that are well suited for the Wikipedia test set. The differences between the C-BiLDA and BiLDA models are not so profound as for the Reuters test set. After performing a qualitative inspection of the topic distributions, we conclude there is a clean mapping between the topics we learned from our training data and the categories of the Wikipedia dataset. The representations of the categories of the Reuters dataset, on the other hand, are more spread out across topics. In the latter case, it is more important to have more clean/coherent topics overall. Therefore, we conclude that C-BiLDA is able to learn "cleaner" per-topic word distributions.

(iv) We observe that for the language pair with the least comparable training data, the C-BiLDA transfer models perform better than the BiLDA model and that the C-BiLDA models with lower χ_m values perform best (recall that a lower χ_m value in fact implies assigning less weight to the a priori *parallel document pair assumption*, see Section 3.3.3). On the other hand, for the EN-FR language pair we observe that the difference between C-BiLDA and BiLDA is less profound and that the higher values for the χ_m parameter perform best. This intuition underpinned by the reported results reveals a link between the comparability of the training data and the performance of the BiLDA model and the C-BiLDA models with different χ_m.

(v) From Figure 3.3 we conclude that the difference between the C-BiLDA transfer model with $\chi_m = 2$ and the BiLDA transfer model are consistent for the lower topic values. For the higher topic values performance begins to drop. This illustrates previously mentioned overfitting problems. More topics lead to more model parameters, for C-BiLDA even more so than BiLDA.

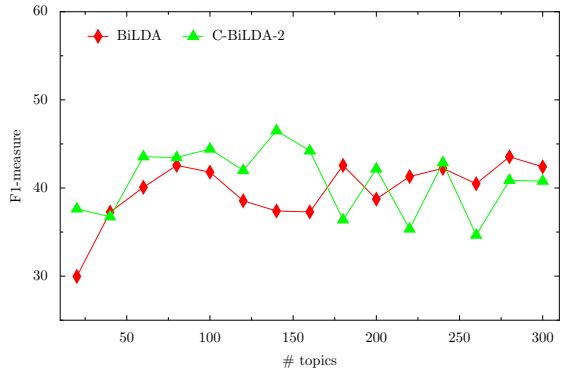
Table 3.5: CLDC with representations trained on Wikipedia. Average F-1 scores on the Wikipedia and Reuters test sets with 8 different transfer models for each language pair. Average F-1 is calculated by macro-averaging the F-1 scores over all category labels and all Ks. The classifier is SVM. A + sign indicates a better F-1 score of a C-BiLDA-TR when compared to the baseline models. The best F-1 scores per language pair are shown in bold.

	EN-ES		EN-FR (most parallel)		EN-IT (least parallel)	
TR-Model	Wiki	Reuters	Wiki	Reuters	Wiki	Reuters
CL-LSI	31.17	27.59	28.44	35.35	26.79	21.06
CL-KCCA	14.03	14.12	24.03	24.28	10.21	8.91
LDA	32.84	7.55	34.65	10.08	30.99	26.86
BiLDA	81.46	39.74	**76.88**	45.30	78.36	45.22
C-BiLDA$_2$	**81.51**+	**40.77**+	76.61	**45.63**+	78.83+	46.21+
C-BiLDA$_1$	80.64	40.27+	74.47	44.92	**79.27**+	45.66+
C-BiLDA$_{0.5}$	80.83	39.70	76.19	45.30	79.09+	**46.76**+
C-BiLDA$_{0.25}$	79.71	40.41+	75.03	44.48	79.06+	46.08+
C-BiLDA$_{0.125}$	79.91	40.41+	75.37	44.02	78.85+	45.60+

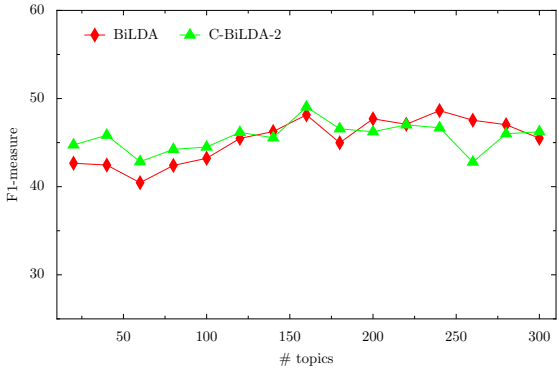
Further Discussion. One may argue that capturing additional phenomena in the data (e.g., document pairs with non-parallel document distributions) leads to an added complexity in the model design. However, the increased design complexity is justified by the need to capture the properties of non-parallel data. Consequently, the final scores in the CLDC task further justify the requirement for a more complex topic model which is better aligned with the given data.

We have reported that the priors placed on the Δ_{jk} variables have a significant influence on the quality of the learned topics. Their values should be high enough to avoid overfitting, though low enough to take into account non-parallelism (i.e., non-shared content) in document pairs. It may be too time/resource consuming to explore what values for the χ priors are appropriate by trying different values and finding out which work best. One approach we intend to investigate in future work is to treat the hyper-parameters as random variables that are learned from the data just like the other parameters. Wallach et al. (2009) have successfully applied this approach to the α hyper-parameters for monolingual LDA.

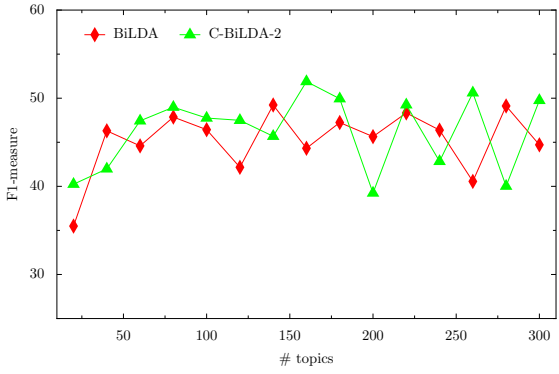
So far we have not talked about the minimum degree of comparability between the corpora in order to learn any useful bilingual knowledge. This is a difficult question in general. For C-BiLDA in particular, the document pairs may exhibit low comparability in case the following conditions hold for the document collection as a whole: (1) the document collection should contain enough cross-lingual information, this means that as the comparability between document pairs goes down, the size of the document collection should go up accordingly; (2) if a theme often reoccurs in the documents



(a)



(b)



(c)

Figure 3.3: The average F-1-scores for a varying amount of topics for the BiLDA transfer model and the C-BiLDA transfer model with $\chi_m = 2$ on the CLDC task with the Reuters dataset: EN-ES (a), EN-FR (b) and EN-IT (c).

of the source language, it should often occur in the documents of the target language. This requirement can be fulfilled by ensuring that the document collection is restricted to a limited domain.

Besides the CLDC task, we believe that the proposed C-BiLDA model and the idea of distinguishing between shared and unique content in related documents may find further application in other tasks. One interesting application is tackled in (Paul and Girju, 2009), where they analyze cultural differences between speakers of the same language across different countries and cultures. A similar idea applied to the analysis of ideological differences is discussed in (Ahmed and Xing, 2010). Another interesting future application is the analysis of differences between Twitter and traditional media (Zhao et al., 2011). The C-BiLDA model and its extensions in future research may be utilized to induce different views on the same subjects/concepts/topics given in different languages and/or in different media, as well as to extract language-specific concepts from blogs, forums, tweets, and online discussions.

3.7 Conclusions

We have studied the problem of extracting cross-lingual topics from non-parallel data. In this chapter, we have presented a new bilingual probabilistic topic model called comparable bilingual LDA (C-BiLDA) which is able to distinguish between shared and unshared content in aligned document pairs to learn more coherent cross-lingual topics. We have demonstrated the utility of C-BiLDA in performing the knowledge transfer for cross-lingual document classification for three language pairs, where our model has outperformed the standard bilingual LDA model (BiLDA) on two benchmarking datasets, indicating that distinguishing between shared and unique content in document pairs leads to better per-topic word distributions when training on non-parallel data. Like other topic models, C-BiLDA can be used in a variety of other natural language processing and information retrieval tasks.

C-BiLDA is completely data-driven and does not require a machine-readable bilingual dictionary or high-quality parallel data. Furthermore, it does not make any language specific assumptions. C-BiLDA's wide applicability in terms of input data makes it an excellent model for learning representations in under-resourced languages and language pairs, as well as in domains with specific terminology for which high-quality (multilingual) data is often not available.

This chapter was published as:

Geert Heyman, Ivan Vulić, Marie-Francine Moens. C-BiLDA extracting cross-lingual topics from non-parallel texts by distinguishing shared from unshared content. *Data Mining and Knowledge Discovery, 30(5), pages 1299–1323*, 2016

Chapter 4

Unsupervised Multilingual Embeddings for Multilingual Downstream Tasks

This chapter further investigates unsupervised representation learning for text by studying how to learn multilingual word representations from monolingual corpora without additional resources. Compared to Chapter 3, we hence move to a more general setting: 1) We learn *multilingual* representations, that is, representations for more than two languages; 2) We no longer require subject-aligned document pairs, but instead rely on very large monolingual corpora that are in the same domain (e.g., entire Wikipedia corpora). Consequently, we switch from the representations induced by probabilistic topic models to word embeddings trained with neural networks. The motivation for this is two-fold. Firstly, word embeddings can be trained very efficiently, making it feasible to train on very large corpora. Secondly, although being less interpretable, word embeddings have been shown to lead to better classification performance compared to PTM-based representations.

In this chapter, we propose two methods for mapping monolingual word embeddings to a multilingual space and evaluate the representations that are induced by the methods on three different tasks (i.e., bilingual lexicon induction, multilingual document classification, and multilingual dependency parsing) using four different benchmark datasets. We show that on these datasets our best method is either competitive (bilingual lexicon induction) or better (document classification and dependency parsing) than the state of the art for multilingual word embedding induction.

4.1 Introduction

Low-dimensional, distributed word representations or *word embeddings* have become the de-facto standard for representing words in NLP. Their success in monolingual tasks quickly led to research on cross-lingual extensions that induce embeddings for two or more languages in the same vector space. Embeddings of translations and words of similar meaning will be geometrically close in this vector space and as such they are effective features for cross-lingual NLP tasks, e.g., cross-lingual document classification (Klementiev et al., 2012), cross-lingual information retrieval (Vulić and Moens, 2015), bilingual lexicon induction (Mikolov et al., 2013c; Gouws et al., 2015; Vulić and Moens, 2016a; Heyman et al., 2017, *inter alia*), or (unsupervised) machine translation (Artetxe et al., 2017b; Lample et al., 2018; Artetxe et al., 2018c).

Most prior work has focused on methods for constructing *bilingual* word embeddings (BWEs), yielding word representations for exactly two languages. For problems such as multilingual document classification, however, it is highly-desirable to represent words in a *multilingual* space. A favourable property is that it enables fitting a single classifier on the union of training datasets in many languages, which results in 1) knowledge transfer across languages that may lead to better classification performance, and 2) a setup that is easier to maintain as it is no longer required to train many different monolingual or bilingual classifiers.

Methods that learn word representations in a multilingual space typically generalize existing BWEs methods by mapping multiple source language spaces to the space of one target language (Ammar et al., 2016), which is used as a pivot language. This approach may lead to a suboptimal solution as it does not account for dependencies between the source languages. Most BWE and multilingual word embedding (MWE) methods rely on some sort of supervision, however: Bilingual lexicons (Mikolov et al., 2013a), parallel corpora (Gouws et al., 2015), or subject-aligned document pairs (Vulić and Moens, 2016a).[1] In such paradigms, modeling dependencies between all languages would be impractical as it would require supervision for all language pair combinations.

Recent research has shown that BWEs can also be learned without cross-lingual supervision and can even outperform their supervised counterparts on bilingual lexicon induction benchmarks (Lample et al., 2018; Artetxe et al., 2018a). Extending the work of Lample et al. (2018), Chen and Cardie (2018) took a first step towards learning multilingual spaces without supervision and incorporating dependencies among all languages, but their approach inherits the limitations of Lample et al. (2018), for which the training objective is not very stable (i.e., it sometimes leads to degenerate solutions) and which does not work for distant language pairs such as English-Finnish (Søgaard et al., 2018).

[1] See Ruder et al. (2018) for a complete overview of BWE model typology.

In this work, we investigate robust methods to induce multilingual word embeddings without any supervision. The robustness of our approach is illustrated in good performance for distant languages such as Finnish and Bulgarian. Specifically, this chapter makes the following contributions:

- Based on a reformulation of the BWE method of Artetxe et al. (2018a), we propose two novel methods for inducing MWEs: the single hub space model (SHS) uses the classical idea of mapping source languages to a single hub language, and the incremental hub space model (IHS) which incorporates dependencies between all languages by incrementally growing the multilingual space. This new strategy results in mappings that are more robust and coherent across languages. Both SHS and IHS only require monolingual word embeddings as an input.
- We evaluate our method on benchmarks for bilingual lexicon induction, multilingual document classification, and multilingual dependency parsing and find that the IHS method is competitive with the state of the art on the bilingual lexicon induction benchmarks and obtains the best results on the multilingual document classification and dependency parsing benchmarks.
- Unlike the majority of prior work (Lample et al., 2018; Artetxe et al., 2018a; Chen and Cardie, 2018, *inter alia*), we do not limit our evaluation to intrinsic tasks such as bilingual lexicon induction. Consequently, we can investigate if embedding reweighting, a recently proposed best practice for BWEs, is useful for extrinsic tasks such as document classification and dependency parsing in multilingual settings.

4.2 Related Work

Cross-lingual word embeddings models have received a lot of attention in recent years. Most methods construct a space shared between two languages using a bilingual signal in the form of bilingual lexicons (Mikolov et al., 2013a; Artetxe et al., 2016; Smith et al., 2017), parallel corpora (Klementiev et al., 2012; Faruqui and Dyer, 2014; Gouws et al., 2015; Luong et al., 2015) or topic-aligned document pairs (Vulić and Moens, 2015, 2016a). See Ruder et al. (2018) for a comprehensive overview.

To enable knowledge transfer across an arbitrary number of languages, research has expanded to methods that map more than two languages. Huang et al. (2015), propose decomposing a matrix with multilingual co-occurrence counts weighted by probabilistic dictionaries and illustrate that this method scales linearly in the number of languages. Ammar et al. (2016) compare this method to three other MWEs models: MultiCluster, MultiCCA, and MultiSkip. MultiCluster uses bilingual dictionaries to

cluster translations and then they train the monolingual Skip-gram model Mikolov et al. (2013a) on a union of monolingual corpora where they replace words with their cluster id such that words in the same cluster get the same representation. MultiCCA is the multilingual extension of the method of Faruqui and Dyer (2014). Using canonical correlation analysis (CCA) and bilingual dictionaries with English as the target language, monolingually trained embeddings are projected to the English embedding space. MultiSkip is a straightforward extension of the BiSkip method (Luong et al., 2015) which generalizes the monolingual skip-gram objective to account for word alignments in parallel corpora. Similarly, Duong et al. (2017), extend their bilingual extension of CBOW to multiple languages. Common to all these methods is that they align spaces using bilingual dictionaries of parallel corpora, which limits their applicability for many languages.

More recently, it was discovered that BWE spaces can also be trained without supervision (Lample et al., 2018; Artetxe et al., 2018a), based on the assumption that the monolingual embedding spaces are approximately isomorphic.[2] Improving on earlier attempts (Cao et al., 2016; Zhang et al., 2017), Lample et al. (2018) propose a two-step framework to map two monolingual word embeddings matrices to the same space. In the first step, they use an adversarial objective to get an initial bilingual space in which the discriminator can no longer distinguish to which language a given word embedding belongs. In the second step, they fine-tune the initial solution. An important limitation is that the adversarial objective is not easy to optimize and sometimes yields degenerate solutions. Furthermore, Søgaard et al. (2018) found that the method does not work for distant language pairs such as English-Finnish.

In parallel to the work of Lample et al. (2018), Artetxe et al. (2018a) proposed another framework with the same intent. Expanding on their earlier work (Artetxe et al., 2017a, 2018b), they use an unsupervised heuristic to obtain an initial seed lexicon which is used to obtain an initial bilingual space. This solution is iteratively improved similar to Artetxe et al. (2017a) and Lample et al. (2018) while using value dropping regularization to escape early local minima. As their method is the starting point for this work it will be explained in detail in Section 4.3.

The approaches of both Lample et al. (2018) and Artetxe et al. (2018a) are limited to finding mappings between a pair of languages. To the best of our knowledge, Chen and Cardie (2018) is the only method that constructs a multilingual embedding space without supervision and incorporates all dependencies between the languages that are being mapped. Their method extends the adversarial pre-training and iterative refinement steps of Lample et al. (2018) to a multilingual setting. As a consequence, the method inherits the aforementioned deficits of Lample et al. (2018): less stable optimization and not applicable to distant language pairs. Furthermore,

[2]One of the necessary conditions for this assumption to hold is that the monolingual corpora on which the embeddings are trained are comparable (Søgaard et al., 2018).

their generalization of the iterative refinement turns it into a non-convex optimization problem.

In contrast, the two multilingual extensions of Artetxe et al. (2018a) that we propose in this work are applicable to distant language pairs and decompose every iteration in the refinement step in multiple convex optimization problems, making them very robust and widely applicable.

4.3 Bilingual Word Embedding Spaces

Before introducing our own multilingual models, we summarize the state of the art on mapping monolingual embeddings to BWEs. Methods that construct a cross-lingual space from two monolingual embedding spaces require a *mapping procedure*, a way to transform the monolingual spaces such that translations become geometrically close. Supervised approaches take translations from a fixed training dictionary of known translations, whereas unsupervised approaches have a heuristic to construct a seed lexicon (or equivalently a bilingual space from which a seed lexicon can be extracted) from scratch and an iterative procedure to refine the seed lexicon and bilingual space.

4.3.1 Mapping Procedure

There have been various mapping procedures (Mikolov et al., 2013c; Dinu et al., 2015; Lazaridou et al., 2015; Vulić and Korhonen, 2016) proposed in the literature, however they can all be explained within a single framework (Artetxe et al., 2018b) which we summarize here.

At its core, each mapping procedure learns the orthogonal transformations $\boldsymbol{W}_x$ and $\boldsymbol{W}_z$ for the monolingual embedding spaces $\boldsymbol{X}$ and $\boldsymbol{Z}$ that minimize the distance between embeddings of translations in the mapped spaces $\boldsymbol{X}\boldsymbol{W}_x$ and $\boldsymbol{Z}\boldsymbol{W}_z$. The orthogonality constraint ensures that the transformations preserve the constellation of embeddings in the respective monolingual spaces. Formally, let $\boldsymbol{D}$ be a matrix representing a bilingual dictionary s.t. $\boldsymbol{D}_{ij} = 1$ if the i^{th} source word is translated by the j^{th} target word and $\boldsymbol{D}_{ij} = 0$ otherwise, then $\boldsymbol{W}_x$ and $\boldsymbol{W}_z$ are found by solving the following optimization problem:

$$\underset{\boldsymbol{W}_x,\boldsymbol{W}_z}{\arg\max} \sum_i \sum_j \boldsymbol{D}_{ij}\big((\boldsymbol{X}(i)\boldsymbol{W}_x)\cdot(\boldsymbol{Z}(j)\boldsymbol{W}_z)\big) = \underset{\boldsymbol{W}_x,\boldsymbol{W}_z}{\arg\max}\ tr(\boldsymbol{X}\boldsymbol{W}_x(\boldsymbol{D}\boldsymbol{Z}\boldsymbol{W}_z)^\intercal) \quad (4.1)$$

$$\text{subject to } \boldsymbol{W}_x\boldsymbol{W}_x^\intercal = \boldsymbol{I}, \boldsymbol{W}_z\boldsymbol{W}_z^\intercal = \boldsymbol{I}$$

where $\boldsymbol{M}(i)$ denotes the i^{th} row of a matrix $\boldsymbol{M}$ and $tr(\boldsymbol{M})$ the trace of a matrix $\boldsymbol{M}$

Equation 4.1 has a closed-form solution based on the singular vectors of $\boldsymbol{X}^\intercal \boldsymbol{D}\boldsymbol{Z}$:

$$\boldsymbol{W}_x = \boldsymbol{U}, \boldsymbol{W}_z = \boldsymbol{V} \tag{4.2}$$
$$\text{with } \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\intercal = SVD(\boldsymbol{X}^\intercal \boldsymbol{D}\boldsymbol{Z})$$

In addition to the orthogonal transformation, there are several optional pre-processing (1-2) and post-processing (3-5) steps:

1) *Normalization*, apply length normalization (normalizing $\boldsymbol{X}$ and $\boldsymbol{Z}$ such that all embeddings have a unit Euclidean norm), or mean centering (ensuring each dimension has zero mean), or a combination of both;
2) *Whitening*, apply ZCA whitening (Bell and Sejnowski, 1997) on $\boldsymbol{X}$ and $\boldsymbol{Z}$ which transforms the monolingual embedding matrices such that each dimension/component has unit variance and such that the dimensions are uncorrelated (for the formula see Equation 4.6 of the next section). The intuition behind this operation is that it could make it easier to align the vector spaces along directions of high variance (Artetxe et al., 2018c);
3) *Re-weighting*, the components according to the singular value matrix $\boldsymbol{S}$ of $\boldsymbol{X}^\intercal \boldsymbol{D}\boldsymbol{Z}$. This is an attempt to further align the embeddings in the multilingual space as each singular value measures how well a dimension in the multilingual space correlates across languages for the given dictionary;
4) *De-whitening*, the inverse transformation of 2). Once the embeddings have been rotated, it has been shown to be important to restore the variance information in case whitening was applied (Artetxe et al., 2018c);
5) *Dimensionality reduction,* which truncates the embedding vectors such that only components with the highest singular values are kept.

4.3.2 Refinement Procedure

The refinement procedure aims at iteratively improving the seed dictionary and the bilingual space with an Expectation Maximization (EM) procedure (Dempster et al., 1977). In every iteration, the mapping procedure is executed using the dictionary from the previous iteration to obtain a new bilingual space, after which a new bilingual dictionary is induced using nearest neighbor retrieval in the cross-lingual similarity matrix $\boldsymbol{M}$. The process is repeated until the (unsupervised) training objective $\sum_i \sum_j \boldsymbol{D}_{ij}\big((\boldsymbol{X}_{i,:}\boldsymbol{W}_x) \cdot (\boldsymbol{Z}_{j,:}\boldsymbol{W}_z)\big)$ stops increasing.

The cross-lingual similarity matrix $\boldsymbol{M}$ is calculated using cosine similarity with cross-domain similarity local scaling (CSLS; Conneau et al. (2018)), a variant to cosine similarity to avoid the hubness problem (Radovanović et al., 2010; Dinu et al., 2015) (i.e., the phenomenon in a high-dimensional vector spaces where there are vectors, known as hubs, that are the nearest neighbors to many vectors in the space). Specifically, the element m_{ij} at row i and column j of M corresponds to the CSLS value between the cross-lingual vectors $\boldsymbol{x}_i^{CL}$ and $\boldsymbol{z}_j^{CL}$ of the i^{th} source word and the j^{th} target word respectively:

$$m_{ij} = CSLS(\boldsymbol{x}_i^{CL}, \boldsymbol{z}_j^{CL}) \tag{4.3}$$

$$CSLS(\boldsymbol{x}, \boldsymbol{z}) = 2\, cos(\boldsymbol{x}, \boldsymbol{z}) - r_Z(\boldsymbol{x}; k) - r_X(\boldsymbol{z}; k) \tag{4.4}$$

Where $r_X(\boldsymbol{x}; k)$ and $r_Z(\boldsymbol{z}; k)$ calculate the average cosine similarity of a vector with its k nearest neighbors (measured by cosine similarity) in the mapped spaces of X, Z respectively. In practice, k is set to 10 (Lample et al., 2018).

It has been shown to be beneficial to jointly infer dictionaries from source to target and from target to source language (Artetxe et al., 2018a).[3] The bilingual mapping is then learned from the concatenation of these two dictionaries. The search space for the bilingual dictionary is then two times the product of the source and target vocabularies. In practice, limiting the search space by truncating both vocabularies and their corresponding embedding matrices to the $C_{refinement}$ most frequent words, results in better solutions and limits the amount of computational effort.

As a measure against getting stuck in early, suboptimal local minima, Artetxe et al. (2018a) propose to randomly drop values from the cross-lingual similarity matrix $\boldsymbol{M}$ with probability $1 - p$ (further called *value dropping*). The value of p is exponentially increased as training progresses. At the start of training p is initialized with a small value (e.g., 0.1). Whenever the objective stops improving for $N_{patience}$ refinement steps p is multiplied with a given factor (e.g., 2) until $p \geq 1$ after which all values in $\boldsymbol{M}$ are kept. Value dropping was shown to be crucial when constructing bilingual spaces between distant language pairs.

4.3.3 Inducing a Seed Lexicon

Artetxe et al. (2018a) obtain a seed lexicon based on the assumption that for a translation pair w_i^X, w_j^Z, the monolingual similarity vectors, $\sqrt{\boldsymbol{X_i X^\intercal}}$ and $\sqrt{\boldsymbol{Z_j Z^\intercal}}$ of translations i, j are (approximately) equal up to a permutation. Therefore, seed translations for a source word i are generated by finding the nearest neighbor of $sorted(\sqrt{\boldsymbol{X}_{i,:}\boldsymbol{X}^\intercal})$ in $sorted(\sqrt{\boldsymbol{Z}\boldsymbol{Z}^\intercal})$.[4] Although this heuristic yields a very noisy seed lexicon, it

[3] Note that $\boldsymbol{z}_j^{CL}$ being the nearest neighbor of $\boldsymbol{x}_i^{CL}$ does not imply that the inverse is also true.

[4] The inclusion of the square root in the formulas is empirically motivated.

has been proven to contain a sufficiently strong bilingual signal to bootstrap the refinement procedure. Similar to the refinement procedure, the seed lexicon is inferred symmetrically (i.e., by concatenating respective seed lexicons from source to target and target to source) and the vocabularies and corresponding embedding matrices are first truncated to the C_{seed} most frequent words.

4.4 Method

In this section, we present two models for learning multilingual embedding spaces without supervision (i.e., without any parallel data or bilingual dictionaries): the single hub space model (SHS) and the incremental hub space model (IHS). The methods generalize the unsupervised bilingual mapping framework of Artetxe (introduced in the previous section) such that we construct a multilingual space without relying on a (less stable) adversarial objective and can leverage the different pre- and post-processing steps in a multilingual setting. Moreover, the IHS model simultaneously incorporates dependencies between multiple languages when mapping monolingual embeddings to the multilingual space. This could not only make the multilingual space more coherent, but could also be a potent regularization mechanism when mapping distant languages.

4.4.1 Multilinguality through a Hub Language

In this section, we present the single hub space model (SHS). This model defines one language as the hub language L_0 and projects the embedding spaces $\boldsymbol{Z_1, ..., Z_N}$ of all other languages $L_1, ..., L_N$ (further secondary languages) to the hub language space $\boldsymbol{X}$. Hence, we reduce the construction of a multilingual space of N languages to the alignment of $N-1$ vector spaces. Learning these projections is similar to the bilingual case, we use the unsupervised iterative refinement procedure and seed lexicon heuristic explained in Sections 4.3.2 and 4.3.3. However, we require the mapping to be asymmetric (in contrast to Equation 4.3 where both the source and target spaces are rotated) as the space of the hub language should either remain unchanged or it should be transformed with the same operation for each of the $N-1$ embeddings pairs. We therefore derive an asymmetric version of the mapping framework of Section 4.3.1 which in the bilingual case leads to an equivalent cross-lingual space.

Let $\boldsymbol{X}$ be the embedding matrix of the hub language; $\boldsymbol{Z}_1$,.., $\boldsymbol{Z}_N$ the embedding matrices of the other languages; and $\boldsymbol{D}^{k,l}$ the dictionary between languages L_k and L_l, then we obtain a multilingual embedding space $\boldsymbol{X}^{multi}$, $\boldsymbol{Z}_1^{multi}$, ... , $\boldsymbol{Z}_N^{multi}$ in three main steps. First, the embeddings of each language are preprocessed by normalizing and whitening the embeddings (see Equations 4.5-4.10). Normalization consists of subsequently performing length normalization, mean centering, and then again length normalization.

$$normalize(\boldsymbol{W}) = length\_norm(mean\_center(length\_norm(\boldsymbol{W}))) \tag{4.5}$$

$$ZCAwhiten(\boldsymbol{W}) = \boldsymbol{W}(\boldsymbol{W}^{\intercal}\boldsymbol{W})^{-0.5} \tag{4.6}$$

$$\boldsymbol{X}^{'} = normalize(\boldsymbol{X}) \tag{4.7}$$

$$\boldsymbol{X}^{''} = ZCAwhiten(\boldsymbol{X}^{'}) \tag{4.8}$$

$$\boldsymbol{Z}_l^{'} = normalize(\boldsymbol{Z}_l) \tag{4.9}$$

$$\boldsymbol{Z}_l^{''} = ZCAwhiten(\boldsymbol{Z}_l^{'}) \tag{4.10}$$

After preprocessing, we rotate each secondary language L_l to a bilingual space between the hub language space and its own embedding space (Equations 4.12-4.14). The calculations are analogous to the bilingual mapping procedure introduced above: the left and right singular vectors $\boldsymbol{U}_l$ and $\boldsymbol{V}_l$ of $\boldsymbol{X}^{''}\boldsymbol{D}^{k,l}\boldsymbol{Z}_l^{''\intercal}$ are the rotation matrices that project the preprocessed matrices $\boldsymbol{X}^{''}$ and $\boldsymbol{Z}_l^{''}$ to their bilingual space (Equations 4.11-4.12).[5] The bilingual projection of $\boldsymbol{Z}_l^{''}$ can be reweighted by multiplying it with a given power q of the singular values matrix $\boldsymbol{S}_l$ of $\boldsymbol{X}^{''}\boldsymbol{D}^{0,l}\boldsymbol{Z}_l^{''\intercal}$ (Equation 4.13). Intuitively this reweighting operation makes the dimensions that correlate better across languages more important. Next, we restore the variance information of $\boldsymbol{Z}_l^{'}$ by performing a dewhitening operation: we project back to the monolingual space, multiply with the inverse of the whitening matrix, and then project back to the bilingual space (Equation 4.14).[6]

$$\boldsymbol{U}_l\boldsymbol{S}_l\boldsymbol{V}_l^{\intercal} = SVD(\boldsymbol{X}^{''\intercal}\boldsymbol{D}^{0,l}\boldsymbol{Z}_l^{''}) \tag{4.11}$$

$$\boldsymbol{Z}_{l,bi(l)} = \boldsymbol{Z}_l^{''}\boldsymbol{V}_l \tag{4.12}$$

$$\boldsymbol{Z}_{l,bi(l)}^{'} = \boldsymbol{Z}_{l,bi(l)}\boldsymbol{S}_l^{q} \tag{4.13}$$

$$\boldsymbol{Z}_{l,bi(l)}^{''} = \boldsymbol{Z}_{l,bi(l)}^{'}\boldsymbol{V}_l^{\intercal}(\boldsymbol{Z}_l^{'\intercal}\boldsymbol{Z}_l^{'})^{0.5}\boldsymbol{V}_l \tag{4.14}$$

[5] In Equation 4.11 *SVD* refers to the singular value decomposition.

[6] Note that the projection matrices that map from the bilingual to the monolingual spaces are given by the inverses of $\boldsymbol{U}_l$ and $\boldsymbol{V}_l$. Because the matrices are orthogonal their inverses are equal to their transposes.

As a final step, we project $\boldsymbol{Z}''_{l,bi(l)}$ to the space of the hub language (Equation 4.15). The multilingual space for the hub language is simply the monolingual embedding space after preprocessing (Equation 4.16).

$$\boldsymbol{Z}_l^{multi} = \boldsymbol{Z}''_{l,bi(l)}\boldsymbol{U}_l^{\intercal} \tag{4.15}$$

$$\boldsymbol{X}^{multi} = \boldsymbol{X}' \tag{4.16}$$

One can easily verify that for the bilingual case this formulation is equivalent to the symmetric mapping introduced in Section 4.3.1 by showing that the dot products between the mapped spaces simplify to the same formula.

4.4.2 Incrementally Constructing the Multilingual Space

In the hub space model most language pairs are aligned indirectly through the hub language. Ideally, we want a mapping algorithm that incorporates the interdependencies between all language pairs. We hypothesize that, especially when mapping a language distant to the hub language, it is beneficial to incorporate the structural similarities with other languages as a kind of regularization mechanism to find a more robust mapping.

We therefore propose the incremental hub space (IHS) model, which incrementally grows the cross-lingual space $\boldsymbol{X}^{multi}$ and takes into account all languages in the cross-lingual space when adding a new language. First, we define an order on the languages and initialize the multilingual space to the preprocessed embedding space of language L_0. Next, we iteratively add new languages to the space by at any given iteration l rotating the preprocessed embedding space $\boldsymbol{Z}''_l$ of language l to the multilingual space by minimizing the dot product between embeddings of the translations between language l and the languages in the multilingual space. The recipe to calculate the cross-lingual embedding $\boldsymbol{Z}_l^{multi}$ is similar to the hub language model: the preprocessing and postprocessing steps are the same, but the rotation matrices are calculated with Equation 4.18 instead of 4.11, conform with the new objective (Equation 4.17). After the self-learning is converged, $\boldsymbol{Z}_l^{multi}$ is added to the cross-lingual space $\boldsymbol{X}^{multi}$: $\boldsymbol{X}_l^{multi} = \boldsymbol{Z}_l^{multi}$.

$$\underset{\boldsymbol{W}_{xl},\boldsymbol{W}_{zl}}{\arg\max} \sum_{k=0}^{l-1} tr(\boldsymbol{X}_k^{multi}\boldsymbol{W}_{xl}(\boldsymbol{D}^{k,l}\boldsymbol{Z}''_l\boldsymbol{W}_{zl})^{\intercal}) \tag{4.17}$$

$$\text{subject to } \boldsymbol{W}_{xl}\boldsymbol{W}_{xl}^{\intercal} = \boldsymbol{I}, \boldsymbol{W}_{zl}\boldsymbol{W}_{zl}^{\intercal} = \boldsymbol{I}$$

$$\boldsymbol{U}_l \boldsymbol{S}_l \boldsymbol{V}_l^{\intercal} = SVD((\boldsymbol{X}_0^{multi})^{\intercal} \boldsymbol{D}^{0,l} \boldsymbol{Z}_l^{''} || \ldots || (\boldsymbol{X}_{l-1}^{multi})^{\intercal} \boldsymbol{D}^{l-1,l} \boldsymbol{Z}_l^{''}) \tag{4.18}$$

where $||$ denotes concatenation along the row axis

In a supervised setting this approach would be impractical as it would require bilingual dictionaries $\boldsymbol{D}^{k,l}$ for all language pairs k,l, not only with the hub language. However, within a self-learning, unsupervised framework there is no such constraint.

4.5 Experimental Setup

4.5.1 Tasks and Datasets

The embeddings are evaluated in three tasks: bilingual lexicon induction (BLI), multilingual dependency parsing, and multilingual document classification. Bilingual lexicon induction is the most widely used method to evaluate bilingual embedding spaces. Although BLI performance is not the primary goal of the multilingual embedding spaces, it provides a fast means to address the following research questions: **1)** Is the incremental construction of multilingual embedding spaces indeed an effective regularization method? Is it still necessary to perform value dropping in this case? Value dropping significantly slows down training time and leads to non-deterministic outcomes, though it has shown to be crucial in the bilingual setting to obtain good results when mapping distant language pairs (Artetxe et al., 2018a); **2)** Is the reweighting of embedding spaces also beneficial for BLI in a multilingual setting?; **3)** Does multilingual training improve bilingual lexicon induction performance? How do the multilingual models compare to each other and the state of the art in unsupervised BLI?

We evaluate BLI performance with *accuracy* and use two BLI datasets:

- DINUARTETXE, the extended version of the dataset of Dinu et al. (2015) used in Artetxe et al. (2018a).[7] It consists of bilingual dictionaries for English-German, English-Italian, English-Spanish and English-Finnish; and of monolingual embeddings trained with the CBOW model on the WaCKy corpora for English, Italian and German, the monolingual WMT Common Crawl corpus for Finnish, and the WMT News Crawl for Spanish. The sizes of the test dictionaries are between 1.869 and 1,993 translations for each language. As our methods are unsupervised, we do not use the training dictionaries. The embeddings are 300-dimensional [8] and were truncated to the 200k most frequent words.

[7] Easy to download with https://github.com/artetxem/vecmap/blob/master/get_data.sh.

[8] Using 300-dimensional word embeddings is standard practice in the word embedding literature.

- EURMUSEWIKI, this dataset is compiled from dictionaries for all combinations of the following European languages: English, German, Spanish, French, Italian, and Portuguese. The sizes of the test dictionaries range between 1,513 and 3,660 translations. We use publicly available[9] monolingual embeddings trained with fastText (Bojanowski et al., 2016) on recent Wikipedia dumps. The embeddings are 300-dimensional and we truncated them to the 200k most frequent words as done in related work (Dinu et al., 2015; Lample et al., 2018, *inter alia*).

With the multilingual dependency parsing and multilingual document classification tasks, we can assess the embeddings w.r.t. their actual goal: enabling transfer learning across multiple languages. The word embeddings are used as the feature vector for classifiers of the respective tasks. We will address the following research questions: **4)** Is reweighting of embedding spaces also beneficial in down-stream tasks?; **5)** How do our methods perform w.r.t. methods that learn multilingual embedding spaces with supervision?

The tasks are evaluated with the convenient online evaluation platform of Ammar et al. (2016)[10] where users can submit their multilingual embeddings and evaluate them on cross-lingual tasks. This ensures that the classifiers we use are identical to the ones used in related work Ammar et al. (2016) and Duong et al. (2017). Specifically, we evaluate the embeddings with the following datasets and corresponding classifiers:

- REUTERSMLDC, the multilingual document classification dataset consists of seven languages: English, German, French, Italian, Spanish, Danish, and Swedish. The classification performance is evaluated using the average accuracy across all languages. The training and test set consist of 7,000 and 13,058 documents respectively. The dataset is well balanced in the number of documents per language.[11] The architecture of the document classifier is the average perceptron used in Klementiev et al. (2012).

- MULTIPARSING, the multilingual dependency parsing dataset is a subset of the Universal Dependencies 1.1 corpus (Agić et al., 2015)[12] containing 12 languages: English, German, French, Spanish, Italian, Bulgarian, Czech, Danish, Swedish, Greek, Finnish, and Hungarian. The training and test set consist of 16,748 and 1,200 sentences respectively. The test set contains 100 sentences for each language, while for the training set the number of sentences for a language ranges between 98 and 6,694. The architecture of the classifier is the LSTM parser from

[9] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

[10] http://128.2.220.95/multilingual/

[11] As the dataset is not publicly available this information was provided by the first author of Ammar et al. (2016).

[12] http://hdl.handle.net/11234/LRT-1478

Dyer et al. (2015) without the use of part-of-speech and morphology features and keeping the word embeddings fixed (i.e., word embeddings are not optimized on the parsing task). The classification performance is evaluated with the average unlabeled attachment score across languages, which is the percentage of words that have the correct head (not evaluating the correctness of the dependency labels).

For a fair comparison with related work, we train 512-dimensional monolingual embeddings by training on the text collections used in Ammar et al. (2016) and Duong et al. (2017).[13] The embeddings were again trained using fastText.

4.5.2 Implementation & Default Hyper-parameters

We implemented the SHS and IHS models in Python 3 using the numpy and cupy libraries, starting from the code of the bilingual unsupervised mapping framework of Artetxe et al. (2018a).[14] In all the experiments, we set the following hyper-parameters to values that were used in prior research (Lample et al., 2018; Artetxe et al., 2018a):

When constructing the seed lexicon the 4,000 most frequent words of each language are considered ($C_{seed} = 4000$), and during the refinement step only the 20,000 most frequent words of each language are considered ($C_{refinement} = 20000$). When using value dropping, the keep probability p is initialized with 0.1, the number of refinement steps $N_{patience}$ without improvements in the objective before increasing p is set to 50, and the stochastic multiplier is set to 2. Dictionaries are constructed symmetrically: from hub language(s) to the secondary language and from the secondary language to the hub language(s). Implying that during refinement each dictionary consists of 2×20000 translations. We use the CSLS similarity metric with $k = 10$ nearest neighbors, following the setup of Lample et al. (2018).

4.6 Experiments

Experiment 1: Value dropping

This experiment is designed to verify if value dropping is a necessary condition for mapping between distant language pairs to be successful in a multilingual setting. In

[13] The web service is available at http://128.2.220.95/multilingual/data/. The source code for the framework is available at https://github.com/wammar/multilingual-embeddings-eval-portal.

[14] https://github.com/artetxem/vecmap

		EN-DE		EN-IT		EN-ES		EN-FI		All
Model	value dropping	av.	max.	av.	max.	av.	max.	av.	max.	total av.
SHS	no	48.00	48.00	45.93	45.93	36.53	36.53	0.14	0.14	32.65
SHS	yes	47.51	47.73	45.60	46.20	36.33	36.87	31.92	32.79	**40.34**
IHS	no	47.93	47.93	45.93	45.93	36.07	36.07	31.04	31.04	**40.24**
IHS	yes	47.45	47.80	45.48	45.67	36.43	36.87	30.97	31.32	40.08

Table 4.1: Comparison of the *accuracy* scores the SHS and IHS models with and without value dropping on the DINUARTETXE BLI dataset.

Table 4.1 we show the results for SHS and IHS models with and without value dropping, using no reweighting (i.e., $q = 0$), and evaluated on the DINUARTETXE dataset. For the SHS model English is taken as the hub language, and for the IHS model we process the languages in the following order: English, German, Italian, Spanish, Finnish. When using value dropping we report the average and best results across five runs.

For the SHS model, we find that value dropping is paramount for mapping distant language pairs, but for the IHS model this is not the case. This supports our hypothesis/interpretation that mapping a language to a space that contains more languages is a type of regularization and hence can replace value dropping. Note that it is important that we do not start with mapping the distant languages. For instance, when using IHS with a language order that starts with English and Finnish, value dropping would still be required to prevent bad performance for Finnish. As we know in advance which languages are more distant this is not a problem in practice.

Experiment 2: Reweighting and Comparison with State of the Art for BLI

In this experiment, we verify if reweighting embedding spaces is still beneficial for BLI in a multilingual setup, and compare our methods w.r.t. state of the art BLI methods. In Table 4.2 we show the results for SHS and IHS with reweighting coefficients q of 0, 0.5 and 1 on the DINUARTETXE dataset and include the state-of-the-art results (Artetxe et al., 2018a) as a reference. Similarly, Table 4.3 reports the results for SHS and IHS with reweighting coefficients q of 0, 0.5 and 1 and the state of the art results of (Chen and Cardie, 2018) on the EURMUSEWIKI dataset. The EURMUSEWIKI benchmark evaluates BLI performance on all language pair combinations of its six languages and this in both directions (EN-DE, DE-EN, EN-ES, ... IT-PT, PT-IT) yielding 28 *accuracy* scores per model. For clarity, we report the average *accuracy* scores per language as well as the global *accuracy* average. Based on the conclusion of Experiment 1, all results for SHS are obtained using value dropping (again averaged across 5 different

		EN-DE		EN-IT		EN-ES		EN-FI		All
Model	q	av.	max.	av.	max.	av.	max.	av.	max.	total av.
Artetxe	2×0.5	48.13	48.53	**48.19**	48.47	37.33	37.60	**32.63**	33.50	**41.57**
SHS	0	47.51	47.73	45.60	46.20	36.33	36.87	31.92	32.79	40.34
IHS	0	47.93	47.93	45.93	45.93	36.07	36.07	31.04	31.04	40.24
SHS	0.5	**48.69**	48.80	47.67	48.00	37.51	37.80	32.40	33.08	**41.57**
IHS	0.5	48.60	48.60	47.73	47.73	**37.53**	37.53	31.74	31.74	41.40
SHS	1	47.77	47.87	47.91	48.13	37.00	37.40	31.82	32.51	41.13
IHS	1	48.00	48.00	48.00	48.00	37.93	37.93	31.46	31.46	41.35

Table 4.2: *Accuracy* scores on the DINUARTETXE BLI dataset: SHS and IHS are evaluated for different values of the reweighting parameter q and the state-of-the-art results of Artetxe et al. (2018a) are added as a reference.

runs), while for IHS we do not use value dropping. The hub language for SHS is English, and the language orders for IHS are EN, DE, IT ES, FI for the DINUARTETXE dataset and EN, DE, ES, FR, IT, PT for the EURMUSEWIKI dataset.

Tables 4.2 and 4.3 reveal that reweighting the target embedding spaces is indeed still beneficial for BLI when mapping to a multilingual space. Both SHS and IHS obtain best results with reweighting coefficient $q = 0.5$. When comparing SHS and IHS, we see that for language pairs involving English (the SHS hub language) SHS obtains slightly better results, but for the other language pairs IHS outperforms SHS slightly. This is no surprise as IHS by design incorporates dependencies between all languages when learning the rotation matrices, though it is striking that mapping to a single hub language is still a strong BLI baseline. For both datasets IHS obtains BLI performance competitive with the state of the art. On the DINUARTETXE dataset, SHS and IHS with $q = 0.5$ obtain (near) identical scores to Artetxe et al. (2018a), on the EURMUSEWIKI dataset IHS $q = 0.5$ slightly outperforms Chen and Cardie (2018) for all languages except Spanish. Although BLI is not the main purpose of multilingual word embeddings, these results illustrate the soundness of our methods. On a final note, we point out that the results on the EURMUSEWIKI dataset are significantly higher than those on the DINUARTETXE. This can be attributed to a combination of factors: the Wikipedia corpora tend to be more comparable across languages than the web crawl data used for DINUARTETXE, the monolingual EURMUSEWIKI embeddings are trained with fastText which tends to provide more fine-grained embeddings than CBOW, and the test dictionaries were constructed with different methods.

Model	q	av. EN	av. DE	av. ES	av. FR	av. IT	av. PT	av. All
Chen and Cardie (2018)		79.57	70.46	**82.88**	82.01	80.69	80.13	79.29
SHS	0	80.04	68.24	80.95	80.10	78.71	77.96	77.67
IHS	0	79.61	69.52	82.35	81.26	80.00	79.02	78.63
SHS	0.5	**80.34**	70.16	82.15	81.65	80.31	79.78	79.07
IHS	0.5	79.91	**70.77**	82.68	**82.08**	**81.08**	**80.47**	**79.50**
SHS	1	79.61	69.59	81.53	81.10	79.74	79.47	78.51
IHS	1	79.05	69.99	81.63	81.23	80.13	79.68	78.62

Table 4.3: *Accuracy* scores on the EURMUSEWIKI BLI dataset averaged per language: SHS and IHS are tested for different values of the reweighting parameter q and the state-of-the-art results of Chen and Cardie (2018) are added as a reference.

Experiment 3: Downstream Tasks

In this experiment, we investigate the effect of reweighting embeddings on performance on downstream tasks that use the multilingual embeddings as input features, and we compare SHS and IHS to several supervised methods that use cross-lingual supervision for learning multilingual embeddings. Table 4.4 reports the results for SHS and IHS with reweighting coefficients q set to 0 and 0.5 on the REUTERSMLDC and MLPARSING benchmarks, along with the results from related work. For SHS the hub language is English and for IHS the language order is English, German, Spanish, Italian, French, Bulgarian, Czech, Danish, Finnish, Greek, Hungarian, and Swedish. Because the languages covered in the MLPARSING is a superset of the languages in REUTERSMLDC, we use the same multilingual embedding space for both tasks. Motivated by Experiment 1, we again use SHS with value dropping and IHS without value dropping. The results in Table 4.4 are comparable as all methods were trained on the same text corpora (i.e., the collections of Ammar et al. (2016)), albeit that our methods do not use the parallel corpora or bilingual dictionaries.

A first interesting result is that, contrary to the BLI task, reweighting the embeddings is not beneficial for multilingual dependency parsing and document classification. This can be explained by the fact that the reweighted embedding space is no longer isomorphic to the original monolingual embedding space, hence important patterns in the embedding space could be distorted. Another important observation is that both the SHS and IHS improve over the best reported results on the REUTERSMLDC and MLPARSING benchmarks. This result is surprising given that the reported baselines all use supervision to train the multilingual embedding spaces. Further, we again find that the best results are obtained with IHS, most notably for dependency parsing for which the difference in unlabeled attachment scores between the best IHS and SHS models is 2.29%.

Model	q	MLPARSING	REUTERSMLDC
Invariance (Huang et al., 2015)		59.80	91.10
MultiSkip (Luong et al., 2015)		57.70	90.40
MultiCluster (Ammar et al., 2016)		61.00	92.10
MultiCCA (Ammar et al., 2016)		58.70	92.10
Duong et al. (2017)		61.20	90.80
SHS	0	63.48	92.59
IHS	0	**65.77**	**92.72**
SHS	0.5	62.23	92.63
IHS	0.5	63.42	92.56

Table 4.4: Results on the MLPARSING (multilingual dependency parsing) and REUTERSMLDC (multilingual document classification) benchmarks: SHS and IHS are compared with and without reweighting and we show the state-of-the-art results of supervised embedding mapping methods as a reference. The results for Invariance, MultiSkip, Multicluster, MultiCCA all come from Ammar et al. (2016).

4.7 Conclusions

Building on recent developments in the induction of bilingual embedding spaces, we proposed two new methods for learning multilingual embedding spaces without using supervision (e.g., in the form of bilingual dictionaries or parallel corpora). By evaluating on different benchmark datasets, we have shown that our Incremental Hub Space (IHS) method infers multilingual embeddings that are competitive with the state of the art when used for bilingual lexicon induction and improve over the state of the art when used for multilingual dependency parsing and multilingual document classification. In contrast to prior research, IHS combines three desirable properties: it incorporates dependencies between all targeted languages, it works for distant languages such as Finnish and Bulgarian, and the method is both deterministic and robust in the sense that it does not produce degenerate solutions. Furthermore, we looked at the influence of reweighting the dimensions of the embedding spaces according to their cross-correlations with the hub language space(s) and found that while it improves performance for the BLI task, it is harmful to downstream tasks such as multilingual dependency parsing and multilingual document classification. This stresses the often overlooked requirement to include comprehensive and heterogeneous evaluation protocols for cross-lingual word embedding models in any future research in the field.

This chapter will be submitted in 2018:
Geert Heyman, Bregt Verreet, Ivan Vulić, Marie-Francine Moens. Unsupervised Multilingual Embeddings for Multilingual Downstream Tasks.

Chapter 5

Automatic Detection and Correction of Context-Dependent Dt-mistakes using Neural Networks

In this chapter, we study representation learning architectures for building an automatic spelling corrector for dt-mistakes, one of the most important Dutch spelling mistakes. Due to a lack of annotated training data (i.e., real-life spelling errors annotated with their corrections), directly estimating the distribution $P(s|\tilde{s})$ of the correct sentence s conditioned on the possibly erroneously-written input sentence $\tilde{s}$ is unfeasible. Because the mistakes we are addressing only concern incorrect usage of verb suffixes (e.g., gebeur**t** vs gebeur**d**), we instead estimate the probability of verb suffixes (e.g., d or t) given their stem (e.g., gebeur). Although this reformulation is more challenging because the system can no longer exploit the fact that the input spelling is usually correct, this circumvents the lack of annotated data: all that is required to train such system is a fair amount of high-quality text. We evaluate our model against four spell checkers, including the spell checker that comes with Microsoft Word, and find our model outperforms all other systems with a large margin.

Whereas Chapters 3 and 4 studied unsupervised RL methods, we now illustrate how representation learning can be used in a weakly-supervised framework by training the

model parameters on a proxy task that circumvents the absence of annotated training examples.

5.1 Introduction

Verbs in the Dutch language get assigned different inflections depending on their grammatical role and position in a sentence. For some verbs different inflections lead to the same pronunciation, making it impossible to *hear* which grammar rule applies. This phenomenon gives rise to one of the most common spelling mistakes in Dutch, commonly referred to as dt-mistakes, which even native speakers and/or language professionals are prone to make (Het Nieuwsblad, 2013, 2017). According to a study of spelling errors on the Internet, dt-errors were the most frequent classical spelling mistake in Dutch (Gheuens, 2012).[1]

An automatic solution for the dt-problem would be very desirable, however, previous investigations found that current grammar and spelling checkers demonstrate low recall when it comes to context-dependent dt-mistakes (Laevaert, 2017). In this chapter, we introduce a new approach that tackles this problem. More specifically, our work has three main contributions:

- We show how we can successfully train a neural network to correct context-dependent dt-mistakes, without annotated training examples, on millions of sentences. We report large improvements over state-of-the-art grammar and spelling checkers on three different benchmarking test sets.
- We propose a generative process for creating dt-mistakes, motivated by cognitive insights (Verhaert and Sandra, 2016). This enables the creation of two large-scale evaluation sets, which we use in addition to the three aforementioned test sets.
- We introduce a method for determining which words in a sentence lead the system to make its prediction. This is a valuable means for providing feedback to the users, especially if they are still not very familiar with the conjugation rules (e.g., for misspelled past participles it identifies the corresponding auxiliary verb, hereby indicating the rule that applies).

The remainder of this chapter is structured as follows. Section 5.2 provides a brief introduction to the dt grammar rules and defines the *context-dependent dt-error correction* task. Section 5.3 presents an overview of related work. Section 5.4 describes the proposed approach and different model architectures that we investigate. Section

[1]The term "classical mistake" refers to the fact that the mistake was not intentional, unlike mistakes related to cyber-slang or dialect usage.

5.5 explains how we construct the training and evaluation datasets. Sections 5.6 and 5.7 report on the setup for our experiments, and the corresponding results and observations. The conclusions and implications of this work are discussed in Section 5.8.

5.2 Dt-rules

In this section, we introduce the Dutch verb conjugation rules (further also referred to as *dt-rules*) and the *context-dependent dt-error correction* task. Table 5.1 illustrates the most important dt-rules. It is not our intention to provide an in-depth explanation of all the rules, though it is important to observe that verb inflections depend on multiple factors: the verb tense, the number of the subject, the position of the subject w.r.t. the verb, the raw stem of the verb (i.e., the infinitive minus the *-en* suffix), etc. Native speakers typically do not have problems applying these rules when they can *hear* the inflection. Verbs that have a homophone form (e.g., *beantwoord* and *beantwoordt*) give rise to many spelling errors, however. Such spelling errors are typically referred to as dt-mistakes.[2] In this work, we focus on correcting context-dependent dt-mistakes. The sentence *"*Ik **beantwoordt** de vraag."* contains an example of a context-dependent dt-mistake: *beantwoordt* is a correct Dutch verb form, hence an interpretation of the rest of the sentence is required to determine which grammar rule applies and detect the mistake. If instead the misspelled verb was *beantwoort*, then it would not have been a context-dependent mistake as *beantwoort* is not part of the Dutch vocabulary and can therefore be identified using dictionary lookups.

We formally define context-dependent dt-correction as the correction of a sentence $\tilde{s}$, consisting of N words $x_{1..N}$ to a sentence s such that incorrectly spelled, context-dependent dt-errors are corrected.

5.3 Related Work

Classical works in context-dependent spelling correction were based on n-gram language models (Atwell and Elliott, 1987; Gale and Church, 1990; Church and Gale, 1991; Mays et al., 1991). An important drawback of these approaches is that the n-gram assumption inhibits learning long-range dependencies. The use of higher order n-gram models to mitigate this issue generalizes badly due to the sparsity of language. Another line of research views context-dependent spelling correction as a disambiguation problem, where different classifiers are trained for each word pair that can be confused (Yarowsky, 1994; Gale et al., 1995; Golding, 1996; Golding et al., 1999; Mangu and Brill, 1997). This approach is ill-suited for dt-correction as it does

[2] Some sources use a more narrow definition of dt-mistakes and only consider rules 1-6 in Table 5.1.

not account for the fact that the correct spelling of any given verb depends on the same set of rules. Learning different classifiers for each confusion set (e.g., {antwoord, antwoordt}, or {vind, vindt}) therefore limits the generalizations that can be made from a training set.

#	tense	usage	subj. position	rule	example + translation
1	present	1st person	anywhere	stem	Ik **beantwoord** je vraag. I **answer** your question.
2	present	2nd person	after the verb	stem	**Beantwoord** je de vraag? Do you **answer** the question?
3	past participle	as verb	anywhere	(ge) + stem + (d/t)†	Hij heeft de vraag **beantwoord**. He has **answered** the question.
4	imperative	/	no subject	stem	**Beantwoord** de vraag! **Answer** the question!
5	present	2nd person	not after the verb	stem + t	Jij **beantwoordt** de vraag. You **answer** the question.
6	present	3rd person	anywhere	stem + t	Hij **beantwoordt** je vraag. He **answers** your question.
7	past participle	adjective	anywhere	(ge) + stem + (d/t)†+ (e)	De **beantwoorde** vraag ... The **answered** question ...
8	past	singular	anywhere	stem + te/de	Hij **beantwoordde** de vraag. He **answered** the question.
9	past	plural	anywhere	stem + ten/den	Zij **beantwoordden** de vraag. They **answered** the question.
10	present	plural	anywhere	stem + en	Zij **beantwoorden** de vraag. They **answer** the question.
11	infinitive	/	anywhere	stem + en	Ik zal de vraag **beantwoorden**. I will **answer** the question.

Table 5.1: Illustration of the main Dutch verb conjugation rules. Although the rules apply to most Dutch verbs, spelling problems only occur for verbs which have homophone verb forms, for which it is impossible to *hear* which rule applies (e.g., *beantwoord* vs *beantwoordt*). Note that this table is by no means a comprehensive overview of all Dutch verb conjugation rules. † (d/t): depending on the last character of the raw stem (infinitive - *en*) *-d*, *-t* or *nothing* should be added.

Prior work in context-dependent dt-error correction typically relies on handcrafted rules (e.g., Vosse (1992) defines an augmented context-free grammar and uses a shift-reduce parser to detect grammatical errors due to morphosyntactic inconsistencies) or uses a shallow statistical model from a limited context window. While rules can be precise, they are too restrictive and fail to detect errors when verb and subject/auxiliary verb are further apart. The work of Stehouwer and Van den Bosch (2009) can be seen as a first

attempt to learn a dt-correction system in a data-driven fashion. They use IGTree, a fast approximation of k-nearest neighbor classification (Daelemans et al., 1997), to identify the correct dt-verb forms based on fixed-length feature vectors constructed from context windows of four words. However, the use of small, fixed-length context windows is an oversimplification that will again run into problems when verb and subject/auxiliary are further apart. The architectures we propose address this issue by using a deep classifier which uses the full sentence rather than a limited window of context words, and learns to represent the context with distributed representations instead of predefined sparse feature vectors, enabling better generalization over contexts.

5.4 Approach

We use a neural network to estimate the conditional probability distribution of the suffix of the verb x_i at position i given the stem of the verb, the other words in the sentence, and the position of the verb in the sentence (see Equation 5.1). During prediction, we use this distribution to select the most likely homophone verb form.

$$P(\mathit{suffix}(x_i) \mid \boldsymbol{x}_{1..i-1}, \mathit{stem}(x_i), \boldsymbol{x}_{i+1..N}) \tag{5.1}$$

Figure 5.1 presents an architectural overview of the system. We train a neural network that estimates the conditional distribution over verb suffixes (no suffix, -t, -d, -e, -de, -te, -en, -den, -ten) given the stem of the verb, the other words in the sentence, and the position of the verb in the sentence. The network can be conceptually divided into three components: a verb encoder, which builds a representation for the stem of the input verb; a context encoder, which builds a representation for the sentence using the position i of the verb within this sentence; and a feed-forward neural network that fuses the verb and sentence representations and transforms them to a probability distribution over suffixes using a softmax layer.

Instead of predicting the suffixes, one could also imagine a model that estimates the probability of the entire verb given the rest of the sentence. There two important drawbacks to this approach: Firstly, such a model is much more computationally expensive as it predicts a distribution over all unique verbs. Secondly, it will be more challenging to spell check infrequent verbs because there is much less data to estimate their probability.

We also investigated an alternative framework where instead of predicting the suffix from the stem, we predict the edit rules that correct a potentially wrongly-spelled verb given its context. This approach has the drawback that it assumes a large annotated set of dt-mistakes. In a preliminary investigation (Laevaert, 2017), we created such a

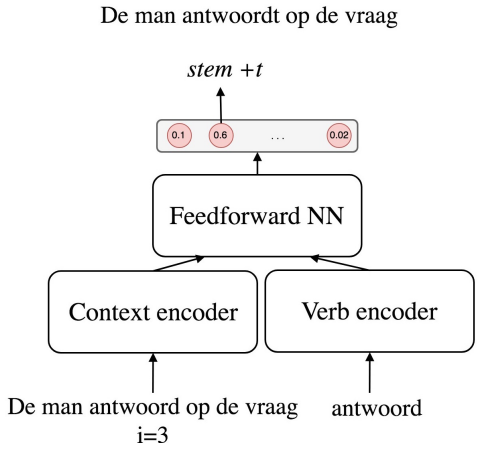


Figure 5.1: Architectural overview of the dt-corrector. The input to the system is the position of the verb *i* and the sentence where the input verb is replaced by its stem. Two neural networks are responsible for encoding the representations of the context and the verb respectively. The resulting representations are concatenated and fused by a feed-forward neural network and transformed to a probability distribution over suffixes using a softmax layer.

dataset by automatically introducing mistakes according to a handpicked distribution of dt-errors. However, we found that the performance drops significantly when the dt-error distributions of the training and test sets do not match, making it infeasible to build a practical system with this approach.

In this work we show it is possible to obtain a highly accurate dt-correction system without making assumptions about the distribution of dt-mistakes. In the remainder of this section, we describe our architecture in further detail.

5.4.1 Verb Stem Representation

We hypothesize that a good verb stem representation is useful for a) determining the suffix of past participles, this is either *-d*, *-t*, or *no suffix* depending on the last letter of the raw verb stem; and b) learning a bias towards the most used suffix for a given stem. In the Europarl dataset for instance, *wordt* is much more frequent than *word*. A bias towards the more frequent verb forms could benefit the precision of a system in cases where it is uncertain about how to interpret the input sentence. Based on this intuition we will experiment with three different verb stem representations:

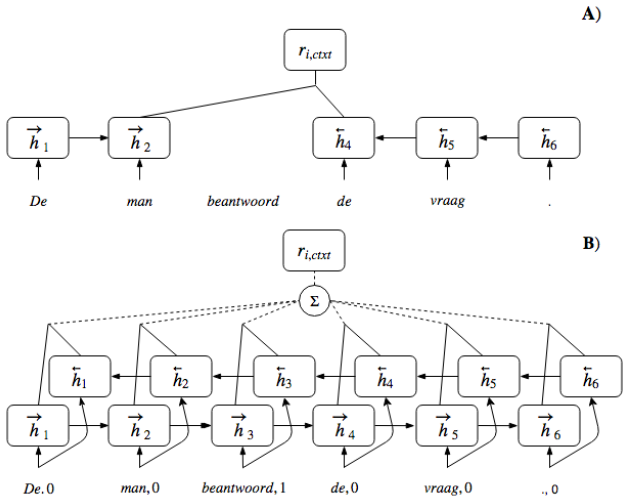


Figure 5.2: Illustration of the context encoders: A) BILSTM, B) BILSTM + ATTENTION.

– CHARS obtains a verb stem representation $\boldsymbol{r}_{i,stem}$ from the last state of an *LSTM* (Hochreiter and Schmidhuber, 1997) to which the verb stem is fed as a sequence of characters. Characters are represented as one-hot vectors;

– CHARS+WORD obtains a verb representation $\boldsymbol{r}_{i,stem}$ by concatenating the CHARS representation with a word embedding of the stem;

– LAST_CHAR+WORD obtains a verb representation $\boldsymbol{r}_{i,stem}$ by concatenating the last character of the stem embedded into a three-dimensional vector with a word embedding of the stem. The three-dimensional character embedding is motivated by the fact that characters can be grouped into three categories based on the suffix that should be added as a past participle.

5.4.2 Context Representation

From the context it should be clear which dt-rule applies (see Table 5.1). It should therefore encode information about the subject, the tense in which the verb is used, and the position of the verb w.r.t. the subject. We experiment with two different context representations (see Figure 5.2):

– *BiLSTM* (Figure 5.2A) obtains a context representation $\boldsymbol{r}_{i,ctxt}$ by concatenating the states, $\overrightarrow{\boldsymbol{h}}_{i-1}$ and $\overleftarrow{\boldsymbol{h}}_{i+1}$ of a bidirectional LSTMs (Schuster and Paliwal, 1997; Graves and Schmidhuber, 2005). A bidirectional LSTM consists of two LSTMs: $LSTM_{forward}$ computes its states $\overrightarrow{\boldsymbol{h}}_1 .. \overrightarrow{\boldsymbol{h}}_N$ by processing the words in order, $LSTM_{backward}$ computes $\overleftarrow{\boldsymbol{h}}_N .. \overleftarrow{\boldsymbol{h}}_1$ by processing the words in reverse order.[3] Each word x_j is represented by its word embedding $\boldsymbol{e}_j$ before feeding it to the LSTMs.

$$\boldsymbol{r}_{i,ctxt} = \overrightarrow{\boldsymbol{h}}_{i-1} \parallel \overleftarrow{\boldsymbol{h}}_{i+1} \tag{5.2}$$

$$\overrightarrow{\boldsymbol{h}}_j = LSTM_{forward}(\overrightarrow{\boldsymbol{h}}_{j-1}, \boldsymbol{e}_j) \tag{5.3}$$

$$\overleftarrow{\boldsymbol{h}}_j = LSTM_{backward}(\overleftarrow{\boldsymbol{h}}_{j+1}, \boldsymbol{e}_j) \tag{5.4}$$

Here $\parallel$ denotes concatenation.

– *BiLSTM + Attention* (Figure 5.2B) obtains a context representation $\boldsymbol{r}_{i,ctxt}$ by first encoding the sentence, for which the relevant verb is replaced by its stem, with a bidirectional LSTM, and then using an attention mechanism to extract the relevant information from its states $\boldsymbol{h}_{1..N}$. A word x_j is fed to the LSTM as the concatenation of its corresponding word embedding $\boldsymbol{e}_j$ and a binary indicator $I_{j=i}$ to identify the position of the verb.

  An attention mechanism summarizes a sequence of vectors to a single vector as a linear combination of these vectors. The motivation for an attention mechanism is twofold. Firstly, attention models are typically beneficial when dealing with long sequences. They could therefore perform better than *BiLSTM* in cases where the subject and the direct verb are far apart. *BiLSTM* always uses the states next to the verb stem, even if the subject is not located nearby. The attention mechanism, on the other hand, calculates a weighted combination of all the states. Second, an attention mechanism could help with gaining insights in what the model has learned. By visualizing the attention weights we might uncover what words/patterns were most relevant for building the context representation.

  There have been different attention mechanisms proposed in the literature. In this chapter, we use single-head attention with an additive attention scoring function (Bahdanau et al., 2015), the best performing attention mechanism for neural machine translation in the comparison of Britz et al. (2017).[4] The attention weights $\boldsymbol{\alpha}$ are calculated by normalizing the scores $score_1$, ..., $score_N$ calculated by the additive scoring function f_{add}, which compares a hidden state $\boldsymbol{h}_j$ with a query vector $\boldsymbol{q}$,

[3] Note that because the model only requires $\overrightarrow{\boldsymbol{h}}_{i-1}$ and $\overleftarrow{\boldsymbol{h}}_{i+1}$ we do not need to actually process the forward and backward sequences all the way to the end, see Figure 5.2.

[4] We also performed preliminary experiments with multi-head attention, but this yielded no improvements.

computes the dot product between the result (after passing it to a *tanh* activation function) and a vector $\boldsymbol{v}$. The vectors $\boldsymbol{q}$ and $\boldsymbol{v}$ are learned jointly with the rest of the network parameters.

$$\boldsymbol{r}_{i,ctxt} = attention(\boldsymbol{h}_{1..N}) \tag{5.5}$$

$$\boldsymbol{h}_j = \overrightarrow{\boldsymbol{h}}_j \parallel \overleftarrow{\boldsymbol{h}}_j \tag{5.6}$$

$$\overrightarrow{\boldsymbol{h}}_j = LSTM_{left}(\overrightarrow{\boldsymbol{h}}_{j-1}, \boldsymbol{e}_j) \tag{5.7}$$

$$\overleftarrow{\boldsymbol{h}}_j = LSTM_{right}(\overleftarrow{\boldsymbol{h}}_{j+1}, \boldsymbol{e}_j) \tag{5.8}$$

$$attention(\boldsymbol{h}_{1..N}) = \sum_{j=1}^{N} \alpha_j \boldsymbol{h}_j \tag{5.9}$$

$$\alpha_j = \frac{exp(score_j)}{\sum_{k=1}^{N} exp(score_k)} \tag{5.10}$$

$$score_j = f_{add}(\boldsymbol{q}, \boldsymbol{h}_j) \tag{5.11}$$

$$f_{add}(\boldsymbol{q}, \boldsymbol{h}_j) = \boldsymbol{v} \cdot tanh(\boldsymbol{W}_{add,q}\, \boldsymbol{q} + \boldsymbol{W}_{add,h}\, \boldsymbol{h}_j) \tag{5.12}$$

5.4.3 Transforming Representations to a Suffix Distribution

The verb and context representations are concatenated and projected to a C-dimensional vector by a feed-forward neural network, where C is equal to the number of suffixes. The softmax function normalizes the resulting vector to a probability distribution over suffixes. For our experiments the feed-forward neural network consists of a single, fully connected hidden layer with 128 dimensions and uses a ReLU activation function, which is defined as $max(0,x)$ and is the default recommendation in modern neural networks (Jarrett et al., 2009; Nair and Hinton, 2010; Glorot et al., 2011; Goodfellow et al., 2016).

$$P(y|x_{1..i-1}, stem(x_i), x_{i+1..N}, i) = softmax(\boldsymbol{l}_i) \tag{5.13}$$

$$\boldsymbol{l}_i = \boldsymbol{W}_l\, \boldsymbol{r}_i + \boldsymbol{b}_l \tag{5.14}$$

$$\boldsymbol{r}_i = feedForward(\boldsymbol{r}_{i,stem} \parallel \boldsymbol{r}_{i,ctxt}) \tag{5.15}$$

5.4.4 Training and Prediction

We use the cross-entropy with the empirical distribution of the training data $\mathscr{D}$ as the training objective:

$$\mathscr{L}(\mathscr{D}) = -\frac{1}{|\mathscr{D}|} \sum_{x,i \in \mathscr{D}} log\, P(suffix(x_i)|x_{1..i-1}, stem(x_i), x_{i+1..N}, i) \tag{5.16}$$

where $|\mathscr{D}|$ denotes the number of training examples

There are nine suffix classes in total, but during prediction we reduce this to only two relevant classes for each instance based on the observation that there can be at most one (real-word) homophone verb form for any given verb. For example *beantwoord* can only be confused with *beantwoordt*, not with for instance *beantwoorde* as their pronunciations differ; similarly, *beantwoorde* can only be confused with *beantwoordde*. Hence, during prediction we identify which two suffixes are relevant based on the input word[5] and force the system to predict the most likely of the two.

5.5 Dataset & Preprocessing

Training neural networks requires a significant amount of data. To our knowledge, no annotated corpus of dt-mistakes is available. By reducing the dt-correction task to a verb suffix prediction problem as described in the previous section, it suffices to have a large dataset of correctly written Dutch text. To this end, we used the Dutch portion of the Europarl corpus (Koehn, 2005), which is written by language professionals.

To have a reliable evaluation of dt-correction systems, we require a significant number of sentences with annotated dt-mistakes. We have therefore decided to automatically introduce mistakes in 20,000 Europarl sentences to create a validation and test set of 10,000 sentences each. To verify how well a system generalizes on out-of-domain test data and to enable a fair comparison with existing systems, we construct three additional out-of-domain test sets from online verb spelling tests. The remainder of this section discusses the different preprocessing steps and our methodology for creating validation and test sets. The datasets and preprocessing scripts are available at `http://liir.cs.kuleuven.be/software.php`.

[5] The last letter of the stem and the suffix of the input word uniquely determine the verb forms that can be confused.

5.5.1 Identifying and Stemming Verbs

We automatically identify verbs in a sentence based on their part-of-speech (PoS) tags using TreeTagger (Schmid, 1994).[6] For some verbs we can easily derive their stem based on their last character(s), e.g., if a verb ends with $-dt$ we can obtain the stem by removing the t at the end. Other word endings are ambiguous, however (e.g., a d-suffix could be the result of rule 3 of Table 5.1, in which case we can not be certain if d is part of the stem). From the infinitive of the verb, in contrast, we can determine the stem unambiguously. We therefore automatically obtain the lemmas using TreeTagger. For verbs for which the lemma is unknown to the tagger, we derive the stem from the dt-rules if possible (e.g., when a verb ends with $-dt$).

5.5.2 Identifying Relevant Verbs

Not all verbs are relevant for training and evaluating a dt-correction system. For evaluation purposes, we are only interested in dt-homophones as these are the verbs for which spelling errors occur. We thus automatically construct a list of dt-homophones, which is used to filter verbs during evaluation and testing. For the training data, there is no need to be so restrictive: as the dt-rules apply for the majority of Dutch verbs, we can significantly increase the amount of training data if we also incorporate regular, non-dt verbs. To this end, in addition to dt-homophones, we use all regular verbs for which TreeTagger can obtain the lemma so that the stem can be determined reliably. We verify if a verb is regular, by constructing a set of irregular verb forms from online grammar sources [7,8], and by checking verbs against this list.

5.5.3 A Generative Process for Dt-Mistakes

To create the validation and test datasets, we need a scheme to introduce dt-mistakes in text. Cognitive research on dt-mistakes has found that people tend to use the more frequent spelling of a homophone verb form when they are distracted or put under time pressure (Verhaert and Sandra, 2016). Based on this insight, we propose a generative process for dt-mistakes, summarized in Algorithm 4.

From every sentence s in the text corpus C, the verbs are identified using TreeTagger. For every verb that has a homophone spelling, we sample a binary variable $i_{focused}$ from a Bernoulli distribution that indicates whether a person is focused when writing the verb. In the scenario where the person is distracted (i.e., $i_{focused} = 0$), we sample the

[6] Frog (Van den Bosch et al., 2007) could have been a good alternative to TreeTagger.

[7] https://educatie-en-school.infonu.nl/taal/28516-regelmatige-en-onregelmatige-werkwoorden-in-de-ott.html

[8] http://users.telenet.be/orandago/nederlands/ww.doc

Algorithm 4: GENERATIVE PROCESS FOR INTRODUCING DT-MISTAKES MOTIVATED BY COGNITIVE INSIGHTS.

initialize: the focus parameter p_f;
for s **in** C **do**
  V = identify_verbs(s)
  for v **in** V **do**
    sample $i_{focused} \sim Bernoulli(p_f)$
    if not *has_homophone_spelling(v)* **or** $i_{focused} = 1$ **then**
      continue
    v_{alt} = $homophone\_spelling(v)$
    sample $p_{error} \sim Beta(freq(v_{alt}), freq(v)))$
    sample $i_{error} \sim Bernoulli(p_{error})$
    if $i_{error} = 1$ **then**
      $replace(s, v, v_{alt})$

probability p_{error} that the person will use the wrong spelling from a Beta distribution where the frequencies of the two spellings are used as the concentration parameters.[9] p_{error} is used as the parameter of a Bernoulli distribution to sample another binary indicator i_{error}. When $i_{error} = 1$, we replace v by its homophone counterpart v_{alt}, hereby introducing a dt-error. This process ensures that we will never introduce mistakes that lead to words that do not exist in Dutch and that mistakes where the used verb form is dominant to the correct verb form are more likely, hence incorporating the insights of Verhaert and Sandra (2016).

We note that the generative process could potentially be made more cognitively plausible by incorporating contextual information as it has been shown that the immediate context in which homophones occur also plays a role in their selection (Daelemans and van den Bosch, 2007). As the main goal of this work is to build a successful dt-correction system, a rigorous comparison of different generative processes falls out of our scope. We leave this for further research.

5.5.4 Out-of-domain Test Sets

To allow an unbiased comparison with existing spell checkers, we construct three out-of-domain (i.e., not related to the domain of the Europarl dataset) test sets from online spelling tests containing 20 verb conjugation exercises each: 1) *Nooit meer dt-fouten* from *de Standaard* (a Flemish news paper)[10]; 2) the *HBO taaltoets, spelling*

[9] Note that when someone is distracted he/she can still use the correct spelling.
[10] https://www.standaard.be/cnt/dmf20141103_01356248

werkwoordsvormen from *Uitgeverij Pak* (a Dutch publishing house)[11]; and 3) a test from *Nederlandse taaltest* also from *Uitgeverij Pak*[12]. The exercises use fill-in-the-blanks type of questions, where a person has to add the right suffix to a given verb stem within a given sentence. To be able to compare with other spelling checkers, we manually fill-in the blanks such that we get the wrongly-spelled homophones. For seven of the sixty verbs no real-word homophone existed (e.g., for *stond*: *stondt* and *stont* are not correct Dutch words). These are retained from the test sets because replacing them would result in context-independent errors.

5.5.5 Statistics

Statistics about the distribution of suffixes in the resulting train, development and test sets are shown in Table 5.2.

dataset	-"	-t	-d	-e	-de	-te	-en	-den	-ten	verbs	errors
train	0.31m	0.76m	0.46m	27k	0.10m	24k	1.1m	22k	2.4k	2.8m	n/a
dev	0.93k	6.7k	3.8k	65	0	3	0.31k	0	0	12k	1.7k
test	0.89k	6.8k	3.9k	69	1	6	0.28k	3	0	12k	1.7k
dS	1	10	9	0	0	0	0	0	0	20	20
HBO	2	5	2	2	0	1	1	1	2	16	16
NLTT	2	8	1	2	0	1	1	1	1	17	17

Table 5.2: Label distribution of the train, development, and test sets. -" denotes that the verb only consists of the stem. The *train*, *dev*, and *test* refer to the train, development and test splits of the Europarl corpus.

5.6 Experimental Setup

We used 100-dimensional word embeddings, pre-trained with continuous Skip-gram using the `word2vec` toolkit with default hyper-parameters. The LSTMs consist of two layers and 128 memory cells for each layer. We use dropout regularization (Srivastava et al., 2014) with dropout probability 0.1. For the LSTMs we use variational recurrent dropout (Gal and Ghahramani, 2016).

We use the Adam optimizer with the recommended hyper-parameters (Kingma and Ba, 2015) and train for a maximum of 500,000 iterations with a mini-batch size of 100,

[11] http://www.hbotaaltoets.nl/spelling-werkwoordsvormen
[12] http://www.nederlandsetaaltest.nl/spellingtest-werkwoorden

saving checkpoints every 1,000 iterations and selecting the checkpoint with the best F1-score on the development set for evaluation on the test sets.

In preliminary experiments, we found that training on plural verb forms (ending with *-en, -den, -ten*) significantly elongates the training time without having a beneficial impact on dt-correction performance due to the large imbalance between the label frequencies of *-en*, *-den* and *-ten* in the training set (see Table 5.2). We therefore did not train on the plural forms in the experiments reported below. During prediction, the system will not check the spelling of plural forms.

We use the following evaluation metrics (Reynaert, 2008): accuracy (*acc*), precision (*prec*), recall (*rec*), and F_1 (F_1). In this context we define true positives (tp) as the instances where the system correctly changes the input spelling; false positives (fp) are the cases where the system introduces an error; false negatives (fn) are wrongly spelled verbs which the system did not correct; true negatives (tn) are correctly spelled which the system left untouched.

$$acc = \frac{tp + tn}{tp + tn + fp + fn} \tag{5.17}$$

$$prec = \frac{tp}{tp + fp} \tag{5.18}$$

$$rec = \frac{tp}{tp + fn} \tag{5.19}$$

$$F_1 = \frac{2\ prec \cdot rec}{prec + rec} \tag{5.20}$$

5.7 Experiments

5.7.1 Influence of Spelling Errors on the PoS-tagger

In this experiment, we verify the robustness of the PoS-tagger w.r.t. dt-spelling errors. Using the development set with annotated dt-mistakes, we verified if TreeTagger could still assign the same PoS tag when a dt-verb was replaced by its homophone. We found that 98.5 % of the wrongly-spelled verbs are still identified as verbs. However, for only 19.5 % of the wrongly-spelled verbs, the correct tense was preserved. In the majority of the cases the tagger will assign the tense that is compatible with the wrongly-spelled verb. This is unsurprising as PoS-taggers are typically trained on high-quality corpora, with a low amount of spelling errors. Hence, taggers trained on such corpora will have

learned that the verb form is a strong predictor for its tense. From this result it is clear that the accuracy of the tenses for wrongly-spelled verbs as recognized by TreeTagger is far too low to be useful in a dt-correction system.

5.7.2 Context-Agnostic Baselines

In this experiment, we investigate the performance of models that do not use context. We compare four models: *majority class* uses the input verb to decide which homophone suffixes it should consider (see Subsection 5.4.4) and then picks the suffix that is most frequent in the training data; the other three models use the different verb representations introduced in Subsection 5.4.1: *chars*, *last_char+word*, and *chars+word*. The results are reported in Table 5.3. We see that the best context-agnostic models (*chars* and *last_char+word*) obtain an F1-score 55.59 on the test set. There is not a lot of difference between verb representations. When comparing *majority class* with the other models, we find support for the hypothesis that using verb representations can help to improve the precision of the system (see Subsection 5.4.1).

	F_1		*acc*		*prec*		*rec*	
model/dataset	dev	test	dev	test	dev	test	dev	test
majority class	26.98	25.81	60.06	60.05	18.38	17.42	50.72	49.79
chars	**56.13**	**55.59**	88.78	88.95	65.11	63.32	49.33	49.55
last_char+word	56.02	**55.59**	88.81	89.04	65.4	64.01	48.99	49.13
chars+word	55.84	55.5	88.77	89.02	65.22	63.88	48.81	49.07

Table 5.3: Results of baseline models that do not use context, evaluated on the Europarl development (dev) and test (test) sets.

5.7.3 Context and Stem Encodings

In this experiment, we explore the different combinations of verb and context representations introduced in Subsections 5.4.1 and 5.4.2. The results are displayed in Table 5.4. We find that models that use attention to encode the context outperform the other models for all verb representations. The verb representation does not seem to have a large impact on the results. With attention, *chars+word* obtains the best results on the development set, but on the test set there are no significant differences between the three encodings. All six architectures' results exhibit large performance gains over the baselines that do not use context (Table 5.3). This shows that models are successfully leveraging the context information.

The fact that these neural networks only use the stem of the verb that is being checked makes these results particularly impressive. It means that the system output is largely independent of the number of mistakes in the input spelling.[13] We can hence expect nearly identical accuracy scores in test sets with a higher or lower ratio of mistakes.

	F_1		*acc*		*prec*		*rec*	
model/dataset	dev	test	dev	test	dev	test	dev	test
BiLSTM+chars	97.98	97.44	99.42	99.29	98.88	97.88	97.1	97
BiLSTM+last_char+word	97.77	97.43	99.36	99.29	98.99	98.17	96.58	96.7
BiLSTM+chars+word	97.75	97.61	99.35	99.34	98.64	98.35	96.87	96.88
BiLSTM+Attn+chars	98.07	**97.85**	99.44	99.4	98.77	98.6	97.39	97.12
BiLSTM+Attn+last_char+word	98.19	97.82	99.48	99.4	99.11	98.48	97.28	97.18
BiLSTM+Attn+chars+word	**98.33**	**97.85**	99.52	99.4	99.29	98.48	97.39	97.24

Table 5.4: Comparison of combinations of verb and context encodings on *Europarl-dev* (dev) and *Europarl-test* (test).

5.7.4 Extending the Training Data

In this experiment, we analyzed the effect of omitting non-dt verbs from the training data (all other experiments include regular, non-dt verbs in the training set, see Subsection 5.5.2). We find F_1-scores of 97.75 and 97.52 on the Europarl development and test sets respectively. This is a decrease compared to when the model is also trained on non-dt verbs, which obtained F_1-scores of 98.33 and 97.85 (see Table 5.4). This confirms the hypothesis that a model that is also trained on non-dt verbs generalizes better.

5.7.5 Comparison with Existing Systems

In this experiment, we compare our system with existing grammar and spell checking systems on the out-of-domain test sets: *de Standaard (dS)*, *HBO taaltoets* (HBO) , and *Nederlandse taaltest* (NLTT). We compare with the following systems:

– *Microsoft Office Word*, we experimented with three versions: Office Professional Plus 2013; Office 365, desktop version; and Office 365, Word online[14]. We report the results for Office 365, Word online which demonstrated the best performance.

[13] With exception of the plural forms (-en, -ten, -den) for which we always leave the input spelling unchanged.

[14] Office 365 was tested in April 2018.

– *Schrijfassistent* (D'Hertefelt et al., 2014; Houthuys, 2016; 201, 2016)[15], a tool for checking grammar, writing style, and spelling in Dutch developed by a collaboration of *het Instituut Levende Talen, KU Leuven*; *de Standaard* (a Flemish newspaper); and *VRT* (the Flemish public broadcasting organization).

– *languatool.org* [16], an open-source grammar, writing style, and spelling checker for several languages including Dutch. For dt-correction it relies on a rule-based system that looks for patterns in the PoS-tags of sentences (OpenTaal, 2014).

– *valkuil.net*, an online spell checker for Dutch which relies on a data-driven approach for dt-rules. This system uses a statistical, context-based system which uses a fixed context window of four words to identify the correct verb form (Stehouwer and Van den Bosch, 2009).

Table 5.5 reports the results. We find that *BiLSTM+Attn+chars+word* outperforms all the other grammar and spelling checkers by a significant margin on all test sets. It is particularly effective on the *de Standaard* test set, where it obtains a perfect score. For the other two test sets the system has perfect precision[17], though the recall is significantly lower than for *de Standaard*. This is due to the fact that these test sets query all verb forms (i.e., verbs ending with *-en, -e, -d,* or *-t*), whereas the *de Standaard* test set focuses on verbs that end with *-d* or *-t*. Homophones that end with *-ten, -den, -te, -de, -e* occur much less frequently and thus are more difficult for our system to predict correctly.[18] Furthermore, some of the input verb forms did not occur at all in the Europarl corpus, and will therefore not be recognized as a dt-verb.

The performance improvement with respect to the existing spell checkers comes from a higher recall. We found that the spell checkers were unable to find any of the errors where the misspelled verb and the relevant context words (e.g., auxiliary verb or subject) were not adjacent to each other. It is striking that valkuil.net did not detect any of the mistakes in the test sets. This is probably related to the fact that the system was tuned to have high precision.[19] The higher recall scores obtained by *BiLSTM+Attn+chars+word* support the intuition that learning distributed representations of the whole context yields better generalization than sparse one-hot representations of fixed context windows. A straightforward way to further improve the recall of our system is to extend our training corpus with other high-quality Dutch texts such as news articles. Furthermore, it could be beneficial to consider a strategy to subsample the frequent suffix classes to boost the recall for the infrequent suffixes.

[15] http://schrijfassistent.standaard.be/index.php

[16] https://languagetool.org/nl

[17] It should be noted that due to the large number of errors in the test sets, high precision scores are not surprising as there is little opportunity to introduce errors in correctly written words.

[18] Recall from the experimental setup that the system was in fact not trained on plural forms because verbs that end with *-ten* and *-den* are too infrequent.

[19] http://valkuil.net/info

Table 5.5: Comparison with other grammar and spelling checkers.

	F_1			*prec*			*rec*		
model/dataset	dS	HBO	NLTT	dS	HBO	NLTT	dS	HBO	NLTT
language tool	0.00	0.00	11.11	0.00	0.00	100.00	0.00	0.00	5.88
Schrijfassistent	33.33	30.00	38.10	100.00	75.00	100.00	20.00	18.75	23.53
valkuil.net	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MS Word	26.09	11.76	21.05	100.00	100.00	100.00	15.00	6.25	11.76
BiLSTM+Attn +chars+word	**100.00**	**57.14**	**66.67**	100.00	100.00	100.00	100.00	40.00	50.00

5.7.6 Acquiring Insight in the Model Predictions

From previous experiments it is clear that the models have learned to successfully leverage context for making accurate dt-corrections. The aim of this experiment is to uncover *how* the models are using the context. In particular, we are interested in measuring the impact of each word in the sentence on the prediction. We experimented with two alternative *word saliency* metrics:

- The weight α_j that the attention mechanism assigns to the state corresponding to the word x_j.
- The difference in the probability of the predicted class when we dropout the word embedding $\boldsymbol{e}_j$ of x_j (i.e., replacing $\boldsymbol{e}_j$ with a vector with all zeros) compared to normal prediction, where we use all words.

We validated the effectiveness of these word saliency metrics by visualizing their output and manually inspecting if it provides a plausible explanation for the model's prediction. We found that the dropout-based probability model yields intuitive results. Figure 5.3 shows heatmaps for the dropout-based word saliency metric for sentences of the *de Standaard* test dataset. For finite verbs in present tense the word with the largest impact is the subject, for past participles this is the auxiliary verb. For the attention-based metric the results were not intuitive. The fact that attention weights have so little explanatory value is surprising, it indicates that the states to which the neural network attends still encode much relevant information from previous and/or subsequent states.

5.8 Conclusion

In this work, we introduced a new approach to automatic correction of context-dependent dt-mistakes, one of the most frequent spelling errors in the Dutch language.

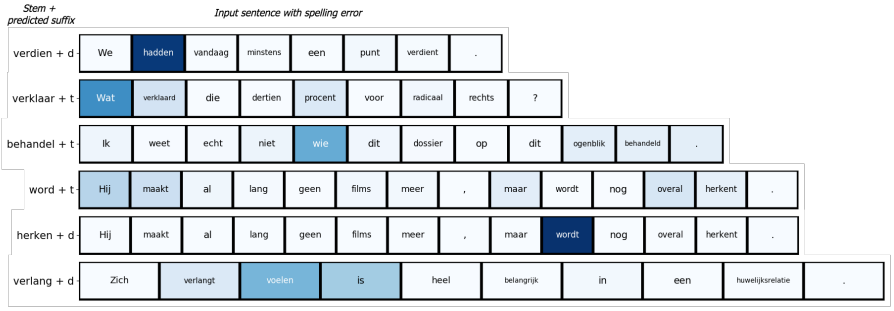


Figure 5.3: Heatmaps of the impact of each word on the prediction of the neural network for wrongly spelled dt-verbs from the *De Standaard* test set. Impact of a word is measured by the decrease in probability mass of the predicted class when we mask the word in the context encoding (i.e., replace its word embedding by a vector of zeros).

By learning a neural network to estimate the probability distribution of a verb's suffix conditioned on its stem and the context in which it occurs, we were able to build a spelling correction system that achieves state-of-the-art performance and perfect precision on three different benchmarking datasets. The method does not require annotated training examples and only relies on basic preprocessing tools to tokenize the text and identify verbs, which allows for training on millions of examples. Furthermore, we proposed a method to determine which words in a sentence cause the system to make corrections, which can be a valuable way of providing feedback to the user. We still see room for further improvement for verb suffixes that occur less frequently (*-den*, *-ten*, *-de*, *-te*, *-e*). A strategy where the more frequent suffix classes are subsampled may help dealing with these cases.

This chapter is accepted for publication:

Geert Heyman, Ivan Vulić, Yannick Laevaert, Marie-Francine Moens. Automatic detection and correction of context-dependent dt-mistakes using neural networks. *CLIN Journal*, 2018

Chapter 6

A Deep Learning Approach to Bilingual Lexicon Induction

In this chapter, we take a representation learning approach to tackle bilingual lexicon induction in domain-specific corpora. BLI was already addressed as a part of Chapter 4, though in this chapter we study BLI with the use-case of domain-specific terminology extraction in mind. As a result, there are several important differences w.r.t. Chapter 4. In this chapter:

- We build a system for extracting words and phrases from up to five words, whereas previously we translated individual words;
- We focus on BLI from domain-specific corpora, which are small in comparison to the large-scale corpora used in Chapter 4;
- The words and phrases are also more morphology-rich because we work with Wikipedia articles in the medical domain, a dynamic area with an abundance of terminology;
- The BLI evaluation sets are a random sample of the words/phrases in the vocabulary, whereas in the BLI evaluation sets in Chapter 4 are biased towards the most frequent words;

For these reasons, we propose a new RL-based BLI. Casting BLI to a classification problem, we combine supervised and unsupervised representation learning techniques: word-level representations are trained on monolingual text corpora without supervision,

character-level representations and their combination with the word-level representations are learned from a bilingual seed lexicon. We find that the inclusion of character-level information results in a significant performance increase and that the RL-based character-level representations are superior to hand-crafted representations that were previously used.

6.1 Introduction

As a result of the steadily growing process of globalization, there is a pressing need to keep pace with the challenges of multilingual international communication. New technical specialized terms such as biomedical terms are generated on almost a daily basis, and they in turn require adequate translations across a plethora of different languages. Even in local medical practices we witness a rising demand for translation of clinical reports or medical histories (Randhawa et al., 2013). Unfortunately, the most comprehensive specialized biomedical lexicons in the English language such as the Unified Medical Language System (UMLS) thesaurus lack translations into other languages for many of the terms.[1] Building bilingual lexicons that contain such terminology by hand is time-consuming and requires trained experts.

As a consequence, we observe interest in automatically learning the translation of terminology from a corpus of domain-specific bilingual texts (Bollegala et al., 2015). What is more in specialized domains such as biomedicine, parallel corpora are often not readily available: therefore, translations are mined from non-parallel, comparable bilingual corpora (Kontonatsios et al., 2014a; Xu et al., 2015). In a comparable corpus, the texts in source and target language contain similar content, but are not exact translations of each other: as an illustration, Figure 6.1 shows a fragment of the biomedical comparable corpus we used in our experiments. In this chapter, we propose a deep learning approach to bilingual lexicon induction (BLI) from a comparable biomedical corpus.

Neural network based deep learning models (LeCun et al., 2015) have become popular in natural language processing tasks. One motivation is to ease feature engineering by making it more automatic or by learning end-to-end. In natural language processing it is difficult to hand-craft good lexical and morpho-syntactic features, which often results in complex feature extraction pipelines. Deep learning models have also made their breakthrough in machine translation (Sutskever et al., 2014; Bahdanau et al., 2015), hence our interest in using deep learning models for the BLI task. Neural networks are typically trained using a large collection of texts to learn distributed representations that capture the contexts of a word. In these models, a word can be

[1] For instance, UMLS currently spans only 21 languages, and only 1.82% of all terms are provided in French.

EN	NL
digitoxin	digitoxine
digoxin	digoxine
cardiac glycoside	hartglycoside

Digitoxin is a cardiac glycoside. It has similar structure and effects to digoxin (though the effects are longer-lasting). Unlike digoxin (which is eliminated from the body via the kidneys), it is eliminated via the liver, so could be used in patients with poor or erratic kidney function. However, it is now rarely used in current Western medical practice. While several controlled trials have shown digoxin to be effective in a proportion of patients treated for heart failure, the evidence base for digitoxin is not as strong, although it is presumed to be similarly effective.

...

Digitoxine is een hartglycoside dat kan geïsoleerd worden uit vingerhoedskruid. Dit wordt gedaan door eerst de bladeren en de knoppen te drogen en deze te vermalen. Vervolgens kan digitoxine door middel van extractie uit de plant worden gehaald. Het is therapeutisch vrijwel volledig verdrongen door digoxine. De structuur lijkt veel op die van digoxine. Het verschil is het aantal hydroxylgroepen en waar deze gebonden zijn. Digitoxine bindt voornamelijk aan plasma-eiwitten, hierdoor kan het gebonden deel de bloedbaan niet verlaten en niet op de receptoren binden.

...

Figure 6.1: Excerpts of the English-Dutch comparable corpus in the biomedical domain that we used in the experiments with a few domain-specific translations indicated in red.

represented as a low-dimensional vector (often referred to as a word embedding) which embeds the contextual knowledge and encodes semantic and syntactic properties of words stemming from the contextual distributional knowledge (Mikolov et al., 2013b).

Lately, we also witness an increased interest in learning character representations, which better capture morpho-syntactic properties and complexities of a language. What is more, the character-level information seems to be especially important for translation mining in specialized domains such as biomedicine as such terms often share common roots from Greek and Latin (see Figure 6.1), or relate to similar abbreviations and acronyms.

Following these assumptions, in this article we propose a novel method for mining translations of biomedical terminology: the method integrates character-level and word-level representations to induce an improved bilingual biomedical lexicon.

6.2 Background and Contributions

BLI in the Biomedical Domain Bilingual lexicon induction (BLI) is the task of inducing word translations from raw textual corpora across different languages. Many information retrieval and natural language processing tasks benefit from automatically induced bilingual lexicons, including multilingual terminology extraction (Bollegala et al., 2015), cross-lingual information retrieval (Lavrenko et al., 2002; Levow et al., 2005; Vulić and Moens, 2015; Mitra et al., 2016), statistical machine translation (Och and Ney, 2003; Zou et al., 2013), or cross-lingual entity linking (Tsai and Roth, 2016). Most existing works in the biomedical domain have focused on terminology extraction from biomedical documents but not on terminology translation. For instance, Hellrich and Hahn (2014) use a combination of off-the-shelf components for multilingual terminology extraction but do not focus on learning terminology translations. The OntoLearn system extracts terminology from a corpus of domain texts and then filters the terminology using natural language processing and statistical techniques, including the use of lexical resources such as WordNet to segregate domain-general and domain-specific terminology (Navigli et al., 2003). The use of word embeddings for the extraction of domain-specific synonyms was probed by Wang et al. (2015).

Other works have focused on machine translation of biomedical documents. For instance, Wołk and Marasek (2015) compared the performance of neural-based machine translation with classical statistical machine translation when trained on European Medicines Agency leaflet texts, but did not focus on learning translations of medical terminology. Recently, Afzal et al. (2015) explored the use of existing word-based automated translators, such as Google Translate and Microsoft Translator, to translate English UMLS terms into French and to expand the French terminology, but do not construct a novel methodology based on character-level representations as we propose

in this thesis. Most closely related to our work is perhaps Xu et al. (2015), where a label propagation algorithm was used to find terminology translations in an English-Chinese comparable corpus of electronic medical records. Different from the work presented in this paper, they relied on traditional co-occurrence counts to induce translations and did not incorporate information on the character level.

BLI and Word-Level Information Traditional bilingual lexicon induction approaches aim to derive cross-lingual word similarity from either context vectors, or bilingual word embeddings. The context vector of a word can be constructed from (1) weighted co-occurrence counts (Rapp, 1995; Fung and Yee, 1998; Gaussier et al., 2004; Laroche and Langlais, 2010; Vulić and Moens, 2013a; Kontonatsios et al., 2014b; Bollegala et al., 2015, *inter alia*), or (2) monolingual similarities (Koehn and Knight, 2002; Vulić and Moens, 2013b; Vulić et al., 2011; Liu et al., 2013) with other words.

The most recent BLI models significantly outperform traditional context vector-based baselines (Gaussier et al., 2004; Tamura et al., 2012) using bilingual word embeddings (BWE) (Baroni et al., 2014). All BWE models learn a distributed representation for each word in the source- and target-language vocabularies as a low-dimensional, dense, real-valued vector. These properties stand in contrast to traditional count-based representations, which are high-dimensional and sparse. The words from both languages are represented in the same vector space by using some form of bilingual supervision (e.g., word-, sentence- or document-level alignments) (Zou et al., 2013; Mikolov et al., 2013c; Hermann, 2014; Chandar et al., 2014; Søgaard et al., 2015; Gouws et al., 2015; Coulmance et al., 2015; Vulić and Moens, 2016b; Duong et al., 2016, inter alia).[2] In this cross-lingual space, similar words, regardless of the actual language, obtain similar representations.

To compute the semantic similarity between any two words, a similarity function, for instance cosine, is applied on their bilingual representations. The target language word with the highest similarity score to a given source language word is considered the correct translation for that source language word. For the experiments in this paper, we use two BWE models that have obtained strong BLI performance using a small set of translation pairs (Mikolov et al., 2013c), or document alignments (Vulić and Moens, 2016b) as their bilingual signals.

The literature has investigated other types of word-level translation features such as raw word frequencies, word burstiness, and temporal word variations (Irvine and Callison-Burch, 2016). The architecture we propose enables incorporating these additional word-level signals. However, as this is not the main focus of our research, it is left for future work.

[2]Note that since this research was completed the field has further advanced. We refer to recent comparative studies (Upadhyay et al., 2016; Vulić and Korhonen, 2016; Ruder et al., 2018) for a thorough explanation and analysis of the differences between recent BWE models.

BLI and Character-Level Information Etymologically similar languages with shared roots such as English-French or English-German often contain word translation pairs with shared character-level features and regularities (e.g., *accomplir:accomplish*, *inverse:inverse*, *Fisch:fish*). This orthographic evidence comes to the fore especially in domains such as legal domain or biomedicine. In such expert domains, words sharing their roots, typically from Greek and Latin, as well as acronyms and abbreviations are abundant. For instance, the following pairs are English-Dutch translation pairs in the biomedical domain: *angiography:angiografie*, *intracranial:intracranieel*, *cell membrane:celmembraan*, or *epithelium:epitheel*. As already suggested in prior work, such character-level evidence often serves as a strong translation signal (Haghighi et al., 2008; Claveau, 2008). BLI typically exploits this through string distance metrics: for instance, Longest Common Subsequence Ratio (LCSR) has been used (Melamed, 1995; Koehn and Knight, 2002), as well as edit distance (Mann and Yarowsky, 2001; Haghighi et al., 2008). What is more, these metrics are not limited to languages with the same script: their generalization to languages with different writing systems has been introduced by Irvine and Callison-Burch (2016). Their key idea is to calculate normalized edit distance only after transliterating words to the Latin script.

As mentioned, previous work on character-level information for BLI has already indicated that character-level features often signal strong translation links between similarly spelled words. However, to the best of our knowledge, our work is the first which learns bilingual character-level representations from the data in an automatic fashion. These representations are then used as one important source of translation knowledge in our novel BLI framework. We believe that character-level bilingual representations are well suited to model biomedical terminology in bilingual settings, where words with common Latin or Greek roots are typically encountered (Montalt Resurrecció and González-Davies, 2014). In contrast to prior work, which typically resorts to simple string similarity metrics (e.g., edit distance (Navarro, 2001)), we demonstrate that one can induce bilingual character-level representations from the data using state-of-the-art neural networks.

Framing BLI as a Classification Task Bilingual lexicon induction may be framed as a discriminative classification problem, as recently proposed by Irvine and Callison-Burch (2016). In their work, a linear classifier is trained which blends translation signals as similarity scores from heterogeneous sources. For instance, they combine translation indicators such as normalized edit distance, word burstiness, geospatial information, and temporal word variation. The classifier is trained using a set of known translation pairs (i.e., training pairs). This combination of translation signals in the supervised setting achieves better BLI results than a model which combines signals by aggregating mean reciprocal ranks for each translation signal in an unsupervised setting. Their model also outperforms a well-known BLI model based on matching canonical correlation analysis from Haghighi et al. (2008). One important drawback of Irvine

and Callison-Burch (2016) concerns the actual fusion of heterogeneous translation signals: they are transformed to a similarity score and weighted independently. Our classification approach, on the other hand, detects word translation pairs by learning to combine word-level and character-level signals in the joint training phase.

Contributions The main contribution of this work is a *novel bilingual lexicon induction framework*. It combines character-level and word-level representations, where both are automatically extracted from the data, within a discriminative classification framework. Similarly to a variety of bilingual embedding models (Ruder et al., 2018), our model requires translation pairs as a bilingual signal for training. However, we show that word-level and character-level translation evidence can be effectively combined within a classification framework based on deep neural nets. Our state-of-the-art methodology yields strong BLI results in the biomedical domain. We show that incomplete translation lists (e.g., from general translation resources) may be used to mine additional domain-specific translation pairs in specialized areas such as biomedicine, where seed general translation resources are unable to cover all expert terminology. In sum, the list of contributions is as follows.

First, we show that bilingual character-level representations may be induced using an RNN model. These representations serve as better character-level translation signals than previously used string distance metrics. Second, we demonstrate the usefulness of framing term translation mining and bilingual lexicon induction as a discriminative classification task. Using word embeddings as classification features leads to improved BLI performance when compared to standard BLI approaches based on word embeddings, which depend on direct similarity scores in a cross-lingual embedding space. Third, we blend character-level and word-level translation signals within our novel deep neural network architecture. The combination of translation clues improves translation mining of biomedical terms and yields better performance than "single-component" BLI classification models based on only one set of features (i.e., character-level or word-level). Finally, we show that the proposed framework is well suited for finding *multi-word translations pairs* which are also frequently encountered in biomedical texts across different languages.

6.3 Methods

As mentioned, we frame BLI as a classification problem as it supports an elegant combination of word-level and character-level representations.

Let V^S and V^T denote the source and target vocabularies respectively, and C^S and C^T denote the sets of all unique source and target characters. The vocabularies contain all unique words in the corpus as well as phrases (e.g., *autoimmune disease*) that are

automatically extracted from the corpus. We use p to denote a word or a phrase. The goal is to learn a function $g : X \rightarrow Y$, where the input space X consists of all candidate translation pairs $V^S \times V^T$ and the output space Y is $\{-1, +1\}$. We define g as:

$$g(p^S, p^T) = \begin{cases} +1 & \text{, if } f(p^S, p^T) > t \\ -1 & \text{, otherwise} \end{cases}$$

Here, f is a function realized by a neural network that produces a classification score between 0 and 1; t is a threshold tuned on a validation set. When the neural network is confident that p^S and p^T are translations, $f(p^S, p^T)$ will be close to 1. The motivation for placing a threshold t on the output of f is twofold. First, it allows balancing between recall and precision. Second, the threshold naturally accounts for the fact that words might have multiple translations: if two target language words/phrases p_1^T and p_2^T both have high scores when paired with p^S, both may be considered translations of p^S.

Note that the classification approach is methodologically different from the classical *similarity-driven* approach to BLI based on a similarity score in the shared bilingual vector space. The cross-lingual similarity between words p^S and p^T is computed as $SF(\boldsymbol{r}_p^S, \boldsymbol{r}_p^T)$, where $\boldsymbol{r}_p^S$ and $\boldsymbol{r}_p^T$ are word/phrase representations in the shared space, and SF denotes a similarity function operating in the space (cosine similarity is typically used[3]). A target language term p^T with the highest similarity score $\arg\max_{p^T} SF(\boldsymbol{r}_p^S, \boldsymbol{r}_p^T)$ is then taken as the correct translation of a source language word p^S.

Since neural network parameters are trained using a set of translation pairs D_{lex}, f in our classification approach can be interpreted as an automatically trained similarity function. For each positive training translation pair $< p^S, p^T >$, we create $2N_s$ *noise* or *negative* training pairs. These negative samples are generated by randomly sampling N_s target language words/phrases $p_{neg,S,i}^T$, $i = 1, \ldots, N_s$ from V^T and pairing them with the source language word/phrase p^S from the true translation pair $< p^S, p^T >$.[4] Similarly, we randomly sample N_s source language words/phrases $p_{neg,T,i}^S$ and pair them with p^T to serve as negative samples. We then train the network by minimizing the cross-entropy loss, a commonly used loss function for classification that optimizes the likelihood of the training data. The loss function is expressed by Equation (6.1), where D_{neg} denotes the set of negative examples used during training, and where y denotes the binary label for $< p^S, p^T >$ (1 for valid translation pairs, 0 otherwise).

$$\mathcal{L}_{ce} = \sum_{< p^S, p^T > \in D_{lex} \cup D_{neg}} -y \log(f(p^S, p^T)) - (1-y) \log(1 - f(p^S, p^T)) \tag{6.1}$$

[3] CSLS, the cosine-based similarity metric that was introduced in Chapter 4 can be an alternative. Note that this work was carried out before the discovery of CSLS.

[4] If we accidentally construct a pair which occurs in the set of positive pairs D_{lex}, we re-sample until we obtain exactly N_s negative samples.

We further explain the architecture of the neural network, the approach to construct vocabularies of words and phrases and the strategy to identify candidate translations during prediction. Four key components may be distinguished: (1) the input layer; (2) the character-level encoder; (3) the word-level encoder; and (4) a feed-forward network that combines the output representations from the two encoders into the final classification score.

6.3.1 Input Layer

The goal is to exploit the knowledge encoded in both the word and character levels. Therefore, the raw input representation of a word/phrase $p \in V^S$ of character length M consists of (1) its *one-hot* encoding on the word level, labeled x_p^S; and (2) a sequence of M one-hot encoded vectors $\boldsymbol{x}_{c0}^S, .., \boldsymbol{x}_{ci}^S, ..\boldsymbol{x}_{cM}^S$ on the character level, representing the character sequence of the word. x_p^S is thus a $|V^S|$-dimensional word vector with all zero entries except for the dimension that corresponds to the position of the word/phrase in the vocabulary. $\boldsymbol{x}_{ci}^S$ is a $|C^S|$-dimensional character vector with all zero entries except for the dimension that corresponds to the position of the character in the character vocabulary C^S.

6.3.2 Character-Level Encoder

To encode a pair of character sequences $\boldsymbol{x}_{c0}^S, .., \boldsymbol{x}_{ci}^S, ..\boldsymbol{x}_{cn}^S, \boldsymbol{x}_{c0}^T, .., \boldsymbol{x}_{ci}^T, ..\boldsymbol{x}_{cm}^T$ we use a two-layer long short-term memory (LSTM) recurrent neural network (RNN) (Hochreiter and Schmidhuber, 1997) as illustrated in Figure 6.2. At position i in the sequence, we feed the concatenation of the i^{th} character of the source language and target language word/phrase from a training pair to the LSTM network. The space character in phrases is threated like any other character. The characters are represented by their one-hot encoding. To deal with the possible difference in word/phrase length, we append special padding characters at the end of the shorter word/phrase (see Figure 6.2). $\boldsymbol{s}_{1i}$, and $\boldsymbol{s}_{2i}$ denote the states of the first and second layer of the LSTM. We found that a two-layer LSTM performed better than a shallow LSTM. The output at the final state $\boldsymbol{s}_{2N}$ is the character-level representation $\boldsymbol{r}_c^{ST}$. We apply dropout regularization (Srivastava et al., 2014) with a keep probability of 0.5 on the output connections of the LSTM (see the dotted lines in Figure 6.2). We will further refer to this architecture as CHARPAIRS .[5]

[5] A possible modification to the architecture would be to swap the (unidirectional) LSTM for a bidirectional LSTM (Schuster and Paliwal, 1997). In preliminary experiments on the development set this did not yield improvements over the proposed architecture, we thus do not discuss it further.

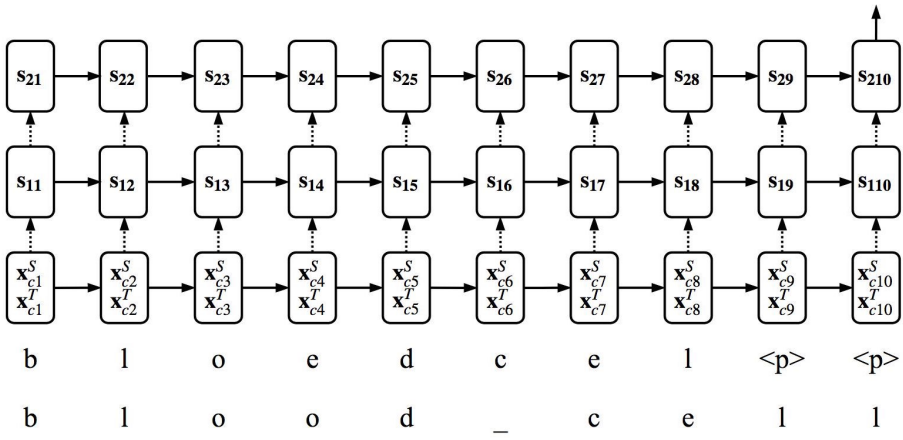


Figure 6.2: An illustration of the character-level LSTM encoder architecture using the example EN-NL translation pair *<blood cell, bloedcel>*.

6.3.3 Word-Level Encoder

We define the word-level representation of a pair $< p^S, p^T >$ simply as the concatenation of the embeddings for p^S and p^T:

$$\boldsymbol{r}_p^{ST} = \boldsymbol{W}^S \cdot \boldsymbol{x}_p^S \quad \| \quad \boldsymbol{W}^T \cdot \boldsymbol{x}_p^T \tag{6.2}$$

Here, $\boldsymbol{r}_p^{ST}$ is the representation of the word/phrase pair, and $\boldsymbol{W}^S$, $\boldsymbol{W}^T$ are word embedding matrices looked up using one-hot vectors $\boldsymbol{x}_p^S$ and $\boldsymbol{x}_p^T$. In our experiments, $\boldsymbol{W}^S$ and $\boldsymbol{W}^T$ are obtained in advance using any state-of-the-art word embedding model, e.g., Mikolov et al. (2013c); Vulić and Moens (2016b) and are then kept *fixed* when minimizing the loss from Equation (6.1).

To test the generality of our approach, we experiment with two well-known embedding models: (1) the model from Mikolov et al. (Mikolov et al., 2013c), which trains monolingual embeddings using skip-gram with negative sampling (SGNS) (Mikolov et al., 2013b); and (2) the model of Vulić and Moens (2016b) which learns word-level bilingual embeddings from document-aligned comparable data (BWESG). For both models, the top layers of our proposed classification network should learn to relate the word-level features stemming from these word embeddings using a set of annotated translation pairs.

6.3.4 Combination: Feed-Forward Network

To combine these word-level and character-level representations we use a fully connected feed-forward neural network $\boldsymbol{r}_h$ on top of the concatenation of $\boldsymbol{r}_p^{ST}$ and $\boldsymbol{r}_c^{ST}$ which is fed as input to the network:

$$\boldsymbol{r}_{h_0} = \boldsymbol{r}_p^{ST} \quad \| \quad \boldsymbol{r}_c^{ST} \tag{6.3}$$

$$\boldsymbol{r}_{h_i} = sigmoid(\boldsymbol{W}_{h_i} \cdot \boldsymbol{r}_{h_{i-1}} + \boldsymbol{b}_{h_i}) \tag{6.4}$$

$$score = sigmoid(\boldsymbol{W}_o \cdot \boldsymbol{r}_{h_H} + \boldsymbol{b}_o) \tag{6.5}$$

H denotes the number of layers between the representation layer and the output layer. In the simplest architecture, H is set to 0 and the word-pair representation $\boldsymbol{r}_{h_0}$ is directly connected to the output layer (see Figure 6.3A). In this setting each dimension from the concatenated representation is weighted independently. This is undesirable as it prohibits learning relationship between the different representations. On the word level, for instance, it is obvious that the classifier needs to combine the embeddings of the source and target word to make an informed decision and not merely calculate a weighted sum of them. Therefore, we opt for an architecture with hidden layers instead (see Figure 6.3B). Unless stated otherwise, we use two hidden layers, while in Experiment V of the Results and Discussion section we further analyze the influence of parameter H.

6.3.5 Constructing the Vocabularies

The vocabularies are the union of all words that occur at least five times in the corpus and phrases that are automatically extracted from it. We opt for the phrase extraction method proposed in Mikolov et al. (2013b).[6] The method iteratively extracts phrases for bigrams, trigrams, etc. First, every bigram is assigned a score using Equation (6.6). Bigrams with a score greater than a given threshold are added to the vocabulary as phrases. In subsequent iterations, extracted phrases are treated as if they were a single token and the same process is repeated. The threshold and the value for δ are set so that we maximize the recall of the phrases in our training set. We performed 4 iterations in total, resulting in N-grams up to a length of 5.

When learning the word-level representations phrases are treated as a single token (following Mikolov et al. (2013b)). Therefore, we do not add words that only occur as part of a phrase separately to the vocabulary, because no word representation is learned

[6] We used the implementation of the gensim toolkit https://github.com/RaRe-Technologies/gensim (Řehůřek and Sojka, 2010).

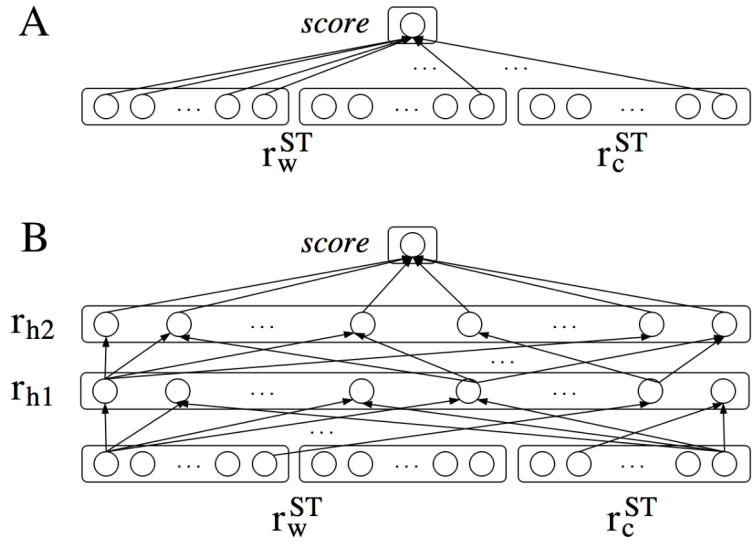


Figure 6.3: Illustrations of the classification component with feed-forward networks of different depths. A: $H = 0$. B: $H = 2$ (our model). All layers are fully connected.

for these words. E.g., for our dataset *"York"* is not included in the vocabulary as it always occurs as part of the phrase *"New York"*.

$$score(w_i, w_j) = \frac{Count(w_i, w_j) - \delta}{Count(w_i) \cdot Count(w_j)} \cdot |V| \tag{6.6}$$

Where $Count(w_i, w_j)$ is the frequency of the bigram $w_i\ w_j$, $Count(w)$ is the frequency of w, $|V|$ is the size of the vocabulary, and δ is a discounting coefficient that prevents that too many phrases consist of very infrequent words.

6.3.6 Candidate Generation

To identify which word pairs are translations, one could enumerate all translation pairs and feed them to the classifier g. The time complexity of this brute-force approach is $O(|V^S| \times |V^T|)$ times the complexity of g. For large vocabularies this can be a prohibitively expensive procedure. Therefore, we have resorted to a heuristic which uses a noisy classifier: it generates $2N_c << |V^T|$ translation candidates for each source language word/phrase p^S as follows. It generates (1) the N_c target words/phrases closest

to p^S measured by the edit distance, and (2) N_c target words/phrases measured closest to p^S based on the cosine distance between their word-level embeddings in a bilingual space induced by the embedding model of Vulić and Moens (2016b). As we will see in the experiments, besides straightforward gains in computational efficiency, limiting the number of candidates is even beneficial for the overall classification performance.

6.3.7 Experimental Setup

Data One of the main advantages of automatic BLI systems is their portability to different languages and domains. However, current standard BLI evaluation protocols still rely on general-domain data and test sets (Mikolov et al., 2013b; Gouws et al., 2015; Lazaridou et al., 2015; Vulić and Moens, 2016b, inter alia). To tackle the lack of quality domain-specific data for training and evaluation of BLI models, we have constructed a new English-Dutch (EN-NL) text corpus in the *medical* domain. The corpus contains topic-aligned documents (i.e., for a given document in the source language, we provide a link to a document in the target language that has comparable content). The domain-specific document collection was constructed from the English-Dutch aligned Wikipedia corpus available online[7], where we retain only document pairs with at least 40% of their Wikipedia categories classified as *medical*.[8] This simple selection heuristic ensures that the main topic of the corpus lies in the medical domain, yielding a final collection of 1198 training document pairs. Following standard practice (Koehn and Knight, 2002; Haghighi et al., 2008; Prochasson and Fung, 2011), the corpus was then tokenized and lowercased, and words occurring less than five times were filtered out.

Translation Pairs: Training, Development, Test We constructed a set of EN-NL translation pairs using a semi-automatic process. We started by translating all the words in our preprocessed corpus. These words were translated by Google Translate and then post-edited by fluent EN and NL speakers.[9] This yields a lexicon with mostly single word translations. In this work we are also interested in finding translations for phrases: therefore, we used IATE (Inter-Active Terminology for Europe), the EU's inter-institutional terminology database, to create a gold standard of domain-specific terminology phrases in our corpus. More specifically, we matched all the IATE phrase terms that are annotated with the *Health* category label to the N-grams in our corpus. This gives a list of phrases in English and Dutch. For some terms a translation was already present in the IATE termbase: these translations were added to the lexicon. The remaining terms are again translated by resorting to Google Translate and post-editing.

[7] http://linguatools.org/tools/corpora/

[8] https://www.dropbox.com/s/hlewabraplb9p5n/medicine_en.txt?dl=0

[9] In case the post-editor was unsure about the automatically acquired translation, he researched the source term on the web and corrected the translation if necessary.

We end up with 20,660 translation pairs. For 8,412 of these translation pairs (40.72%) both source and target words occur in our corpus.[10] We perform a 80/20 random split of the obtained subset of 8,412 translation pairs to construct a training and test set respectively. We make another 80/20 random split of the training set into training and validation data. 7.70% of the translation pairs have a phrase on both source and target side, 2.31% of the pairs consists of a single word and a phrase, 90.00% of the pairs consist of single words only. We note that 21.78% of the source words have more than one translation. In our corpus, the English phrases in the lexicon have an average frequency of 20. For Dutch phrases this is 17. English words in the lexicon have an average frequency of 59, for Dutch this number is 47.

Word-Level Embeddings Skip-gram word embeddings with negative sampling (SGNS) (Mikolov et al., 2013c) are induced using the `word2vec` toolkit with the subsampling threshold set to $10e$-4 and window size set to 5. BWESG embeddings (Vulić and Moens, 2016b) are learned by merging topic-aligned documents with length-ratio shuffling and then training the SGNS model over the merged documents with the subsampling threshold set to $10e$-4 and the window size set to 100. The dimensionality of all word-level embeddings in all experiments is $d = 50$, and similar trends in results were observed with $d = 100$.

Classifier The model is implemented in Python using Tensorflow (Abadi et al., 2015). For training we use the Adam optimizer with default values (Kingma and Welling, 2014) and mini-batches of 10 examples. The number of negative samples $2N_s$ and candidate translation pairs during prediction $2N_c$ are tuned on the development set for all models except CHARPAIRS and CHARPAIRS -SGNS (see Experiments II, IV, and V) for which we opted for default non-tuned values of $2N_c = 10$ and $2N_s = 10$.[11] The classification threshold t is tuned measuring F_1 scores on the validation set using a grid search in the interval $[0.1, 1]$ in steps of 0.1.[12]

Evaluation Metric The metric we use is F_1, the harmonic mean between recall and precision. While prior work typically proposes only one translation per source word and reports *accuracy* scores accordingly, here we also account for the fact that words can have multiple translations. We evaluate all models using two different modes: (1) *top* mode, as in prior work, identifies only one translation per source word (i.e., it is the

[10] Since we work with a comparable corpus in our experiments, not all translations of the English vocabulary words occur in the Dutch part of the corpus and vice versa.

[11] It takes more time to train and hence tune the models with the character-LSTM.

[12] The code and dataset used in this work can be downloaded from `http://liir.cs.kuleuven.be/software.php`.

target word with the highest classification score), (2) the *all* mode identifies as valid translation pairs all pairs for which the classification score exceeds the threshold t.

6.4 Results and Discussion

A Roadmap to Experiments We start by evaluating the phrase extraction (Experiment I) as it places an upper bound on the performance of the proposed system. Next, we report on the influence of the hyper-parameters $2N_c$ and $2N_s$ on the performance of the classifiers (Experiment II). We then study automatically extracted word-level and character-level representations for BLI separately (Experiment III and IV). For these single-component models Equation 6.3 simplifies to $\boldsymbol{r}_{h_o} = \boldsymbol{r}_w^{ST}$ (word-level) and $\boldsymbol{r}_{h_o} = \boldsymbol{r}_c^{ST}$ (character-level). Following that, we investigate the synergistic model presented in the Methods section which combines word-level and character-level representations (Experiment V). We then analyze the influence on performance of: the number of hidden layers of the classifier, the training data size, and word frequency. We conclude this section with an experiment that verifies the usefulness of our approach for inducing translations with Greek/Latin roots.

6.4.1 Experiment I: Phrase Extraction

The phrase extraction module puts an upper bound on the system's performance as it determines which words and phrases are added to the vocabulary - translation pairs with a word or phrase that do not occur in the vocabulary can of course never be induced. To maximize the recall of words and phrases in the ground truth lexicon w.r.t. the vocabularies, we tune the threshold of the phrase extraction on our training set. The thresholds were set to 6 and 8 for English and Dutch respectively, and the value for δ was set to 5 for both English and Dutch. The resulting English vocabulary contains 13,264 words and 9,081 phrases, the Dutch vocabulary contains 6,417 words and 1,773 phrases.

Table 6.1 shows the recall of the words and phrases in the training and test lexicons w.r.t. the extracted vocabularies. We see that the phrase extraction method obtains a good recall for translation pairs with phrases (around 80%) without hurting the recall of single word translation pairs.[13] The recall difference between English and Dutch phrase extraction can be explained by the difference in size of their respective corpora.[14]

[13] Note that when a word is always extracted as part of a phrase then it would not occur separately in the vocabulary.

[14] The English corpus consists of ≈1246k word occurrences, the Dutch corpus of ≈ 413k word occurrences.

	EN		NL		EN-NL	
	Phrases	**Words+Phrases**	**Phrases**	**Words+Phrases**	**Phrases**	**Words+Phrases**
Train	86.26	97.03	72.06	95.31	80.96	99.51
Test	88.60	97.12	67.44	95.62	79.69	99.11

Table 6.1: Recall of the words and phrases in the training and test lexicons w.r.t. the extracted vocabularies. In the EN-NL column, we show the percentage of translation pairs for which both source and target words/phrases are present in the vocabulary. In the EN/NL columns we show the percentage of English/Dutch words/phrases that are present in the vocabulary.

6.4.2 Experiment II: Hyper-parameters $2N_c$ and $2N_s$

Figure 6.4 shows the relation between the number of candidates $2N_c$ and precision, recall and F_1 of the candidate generation (without using a classifier). We see that the candidate generation works reasonably well with a small number of candidates and that the biggest gains in recall are seen when $2N_c$ is small (notice the log scale).

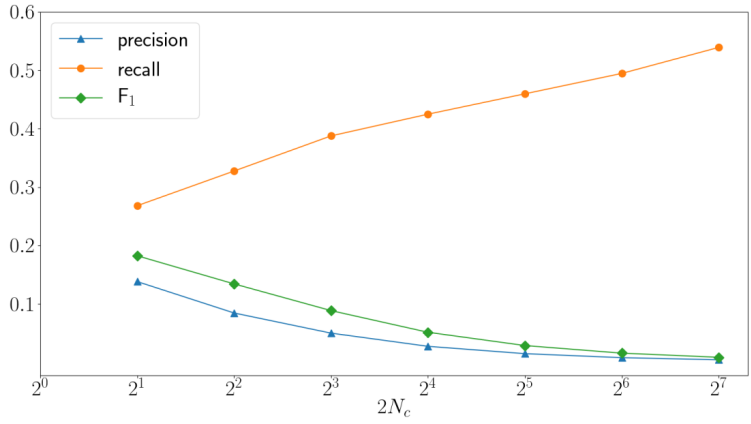
From the tuning experiments for Experiment III and IV we observed that using large values for $2N_c$ gives a higher recall, but that the best F_1 scores are obtained using small values for $2N_c$; The best performance on the development set for the word-level models was obtained with $2N_c = 2$ (Experiment III), for the character-level models this was with $2N_c = 4$ (Experiment IV). The low optimal values for $2N_c$ can be explained by the strong similarity between the features that the candidate generation and the classifiers use respectively. Because of this close relationship, translations pairs that are lowly ranked in the list of candidates should also be difficult instances for the classifiers. Increasing the number of candidates will result in a higher number of false positives, which is not compensated by a sufficient increase of the recall.

We found that the value of $2N_s$ is less critical for performance. The optimal value depends on the representations used in the classifier and on the value used for $2N_c$.

6.4.3 Experiment III: Word Level

In this experiment, we verify if word embeddings can be used for BLI in a classification framework. We compare the results with the standard approach that computes cosine similarities between embeddings in a cross-lingual space. For SGNS-based embeddings, this cross-lingual space is constructed following Mikolov et al. (2013c): a linear transformation between the two monolingual spaces is learned using the same set of training translation pairs that are used by our classification framework. For the BWESG-

Figure 6.4: Precision, recall and F_1 for candidate generation with $2N_c$ candidates.



based embeddings, no additional transformation is required, as they are inherently cross-lingual. The neural network classifiers are trained for 150 epochs.

The results are reported in Table 6.2. The SIM header denotes the baselines models that score translation pairs based on cosine similarity in the cross-lingual embedding space; The CLASS header denotes the models that use the proposed classification framework.

The results show that exploiting word embeddings in a classification framework has strong potential as the classification models significantly outperform the similarity-based approaches. The classification models yield best results in *all*-mode, this means they are good at translating words with multiple translations. For BWESG in the similarity-based approach, the inverse is true, it works better when only it proposes a single translation per source word.

We also find that when using the standard similarity method on SGNS embeddings that are mapped with a linear transformation (Mikolov et al., 2013c) yield extremely low results.[15] In this setup, where the embedding spaces are induced from small monolingual corpora and where the mapping is learned using infrequent translation pairs, the model seems unable to learn a decent linear mapping between the monolingual spaces. This is in line with the findings of Vulić and Korhonen (2016).

We observe that in the classification framework SGNS embeddings outperform BWESG embeddings. This could be because SGNS embeddings can better represent features related to the local context of words such as syntax properties, as SGNS is typically

[15]The NaN values in Table 6.2 are caused by an absence of true positives.

		Development					
		Words		Phrases		Words + Phrases	
	representation	F_1 (top)	F_1 (all)	F_1 (top)	F_1 (all)	F_1 (top)	F_1 (all)
SIM	BWESG	13.48	9.15	21.95	15.84	14.24	9.73
	SGNS	0.55	0.88	NaN	NaN	0.51	0.80
CLASS	BWESG	17.08	21.19	24.04	26.47	17.59	21.56
	SGNS	**23.83**	**25.05**	**25.77**	**27.27**	**23.99**	**25.22**

		Test					
		Words		Phrases		Words + Phrases	
	representation	F_1 (top)	F_1 (all)	F_1 (top)	F_1 (all)	F_1 (top)	F_1 (all)
SIM	BWESG	12.78	10.03	21.43	12.52	13.52	10.31
	SGNS	0.22	0.69	NaN	0.93	0.20	0.71
CLASS	BWESG	16.47	21.50	23.48	23.75	17.01	21.68
	SGNS	**22.80**	**24.41**	**26.74**	**27.14**	**23.10**	**24.62**

Table 6.2: Comparison of word-level BLI systems.

trained with much smaller context windows compared to BWESG.[16] Another general trend we see is that word-level models are better in finding translations of phrases. This is explained by the observation that the meaning of phrases tends to be less ambiguous, which makes word-level representations a reliable source of evidence for identifying translations.

6.4.4 Experiment IV: Character Level

This experiment investigates the potential of learning character-level representations from the translation pairs in the training set. We compare this approach to commonly-used, hand-crafted features. The following methods are evaluated:

- CHARPAIRS , uses the representation $\boldsymbol{r}_c^{ST}$ of the character-level encoder as described in the *Methods* section and illustrated in Figure 6.2.

[16] Note that BWESG uses large window sizes by design.

- ED_{norm}, uses the edit distance between the word/phrase pair divided by the average character length of p_s and p_t, following prior work (Irvine and Callison-Burch, 2013, 2016).
- $\log(\text{ED}_{rank})$, uses the logarithm of the rank of p_t in a list sorted by the edit distance w.r.t. p_s. For example, a pair for which p_t is the closest word/phrase in edit distance w.r.t. p_s, will have a feature value of $log(1) = 0$.
- $\text{ED}_{norm} + \log(\text{ED}_{rank})$, concatenates the ED_{norm} and $\log(\text{ED}_{rank})$ features.

The ED-based models comprise a neural network classifier similar to CHARPAIRS, though for ED_{norm} and $\log(\text{ED}_{rank})$ no hidden layers are used because the features are one-dimensional. For the ED-based models, the optimal values for the number of negative samples $2N_s$ and the number of generated translation candidates $2N_c$ were determined by performing a grid search, using the development set for evaluation. For the CHARPAIRS representation, the parameters $2N_s$ and $2N_c$ were set to the default values (10) without any additional fine-tuning, and the number of LSTM cells per layer was set to 512. We train the ED-based models for 25 epochs, the CHARPAIRS model takes more time to converge and is trained for 250 epochs.

The results are shown in Table 6.3. We observe that the performance of the character-level models is quite high w.r.t. the results of the word-level models in Experiment III. This supports our claim that character-level information is of crucial importance in this dataset and is explained by the high presence of medical terminology and expert abbreviations (e.g., *amynoglicosides, aphasics, nystagmus, EPO, EMDR* in the data; see also Figure 6.1), which because of its etymological processes, often contain morphological regularities across languages. This further illustrates the need for fusion models that exploit both word-level and character-level features. Another important finding is that the CHARPAIRS model systematically outperforms the baselines, which use hand-crafted features, indicating that learning representations on the character level is advantageous. Unlike the word-level models, translation pairs with phrases have lower performance than translations with single words. This is to be expected as phrases usually consist of a longer character sequence and hence are more difficult to represent.

6.4.5 Experiment V: Combined Model

On their own, the single-component word-level and character-level BLI models already perform very well in the task of biomedical BLI. In this experiment, we report the results of the combined model. In this setup, the LSTM network has 256 memory cells in each layer[17], and SGNS embeddings were selected as word-level representations.

[17] We found that in the combined setting of using both word-level and character-level representations, it is beneficial to use an LSTM of smaller size than in the character-level only setting.

	Development					
	Words		Phrases		Words + Phrases	
representation	F_1 (top)	F_1 (all)	F_1 (top)	F_1 (all)	F_1 (top)	F_1 (all)
ED_{norm}	24.49	19.53	15.62	19.87	23.83	19.55
$\log(\text{ED}_{rank})$	28.57	28.17	18.05	17.27	27.86	27.46
$\text{ED}_{norm} + \log(\text{ED}_{rank})$	25.99	11.20	18.40	14.35	25.49	11.31
CHARPAIRS	**31.95**	**32.32**	**23.70**	**25.97**	**31.39**	**31.92**

	Test					
	Words		Phrases		Words + Phrases	
representation	F_1 (top)	F_1 (all)	F_1 (top)	F_1 (all)	F_1 (top)	F_1 (all)
ED_{norm}	28.10	28.29	8.70	8.63	26.97	27.24
$\log(\text{ED}_{rank})$	29.30	28.95	19.48	19.35	28.70	28.39
$\text{ED}_{norm} + \log(\text{ED}_{rank})$	29.76	29.65	17.57	17.45	29.05	29.00
CHARPAIRS	**30.70**	**32.19**	**31.82**	**30.61**	**30.81**	**32.15**

Table 6.3: Comparison of character-level BLI methods from prior work (Irvine and Callison-Burch, 2016; Haghighi et al., 2008) with automatically learned character-level representations.

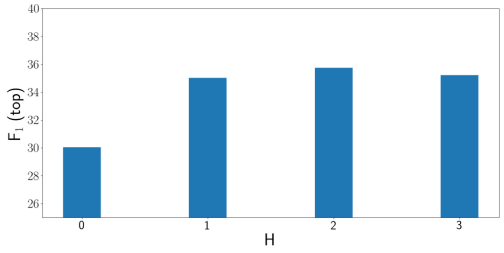


Figure 6.5: The influence of the number of layers H between the representations and the output layer on the BLI performance.

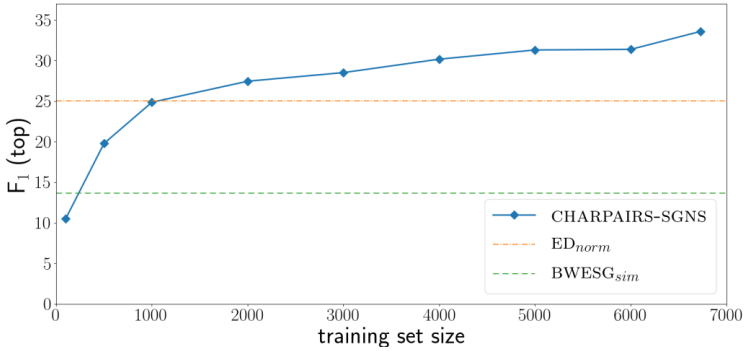


Figure 6.6: The influence of the training set size (the number of training pairs).

The embeddings are trained a priori, whereas the character-level representations are trained jointly with the rest of the network. This configuration will encourage the network to learn new character-level information which is distinctive from the word-level representations.

Table 6.4 shows the results of the combined model together with the best single component models. As hypothesized, we obtain the best results with the combined model. For phrases however, CHARPAIRS -SGNS 's performance is lower than the single component models. Our hypothesis for this behavior is that the LSTM in the combined model has fewer memory cells in the LSTM layers. We found that having 256 memory cells, rather than 512 cells as in the CHARPAIRS model, gives best results overall. However, for a combined model with 512 cells, we get an improved performance for the phrases. Table 6.5 shows translations induced by the different models that illustrate the advantage of a hybrid model. We observe that the CHARPAIRS model has learned that the first characters of words/phrases are very informative, though this sometimes results in false positives. The SGNS model sometimes confuses words that are semantically related, e.g., *zwangerschap (pregnancy)* and *miskraam (miscarriage)*. The CHARPAIRS -SGNS model is able to filter out false positives by exploiting both representations simultaneously. Even in cases where both single component models predict the wrong translations, it is possible that the combined model induces the correct translation(s) (e.g., *injected-ingespoten*).

Influence of the Number of Hidden Layers H The number of hidden layers H is a pertinent hyper-parameter. Figure 6.5 shows the influence of H on the performance measured by F_1 in *top* mode. We see a large improvement when H ranges from 0 to 1. When there are no hidden layers ($H = 0$), the network is unable to incorporate

	Development					
	Words		**Phrases**		**Words + Phrases**	
representation	F_1 (top)	F_1 (all)	F_1 (top)	F_1 (all)	F_1 (top)	F_1 (all)
CHARPAIRS	31.95	32.32	**23.70**	**25.97**	31.39	31.92
SGNS	23.83	26.36	17.37	17.08	25.77	25.81
CHARPAIRS -SGNS	**34.57**	**33.61**	18.18	23.29	**33.47**	**32.99**

	Test					
	Words		**Phrases**		**Words + Phrases**	
representation	F_1 (top)	F_1 (all)	F_1 (top)	F_1 (all)	F_1 (top)	F_1 (all)
CHARPAIRS	30.70	32.19	**31.82**	**30.61**	30.81	32.15
SGNS	22.80	24.41	26.74	27.14	23.10	24.62
CHARPAIRS -SGNS	**34.34**	**34.60**	23.17	26.59	**33.60**	**34.15**

Table 6.4: Results of the model that combines word-level and character-level representations (CHARPAIRS -SGNS) and the best performing single component models (CHARPAIRS and SGNS).

source word	**predictions** CHARPAIRS	**predictions** SGNS	**predictions** CHARPAIRS -SGNS
miscarriage	/	zwangerschap, miskraam , cardiale	miskraam
contractions	contraststof	samentrekkingen	samentrekkingen
injected	injecties, injectie	naald	ingespoten
desensitization	desensitisatie	injecties, desensibilisatie, ventilation	desensibilisatie, desensitisatie
hart attack	hartinfarct, hartaanval, hartmassage	hartaanval, atherosclerose, tia	hartinfarct, hartaanval
multifocal	multiple, multifocale	dominante	multifocale

Table 6.5: Predicted translations of single component models and the combined model, illustrating the advantage of the combined model. Correct translations are underlined.

dependencies between features. In case the number of hidden layers is larger than one, we notice no large effect of the number of hidden layers on performance.

Influence of Training Set Size In many realistic settings, especially when dealing with languages and domains that have limited translation resources, we lack large numbers of readily available translation pairs. Figure 6.6 illustrates the influence of training set size on the performance of CHARPAIRS -SGNS . We also plot the performance of two of our baseline models that only use training data to tune the threshold t: BWESG embeddings combined with cosine similarity (see Table 6.2) and normalized edit distance (ED_{norm}, see Table 6.3). We plot the performance of the baselines on the complete training set and assume it constant over the training examples. Unsurprisingly, the CHARPAIRS -SGNS performance increases with more training examples. Already from a seed lexicon size of 2000 translations it starts outperforming the baseline models.

Influence of Frequency In Figure 6.7 we see the effect of word/phrase frequency on performance. We plot F_1 scores after filtering the predicted translations and test set with a minimum word frequency cut-off. For example, for a cut-off frequency of 10, we only evaluate the translation pairs for which source and target words/phrases occur at least 10 times. Until a cut-off value of 125 performance for the three representations fluctuates but remains roughly level. When we only evaluate on high-frequency words (> 125) we see a performance drop for all models, especially for the character-level only model. From a manual inspection of these words we find that they typically have a broader meaning and are not particularly related to the medical domain (e.g., *consists-bestaat*, *according-volgens*, etc.). For these words, character-level information turns out to be less important.

Translation pairs derived from Latin or Greek We conclude the evaluation by verifying the hypothesis that our approach is particularly effective for translation pairs derived from Latin or Greek. Table 6.6 presents the F_1 scores on a subset of the test data in which only translation pairs for which the English word or phrase has clear Greek or Latin roots are retained. The results reveal that character-level modeling is indeed successful for these type of translation pairs. All models scored significantly higher on this subset, surprisingly also the SGNS model. The higher scores of the SGNS model, which operates on the word-level, could be attributed to an increased performance of the candidate generation, as it uses both word- and character-level information. Regarding the differences between models, the same trends as in previous model comparisons are apparent: the CHARPAIRS model improves nearly 5% over the edit distance baseline and the CHARPAIRS -SGNS model achieves the best results.

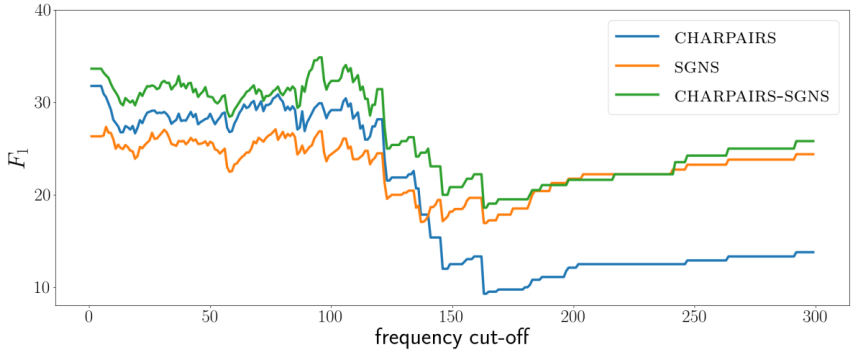


Figure 6.7: This plot shows how performance varies when we filter out translation pairs with a frequency below the specified cut-off point (on x axis).

	ED_{norm}	CHARPAIRS	SGNS	CHARPAIRS -SGNS
F_1 (top)	50.25	54.46	42.92	**57.20**
F_1 (all)	50.23	55.04	48.14	**56.41**

Table 6.6: Results on a subset of the test data consisting of translation pairs with Greek or Latin origin.

6.5 Conclusions

We have proposed a neural network based classification architecture for automated bilingual lexicon induction (BLI) from biomedical texts. Our model comprises both a word-level and character-level component. The character-level encoder has the form of a two-layer long short-term memory network. On the word level, we have experimented with different types of representations. The resulting representations were used in a deep feed-forward neural network. The framework that we have proposed can induce bilingual lexicons which contain both single words and multi-word expressions. Our main findings are that (1) taking a deep learning approach to BLI where we learn representations on word-level and character-level is superior to relying on handcrafted representations like edit distance (ED) and (2) the combination of word- and character-level representations proved to be very successful for BLI in the biomedical domain because of a large number of orthographically similar words (e.g., words stemming from the same Greek or Latin roots).

The proposed classification model for BLI leaves room for integrating additional translation signals that might improve biomedical BLI such as representations learned from available biomedical data or knowledge bases.

This chapter is based on the following publications:

- Geert Heyman, Ivan Vulić, Marie-Francine Moens. A Deep Learning Approach to Bilingual Lexicon Induction in the Biomedical Domain *BMC Bionformatics, 19(259) pages 1–15*, 2018
- Geert Heyman, Ivan Vulić, Marie-Francine Moens. Bilingual Lexicon Induction by Learning to Combine Word-Level and Character-Level Representations. *In Proceedings of the 15th International Conference of the European Chapter of the Association for Computational Linguistics (EACL), pages 1085–1095*, 2017

Chapter 7

Conclusion

In this final chapter, we summarize the work of this thesis, restate the main contributions, and discuss perspectives for further research.

7.1 Summary

In recent years, supervised representation learning has revolutionized several artificial intelligence subdomains including natural language processing. However, its success has been tightly correlated with the amount of annotated training data. As for many NLP problems such data is scarce and expensive to obtain, the main goal of this thesis was to further develop representation learning for weakly-supervised NLP problems.

After introducing the main thesis subjects and context (Chapter 1) and summarizing the fundamentals on which the contributions of this thesis are based (Chapter 2), we first studied two different paradigms for learning cross-lingual text representations without supervision (Chapters 3 and 4 respectively). Both these chapters learn representations that are valuable for transferring knowledge across languages. That is, a lack of annotated training data in a language can be compensated with training examples in one or more resource-rich languages by employing cross-lingual input representations.

In Chapter 3 we investigated probabilistic topic modeling as a means to obtain bilingual representations for documents and words. We argued that the probabilistic topic modeling framework has the advantage of being very transparent because the model parameters and consequently also the induced representations are easily interpretable by humans. We proposed C-BiLDA, a new probabilistic topic model that generalizes LDA so that it can induce bilingual representations from a collection of non-parallel

document pairs. In particular, we designed the model to distinguish between shared and unique content in an aligned document pair and put a prior distribution on the random variables that determine the ratio of shared and unique content. C-BiLDA is completely data-driven and does not require a machine-readable bilingual dictionary or high-quality parallel data. We showed that the representations are a valuable vehicle for knowledge transfer by evaluating them on two cross-lingual document classification datasets comprised of three language-pairs each (i.e., English-Spanish, English-French, and English-Italian), where we trained a shallow classifier on annotated English documents and evaluate performance on the documents in the respective target languages. C-BiLDA outperforms four well-known bilingual topic models, most notably on training corpora with a lower degree of parallelism. Its wide applicability in terms of input data makes C-BiLDA a useful model for learning representations in under-resourced languages and language pairs, as well as in domains with specific terminology for which parallel data is often not available.

In Chapter 4 we turned our attention to unsupervised multilingual text representations based on word embeddings. Word embeddings are not as interpretable as the PTM-based representations, but they can be trained very efficiently and have been proven superior to PTM representations in bilingual lexicon induction and information retrieval. With the switch to word embeddings, we also moved to a setting that is more general than the one in Chapter 3 in two respects: 1) We designed methods that learn representations for a variable number of languages; 2) We further relaxed the data requirements by only relying upon monolingual corpora in the same domain.

More specifically, Chapter 4 introduces two self-learning methods that map monolingually trained word embeddings in the same vector space such that related words and translations are represented by vectors that are positioned close to each other. We validate the models by using their induced representations in bilingual lexicon induction, multilingual document classification, and multilingual dependency parsing, using four datasets. The results show that our best method is competitive with (bilingual lexicon induction) or better than (document classification and dependency parsing) the state of the art on these datasets. In contrast to related work, this method combines the following key properties: it incorporates the dependencies between the vector spaces of all the involved languages, it is robust w.r.t. exotic languages such as Finnish and Bulgarian, and it has no risk of producing degenerate solutions.

Chapter 5 proposed an RL-based solution to an important Dutch verb spelling problem. We tackled the lack of annotated spelling mistakes by designing a system that is largely independent of the input spelling of the verb that is being spell-checked. This bypasses the need for annotated data and enabled training an advanced neural network on a large collection of (high-quality) Dutch texts. We evaluated our system on 1) a big synthetic test set, which was created by automatically introducing mistakes in correctly-written text according to a cognitively-inspired generative process, and 2) three online verb spelling tests. The results indicate that our system is highly accurate except for

infrequent mistakes (i.e., confusion between verb forms that end with *-en*). Moreover, the proposed model achieved significant performance gains compared to four existing spell checkers. Despite neural network parameters being hard to interpret, we also provided a method to gain insight into the model's predictions. The method assigns a score to each word in the sentence that indicates its importance on the predicted spelling.

Whereas the previous chapters focussed either on unsupervised RL (Chapters 3-4) or supervised RL (Chapter 5), Chapter 6 is positioned at the intersection of both. In this chapter, we tackled bilingual lexicon induction from domain-specific corpora by integrating character-level representations that are learned from a seed lexicon with word/phrase-level representations that are learned from monolingual corpora without additional supervision. The goal of this work was to study BLI in a setting that closely resembles industrial needs, i.e., find translations in domain-specific, terminology-rich texts. To this end, we constructed a new benchmark dataset from Dutch and English Wikipedia articles in the medical domain and bilingual dictionaries containing words and phrases that occur in these articles.

The key findings of this work were 1) that word-/phrase- and character-level representations for BLI can be elegantly combined by framing BLI as a classification problem and that integration of character-level representations significantly lifts performance for medical terms, and 2) that it is beneficial to learn character-level representations from a seed lexicon rather than to use hand-crafted features (e.g., edit distance) as standard practice dictated.

In sum, with the work presented in this thesis we made the following key contributions:

- **We proposed a model that induces easy-to-interpret representations for words and documents from a collection of subject-aligned bilingual document pairs:** C-BiLDA is a bilingual probabilistic topic model that induces soft word clusters, called topics, from subject-aligned bilingual document pairs. The topics learned by C-BiLDA can be used to obtain interpretable bilingual representations for words or documents. We hence positively answered the first research question (RQ 1) that was posed in the introduction.
- **We investigated new methods for learning multilingual word representations from monolingual corpora only:** We found that we can construct multilingual representations without cross-lingual supervision that may serve as useful cross-lingual transfer learning features, even for languages with unique characteristics such as Finnish (answering RQ 2). Furthermore, the state of the art results of IHS confirm the hypothesis of RQ 3 as they illustrate the benefits of incrementally growing the multilingual space.
- **We designed a very accurate RL model to tackle one of the most prominent Dutch spelling mistakes:** We showed that by making a model largely

independent from the input spelling of the spell-checked words, we can fruitfully train an advanced neural network model on large amounts of high-quality text (answering RQs 4-5). The resulting model significantly outperforms existing spell-checkers, including the commercial spell-checker that comes with Microsoft Word, and obtains a perfect score on a spelling test published by *de Standaard*, a Flemish newspaper. In addition, to address RQ 6, we proposed a technique that provides useful feedback to the user by indicating which parts of the input sequence were most influential on the model predictions.

- **We proposed an RL approach to bilingual lexicon induction:** By formulating BLI as a classification problem, we found that we can naturally combine word/phrase-level representations and character-level representations (answering RQ 7). Furthermore, we showed that instead of using hand-crafted features such as edit distance it is better to *learn* character-level features (answering RQ 8).
- **We constructed new benchmark datasets for bilingual lexicon induction and dt-correction:** With our English-Dutch bilingual lexicon induction dataset constructed from Wikipedia articles in the medical domain, BLI methods can be evaluated in a setting that is more closely aligned with industrial needs; Furthermore, we created the first large-scale evaluation dataset for dt-mistakes by introducing errors into the proceedings of the European Parliament with a generative process motivated by cognitive science insights. The latter is supplemented with three online verb spelling tests.

7.2 Perspectives for Future Research

The work of this thesis has opened several promising avenues for further research on representation learning in weakly-supervised NLP settings.

Concerning the unsupervised training of cross-lingual text representations, we spoke about the differences between PTM-based representations and word embeddings: The former are more interpretable because their dimensions correspond to topics that can be represented by a set of keywords, while the latter are more expressive and are efficiently trained. Interpretability, computational efficiency, and expressiveness are all relevant characteristics when training representations for weakly-supervised settings. Towards the goal of inducing representations that combine transparency, computational efficiency and expressiveness, we naturally identify two strategies: improving PTM's training time and expressiveness, or adapting word embedding models such that it is easier to associate meaning to individual feature dimensions.

From the PTM perspective, the slower training times are due to the use of optimization algorithms (e.g., Gibbs sampling, mean-field variational inference (Hoffman et al.,

2013)) that are less efficient compared to optimization in word embedding models. Recent works (Miao et al., 2016; Srivastava and Sutton, 2017; Miao et al., 2017) that propose variational auto-encoders (Kingma and Welling, 2014) as an alternative inference method seem very promising, however. They estimate the posterior distribution over the latent variables by a neural network which results in fast training times (e.g., Srivastava and Sutton (2017) report a training time of 80 minutes for fitting a topic model on 1 million documents on a modern GPU) and have the additional advantage that the optimization method makes little model-specific assumptions. The latter enabled Srivastava and Sutton (2017) to propose a variation of LDA that obtains better topic coherence scores, without any major changes to the optimization algorithm.[1]

Hence, if we could exchange Gibbs sampling for an approach based on variational auto-encoders, the C-BiLDA model would become scalable to entire Wikipediae and it could be trivially extended to more than two language pairs and trained on many more document pairs. Moreover, it may facilitate relaxing some of the assumptions that PTM's make. For instance, incorporating document structure in the generative process instead of viewing documents as a bag-of-words or imposing some sort of hierarchical structure on the topics, which in turn could improve the quality of the representations.

From the perspective of word embeddings models, the word representations are hard to interpret because they are dense, real-valued vectors of which the dimensions have no obvious meaning. One strategy to attack this problem is forcing representations to be sparser, for instance by using a regularization term in the training objective that stimulates sparsity. Despite yielding encouraging results in initial studies (Faruqui et al., 2015; Sun et al., 2016), such approaches have yet to become common practice.

Another interesting research angle for learning unsupervised multilingual text representations is studying the effect of using multilingual word representations for problems sensitive to word order and studying algorithms that learn multilingual representations beyond the word-level (i.e., representations for phrases, sentences or documents) that do not use the bag-of-words assumption. It is clear that this is a very challenging problem, for instance for semantic role labeling, the task that aims to extract *"who does what to whom"* in a given sentence, empirical results indicate that the most appropriate model architecture is language-dependent (Do et al., 2018). However, the payoff would also be substantial, consider for example how it could facilitate porting virtual assistants such as Alexa, Siri, or Google Assistant to different languages.

Furthermore, with regard to the proposed dt-correction model, we see two interesting follow-up studies. First, because the method effectively learns representations for a syntactically-oriented task without annotated examples, it would be interesting to

[1] Note that the same does not hold for Gibbs sampling because it requires deriving new inference formulas (see the Gibbs sampling derivations for C-BiLDA in Chapter 3 for instance).

investigate if its induced representations can be useful for related tasks such as part-of-speech tagging, constituency parsing, or dependency parsing. Note that if this hypothesis were to hold, it could benefit more languages than Dutch alone: Any language for which there is a non-trivial relationship between word inflections and syntactical relations in sentences could potentially benefit from representations that are induced by predicting word inflections. Second, though the model was very accurate on the most common mistakes, it was not able to predict the spelling of plural verb forms (e.g., beantwoorden vs beantwoordden), this is due to the large imbalance between the frequencies of the *"stem + en"* and *"stem + den/ten"* forms. We hypothesize this can be addressed by subsampling the more frequent *"stem + en"*, but further investigation is required.

The representation learning framework for bilingual lexicon induction has already inspired further research. Starting from the code and using the dataset we had released, Hangya et al. (2018) proposed a simple but effective approach to leverage large general-purpose corpora in addition to the domain-specific medical texts. Training on this additional data leads to improved representations of the general-purpose words that occur in the medical texts, which in turn leads to higher quality representations for the domain-specific words.

Another extension to the BLI framework that comes to mind is the incorporation of additional translation signals because one of its key advantages is that it facilitates fusing heterogeneous features. For instance, considering its application in the medical domain, it could be interesting to incorporate structured data from biomedical databases or knowledge bases. Furthermore, it would be interesting to investigate temporal representations, which encode word usage over time (Irvine and Callison-Burch, 2013). This could be particularly beneficial for translating terminology as their usage is very dynamic (e.g., new terms are coined as a consequence of new technologies or discoveries, and terms can become trending because of certain news events).

7.3 Epilogue

The main conclusion of this dissertation is that representation learning is a potent paradigm for addressing the scarcity of annotated training data in natural language processing. With the models proposed in this thesis, we demonstrated the value of RL for weakly-supervised NLP problems as a means to a) inject prior knowledge into models by representing the model inputs with textual representations that are extracted from large corpora, and b) extract abstractions for weakly-annotated text data that are not expressed by classical, manually-extracted NLP features.

To further increase the synergy between natural language processing and representation learning, we think it is key to research new representation learning models that combine

the interpretability of classical probabilistic models with the expressiveness of neural networks. Another fundamental challenge concerns the induction of sentence/document representations that are sensitive to word order which can be utilized for cross-lingual transfer learning - this to address a lack of annotated training data in tasks such as semantic role labeling.

Bibliography

VRT lanceert online schrijfhulp, 2016. URL https://www.vrt.be/nl/over-de-vrt/nieuws/2018/05/17/vrt-lanceert-online-schrijfhulp/.

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. URL http://tensorflow.org/.

Zubair Afzal, Saber Akhondi, Herman van Haagen, Erik van Mulligen, and Jan Kors. Biomedical Concept Recognition in French Text Using Automatic Translation of English Terms. In *Conference and Labs of the Evaluation Forum (CLEF; Working Notes)*, 2015.

Željko Agić, Maria Jesus Aranzabe, Aitziber Atutxa, Cristina Bosco, Jinho Choi, Marie-Catherine de Marneffe, Timothy Dozat, Richárd Farkas, Jennifer Foster, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Jan Hajič, Anders Trærup Johannsen, Jenna Kanerva, Juha Kuokkala, Veronika Laippala, Alessandro Lenci, Krister Lindén, Nikola Ljubešić, Teresa Lynn, Christopher Manning, Héctor Alonso Martínez, Ryan McDonald, Anna Missilä, Simonetta Montemagni, Joakim Nivre, Hanna Nurmi, Petya Osenova, Slav Petrov, Jussi Piitulainen, Barbara Plank, Prokopis Prokopidis, Sampo Pyysalo, Wolfgang Seeker, Mojgan Seraji, Natalia Silveira, Maria Simi, Kiril Simov, Aaron Smith, Reut Tsarfaty, Veronika Vincze, and Daniel Zeman. Universal Dependencies 1.1, 2015. URL http://hdl.handle.net/11234/LRT-1478.

Amr Ahmed and Ep Xing. Staying Informed: Supervised and Semi-supervised Multi-view Topical Analysis of Ideological Perspective. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2010.

Massih Reza Amini and Cyril Goutte. A Co-classification Approach to Learning from Multilingual Corpora. *Machine Learning*, 79(1-2):105–121, 2010.

Massih-Reza Amini, Nicolas Usunier, and Cyril Goutte. Learning from Multiple Partially Observed Views - an Application to Multilingual Text Categorization. In *Proceedings of the 23rd Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 28–36, 2009.

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A. Smith. Massively Multilingual Word Embeddings. 2016.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning Principled Bilingual Mappings of Word Embeddings while Preserving Monolingual Invariance. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2289–2294, 2016.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning Bilingual Word Embeddings with (Almost) no Bilingual Data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, 2017a.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised Neural Machine Translation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–11, 2017b.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A Robust Self-learning Method for Fully Unsupervised Cross-lingual Mappings of Word Embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018a.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Generalizing and Improving Bilingual Word Embedding Mappings with a Multi-Step Framework of Linear Transformations. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 5012–5019, 2018b.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Unsupervised Statistical Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018c.

Eric Atwell and Stephen Elliott. Dealing with Ill-formed English Text. *The Computational Analysis of English: A Corpus-Based Approach*, 1987.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–15, 2015.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't Count, Predict! A Systematic Comparison of Context-counting vs. Context-predicting Semantic Vectors. In *Proceedings of ACL*, pages 238–247, 2014.

Núria Bel, Cornelis H A Koster, and Marta Villegas. Cross-Lingual Text Categorization. In *Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, pages 126–139, 2003.

Anthony Bell and Terrence Sejnowski. The 'Independent Components' of Natural Scenes are Edge Filters. *Vision Research*, 1997.

Christopher M Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.

David M Blei and Jon D McAuliffe. Supervised Topic Models. In *Proceedings of the 21st Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 121–128, 2007.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information. *CoRR abs/1607.04606*, 2016.

Danushka Bollegala, Georgios Kontonatsios, and Sophia Ananiadou. A Cross-lingual Similarity Measure for Detecting Biomedical Term Translations. *PLoS ONE*, 2015.

Jordan Boyd-Graber and David M Blei. Multilingual Topic Models for Unaligned Text. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 75–82, 2009.

Jordan Boyd-Graber and Philip Resnik. Holistic Sentiment Analysis Across Languages: Multilingual Supervised Latent Dirichlet Allocation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 45–55, 2010.

Denny Britz, Anna Goldie, Thang Luong, and Quoc Le. Massive Exploration of Neural Machine Translation Architectures. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1442–1451, 2017.

Hailong Cao, Tiejun Zhao, Shu Zhang, and Yao Meng. A Distribution-based Model to Learn Bilingual Word Embeddings. *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, 2016.

Giovanni Cavallanti, Nicolò Cesa-Bianchi, and Claudio Gentile. Linear Algorithms for Online Multitask Classification. *The Journal of Machine Learning Research*, 11: 2901–2934, 2010.

Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. An Autoencoder Approach to Learning Bilingual Word Representations. In *Proceedings of the 27th Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 1853–1861, 2014.

Xilun Chen and Claire Cardie. Unsupervised Multilingual Word Embeddings. In *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMLP)*, 2018.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

Kenneth W Church and William A Gale. Probability scoring for spelling correction. *Statistics and Computing*, 1(2):93–103, dec 1991.

Vincent Claveau. Automatic Translation of Biomedical Terms by Supervised Machine Learning. In *Proceedings of the International Language Resources and Evaluation Conference (LREC)*, pages 684–691, 2008.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word Translation Without Parallel Data. In *In Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–14, 2018.

Jocelyn Coulmance, Jean-Marc Marty, Guillaume Wenzek, and Amine Benhalloum. Trans-gram, Fast Cross-lingual Word Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1109–1113, 2015.

George Cybenko. Approximisation by Superpositions of Sigmoidal Function. *Mathematics of Control, Signals and Systems*, 1989. ISSN 10009221. doi: 10.1007/BF02836480.

Walter Daelemans and Antal van den Bosch. Dat gebeurd mei niet: computationele modellen voor verwarbare homofonen. In *Tussen taal, spelling en onderwijs : essays bij het emeritaat van Frans Daems. Gent : Academia Press*, pages 199—-210. 2007.

Walter Daelemans, Antal Van Den Bosch, and Ton Weijters. IGTree: Using trees for compression and classification in lazy learning algorithms. *Artificial Intelligence Review*, 11:407–423, 1997. ISSN 02692821. doi: 10.1.1.29.4517.

Dipanjan Das and Slav Petrov. Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections. In *Proceedings of the 49th Annual Meeting of the*

Association for Computational Linguistics: Human Language Technologies (ACL-HLT), pages 600–609, 2011.

Wim De Smet and Marie-Francine Moens. Cross-language linking of news stories on the Web using interlingual topic modeling. In *Proceedings of the CIKM 2009 Workshop on Social Web Search and Mining (SWSM@CIKM)*, pages 57–64, 2009.

Wim De Smet, Jie Tang, and Marie-Francine Moens. Knowledge transfer across multilingual corpora via latent topics. In *Proceedings of the 15th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 549–560, 2011.

Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1): 1–38, 1977.

Margot D'Hertefelt, Lieve De Wachter, and Serge Verlinde. Writing Aid Dutch. Supporting students' writing skills by means of a string and pattern matching based web application. In *Proceedings of the 6th International Conference on Computer Supported Education*, pages 486–491, 2014.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. Improving zero-shot learning by mitigating the hubness problem. In *Proceedings of the International Conference on Learning Representations (ICLR), workshop track*, 2015.

Quynh Ngoc Thi Do, Artuur Leeuwenberg, Geert Heyman, and Marie-Francine Moens. A Flexible and Easy-to-use Semantic Role Labeling Framework for Different Languages. In *COLING 2018, The 27th International Conference on Computational Linguistics: System Demonstrations, Santa Fe, New Mexico, August 20-26, 2018*, pages 161–165, 2018.

John Duchi. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. 12:2121–2159, 2011.

Kevin Duh, Akinori Fujino, and Masaaki Nagata. Is Machine Translation Ripe for Cross-Lingual Sentiment Classification? In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 429–433, 2011.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. Learning Crosslingual Word Embeddings without Bilingual Corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1285–1295, 2016.

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. Multilingual Training of Crosslingual Word Embeddings. *Proceedings of the*

15th Conference of the European Chapter of the Association for Computational Linguistics, 1:894–904, 2017.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. Transition-Based Dependency Parsing with Stack Long Short-Term Memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, 2015. ISBN 9781941643723. doi: 10.3115/v1/P15-1033.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 1990. ISSN 03640213. doi: 10.1016/0364-0213(90)90002-E.

Manaal Faruqui and Chris Dyer. Improving Vector Space Word Representations Using Multilingual Correlation. *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, 2014. doi: 10.3115/v1/E14-1049.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A Smith. Sparse Overcomplete Word Vector Representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1491–1500. Association for Computational Linguistics, 2015. doi: 10.3115/v1/P15-1144.

Blaž Fortuna and John Shawe-Taylor. The use of machine translation tools for cross-lingual text mining. In *Proceedings of the ICML 2005 KCCA Workshop (KCCA)*, 2005.

Pascale Fung and Lo Yuen Yee. An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (ACL-COLING)*, pages 414–420, 1998.

Yarin Gal and Zoubin Ghahramani. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. In *Proceedings of the 29th Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 1019–1027, 2016.

William A Gale and Kenneth W Church. Estimation Procedures for Language Context: Poor Estimates are Worse than None. In *Compstat*, pages 69–74, Heidelberg, 1990. Physica-Verlag HD.

William A Gale, Kenneth W Church, and David Yarowsky. Discrimination Decisions for 100,000-dimensional Spaces. *Annals of Operations Research*, 55(2):323–344, jun 1995.

Kuzman Ganchev and Dipanjan Das. Cross-Lingual Discriminative Learning of Sequence Models with Posterior Regularization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1996–2006, 2013.

Debasis Ganguly, Johannes Leveling, and Gareth Jones. Cross-lingual Topical Relevance Models. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 927–942, 2012.

Éric Gaussier, Jean-Michel Renders, Irina Matveeva, Cyril Goutte, and Hervé Déjean. A Geometric View on Bilingual Lexicon Extraction from Comparable Corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 526–533, 2004.

Stuart Geman and Donald Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

Koen Gheuens. Spelling op het internet; de chaos becijferd. *Levende Talen Tijdschrift*, 13(1):26–35, 2012. ISSN 1566-2713.

Alfio Massimiliano Gliozzo and Carlo Strapparava. Exploiting Comparable Corpora and Bilingual Dictionaries for Cross-Language Text Categorization. In *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics and the 21st International Conference on Computational Linguistics (ACL-COLING)*, 2006.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep Sparse Rectifier Neural Networks. *AISTATS '11: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 15:315–323, 2011. ISSN 15324435. doi: 10.1.1.208.6449.

Andrew R Golding. A Bayesian Hybrid Method for Context-sensitive Spelling Correction. *Proceedings of the Third Workshop on Very Large Corpora*, 3:15, 1996.

Andrew R Golding, Dan Roth, Claire Cardie, and Raymond Mooney. A Winnow-Based Approach to Context-Sensitive Spelling Correction*. *Machine Learning*, 34: 107–130, 1999.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. BilBOWA: Fast Bilingual Distributed Representations without Word Alignments. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 748–756, 2015.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings of the International Joint Conference on Neural Networks*, volume 4, pages 2047–2052, 2005.

Yuhong Guo and Min Xiao. Cross Language Text Classification via Subspace Co-regularized Multi-view Learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012a.

Yuhong Guo and Min Xiao. Transductive Representation Learning for Cross-Lingual Text Classification. In *Proceedings of the 12th IEEE International Conference on Data Mining (ICDM)*, pages 888–893, 2012b.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. Learning Bilingual Lexicons from Monolingual Corpora. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 771–779, 2008.

Viktor Hangya, Fabienne Braune, Alexander Fraser, and Hinrich Schütze. Two Methods for Domain Adaptation of Bilingual Tasks: Delightfully Simple and Broadly Applicable. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 810–820, 2018.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical Correlation Analysis: An overview with Application to Learning Methods. *Neural computation*, 16(12):2639–2664, 2004.

Johannes Hellrich and Udo Hahn. Exploiting Parallel Corpora to Scale Up Multilingual Biomedical Terminologies. In *Proceedings of Medical Informatics Europe (MIE)*, pages 575–578, 2014.

Karl Moritz Hermann. Distributed Representations for Compositional Semantics. 2014.

Karl Moritz Hermann and Phil Blunsom. Multilingual Models for Compositional Distributed Semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 58–68, Baltimore, Maryland, 2014a. Association for Computational Linguistics.

Karl Moritz Hermann and Phil Blunsom. Multilingual Distributed Representations without Word Alignment. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014b.

Het Nieuwsblad. Dt-fout in Het Journaal, feb 2013. URL https://www.nieuwsblad.be/cnt/dmf20130213_033.

Het Nieuwsblad. Paleis maakt dt-fout in kerstboodschap, 2017. URL https://www.nieuwsblad.be/cnt/dmf20171211_03236553.

Geert Heyman, Ivan Vulić, and Marie-Francine Moens. Bilingual Lexicon Induction by Learning to Combine Word-Level and Character-Level Representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1085–1095, 2017.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Lecture 6a Overview of Minibatch Gradient Descent, 2012.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Thomas Hofmann. Probabilistic Latent Semantic Analysis. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 289–296, 1999.

Astrid Houthuys. 'De Schrijfassistent': uw nieuwe personal schrijfcoach, 2016. URL http://www.standaard.be/cnt/dmf20161029_02546399.

Yuening Hu, Ke Zhai, Vladimir Eidelman, and Jordan L Boyd-Graber. Polylingual Tree-Based Topic Models for Translation Domain Adaptation. In *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1166–1176, 2014.

Kejun Huang, Matt Gardner, Evangelos Papalexakis, Christos Faloutsos, Nikos Sidiropoulos, Tom Mitchell, Partha P Talukdar, and Xiao Fu. Translation Invariant Word Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015. ISBN 9781941643327.

Ann Irvine and Chris Callison-Burch. Supervised Bilingual Lexicon Induction with Multiple Monolingual Signals. In *Proceedings of the 14th Meeting of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 518–523, 2013.

Ann Irvine and Chris Callison-Burch. A Comprehensive Analysis of Bilingual Lexicon Induction. *Computational Linguistics*, 2016.

Jagadeesh Jagarlamudi and Hal Daumé III. Extracting Multilingual Topics from Unaligned Comparable Corpora. In *Proceedings of the 32nd Annual European Conference on Advances in Information Retrieval (ECIR)*, pages 444–456, 2010.

Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the Best Multi-stage Architecture for Object Recognition? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2146–2153, 2009. ISBN 9781424444205. doi: 10.1109/ICCV.2009.5459469.

Yu Jiang, Jing Liu, Zechao Li, and Hanqing Lu. Collaborative PLSA for multi-view clustering. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR)*, pages 2997–3000. IEEE, 2012.

Thorsten Joachims. Making large-scale SVM Learning Practical. In B Schölkopf, C Burges, and A Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, 1999.

Sungchul Kim, Kristina Toutanova, and Hwanjo Yu. Multilingual Named Entity Recognition using Parallel Data and Metadata from Wikipedia. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 694–702, 2012.

Diederik Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. *In Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–15, 2015.

Diederik Kingma and Max Welling. Auto-Encoding Variational Bayes. In *In Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–14, 2014.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. Inducing Crosslingual Distributed Representations of Words. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 1459–1474, 2012.

Tomáš Kočiský, Karl Moritz Hermann, and Phil Blunsom. Learning Bilingual Word Representations by Marginalizing Alignments. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 224–229, Baltimore, Maryland, jun 2014. Association for Computational Linguistics.

Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the 10th Machine Translation Summit (MT SUMMIT)*, pages 79–86, 2005.

Philipp Koehn and Kevin Knight. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL Workshop on Unsupervised Lexical Acquisition (ULA)*, pages 9–16, 2002.

Georgios Kontonatsios, Ioannis Korkontzelos, Jun'ichi Tsujii, and Sophia Ananiadou. Using a Random Forest Classifier to Compile Bilingual Dictionaries of Technical Terms from Comparable Corpora. In *Proceedings of the 14h Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 111–116, 2014a.

Georgios Kontonatsios, Ioannis Korkontzelos, Jun'ichi Tsujii, and Sophia Ananiadou. Combining String and Context Similarity for Bilingual Term Alignment from Comparable Corpora. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1701–1712, 2014b.

Kriste Krstovski and David A Smith. Online Polylingual Topic Models for Fast Document Translation Detection. In *Proceedings of the Workshop on Statistical MT*, 2013.

Yannick Laevaert. Automatische detectie en correctie van dt-fouten met recurrente neurale netwerken. *Master Thesis. KU Leuven.*, 2017.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised Machine Translation Using Monolingual Corpora Only. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–12, 2018.

Audrey Laroche and Philippe Langlais. Revisiting Context-based Projection Methods for Term-translation Spotting in Comparable Corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 617–625, 2010.

Victor Lavrenko, Martin Choquette, and W Bruce Croft. Cross-lingual Relevance Models. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 175–182, 2002.

Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. Hubness and Pollution: Delving into Cross-Space Mapping for Zero-Shot Learning. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 270–280, 2015.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

Gina-Anne Levow, Douglas W Oard, and Philip Resnik. Dictionary-based Techniques for Cross-language Information Retrieval. *Information Processing and Management*, 41(3):523–547, 2005.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. RCV1: a new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

Xiao Ling, Gui-Rong Xue, Wenyuan Dai, Yun Jiang, Qiang Yang, and Yong Yu. Can Chinese Web pages be classified with English data source? In *Proceedings of the 17th International Conference on World Wide Web (WWW)*, pages 969–978, 2008.

Michael Littman, Susan T Dumais, and Thomas K Landauer. Automatic Cross-Language Information Retrieval using Latent Semantic Indexing. In *Chapter 5 of Cross-Language Information Retrieval*, pages 51–62. Kluwer Academic Publishers, 1998.

Xiaodong Liu, Kevin Duh, and Yuji Matsumoto. Topic Models + Word Alignment = A Flexible Framework for Extracting Bilingual Dictionary from Comparable Corpus. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL)*, pages 212–221, 2013.

Bin Lu, Chenhao Tan, Claire Cardie, and Benjamin K. Tsou. Joint Bilingual Sentiment Classification with Unlabeled Parallel Corpora. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 320–330, 2011.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Bilingual Word Representations with Monolingual Quality in Mind. In *Proceedings Workshop on Vector Modeling for NLP, NAACL 2015*, 2015. ISBN 9781906740009.

Lidia Mangu and Eric Brill. Automatic Rule Acquisition for Spelling Correction. *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 187–194, 1997.

Gideon S Mann and David Yarowsky. Multipath Translation Lexicon Induction via Bridge Languages. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1–8, 2001.

Eric Mays, Fred J. Damerau, and Robert L. Mercer. Context Based Spelling Correction. *Information Processing and Management*, 27(5):517–522, 1991.

I Dan Melamed. Automatic Evaluation and Uniform Filter Cascades for Inducing n-best Translation Lexicons. In *Proceedings of Third Workshop on Very Large Corpora*, 1995.

Yishu Miao, Lei Yu, and Phil Blunsom. Neural Variational Inference for Text Processing. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, pages 1727–1736, 2016.

Yishu Miao, Edward Grefenstette, and Phil Blunsom. Discovering Discrete Latent Topics with Neural Variational Inference. 2017.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 27th Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 3111—-3119, 2013a.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations (ICLR), workshop track*, 2013b.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. In *CoRR, abs/1309.4168*, 2013c.

David Mimno, Hanna Wallach, Jason Naradowsky, David A Smith, and Andrew McCallum. Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 880–889, 2009.

Bhaskar Mitra, Eric T Nalisnick, Nick Craswell, and Rich Caruana. A Dual Embedding Space Model for Document Ranking. *CoRR*, abs/1602.0, 2016.

Vicent Montalt Resurrecció and Maria González-Davies. Medical Translation Step by Step: Learning by Drafting, 2014.

Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning*, (3):807–814, 2010.

Gonzalo Navarro. A Guided Tour to Approximate String Matching. *ACM Computing Surveys*, 33(1):31–88, 2001.

Roberto Navigli, Paola Velardi, and Aldo Gangemi. Ontology Learning and its Application to Automated Terminology Translation. *Intelligent Systems, IEEE*, 18(1):22–31, 2003. ISSN 1541-1672.

Xiaochuan Ni, Jian-Tao Sun, Jian Hu, and Zheng Chen. Mining Multilingual Topics from Wikipedia. In *Proceedings of the 18th International World Wide Web Conference (WWW)*, pages 1155–1156, 2009.

Xiaochuan Ni, Jian-Tao Sun, Jian Hu, and Zheng Chen. Cross-lingual Text Classification by Mining Multilingual Topics from Wikipedia. In *Proceedings of the 4th International Conference on Web Search and Web Data Mining (WSDM)*, pages 375–384, 2011.

Franz Josef Och and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, 2003.

J Scott Olsson, Douglas W Oard, and Jan Hajič. Cross-language Text Classification. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 645–646, 2005.

OpenTaal. Grammaticacontrole met LanguageTool, 2014. URL https://www.opentaal.org/begrippenlijst/216-grammaticacontrole-met-languagetool.

Junfeng Pan, Gui-Rong Xue, Yong Yu, and Yang Wang. Cross-Lingual Sentiment Classification via Bi-view Non-negative Matrix Tri-Factorization. In *Proceedings of the 15th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*, pages 289–300, 2011.

Michael J Paul and Roxana Girju. Cross-Cultural Analysis of Blogs and Forums with Mixed-Collection Topic Models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1408–1417, 2009.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

John Platt, Kristina Toutanova, and Wen-Tau Yih. Translingual Document Representations from Discriminative Projections. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 251–261, 2010.

Peter Prettenhofer and Benno Stein. Cross-Language Text Classification Using Structural Correspondence Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1118–1127, 2010.

Emmanuel Prochasson and Pascale Fung. Rare Word Translation Extraction from Aligned Comparable Documents. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 1327–1335, 2011.

Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. Hubs in Space: Popular Nearest Neighbors in High-dimensional Data. *Journal of Machine Learning Research*, 11:2487–2531, 2010.

Gurdeeshpal Randhawa, Mariella Ferreyra, Rukhsana Ahmed, Omar Ezzat, and Kevin Pottie. Using Machine Translation in Clinical Practice. *Canadian Family Physician*, 59(4):382–383, 2013.

Reinhard Rapp. Identifying Word Translations in Non-parallel Texts. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 320–322, 1995.

Martin Reynaert. All, and only, the Errors: More Complete and Consistent Spelling and OCR-error Correction Evaluation. In *Proceedings of the Sixth International Language Resources and Evaluation Conference (LREC)*, Marrakech, Morocco, 2008.

Leonardo Rigutini, Marco Maggini, and Bing Liu. An EM Based Training Algorithm for Cross-Language Text Categorization. In *Proceedings of the 2005 ACM International Conference on Web Intelligence (WIC)*, pages 529–535, 2005.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. A Survey of Cross-lingual Embedding Models. *Journal of Artificial Intelligence Research (JAIR)*, page accepted, 2018.

Stuart J Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach, 3rd Edition*. Pearson Education, 2010.

Helmut Schmid. Probabilistic Part-of-speech Tagging using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, 1994.

Peter Schönemann. A Generalized Solution of the Orthogonal Procrustes Problem. *Psychometrika*, 31(1):1–10, 1966.

Mike Schuster and Kuldip Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

Samuel Smith, David Turban, Steven Hamblin, and Nils Hammerla. Offline Bilingual Word Vectors, Orthogonal Transformations and the Inverted Softmax. pages 1–10, 2017.

Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. Inverted Indexing for Cross-lingual NLP. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1713–1722, 2015.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. On the Limitations of Unsupervised Bilingual Dictionary Induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.

Hubert Soyer, Pontus Stenetorp, and Akiko Aizawa. Leveraging Monolingual Data for Crosslingual Compositional Word Representations. In *Proceedings of*

the International Conference on Learning Representations (ICLR), San Diego, California, USA, 2015.

Akash Srivastava and Charles Sutton. Autoencoding Variational Inference For Topic Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–12, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. ISSN 15337928. doi: 10.1214/12-AOS1000.

Herman Stehouwer and Antal Van den Bosch. Putting the t Where it Belongs: Solving a Confusion Problem in Dutch. *Computational Linguistics in the Netherlands 2007: Selected Papers from the 18th CLIN Meeting*, pages 21–36, 2009.

Mark Steyvers and Tom Griffiths. Probabilistic Topic Models. *Handbook of Latent Semantic Analysis*, 427(7):424–440, 2007.

Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. Sparse Word Embeddings Using ‘ 1 Regularized Online Learning. *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pages 2915–2921, 2016. ISSN 10450823. doi: 10.1145/2939672.2939823.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 28th Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 3104–3112, 2014.

Oscar Täckström, Ryan McDonald, and Joakim Nivre. Target Language Adaptation of Discriminative Transfer Parsers. In *Proceedings of the 14th Meeting of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 1061–1071, 2013.

Tuomas Talvensaari, Ari Pirkola, Kalervo Järvelin, Martti Juhola, and Jorma Laurikkala. Focused web crawling in the acquisition of comparable corpora. *Information Retrieval*, 11(5):427—-445, 2008. ISSN 13864564. doi: 10.1007/s10791-008-9058-8.

Akihiro Tamura, Taro Watanabe, and Eiichiro Sumita. Bilingual Lexicon Extraction from Comparable Corpora Using Label Propagation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 24–36, 2012.

Tao Tao and ChengXiang Zhai. Mining Comparable Bilingual Text Corpora for Cross-language Information Integration. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 691–696, 2005.

Chen-Tse Tsai and Dan Roth. Cross-lingual Wikification Using Multilingual Embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 589–598, 2016.

Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. Cross-lingual Models of Word Embeddings: An Empirical Comparison. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1661–1670, 2016.

Masao Utiyama and Hitoshi Isahara. Reliable Measures for Aligning Japanese-English News Articles and Sentences. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 72–79, 2003.

Takehito Utsuro, Takashi Horiuchi, Yasunobu Chiba, and Takeshi Hamamoto. Semi-automatic Compilation of Bilingual Lexicon Entries from Cross-lingually Relevant News Articles on WWW News Sites. In *Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 165–176. Springer, 2002.

Antal Van den Bosch, Bertjan Busser, Sander Canisius, and Walter Daelemans. An Efficient Memory-based Morphosyntactic Tagger and Parser for Dutch. *Selected Papers of CLIN 2007*, 2007. ISSN 1572-199X.

Lonneke van der Plas, Paola Merlo, and James Henderson. Scaling up Automatic Cross-Lingual Semantic Role Annotation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 299–304, 2011.

Nina Verhaert and Dominiek Sandra. Homofoondominantie Veroorzaakt Dt-fouten Tijdens het Spellen en Maakt er ons Blind voor Tijdens het Lezen. *Levende Talen Tijdschrift*, 17(4):37—-46, 2016.

Alexei Vinokourov, Nello Cristianini, and John Shawe-Taylor. Inferring a Semantic Representation of Text via Cross-language Correlation Analysis. In *Proceedings of the 16th Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 1473–1480, 2002.

Theo Vosse. Detecting and correcting morpho-syntactic errors in real texts. In *Proceedings of the third conference on Applied natural language processing*, pages 111–118. Association for Computational Linguistics, 1992.

Thuy Vu, Ai Ti Aw, and Min Zhang. Feature-based method for document alignment in comparable news corpora. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 843–851, 2009.

Ivan Vulić and Anna Korhonen. On the Role of Seed Lexicons in Learning Bilingual Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 247–257, 2016.

Ivan Vulić and Marie-Francine Moens. A Study on Bootstrapping Bilingual Vector Spaces from Non-Parallel Data (and Nothing Else). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1613–1624, 2013a.

Ivan Vulić and Marie-Francine Moens. Cross-Lingual Semantic Similarity of Words as the Similarity of Their Semantic Word Responses. In *Proceedings of the 14th Meeting of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 106–116, 2013b.

Ivan Vulić and Marie-Francine Moens. Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 363–372, 2015.

Ivan Vulić and Marie-Francine Moens. Bilingual Word Embeddings from Comparable Data with Application to Bilingual Lexicon Extraction and Word Translation Disambiguation. *Journal of Artifical Intelligence Research*, 2016a.

Ivan Vulić and Marie-Francine Moens. Bilingual Distributed Word Representations from Document-Aligned Comparable Data. *Journal of Artificial Intelligence Research*, 55:953–994, 2016b.

Ivan Vulić, Wim De Smet, and Marie-Francine Moens. Identifying word translations from comparable corpora using latent topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 479–484, 2011.

Ivan Vulić, Wim De Smet, and Marie-Francine Moens. Cross-language information retrieval models based on latent topic models trained with document-aligned comparable corpora. *Information Retrieval*, 16(3):331–368, 2013.

Ivan Vulić, Wim De Smet, Jie Tang, and Marie-Francine Moens. Probabilistic topic modeling in multilingual settings: An overview of its methodology and applications. *Information Processing and Management*, 51(1):111–147, 2015.

Hanna M Wallach, David M Mimno, and Andrew McCallum. Rethinking LDA: Why priors matter. In *Advances in neural information processing systems*, pages 1973–1981, 2009.

Chang Wan, Rong Pan, and Jiefei Li. Bi-Weighting Domain Adaptation for Cross-Language Text Classification. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1535–1540, 2011.

Xiaojun Wan. Co-Training for Cross-Lingual Sentiment Classification. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 235–243, 2009.

Chang Wang, Liangliang Cao, and Bowen Zhou. Medical Synonym Extraction with Concept Space Models. In *Proceedings of IJCAI*, pages 989–995, 2015.

Hua Wang, Heng Huang, Feiping Nie, and Chris Ding. Cross-language Web Page Classification via Dual Knowledge Transfer Using Nonnegative Matrix Trifactorization. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 933–942, 2011.

Bin Wei and Christopher J Pal. Cross Lingual Adaptation: An Experiment on Sentiment Classifications. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 258–262, 2010.

Krzysztof Wołk and Krzysztof Marasek. Neural-based Machine Translation for Medical Text Domain. Based on European Medicines Agency leaflet texts. *Procedia Computer Science*, 64:2–9, 2015.

Min Xiao and Yuhong Guo. Semi-Supervised Representation Learning for Cross-Lingual Text Classification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1465–1475, 2013a.

Min Xiao and Yuhong Guo. A Novel Two-Step Method for Cross Language Representation Learning. In *Proceedings of the 27th Annual Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 1259–1267, 2013b.

Yan Xu, Luoxin Chen, Junsheng Wei, Sophia Ananiadou, Yubo Fan, Yi Qian, Eric I-Chao Chang, and Junichi Tsujii. Bilingual Term Alignment from Comparable Corpora in English Discharge Summary and Chinese Discharge summary. *BMC Bioinformatics*, 16(1):149:1—-149:10, 2015. ISSN 1471-2105. doi: 10.1186/s12859-015-0606-0.

David Yarowsky. Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. 1994.

Duo Zhang, Qiaozhu Mei, and ChengXiang Zhai. Cross-Lingual Latent Topic Extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1128–1137, 2010.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. Adversarial Training for Unsupervised Bilingual Lexicon Induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1959–1970, 2017.

Tao Zhang, Kang Liu, and Jun Zhao. Cross Lingual Entity Linking with Bilingual Topic Model. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2218–2224, 2013.

Hai Zhao, Yan Song, Chunyu Kit, and Guodong Zhou. Cross Language Dependency Parsing using a Bilingual Lexicon. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 55–63, 2009.

Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing Twitter and Traditional Media Using Topic Models. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval (ECIR)*, pages 338–349, 2011.

Will Zou, Richard Socher, Daniel Cer, and Christopher Manning. Bilingual Word Embeddings for Phrase-Based Machine Translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1393–1398, 2013.

Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the International Language Resources and Evaluation (LREC) Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, may 2010. ELRA.

List of publications

Journal articles

Geert Heyman, Ivan Vulić, Yannick Laevaert, Marie-Francine Moens. Automatic detection and correction of context-dependent dt-mistakes using neural networks. *CLIN Journal, accepted*

Geert Heyman, Ivan Vulić, Marie-Francine Moens. A Deep Learning Approach to Bilingual Lexicon Induction in the Biomedical Domain *BMC Bionformatics, 19(259) pages 1–15*, 2018

Geert Heyman, Ivan Vulić, Marie-Francine Moens. C-BiLDA extracting cross-lingual topics from non-parallel texts by distinguishing shared from unshared content. *Data Mining and Knowledge Discovery, 30(5), pages 1299–1323*, 2016

Susana Zoghbi, **Geert Heyman**, Juan Carlos Gomez, Marie-Francine Moens. Fashion meets computer vision and NLP at e-commerce search. *International Journal of Computer and Electrical Engineering, 8 (1), pages 31-43*, 2016

Susana Zoghbi, **Geert Heyman**, Juan Carlos Gomez, Marie-Francine Moens. *Cross-modal fashion search. Lecture Notes in Computer Science, 9517, pages 367-373*, 2016.

Peer-reviewed international conference articles

Quynh Ngoc Thi Do, Artuur Leeuwenberg, **Geert Heyman**, Marie-Francine Moens. A Flexible and Easy-to-use Semantic Role Labeling Framework for Different Languages. *In Proceedings The 27th International Conference on Computational Linguistics (COLING): System Demonstrations, pages 161-165*, 2018

Geert Heyman, Ivan Vulić, Marie-Francine Moens. Bilingual Lexicon Induction by Learning to Combine Word-Level and Character-Level Representations. *In Proceedings*

of the 15th International Conference of the European Chapter of the Association for Computational Linguistics (EACL), pages 1085–1095, 2017

Meeting abstracts, presented at internationcal conferences and symposia

Quynh Ngoc Thi Do, Artuur Leeuwenberg, **Geert Heyman** and Marie-Francine Moens. How to use DameSRL: A framework for deep multilingual semantic role labeling. *CLARIN2018 Book of Abstracts*, 2018

Yannick Laevaert, **Geert Heyman**, Marie-Francine Moens. Automatic detection and correction of real-word dt-mistakes in Dutch using recurrent neural networks. *Book of Abstracts Computational Linguistics in the Netherlands (CLIN 28)*, 2018

Vincent Vandeghinste, Tom Vanallemeersch, Bram Bulté, Liesbeth Augustinus, Frank Van Eynde, Joris Pelemans, Lyan Verwimp, Patrick Wambacq, **Geert Heyman**, Marie-Francine Moens, Iulianna van der Lek-Ciudin, Frieda Steurs, Ayla Rigouts Terryn, Els Lefever, Arda Tezcan, Lieve Macken, Sven Coppers, Jens Brulmans, Jan Van den Bergh, Kris Luyten, Karin Coninx. *The SCATE project: Highlights. 21st annual conference of the European Association for Machine Translation (EAMT)*, 2018

Geert Heyman, Ivan Vulić, Marie-Francine Moens. Bilingual Lexicon Induction by Learning to Combine Word-Level and Character-Level Representations. *Book of Abstracts Computational Linguistics in the Netherlands (CLIN 27), pages 66-67*, 2017

Iulianna van der Lek-Ciudin, Ayla Rigouts Terryn, **Geert Heyman**, Els Lefever and Frieda Steurs. Translator's methods of acquiring domain-specific terminology. Information retrieval in terminology using lexical Knowledge Patterns. *In Proceedings of the 21st European Symposium on Languages for Special Purposes (LSP)*, 2017

Vincent Vandeghinste, Tom Vanallemeersch, Liesbeth Augustinus, Frank Van Eynde, Joris Pelemans, Lyan Verwimp, Patrick Wambacq, **Geert Heyman**, Marie-Francine Moens, Iulianna van der Lek-Ciudin, Frieda Steurs, Ayla Rigouts Terryn, Els Lefever, Arda Tezcan, Lieve Macken, Sven Coppers, Jan Van den Bergh, Kris Luyten and Karin Coninx. SCATE - Smart Computer-Aided Translation Environment. *In Proceedings of the 20th Annual Conference of the European Association for Machine Translation (EAMT)*, 2017.

Vincent Vandeghinste, Tom Vanallemeersch, Liesbeth Augustinus, Joris Pelemans, **Geert Heyman**, Iulianna van der Lek-Ciudin, Arda Tezcan, Donald Degraen, Jan Van den Bergh, Lieve Macken, Els Lefever, Marie-Francine Moens, Patrick Wambacq, Frieda Steurs, Karin Coninx, Frank Van Eynde. SCATE - Smart Computer-Aided Translation Environment. *In Proceedings of the 19th Annual Conference of the European Association for Machine Translation (EAMT)*, 2016

Susana Zoghbi, **Geert Heyman**, Juan Carlos Gomez, Marie-Francine Moens. Cross-modal attribute recognition in fashion. *NIPS Multimodal Machine Learning workshop*, 2015.

Vincent Vandeghinste, Tom Vanallemeersch, Frank Van Eynde, **Geert Heyman**, Marie-Francine Moens, Joris Pelemans, Patrick Wambacq, Iulianna Van der Lek-Ciudin, Arda Tezcan, Lieve Macken, Véronique Hoste, Eva Geurts and Mieke Haesen. Smart Computer Aided Translation Environment. *In Proceedings of the 18th Annual Conference of the European Association for Machine Translation (EAMT)*, 2015

FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
LANGUAGE INTELLIGENCE AND INFORMATION RETRIEVAL
Celestijnenlaan 200A box 2402
B-3001 Leuven
geert.heyman@cs.kuleuven.be
http://www.people.cs.kuleuven.be/~geert.heyman