Data Min Knowl Disc
DOI 10.1007/s10618-012-0270-1

Representations for multi-document event clustering

Wim De Smet · Marie-Francine Moens

Received: 27 October 2008 / Accepted: 11 May 2012

Abstract We study several techniques for representing, fusing and comparing content representations of news documents. As underlying models we consider the vector space model (both in a term setting and in a latent semantic analysis setting) and probabilistic topic models based on latent Dirichlet allocation. Content terms can be classified as topical terms or named entities, yielding several models for content fusion and comparison. All used methods are completely unsupervised. We find that simple methods can still outperform the current state-of-the-art techniques.

Keywords Text mining · Probabilistic content models · Clustering

1 Introduction

When processing news stories of several accounts of a certain happening, it is often relevant to determine whether two stories report on the same event. An event is defined here as a well-specified happening at a certain moment in time (a single day or a short period) which deals with a certain set of topics (e.g., a hurricane and inundations, an earthquake and lack of drinking water) and involves some named entities. Those entities are, for instance, the actors (such as the names of the leading persons or companies) and the location where the event occurred. News stories are typical examples. Broadcast news can be segmented in different stories that each report on a single event.

Responsible editor: R. Bayardo.

W. De Smet (✉) · M.-F. Moens
Department of Computer Science, K.U. Leuven, Leuven, Belgium
e-mail: wim.desmet@cs.kuleuven.be

M.-F. Moens
e-mail: marie-francine.moens@cs.kuleuven.be

Published online: 20 June 2012

Written news typically is recorded per story, where each story reports on one event. However, different sources or even the same source can produce several stories on the same event, which we might group as a preprocessing step for mining, summarizing or searching purposes. In this article we focus on the clustering of textual news stories coming from different sources that describe the same event. We use the words "story" and "document" interchangeably.

Any clustering depends on the quality of the distinction between the elements, and the quantitative representation thereof, i.e. the distance or dissimilarity function. Our main goal is to investigate the suitability of existing document representations for the event detection task in a repository of news stories. As underlying models we consider the vector space model [both in a term setting as in a latent semantic analysis (LSA) setting (Hofmann 1999)] and probabilistic topic models based on latent Dirichlet allocation (LDA) (Blei et al. 2003). These methods also include representing documents along different angles. Views or *aspects*[1] of their content enhance these distance computations. Indeed, documents can contain different kinds of information. Based on the definition of a news event, these aspects comprise the event's topics and its entities.

There is the long standing *vector space model* (Salton 1989) for document representation where the importance of topics is signalled by the term weights usually computed as the term frequency multiplied by the inverse document frequency. Similarity and distance computations rely on real distances in a geometric vector space. Latent class models have recently become popular for representing content. Among them is the LSA model (Deerwester et al. 1990), a vector space model that represents the terms of a document in a lower dimensional space representing semantic concepts or topics. There are the newer *probabilistic topic models*, such as probabilistic LSA (Hofmann 1999) and LDA (Blei et al. 2003), which see a document as a mixture of topics and topics as mixtures of words. Similarity or dissimilarity are respectively seen here as convergence or divergence of probability distributions. Current information extraction and analysis techniques enable the detection of the aspects. For instance, we already have reliable named entity recognizers for common languages such as English that classify proper names into their semantic categories. Typical semantic categories are locations, persons and organizations. Similarity or dissimilarity computations can alternatively be based on the extracted content elements.

When documents are represented by different aspects (i.e. different parts of their content), one can compute their dissimilarity based on a single aspect, or based on the combination of similarities each obtained by considering a different aspect. In a first model, what we could call *early fusion* or full text, we do not split the information found, i.e. we combine all the features (in our case the topical words and the named entities) in a vector or probabilistic model and compute the similarity. In a *late fusion* model, for each aspect there is a different representation, for each aspect representation the dissimilarity between two documents is computed, and the dissimilarities are then fused with an evidence combination function. We will also use an *intermediate fusion* model, where the representations of the different aspects are used together to calculate hidden variables that can describe a document; the dissimilarity is then expressed in

[1] "Aspects" here have a broader meaning than the latent probabilistic topics generally meant, which we discuss further in this article.

terms of these latent variables. All used methods are completely unsupervised. We compare methods that are trained either on the same, or on a different corpus than the one on which they are applied.

The contributions of this article are the benchmarking study of the value of different document representations (algebraic and probabilistic) in an event clustering task and the value of different fusion models that can be applied for computing the dissimilarity in content between two document pairs. An extensive study of the representation models and of different models of fusion in the dissimilarity computations is to our knowledge non-existing in the literature. This study shows that simple models of text representation, based on term frequency and inverse document frequency, are still competitive compared to current latent semantic models.

The techniques proposed in this article offer many avenues for future expansions and applications. The methods, although tested in the field of news event clustering, could well be applied on other types of texts or media in different comparison or clustering tasks. It would be interesting to assess how the resulting representations influence the computation of similarity of textual content. Documents very often are analyzed and compared. For example, in patent files one looks for similarity in subject domains and methods. Facts in police and intelligence reports match based on names, locations, car brands, modus operandi and other extracted information. Automatic grading of essay type exams of students and detection of plagiarism constitute another domain, where content comparison is important. In law, precedent reasoning compares cases on the basis of similar facts and correspondence in legal factors and issues. Opinions in political speeches can be compared focusing on specific issues. In medical patient reports common patterns based on certain extracted information can be found. In each of these domains, the text could be divided into different aspects that might be important for text comparison. This article provides a framework for such a comparative analysis.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes our methodology focusing on document representations, named entity recognition (NER), dissimilarity metrics and clustering. Our methods are both generically and more specifically described, the latter focusing on the event clustering task. Comparative tests and evaluations are presented in Sect. 4. We conclude by citing the main findings and the many applications of our research.

2 Related research

Assessing similarities and differences between text documents is a long standing problem. Established algebraic approaches represent documents as term vectors (where terms are possibly weighted by a $tf \times idf$ factor) and compute the cosine of the angle of the term vectors (Salton 1989) (vector space model). This model assumes that the vectors that span the geometric space are pairwise orthogonal, an assumption which is violated in real texts (Wang et al. 1992).

In order to cope with synonym and related terms, an alternative vector space model incorporates document representations based on LSA (Deerwester et al. 1990) and singular value decomposition of the term-by-document matrix of a document

collection. The LSA model is also a kind of aspect model as documents are represented by several "topics". (Gong and Liu 2001; Steinberger et al. 2007, 2010; Steinberger and Ježek 2009) perform a singular value decomposition on the term-by-sentence matrix of a set of documents, from which a multi-document summary is obtained.

Recently probabilistic topic models have been proposed as an alternative to LSA. Proponents claim topic models deal with polysemy in a more natural way. These models have rapidly gained popularity as an unsupervised way of describing documents. The main idea is that documents are viewed as a mixture of topics and each topic as a mixture of words. Several latent topic models exist, such as probabilistic latent semantic analysis or pLSA (Hofmann 1999) and LDA (Blei et al. 2003). In both cases the topic and word distributions are learned from a large training corpus, but newer models such as LDA learn additional latent variables that are independent of the training corpus, so that also the topic distributions of new, previously unseen documents can be inferred. In addition, LDA models require less parameters to train, whereas this number of parameters grows linearly with the number of documents in a pLSA model. Variant models have been studied by Buntine and Jakulin (2006). Li et al. (2005) build a probabilistic generative model for retrospective news events detection, where an event generates persons, locations, keywords as named entities apart from a time pointer, in this way combining the different aspects. We build further upon these models, but investigate the effect of splitting content representations and fusion of the dissimilarity values obtained with the content representations.

Recent work on probabilistic topic models combines metadata content with topic models as is done by Mccallum et al. (2005) who steer the discovery of topics according to the relationships between people. These models, although very valuable in other ways, appear to augment the words in a document with semantics in a limited way, but the document representation is still quite a rudimentary reflection of its semantics. Structured models that take into account topic correlations have been proposed by Li and Mccallum (2006). This model does not yet take into account extracted information such as named entities.

In the computational linguistics domain, paraphrasing techniques have been developed in order to detect similar content by considering matching of word co-occurrences, matching noun phrases, verb classes, proper nouns, etc. (Hatzivassiloglou 1998; Barzilay and Lee 2003), where the matching patterns might be learned in an unsupervised way using sentences that already describe comparable content. From a more cognitive point of view, automatic text understanding and similarity detection, compared to the human capability of doing so has been discussed in Lee and Welsh (2005), Tsatsaronis et al. (2010) and Stone et al. (2011). As a variation of the paraphrasing models, researchers have attempted to detect contradictions in natural language statements, for instance, by means of handcrafted rules (Mckeown and Radev 1995) or learning contradiction models from annotated sentences (de Marneffe et al. 2008). These techniques are usually confined to finding similarities or differences in a fine grained way, but their use is currently still restricted by a rather low performance, making them less suited for content comparison.

Our work presented here introduces semantic representations of documents which move beyond simple topic models and incorporate specific information extracted from the documents, but are more robust than the fine-grained representations obtained

through an in depth natural language processing. Information extraction technologies that semantically classify certain information in the documents (such as NER) in combination with probabilistic topic models offer many interesting possibilities for representing and comparing texts possibly along different aspects of content.

Event detection has received substantial interest in information retrieval research (often as part of topic detection and tracking[2] (TDT) tasks. Early work on retrospective event detection based on a hierarchical agglomerative clustering (group average clustering) is done by Yang et al. (1999) (building further on Cutting et al. 1992). The events are clustered based on lexical (single words) similarity of the documents and temporal proximity. The temporal proximity parameter avoids clustering documents that are too far apart in time. Many different studies on event detection followed these initial efforts (see Allan et al. 2002 for the main approaches). Many of them rely on a vector space representation of the documents, where more recent approaches make a distinction between named entities and non named entity words (e.g., Kumaran and Allan 2004). In such a scheme each term type might receive a different weight, possibly learned from a training corpus (Zhang et al. 2007). As mentioned above we use the vector space model, but our task is different from the TDT task. We do not consider a live stream of documents which are ordered in time. Our goal is only to assess similarities in content between documents in the most effective way.

Probabilistic models for representing events in documents are scarce. In Allan et al. (2003), a simple probabilistic language model is used as a document representation. Other research on integrating named entities in an event detection task include Makkonen et al. (2002) and Zhang et al. (2007), where Zhang et al. (2007) demonstrated correlations between named entity types and news classes.

In this paper we also benchmark models characterized by an intermediate or late fusion of evidence obtained from the documents. In the intermediate fusion model we split a document in entities and other words and train a parallel LDA model where topics are learned paired on the paired entity and other words documents. In the late fusion model documents are compared by means of their different aspects, and these similarity results are fused. In classification, when dealing with heterogeneous information, late fusion approaches are common [e.g., in semantic analysis of video (Snoek 2005) or in spam email filtering (Hershkop and Stolfo 2005)]. When fusion information from different sources, the impact of each of the sources might differ. Such weight variables (e.g., used in a linear interpolation of the evidence) can be learned from a training set. We did not follow this approach, because we did not include any supervision in our model. We fused dissimilarity values by using either their average or maximum scores. Other methods are here possible including probabilistic models (Pearl 1991) or models based on Dempster-Shafer evidential theory (Shafer 1976). These "split models" are in line with recent work in information retrieval where different language models are built from the different parts of a document (e.g., title, HTML anchor texts, body) and combined in a ranking function (Wang et al. 2010).

[2] Note that *topic* here is used here in a different sense than in latent probabilistic topic models (Nallapati et al. 2004).

3 Methodology

Our first task is to create a document representation d_i. We begin by defining the aspects that we consider in a news document, and continue with the different techniques to represent them. These techniques include a term vector or a set of term vectors (*vector space term model*), a lower rank projection of these term vectors (using LSA) or probabilistic content models built from the document (using LDA).

Since we have defined a document as containing several aspects, we categorize each of the models that we investigate into one of the following three classes, depending on when the information extracted from the aspects is fused into a final document comparison.

- *Early fusion* models create a document description that is extracted from all terms combined, effectively fusing the aspects before creating a description and comparing the documents in full.
- *Late fusion* models create a description for each aspect, compare each aspect with its counterpart in another document, and then fusing the aspect comparisons into a document comparison.
- *Intermediate fusion* models, finally, use the aspects separately to create a document description, fusing the aspect information before comparing.

This section also describes the second task: how the documents characterized by the different content representations are grouped into event clusters.

3.1 Aspects

Topical words The topical words of a news event are those terms that express the generally applicable subjects. For example, in a story about an earthquake, subjects may be the earthquake itself, damage to houses, flooding, etc. With "generally applicable", we mean that every story on earthquakes might contain these words. The information that unambiguously separates one event from another similar event lies in the named entities.

Named entities Named entities are entities in the real world that have unique names. Different types of named entities occurring often in news reports are for instance *persons* and *organizations* (the actors of an event), *locations* (where the event takes place) and *timestamps*. NER detects and classifies the entities.

We use the OpenNLP[3] package, which detects noun phrase chunks in the sentences that represent persons, locations, organizations and dates. To improve the recognition of person names, we use a dictionary of names, which we have extracted from the Wikipedia website.

[3] http://opennlp.sourceforge.net.

3.2 Underlying representations

3.2.1 Vector space model

In the *vector space model* (Salton 1989), a document is represented as a vector in a n-dimensional space: $d_i = [w_i^1, w_i^2, \ldots, w_i^n]$, where $n =$ the number of used features. The features w_i commonly represent the terms of the vocabulary by which the documents in the collection are indexed. Term weights might be binary, indicating term presence or absence, or have a numerical value to indicate the importance of the terms in the document. For instance, weights are often computed by a $tf \times idf$ weighting scheme, where the term weight is proportional with the number of times the term occurs in the considered document text (tf) and inversely proportional with the number of documents of a reference collection in which the term occurs (idf). Term vectors are normalized by division with their Euclidean norm, $||d_i||$.

Topical Words Representation The terms used in the vector representation of the topical words are all the topical words in the documents weighted with their $tf \times idf$ values, minus predefined stop words and words that are filtered away because of a low idf.

NER For named entities, we consider two types of representations. In the first, all named entities in a document are represented by one t-dimensional vector: $d_i = [e_i^1, e_i^2, \ldots, e_i^t]$. In the second approach, we separate the named entities by the named entity classes of persons, locations and organizations. This yields three separate named entity vectors per document: $d_i = \{[p_i^1, p_i^2, \ldots, p_i^u], [l_i^1, l_i^2, \ldots, l_i^v], [o_i^1, o_i^2, \ldots, o_i^w]\}$, representing respectively the identified persons, locations and organizations.

Named entity vectors use solely the term frequency as weighting factor. The weight of a named entity does not need to be demoted by an inverse document frequency factor in the dataset used for our experiments: there is no such thing as an unimportant named entity. Even if the entity occurs in many documents, it is considered as discriminative for the event considered.

Dissimilarity The similarity between two vectors is computed as the cosine of the angle between the two normalized vectors, thus the distance dis between two documents d_i and d_j: $dis(d_i, d_j) = 1 - \cos(\widehat{d_i, d_j}) = 1 - \frac{d_i \cdot d_j}{\|d_i\| \cdot \|d_j\|}$. This distance can be computed considering the term vector of the documents containing all terms, or by considering separate representations (e.g., named entities).

3.2.2 LSA

Because there are a large number of features (words) and their occurrence in documents is sparse, models of LSA have been proposed (Deerwester et al. 1990). The documents in a collection are represented by their term by document matrix, which captures the weight of a term in a document using the same $tf \times idf$ value used in the vector space model. A singular value decomposition of this matrix is computed and a low-rank approximation of the term-document matrix is constructed. The original documents are then projected into the obtained S-dimensional space. This S, the rank

or number of dimensions is set a priori and is usually chosen as a value between 100 and 300 for the English language for document representation. We have performed tests within and outside of this range, and have seen that generally these boundaries yield the best results.

Dissimilarities in the LSA model are also calculated using the cosine metric, as in the vector space model.

3.2.3 LDA

In the example of an earthquake event, we mentioned that it may cover topics such as flooding, damage, etc. Probabilistic models define a mathematical basis for this idea. For a particular language, one can define a number of topics, each characterized by a probability distribution over words. An event can be seen as a mixture of these topics, where some topics are prominently and others only marginally present. As we want an unsupervised approach, we need a way to automatically define and detect these topics. For this purpose, LDA is used.

LDA is a statistical model for document generation, presented in Blei et al. (2003). The idea is that documents are created according to a random mixture of topics, sampled from a *topic distribution*. These topics generate a random set of words, sampled from each topic's *word distribution*. LDA learns both kinds of distributions in an unsupervised way, based on a training set of documents. Due to the shared use of the term "topic", both meaning a word-distribution in LDA and the content of a news story, confusion may arise. The context will help to disambiguate between the two, as well as our use of "distribution" or "probability", when referring to the first. "Topic representation" will always refer to the representation of the topical words without named entities.

The algorithm, as described in Blei et al. (2003), is based on a generative process of a document (Fig. 1). A corpus has two variables associated with it: α and β. For each document in the corpus, a multinomial S-dimensional parameter θ is sampled from the Dirichlet distribution with parameter α. θ refers to the multinomial distribution with S dimensions, where S is set a priori. Then, for each of the N word positions in the document, a topic z_n is assigned by sampling the multinomial distribution based on θ. For each topic, β is another multinomial distribution over the vocabulary V. So, each word w_n is selected according to $p(V|\beta, z_n)$.

In summary:

1. For each of the M documents d_i, choose $\theta \sim Dir(\alpha)$
2. For each of the N word positions w_n:
   (a) Choose a topic $z_n \sim Multinomial(\theta)$
   (b) Choose a word w_n from $p(V|z_n, \beta)$

The probability of a document, using these parameters, becomes:

$$p(d|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta$$

The values for α and β are chosen in accordance with the guidelines presented in Griffiths et al. (2007), i.e. $\alpha = 0.01$ and $\beta = \frac{50}{S}$.

Since θ and β appear together in elaborate equations because of their dependency, its estimation becomes intractable. Several solutions have been described in the literature. We employed Gibbs sampling, which calculates iteratively

$$p(z_{ji}|\mathbf{w}_j, \mathbf{z}_j^{\neg i}) = \frac{n_{d_j z_{ji}}^{\neg i} + \alpha}{\sum_{z'} n_{d_j z'}^{\neg i} + K \cdot \alpha} \cdot \frac{v_{z_{ji} w_{ji}}^{\neg i} + \beta}{\sum_w (v_{z_{ji} w}^{\neg i} + \beta)} \tag{1}$$

where $n_{d_{jz}}$ is the number of times that topic z has been sampled from the multinomial distribution specific to document d_j and v_{zw} is the number of times that word w has been generated by topic z. The superscript $^{\neg i}$ means that the current word being sampled does not appear in those counts. By updating this formula for each word in each document repeatedly, we converge towards the correct distributions.

In order to obtain clean word distributions, we first remove stop words and low-idf words from the training set, as we did with the vector space model. It has been shown in the literature that, if the training set is large and diverse enough, the topic-word distributions are stable, as our results later will confirm. A typical value for S to obtain valid topics is usually chosen as a value between 100 and 300 for the English language (Griffiths et al. 2007).

The power of LDA lies in the natural modeling of synonymous and related words and of polysemous words. Another advantage is the possibility of *inferring* the topic distributions of new documents. In certain settings this inference is very useful. For instance, when dealing with a stream of news stories, new events are added continuously, making a frequent retraining of the system inconvenient. In that case it is computationally interesting to train a model once, and then infer it on the newly added events. We will compare inferring from a pre-trained model with training on the news stream itself.

Topic Representation In LDA defined on the document's full text, the entities are part of the topic distributions. This has the undesirable property that entities that were not apparent in the training set (which, given the dynamic nature of news, occurs often) cannot influence the topic inference of a new event.

Named Entity Representation Because named entities in news change dynamically (e.g. person, location and organization names occur which never had been mentioned before), named entity models are difficult to learn from text corpora. Therefore, we chose a different probabilistic representation of entities.

We create a probabilistic distribution, much in the same way as we would create a vector in the vector space model. Normalization (i.e. division by $\sum_i d_{ji}$ rather than $||d_j||$, where d_{ji} is the weight of the ith dimension of d_j) ensures the property of summation to 1. This applies both for the models based on all entities together, as those where entities are separated by their class.

Dissimilarity To compare two probability distributions, we use the *symmetric Kullback–Leibler divergence* (KL) of the n-dimensional probability distributions d_i and d_j, defined as

$$KL(d_i, d_j) = \frac{1}{2}\sum_{l=1}^{n} d_{il} \log\left(\frac{d_{il}}{d_{jl}}\right) + \frac{1}{2}\sum_{l=1}^{n} d_{il} \log\left(\frac{d_{il}}{d_{jl}}\right)$$

where d_{il} is the probability of the lth dimension of d_i. In case of entities, d_i is the term vector normalized by its sum, for LDA generated topics it is the θ associated with the document.

The KL is a common measure when comparing probability distributions. However, in its natural form it is asymmetrical, i.e. $KL(d_i, d_j) \neq KL(d_j, d_i)$. An asymmetrical distance function is an undesirable feature in a clustering task, as it would mean that when two elements are compared in a different.

Whereas the cosine-measure is limited between 0 and 1, Kullback–Leibler is theoretically unbound. To be able to compare the distances and divergences, we need to normalize the KL measures between a set of documents. We do this by simply dividing them by the largest KL in the set.

Normalizing the divergences yield a value between 0 and 1, being 0 if and only if the distributions are equal, and 1 if they reach maximum divergence.

3.3 Fusion models

Now that we have defined the aspects of a document, the different underlying representations and the distance metric for each representation, we can fuse all this information for document comparison.

3.3.1 Early fusion

Early fusion, as defined in the beginning of this section, fuses the information from the aspects before a representation is built. This implies that we simply use the full text of a document as input for the models. For the vector space model, this means using all terms in the vector. The LSA approach calculates its topics from the same vectors, as does LDA in training mode. When inferring an existing LDA model however, we do not include the named entities of a document: as explained in Sect. 3.2.3, a pre-trained model does not contain named entities, as it would become outdated quickly in a news setting.

When the full documents' representations are calculated, we can compare the documents using the appropriate distance metrics.

3.3.2 Late fusion

In late fusion, to compare two documents we first calculate the distances between the respective aspects of each document, and then combine this information into one value. Formally, for a document d_i we have defined the aspects $\mathbf{A_{d_i}}$ of *topical words* ($A^t_{d_i}$), *entities* ($A^e_{d_i}$), where alternatively $A^e_{d_i}$ can be split in *persons* ($A^p_{d_i}$), *locations* ($A^l_{d_i}$) and *organizations* ($A^o_{d_i}$). Dividing d_i into its topical words and entities will from now on be referred to as a "*two-split*", while topical words, persons, locations and organizations constitutes a "*four-split*".

The obtained dissimilarities between different aspects can be combined in several ways to obtain a global document dissimilarity. We propose two ways of combining them:

1. $max\_dis(d_i, d_j) =$

$$\max_k \; dis(A^k_{d_i}, A^k_{d_j}), \quad k = 1 \rightarrow N,$$

2. $average\_dis(d_i, d_j) =$

$$\frac{1}{N} \sum_{k=1}^{N} dis(A^k_{d_i}, A^k_{d_j}),$$

where N is the number of aspects the document is split into: $N = 2$ for the two-split, $N = 4$ for the four-split.

Each of these combination functions imposes different views of what is important when comparing documents. The max-function, which computes the maximum of the two dissimilarities, ensures that two documents are dissimilar when at least one of the aspects has dissimilar distributions: if two documents differ too much in one aspect, then it does not matter whether the other distribution is close or not. In an event setting, this translates into the following: if we detect different actors or locations, then we assume that we deal with different events, even when their topics are similar. Analogically, events with different topics that happen at the same location will be treated as different events.

The ave-function, which computes the average of the two dissimilarities, is more tolerant towards differences. Even when covering the same event, different sources may stress different locations, interview different persons, etc. However, as NER is not yet perfect, it is possible (as we have encountered in our evaluations) that essential, shared entities are not recognized. This makes the named entity distribution's divergence larger then it should be. Averaging with the topic distribution dissimilarity smooths these differences, but can be too forgiving in other cases. Late fusion performs first the representation step on each aspect of a document separately, i.e. either on the topical words and then on the named entities (in a two-split late fusion), or on the topical words, persons, locations and organizations (in a four-split late fusion).

3.3.3 Intermediate fusion model

As a third level of fusion, we have also implemented an intermediate fusion model, which splits a document into its aspects, but generates only one representation for the document. The model we implemented with these characteristics is a generative model that trains topics on the different aspects separately. Each document is modeled as a distribution of topics over words and over named entities. A document is then defined by one representation to which all aspects have contributed, rather than by every aspect separately.

This model is different from late fusion in that both the named entities as the topical words contribute to the hidden variables of the document (its topic distribution) instead of only the topical words, and it is different from an early fusion LDA model in that

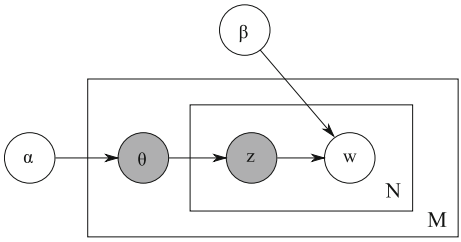

Fig. 1 Graphical model of LDA

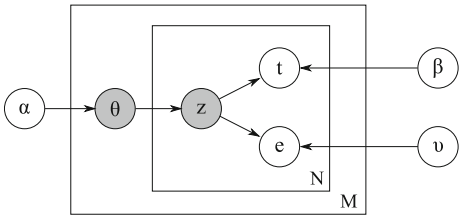

Fig. 2 Graphical model representing intermediate topics (z), topic words (t) and entities (e)

the weight of the named entities is increased as they now appear in their own "entity topics", which are linked but not equal to the "word topics" (see further) (Fig. 1).

This intermediate fusion model is presented in Fig. 2. It's generative process now assumes that news documents are created by the following steps:

1. For each of the M documents d_i, choose $\theta \sim Dir(\alpha)$
2. For each of the N_t topic word positions t_n
   (a) Choose a topic $z_n \sim Multinomial(\theta)$
   (b) Choose a topical word t_n from $p(T|z_n, \beta)$
3. For each of the N_e named entity positions e_n
   (a) Choose a topic $z_n \sim Multinomial(\theta)$
   (b) Choose a named entity e_n from $p(E|z_n, \nu)$,

where N_t is the number of topic words in a document, and N_e is the number of entities ($N = N_t + N_e$). β and ν are respectively the word topics and entity topics (but will not play a further role, as θ is the only distribution we are interested in).

This method, when trained on a corpus of news documents, learns what each document's topics are as the hidden parameter z, and assigns topical words (sampled from the topic vocabulary T) and entities (sampled from the named entity vocabulary E) to each topic. Comparing whether two documents cover the same event now merely requires comparing both topic-distributions z, instead of the topic distributions and the named entity distributions.

This model is a variation of work presented in Li et al. (2005). In their model, besides topic words and entities, timestamps are also sampled conditioned on the event, which allows for annual returning events to be learned.

While a very elegant model (for example, the dependency of topic words and entities is now covered by their mutual dependency on the event), there are a few drawbacks.

1. The number of topics has to be set fixed before training. Changing that number again requires retraining. In news, it is very difficult to foresee the number of topics in advance. In the split models, discussed above, we do not encounter this

problem. Topic models that exclude named entities are stable when trained on a large reference corpus.
2. As the model is trained on the data set, it does not allow for other topics to be recognized because the entities and topic words change dynamically. Therefore, when new documents that possibly represent new events are added, this model requires complete retraining.
3. As mentioned in Sect. 3.2.3, a large and diverse dataset is necessary to learn meaningful word-distributions. If the dataset is too small, or the desired events are too specific, this method may fail to learn useful topic distributions.

Comparison between two documents in this representation is again done with the Kullback–Leibler metric.

3.4 Clustering

The document dissimilarity $dis(d_i, d_j)$, which is a fused dissimilarity in case documents are represented with different aspects, is used in a clustering algorithm. We used a hierarchical agglomerative clustering with complete linkage, as it is mentioned in the literature as one of the best performing document clustering algorithms (Voorhees 1986). The hierarchical clustering algorithm does not require the number of clusters to be chosen a priori, a very important property in our dynamic environment. We can use a fitness-condition on the clustering to create a natural, unsupervised stopping criterion. This *natural* clustering is the most logical extension of our unsupervised approach: the data provides the number of clusters itself.

The clustering's fitness is calculated using the "*silhouette*" of the clustering, defined in Rousseeuw (1987) as follows. For every document d_i in our corpus, we calculate its fitness in cluster C_i as the normalized difference between the distance of d_i to the second best cluster C_j, and the average distance of d_i to the other documents in C_i:

$$f(d_i) = \frac{b(d_i) - a(d_i)}{max\,\{a(d_i), b(d_i)\}}$$

where

$$a(d_i) = \frac{1}{|C_i| - 1} \sum_{d_j \in C_i} dis(d_i, d_j)$$

and

$$b(d_i) = \arg\min_{C_j} \frac{1}{|C_j|} \sum_{d_j \in C_j} dis(d_i, d_j).$$

If C_i is a singleton cluster (containing only d_i), we assign $f(d_i)$ the default value 0. We search for the clustering that maximizes the average of f over all documents, over all possible stops in the hierarchy. This will be our *natural* clustering.

As a comparison measure, we also search manually for the threshold value that, when used as a stop criterion, yields the clustering with maximum performance (in F1 measure, see Sect. 4.2). This supervised evaluation is presented as our *maximum* clustering.

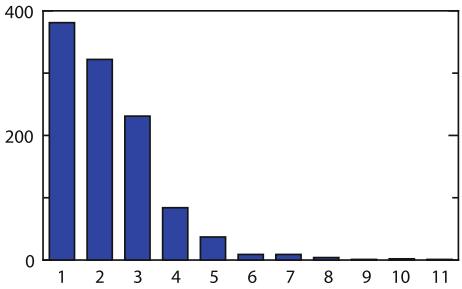

Fig. 3 Numbers of stories plotted by number of covering documents

4 Evaluation

We will first give details on the datasets used in the evaluation of the event clustering. Then follows a short section on our clustering algorithms and cluster evaluation techniques. After that, we present the results and discussions.

4.1 Datasets

For our evaluation, we used two different datasets:

Wikinews To test our different techniques for event clustering, we need a corpus for which we know of every document which event it covers, and to which other documents it relates. We considered using the TDT4 corpus. The large number of documents (28,500) is a positive point; however, only 160 separate events, and therefore a small portion of all documents, are annotated. This makes a correct computation of precision and recall impossible. Therefore we created our own ground truth corpus[4] from Wikinews that is used in the evaluation of the clustering.

On this news website, every reported event comes with several links to sources from different news-providers, thus providing a set of documents which cover the same event. We collected 2,428 documents, covering 1,081 separate events. Each event is covered by an average of 2.2 documents, with the number of covering stories for each event ranging from 1 to 11.

The distribution of the number of events over the number of covering documents is shown in Fig. 3.

TREC The training set for the LDA topic model, which we will infer on our news documents, needs to cover a wide range of topics in order to have clean word distributions. From the Text Retrieval Conferences' TREC Vol. 5, we randomly selected over 30,000 documents out of the LA Times corpus, reporting events from areas as different as the political, financial or scientific world, the world of media and entertainment, etc. After removal of stop words, low-idf words ($idf \leq 2.0$) and named entities, we end up with a word-list of 17,000 elements.

[4] This corpus will be available on request.

4.2 Evaluation metrics

Each dissimilarity function described in our methodology is used in two different hierarchical agglomerative clustering applications: *natural* clustering (which uses the silhouette fitness function to determine the "ideal" number of topics) and *maximum* clustering. Evaluation of these clusterings is done using the B-Cubed metric (Bagga and Baldwin 1998). Let C_i be the symbol for the cluster that document d_i gets clustered in, and M_i be its manual cluster (i.e. from the ground truth). The B-Cubed metric then calculates for each document its precision (how many of the other documents in its automatic cluster should be there?) as $\frac{|C_i \cap M_i|}{|C_i|}$, and its recall (how many of the documents in its manual cluster are in its automatic cluster?) as $\frac{|C_i \cap M_i|}{|M_i|}$. The total clustering's precision and recall are taken as the average over all documents.

We have to use a metric for cluster evaluation, which is capable of working with a different number of detected events as in the ground truth. Since we don't know how many events would be present, we cannot use a classification evaluation metric.

Our main remark on the B-Cubed metric is the fact that it rewards a singleton clustering (each document in its own cluster) with a precision of 100 %, as no document is clustered together with an unrelated one. Of course, recall will be very low in that case. Therefore we urge the reader to focus on the F1 values, as this gives a clear view on both precision and recall. The *maximum* clustering results shown in the tables are the ones that maximize the F1 measure.

4.3 NER

To estimate the influence of the NER, we have manually evaluated performance of NER on a small validation set and found that performance was satisfying: we obtained a precision of 93.37 % and a recall of 97.69 %. Precision is the percentage of identified person names by the system that corresponds to correct person names, and recall is the percentage of person names in the text that have been correctly identified by the system.

4.4 Results of event detection in Wikinews: natural vs. maximum clustering

This section contains the results of our tests. They are gathered per type of fusion. First we show the early fusion models. These contain the vector space model, the LSA model, and the LDA models (both trained from the full text, as well as inferred from the TREC model). Next we show the two-split between words and entities. In this respect we discuss both the late fusion combination models as the early fusion topic model. And last, we also have the results for the four-split of words, persons, locations and organizations.

The following abbreviations are used in the evaluation tables:

Full	All terms in the document
Wor	The topical words in the document (i.e. not including the named entities)

Ent	All named entities
Per	Only the persons
Loc	Only the locations
Org	Only the organizations
S	Number of topics LDA or intermediate model; rank of LSA
Top	Topics trained or inferred using topic models (default: $S = 100$)
max	Maximum over all used aspects
ave	Average over all used aspects
P	Precision
R	Recall
F1	F1 measure

Depending on the type of experiment, these features are used in different distance metrics: the cosine distance for all vector space term models and LSA models, and Kullback–Leibler in the case of the probabilistic topic models.

In every table, we highlighted the model with the highest F1 performance, both for the natural as for the maximum clustering. We also give the number of clusters at which this performance was measured.

4.4.1 Early fusion models

Table 1 shows the results when we used the full text as input for the different models. Surprisingly, the basic vector space term model (73.3 %) outperforms the other, topic-based models (both probabilistic and algebraic). The most advanced models (based on LDA) perform worst, the inferred model doing slightly better than the trained model.

The vector space model using the full text provides the most fine-grained representation of the text. This is apparently advantageous in a news clustering task as opposed to a broader topic model representation.

In the next section, we will investigate the role of entities, and how the information gained from them can best be incorporated in the document representation.

Table 1 Results of the early fusion text models

	Natural clustering				Maximum result			
	P (%)	R (%)	F1 (%)	# Events	P (%)	R (%)	F1 (%)	# Events
Vector space	72.6	74.0	**73.3**	1,086	74.8	72.9	**73.8**	1,148
LSA	99.7	44.5	61.6	2,421	74.1	61.3	67.1	1,796
LDA trained	25.8	80.7	39.2	296	83.2	54.3	65.7	1,791
LDA inferred	54.9	49.7	52.1	1,196	100.0	44.5	61.6	2,428

Bold values are the highest F1 measures

4.4.2 *Late fusion two-split models*

Tables 2, 3, 4, 5, and 6 show the results when we first split each document into its topical words and named entities, and input them separately into each model.

Table 2 evaluates the vector space model. Unlike the other models (as the other tables will show), the vector space model doesn't benefit from the split. Whereas the early fusion vector space model achieves 73.3 %, the best result achieved by the two-split vector model is 71.1 % using the *max*-combination function. Most information here comes from the topical words (70.2 %).

In Table 3, the performance of the split LSA model is presented. Unlike the other two-split models, the *max*-function does not improve on the aspects: it relies only on the information coming from the topics. The model also fails to improve on the early fusion LSA, which reached 61.6 %. Its natural number of clusters (2,316) however is

Table 2 Results of the two-split for the vector space model

	Natural clustering				Maximum result			
	P (%)	R (%)	F1 (%)	# Events	P (%)	R (%)	F1 (%)	# Events
Wor	69.9	70.5	70.2	1,096	77.0	66.9	71.6	1,301
Ent	61.4	63.7	62.5	1,096	78.8	56.0	65.5	1,620
max	72.0	70.2	**71.1**	1,151	76.7	67.7	**71.9**	1,287
ave	67.6	71.2	69.3	1,051	78.0	65.2	71.0	1,369

Bold values are the highest F1 measures

Table 3 Results of the two-split LSA model

	Natural clustering				Maximum result			
	P (%)	R (%)	F1 (%)	# Events	P (%)	R (%)	F1 (%)	# Events
Top	95.4	44.5	**60.7**	2,316	100.0	44.5	61.6	2,428
Ent	31.3	74.1	44.0	411	78.1	61.3	68.7	1,450
ave	35.9	71.9	47.9	505	77.5	59.0	67.0	1,486
max	95.3	44.5	60.6	2,314	100.0	44.5	61.6	2,428

Bold value is the highest F1 measure

Table 4 Results of the two-split for the trained LDA model

	Natural clustering				Maximum result			
	P (%)	R (%)	F1 (%)	# Events	P (%)	R (%)	F1 (%)	# Events
Top	9.5	79.3	17.0	101	78.5	52.6	63.0	1,762
Ent	31.3	74.1	44.0	411	78.1	61.3	68.7	1,450
max	68.1	68.1	68.1	1,106	76.3	64.0	69.6	1,357
ave	69.5	71.1	**70.3**	1,101	78.1	66.2	**71.7**	1,358

Bold values are the highest F1 measures

Table 5 Results of the two-split for the inferred LDA model

	Natural clustering				Maximum result			
	P (%)	R (%)	F1 (%)	# Events	P (%)	R (%)	F1 (%)	# Events
Top	54.9	49.7	52.1	1, 196	100.0	44.5	61.6	2,428
Ent	31.3	74.1	44.0	411	78.1	61.3	68.7	1,450
max	71.2	65.1	68.0	1, 221	74.8	63.4	68.6	1,329
ave	71.4	67.5	**69.4**	1, 771	76.9	64.1	**69.9**	1,357

Bold values are the highest F1 measures

Table 6 Results of the two-split intermediate fusion topic model, for different values of S

	S	Natural clustering				Maximum result			
		P (%)	R (%)	F1 (%)	# Events	P (%)	R (%)	F1 (%)	# Events
Intermediate	100	52.6	44.6	**48.3**	1,276	100.0	44.5	61.6	2,428
Fusion topic	300	19.5	78.9	31.3	226	78.8	54.5	64.4	1,718
Model	500	16.4	77.3	27.0	181	77.8	53.1	**63.1**	1,722
	700	12.0	75.8	20.8	131	92.7	47.0	62.4	2,182

Bold values are the highest F1 measures

far from the real number (1,081), indicating that the topic distribution learned by LSA do not provide reliable information for the clustering (in a natural clustering setting).

Tables 4 and 5 show the results when the topics and the named entities are represented by probabilistic distributions. In the former table the feature of the topical words is created by training LDA on the test corpus itself (using only the topical words). In the latter case the probabilistic topic model is trained on an external large corpus (here the TREC corpus), which has the advantage that the topic model does not need to be retrained when new documents are available for which the events need to be detected.

In contrast to the results of the vector space model and the LSA model, the natural clustering's performance in terms of the B-Cubed F1 measure is improved, over using the full document, by considering the different aspects of the news document and combining them. The feature of the probabilistic topic model Top is the document's topic distribution, inferred from one of the LDA models, covering 100 topics. Using the combination of the document's topic model with the named entity model boosts the recognition of the events up to 69.4 %, higher than both aspects separately. When the probabilistic topic model is trained on the test set, this goes up to 70.3 %. In both cases it is also higher than the early fusion LDA models, which gave 52.1 and 39.2 %.

We assume that the added entity aspect provides relevant, fine-grained information needed in the event clustering task that topic models lack.

Figure 4 plots the results for an inferred LDA topic model when a different number than 100 topics is used. The maximum F1 score for the combination functions is quite insensitive to the chosen number of topics. This result makes it easy to rely on LDA: there is freedom to select a number without worrying about negative influence on the performance.

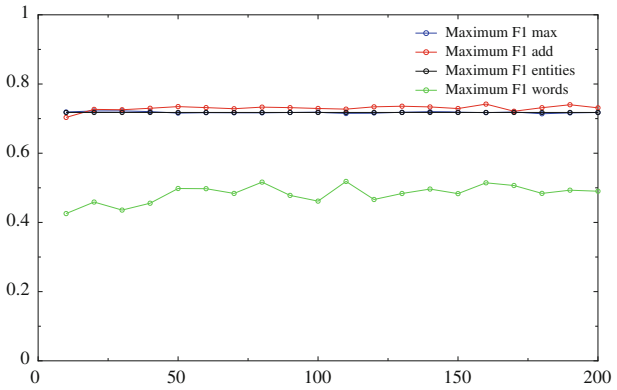

Fig. 4 Maximum F1 score (vertical) for the two-way split for the inferred probability model for different numbers of topics (horizontal)

The results of the last two-split model we tested, the intermediate fusion topic model, are shown in Table 6. Instead of combining topics learned from words and adding the entity information afterwards, this model incorporates the aspects into the training, but giving them more importance than a full text topic model by considering their distributions separately. For a natural clustering this improves slightly the performance (from 39.2 to 48.3 % for 100 topics), although it's still not reliable enough to be used effectively. The maximum clustering does not benefit from the split either. When compared with Table 4, it is clear that a late fusion topic model is a better choice than an intermediate fusion model.

Although the results of similar experiments presented in Li et al. (2005) gave good results, the method's performance in our case compares unfavorably. An important cause for this behavior appears to be the dataset, and its high number of events and a relatively small number of documents per event. The data sets used by Li et al. (2005) regard the TDT4 corpus where eighty important events are annotated in the 28,500 documents and a collection of CNN, MSNBC and BBC news stories, where again only a very limited number of major events were selected. Through this selection in both cases the average document per event ratio is artificially increased. Our dataset with a much lower average document per event ratio, however, more reliably reflects real news streams. The sparseness of data per event makes the split event model sensitive to its random initialization, so that it converges to an incorrect solution. The late fusion model is more robust to this problem, as can be seen by the previous results.

4.4.3 Four-split models

For our final evaluation of the influence of entities, we split up the news stories further. This time we take four aspects into account, by subdividing the aspect "named entities" in "persons", "locations" and "organizations".

Although splitting a document into topical words and named entities improves the natural clustering, further splitting of the named entities into persons, locations and organizations decreases the quality of the natural clustering for the *max*-function

Table 7 Four-split for the vector space model

	Natural clustering				Maximum result			
	P (%)	R (%)	F1 (%)	# Events	P (%)	R (%)	F1 (%)	# Events
Wor	69.9	70.5	**70.2**	1,096	77.0	66.9	**71.6**	1,301
Per	68.1	60.8	64.3	1,281	81.4	55.6	66.1	1,687
Loc	29.4	64.5	40.4	566	98.5	45.1	61.9	2,375
Org	47.9	55.4	51.4	981	89.4	47.4	61.9	2,098
max	90.3	51.2	65.3	1,976	90.3	51.2	65.4	1,973
ave	76.5	63.8	69.6	1,356	77.1	63.8	69.8	1,369

Bold values are the highest F1 measures

Table 8 Four-split for the trained LDA model

	Natural clustering				Maximum result			
	P (%)	R (%)	F1 (%)	# Events	P (%)	R (%)	F1 (%)	# Events
Top	9.5	79.3	17.0	101	78.5	52.6	63.0	1,762
Per	31.9	71.0	44.0	466	78.2	56.4	**65.5**	1,572
Loc	78.7	47.5	59.3	1,866	97.4	45.3	61.8	2,353
Org	83.0	46.7	59.7	1,981	99.2	45.3	61.6	2,406
max	84.8	53.0	**65.3**	1,786	84.5	53.2	65.3	1,776
ave	78.9	54.6	64.6	1,611	68.6	64.0	66.2	1,185

Bold values are the highest F1 measures

Table 9 Four-split for the inferred LDA model

	Natural clustering				Maximum result			
	P (%)	R (%)	F1 (%)	# Events	P (%)	R (%)	F1 (%)	# Events
Top	54.9	49.7	52.1	1,196	100.0	44.5	61.6	2,428
Per	31.9	71.0	44.0	466	78.2	56.4	**65.5**	1,572
Loc	78.7	47.5	59.3	1,866	97.4	45.3	61.8	2,353
Org	83.0	46.7	59.7	1,981	99.2	45.3	61.6	2,406
max	86.9	51.3	**64.5**	1,881	84.3	53.1	65.1	1,769
ave	79.4	53.9	64.2	1,641	70.9	60.4	62.2	1,293

Bold values are the highest F1 measures

(65.3 %) in the vector space term model (see Table 7), although it increases slightly for when using the *ave*-function (69.6 %). In the case of the LDA models (both trained and inferred), the decrease in performance applies on both the combination functions (Tables 8, 9).

To understand this behavior for different features and combination functions, we plotted in Fig. 5 the aspects' or combined dissimilarities for a sample of the test set. The documents are ordered per event, so in the ideal case we should see a block diag-

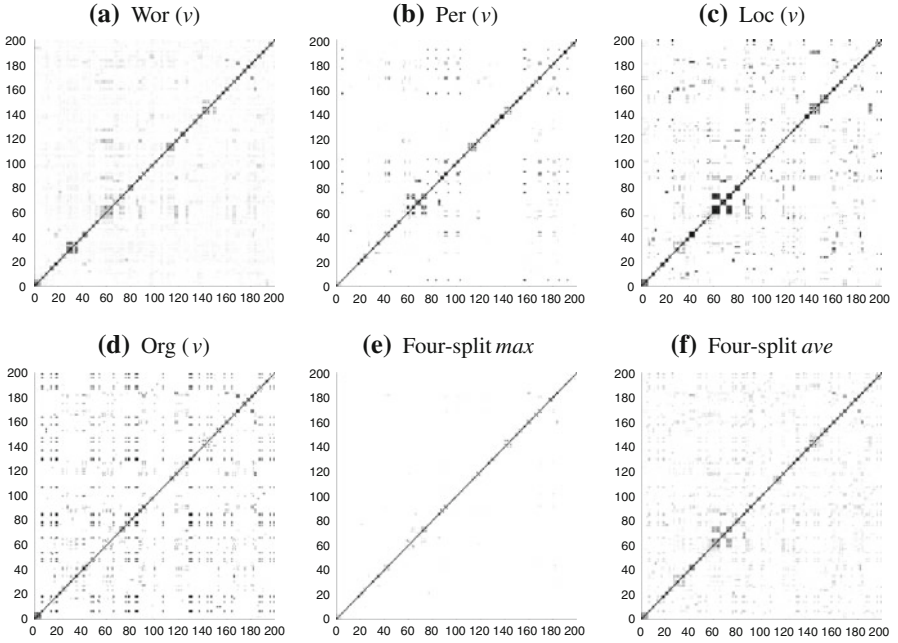

Fig. 5 Plots of aspects' distances and their combinations in a four-split model

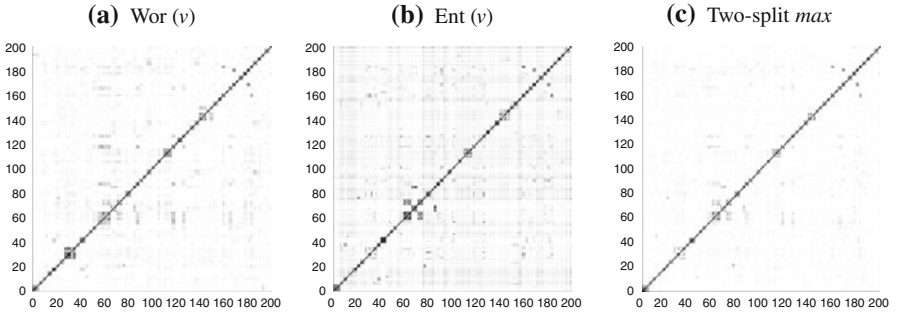

Fig. 6 Plots of aspects' distances and their combination in a two-split model

onal matrix, one block per event. For the *words*-feature (Fig. 4a), we have a good approximation. For *persons*, *locations* and *organizations* (Fig. 4b–d) however, we get increasingly more noise in other parts of the matrices. This is caused by stories that have no named entities from one of the named entity classes, which results in a sparse dataset. The effect is that the fitness function will yield a too *high* dissimilarity threshold, causing many unrelated stories to be clustered together.

When we look at the plot for the *max*-function (Fig. 4e), we see that we deal with the same problem, although its cause is now the opposite. No noise appears in the upper and lower triangle area, but we also lost the blocks around the diagonal. Figure 4f shows the effect the *ave*-function has: it is clear that it is easier to distinguish correct clusters in this setting.

As a comparison, we also included the same plot for the two-split in Fig. 6. Comparing Fig. 4e with Fig. 5c clearly shows that the two-split has cleaner, easier recognizable clusters.

Since the sparseness problem appeared in the tested models so far, we did not test it on the LSA or intermediate fusion topic models, they would suffer equally from it.

4.4.4 Real clustering

If the correct number of events should be known a priori, it makes sense to cluster the data into that number of clusters, stopping the hierarchical clustering method when reached this number (here 1,081 clusters). Table 10 shows the performance of each

Table 10 Results of clustering with the correct number of events for all used representations

	P (%)	R (%)	F1 (%)
Full text vector model	72.5	74.1	**73.3**
Full text LSA model	44.7	78.8	57.0
Full text LDA model	59.9	67.0	63.3
Two-split vector model			
Wor	69.3	70.5	69.9
Ent	60.9	64.0	62.5
ave	68.5	70.3	69.4
max	69.5	71.3	**70.4**
Two-split LSA models			
Top	44.6	78.8	57.0
Ent	65.3	66.9	66.1
ave	64.2	65.5	64.8
max	65.3	67.0	**66.1**
Two-split early fusion topic model	56.7	64.2	60.3
Two-split early fusion topic model ($S = 1{,}081$)	50.3	55.4	52.7
Two-split inferred LDA model			
Top	50.3	50.5	50.5
Ent	65.3	66.9	66.1
ave	68.0	68.9	**68.5**
max	66.2	67.5	66.9
Two-split trained LDA model			
Top	54.0	62.7	58.0
Ent	65.3	66.9	66.1
ave	68.8	71.4	**70.1**
max	67.3	68.6	67.9
Four-split vector model			
Wor	69.3	70.5	**70.0**
Per	60.6	63.3	62.0
Loc	52.7	56.9	54.8

Table 10 continued

	P (%)	R (%)	F1 (%)
Org	52.1	54.6	53.3
ave	66.3	67.6	67.0
max	50.6	69.4	58.6
Four-split trained LDA model			
Top	54.0	62.7	58.0
Per	61.9	63.1	62.5
Loc	53.3	55.2	54.3
Org	52.8	54.3	53.6
ave	64.9	65.5	**65.2**
max	59.3	58.5	58.9
Four-split inferred LDA model			
Top	50.3	50.5	50.5
Per	61.9	63.1	62.5
Loc	53.3	55.2	54.3
Org	52.8	54.3	53.6
add	63.8	63.2	**63.6**
max	59.0	58.1	58.6

Bold values are the highest F1 measures

model in this case. Again, the full text vector space term model outperforms all others, followed by the two-split vector space model, the two-split trained LDA model and the four-split vector space model. We also report the results of the two-split intermediate fusion topic model when trained with 1,081 topics, to investigate if our intermediate model could learn the events themselves as topics. This setup however yields a very low result. Topic models are not capable of learning such fine-grained distinctions, especially if the training-set is relatively small (only 2.2 documents per topic).

5 Conclusions

In this paper we have performed a benchmarking study of several unsupervised methods in order to detect similarities and differences between news documents. Our methods were evaluated in the setting of news event clustering. We performed an extensive study to investigate the influence of representation models, including algebraic vector space models (among which is LSA) and probabilistic models (mainly based on LDA). In addition, we evaluated the influence of content splitting into aspects and compared models of late and intermediate fusion of evidence against "full text" or early fusion models where all content words are jointly considered in a similarity function. Late fusion models allow aspects to better control the contribution of the different aspects in the similarity computations.

The benchmarking was realized by means of an extensive evaluation of event detection on 2,400 Wikinews documents. Our most surprising result is that the vector space

model, with its simple techniques of $tf \times idf$-weighting and cosine metric for distance, still outperforms the more advanced LSA and LDA techniques. It also does this while only using a fraction of the computation time required to train LSA or LDA. The use of synonyms, one of the main problems of the vector space model that LSA and LDA aim to resolve, turns out to be not as much of an issue as would be expected in a setting of news texts coming from different sources. This may be caused by news agencies, which cause the different related articles to be based on the same text, only to be rewritten slightly.

This lack of synonyms does not explain the low results of LDA topics when grouping news into events both in a trained and in an inferred LDA setting. The probabilistic, high-level representation of the news articles in term of topics does not provide the necessary fine-grained view of a document's content to distinguish between identical, similar and unrelated news stories. Especially the topics created from the inferred LDA, which is a better option compared to the trained LDA from a computational point of view, are much too coarse as clustering features. The trained LDA performs slightly better as it can learn topics specific to the news stories themselves, but has the disadvantage that it needs to be recalculated for each dataset.

The probabilistic topics do, however, provide enough clues to aid the story identification based on named entity comparison, in the late fusion scheme. Again the trained LDA is better than the inferred, although the difference between the two becomes smaller, and may be worth the trade-off. In this setting, they both perform considerably better than LSA, which does better when taking only the topics into account, but has less improvement when entities are added in the late fusion.

We conclude that, despite the advance in complexity (and associated increase in computation time) of representation models for text, it is not yet time to discard the simple, tried and true techniques especially when the similarity depends on the correspondences in fine-grained content such as names or specific terminology used. $Tf \times idf$ is still valid as a weighting scheme for content representation and many orders of magnitudes faster. Probabilistic topic models, although being heralded for their representative power and the intrinsic quality of the learned topics (usually stated through measurements as perplexity and the human evaluation of said topics) do not in practice have the same fine-grained representative power as a simple vector space model has.

Every document contains its own information, easily divisible into topical words and named entities (and other types of documents may have different divisions). Although probabilistic models have been adapted gracefully to take complex dependencies into account (such as link structure, temporal ordering, etc.), if such information is not available then the old techniques should still be considered a valid alternative. However, the generative topic models have the potential to generate synonym words in specific contexts, as for instance is already done by the Latent Word Language Model, which generates replacement words at a certain position in a sentence (Deschacht and Moens 2012). Such an approach could be useful in fine-grained comparisons.

This work has contributed to the study of text similarity metrics for one specific task, i.e. event clustering. This work could be expanded to a study of such metrics in different text comparison tasks, in order to draft guidelines for using certain methods given the requirements of the task and the properties of the dataset.

References

Allan J, Lavrenko V, Swan R (2002) Explorations within topic tracking and detection. Kluwer, Norwell, ir 20, pp 197–224

Allan J, Wade C, Bolivar A (2003) Retrieval and novelty detection at the sentence level. In: SIGIR '03, ACM, New York, pp 314–321

Bagga A, Baldwin B (1998) Algorithms for scoring coreference chains. In: The first international conference on language resources and evaluation workshop on linguistics coreference, Granada, pp 563–566

Barzilay R, Lee L (2003) Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In: HLT-NAACL '03: main proceedings, Edmonton, pp 16–23

Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3:993–1022

Buntine W, Jakulin A (2006) Discrete component analysis. In: Saunders C, Grobelnik M, Gunn S, Shawe-Taylor J (eds) Subspace, latent structure and feature selection techniques. Springer, Heidelberg, pp 237–247

Cutting DR, Pedersen JO, Karger D, Tukey JW (1992) Scatter/gather: a cluster-based approach to browsing large document collections. In: SIGIR '92, Seattle, pp 318–329

de Marneffe MC, Rafferty AN, Manning CD (2008) Finding contradictions in text. In: ACL'08: HLT, Association for Computational Linguistics, Columbus, pp 1039–1047

Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Inf Sci 41:391–407

Deschacht K, De Belder J, Moens MF (2012) The latent words language model. Comput Speech Lang 26(5):384–409

Gong Y, Liu X (2001) Generic text summarization using relevance measure and latent semantic analysis. In: SIGIR '01, ACM, New York, pp 19–25

Griffiths T, Steyvers M, Tenenbaum J (2007) Topics in semantic representation. Psychol Rev 114(2): 211–244

Hatzivassiloglou V (1998) Automatic acquisition of lexical semantic knowledge from large corpora: the identification of semantically related words, markedness, polarity, and antonymy. PhD thesis, New York

Hershkop S, Stolfo SJ (2005) Combining email models for false positive reduction. In: KDD '05, ACM, New York, pp 98–107

Hofmann T (1999) Probabilistic latent semantic analysis. In: Proceedings of uncertainty in artificial intelligence, Stockholm

Kumaran G, Allan J (2004) Text classification and named entities for new event detection. In: SIGIR '04, ACM, New York, pp 297–304

Lee MD, Welsh M (2005) An empirical evaluation of models of text document similarity. In: CogSci2005, Erlbaum, pp 1254–1259

Li W, Mccallum A (2006) Pachinko allocation: DAG-structured mixture models of topic correlations. In: ICML '06, ACM, New York, pp 577–584

Li Z, Wang B, Li M, Ma WY (2005) A probabilistic model for retrospective news event detection. In: SIGIR '05, ACM, New York, pp 106–113

Makkonen U, Ahonen-Myka H, Marko (2002) Applying semantic classes in event detection and tracking. In: Proceedings of the International Conference on Natural Language Processing (ICON'02), Bombay, pp 175–183

Mccallum A, Corrada-Emmanuel A, Wang X (2005) Topic and role discovery in social networks. In: Proceedings of the 19th international joint conference on artificial intelligence, Edinburgh, pp 786–791

Mckeown K, Radev DR (1995) Generating summaries of multiple news articles. In: SIGIR '95, Seattle, pp 74–82

Nallapati R, Feng A, Peng F, Allan J (2004) Event threading within news topics. In: CIKM '04, Washington, pp 446–453

Pearl J (1991) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Mateo

Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65

Salton G (1989) Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley, Boston

Shafer G (1976) A mathematical theory of evidence. Princeton University Press, Princeton
Snoek CGM (2005) Early versus late fusion in semantic video analysis. In: ACM multimedia, New York, pp 399–402
Steinberger J, Ježek K (2009) Update summarization based on novel topic distribution. In: DocEng'09, ACM, New York, pp 205–213
Steinberger J, Poesio M, Kabadjov MA, Jeek K (2007) Two uses of anaphora resolution in summarization. Inf Process Manag 43(6):1663–1680
Steinberger J, Turchi M, Kabadjov M, Steinberger R, Cristianini N (2010) Wrapping up a summary: from representation to generation. In: Proceedings of the ACL 2010 conference short papers, Association for Computational Linguistics, Uppsala, pp 382–386. http://www.aclweb.org/anthology/P10-2070
Stone B, Dennis S, Kwantes PJ (2011) Comparing methods for single paragraph similarity analysis. Top Cogn Sci 3(1):92–122. doi:10.1111/j.1756-8765.2010.01108.x
Tsatsaronis G, Varlamis I, Vazirgiannis M (2010) Text relatedness based on a word thesaurus. J Artif Intell Res 37:1–39
Voorhees EM (1986) Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. Technical Report, Ithaca
Wang ZW, Wong SKM, Yao YY (1992) An analysis of vector space models based on computational geometry. In: SIGIR '92, ACM, New York, pp 152–160
Wang K, Li X, Gao J (2010) Multi-style language model for web scale information retrieval. In: SIGIR '10, ACM, New York, pp 467–474
Yang Y, Carbonell JG, Brown RD, Pierce T, Archibald BT, Liu X (1999) Learning approaches for detecting and tracking news events. IEEE Intell Syst 14(4):32–43
Zhang K, Zi J, Wu LG (2007) New event detection based on indexing-tree and named entity. In: SIGIR '07, ACM, New York, pp 215–222