From Names to Entities using Thematic Context Distance

Anja Pilz
Fraunhofer IAIS
Schloss Birlinghoven
Sankt Augustin, Germany
anja.pilz@iais.fraunhofer.de

Gerhard Paaß
Fraunhofer IAIS
Schloss Birlinghoven
Sankt Augustin, Germany
gerhard.paass@iais.fraunhofer.de

ABSTRACT

Name ambiguity arises from the polysemy of names and causes uncertainty about the true identity of entities referenced in unstructured text. This is a major problem in areas like information retrieval or knowledge management, for example when searching for a specific entity or updating an existing knowledge base.

We approach this problem of named entity disambiguation (NED) using thematic information derived from Latent Dirichlet Allocation (LDA) to compare the entity mention's context with candidate entities in Wikipedia represented by their respective articles. We evaluate various distances over topic distributions in a supervised classification setting to find the best suited candidate entity, which is either covered in Wikipedia or unknown. We compare our approach to a state of the art method and show that it achieves significantly better results in predictive performance, regarding both entities covered in Wikipedia as well as uncovered entities.

We show that our approach is in general language independent as we obtain equally good results for named entity disambiguation using the English, the German and the French Wikipedia.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Text analysis; I.5.2 [**Design Methodology**]: Classifier design and evaluation, Feature evaluation and selection; H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation

Keywords

named entities, named entity disambiguation, topic modeling, named entity resolution, classification

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

1. INTRODUCTION

Recently several analysts have estimated that more than 70%-80% of all data in organizations is unstructured information [24]. The enrichment of knowledge bases from unstructured text can hence gain from reliable information extraction methods. Among those are text mining techniques such as entity recognition, relation extraction or entity disambiguation.

Named entity recognition identifies name phrases in a text as named entities by labeling the words with type labels such as person or location. Subsequently *named entity disambiguation* (NED) or *name disambiguation* aims to assign a *name mention* in a text to the underlying *real-world entity*, e.g. to a unique entity description. This is a critical step in the construction of semantic knowledge bases from unstructured text allowing entity-based retrieval instead of keyword search.

It is well known that names are not unique. The name mention *John Taylor*, for example, can denote a South Carolina governor, an athlete, a racing driver, a jazz musician, a bass guitarist and so on. Note that the last two entities also have a similar artistic profession (both are musicians) which makes their distinction even more difficult. Additionally, polysemy of names spans across entity types. The term *Bush*, for example, may be used to refer to a large number of persons, a shrub, to undeveloped landscape, to a number of locations, brands, as well as to two rock bands.

The effects resulting from name ambiguity can easily be seen when carrying out web searches or retrieving articles from an archive of newspaper texts. For example, the top ten hits of a Google search for the string *John Taylor* contain ten different entities, among those a professor, a jazz guitarist, a college and so on. While it may be clear to a human that these mentions do not refer to the same real-world entity, it is difficult to draw this conclusion automatically, using natural language processing techniques.

Most approaches to name disambiguation estimate the identity of a name mention by comparing the words in the neighborhood of the mention with the words in the descriptions of real-world entities. This reflects the observation of [17] that words with similar meanings are often used in similar contexts.

While the comparison of (bag-of-word-) word vectors for NED is quite successful [1, 8, 5, 7] it is not robust if different words with similar meaning are used in the neighborhood of a name mention and the corresponding descriptions of the real-world entity. Figure 1 depicts examples for some contexts mentioning the name *John Taylor*. Consider for exam-

T1 (John Taylor (bass guitarist)):
*Duran Duran was chosen to do the song after bassist **John Taylor** (a lifelong Bond fan) approached producer Cubby Broccoli at a party, and somewhat drunkenly asked "When are you going to get someone decent to do one of your theme songs?"*

T2 (John Taylor (poet)):
*Thames watermen played an important part in the very early movements that ultimately led to the creation of modern trade unionism in the United Kingdom, most notably in the writings of pamphleteer **John Taylor** (1580-1653) and later with the use of petitions or "petitions of grievances" in particular the petitions supporting the curtailment of the growth of hackney coaches in the 17th century.*

T3 (John Taylor (poet)):
*In 1630, **John Taylor**, a poet wrote "At Bow, the Thursday after Pentecost, There is a fair of green geese ready rost, Where, as a goose is ever dog cheap there, The sauce is over somewhat sharp and deare".*

T4 (Elizabeth Taylor):
*Critical reaction to the film was mostly negative, with much being made of **Taylor**'s wildly fluctuating weight from scene to scene.*

Figure 1: Context examples for the name *Taylor* referring to three different entities (extracted from Wikipedia).

ple T2 and T3: while the thematic information is similar, the overlap of common terms is very low, even though both contexts refer to the same person. Additionally, approaches that are based on word comparisons are also often unable to reason on previously unseen attributes (e.g. new terms), and thus are likely to suffer from a lower recall for new data.

There have been several approaches to overcome the problem of synonymy. [19], for example, generate probable words according to a topic model of a document. In this paper, instead of representing topics by some words, we used topic model probabilities directly to describe the document content that a name mention appears in. This is motivated by the idea that topic models automatically disambiguate terms based on the co-occurrences with other terms. If LDA is trained on a sufficiently large corpus, most words of a new document will already be present in the LDA model. Thus, we assume that in a new document we can infer a topic probability distribution with sufficient quality which may be used to assess semantic similarity to Wikipedia articles.

The approach is able to assign name mentions in new documents to unseen entity descriptions (Wikipedia articles). The topic models may be applied to new documents as well as to new Wikipedia articles and yield useful semantic characterizations even if some words are not covered by the topic model. Hence it is not necessary to train the topic model with all Wikipedia articles as long as a sufficiently large sample is used.

As topic models do not require labeled training data, the approach presented in this paper can be applied to any language as long as unique descriptions of named entities are available (currently there exist Wikipedias in 269 languages). To demonstrate this, we apply our method to the English, the German, and the French version of Wikipedia, which are currently the three largest versions, and achieve quite similar performance figures.

In the next section we describe prior work in name disambiguation, which is also often referred to as entity resolution or in the field of databases as record linkage. Then we outline the properties of Wikipedia as a reference for unique named entities and the construction of a disambiguated dataset of corresponding entity mentions. Next, we outline the different topic distances used for thematic context comparison. Subsequently we describe the classification approach used in this paper. Then we present the experimental setup and discuss the obtained results.

2. RELATED WORK

2.1 Problem Description

In this work, we study the task of name disambiguation using Wikipedia as a knowledge base with describing contexts. Given a *name mention* m with surrounding text $T(m)$, we want to assign this mention to the corresponding true entity $e(m)$ described by a Wikipedia article. By matching the surface forms of names (e.g. first and last name) m we get a set of *candidate entities* or candidate articles $\mathcal{E}(m) = \{e_1, \ldots, e_{|\mathcal{E}|}\}$ from Wikipedia such that for all $e_j \in \mathcal{E}(m)$ we have $m = name(e_j)$ in the case of an exact name match or $m \subset name(e_j)$ in the case of a partial name match. We denote the Wikipedia text around entity e_j as $T(e_j)$. The task is to select one of the entities in $\mathcal{E}(m)$ as the correct entity ($e(m) = e_j$) or to decide that $e(m)$ is *not covered* by Wikipedia ($e(m) \notin \mathcal{E}(m)$).

2.2 Prior Work on Name Disambiguation

Name disambiguation is closely related to the task of word sense disambiguation, which aims at resolving the ambiguity of words in a text, as both rely on the assumption that the meaning of a mention is strongly dependent on the context it appears in [17]. However, studies in the fields of word sense disambiguation and entity resolution do often not assume the presence of a reference knowledge base and the applied algorithms are often unsupervised.

One of the first studies in the field of name disambiguation is [1]. The authors propose to create context vectors for each target name mention m, where each vector contains exactly the words that occur within a fixed-sized window around the ambiguous name. To cluster contexts referring to different entities, the authors measure the similarity among these vectors by the cosine measure. This algorithm was extended to a large corpus by [8].

[6] combined lexical context features and extracted information in a vector space model: non-stop words appearing within a fixed window around any mention m of the entity, noun phrases in the sentences containing the ambiguous name, and named entities (persons other than the ambiguous name, organizations, etc.) in the entire document. The authors reported better performance compared to previously published results.

The outputs of relation extractors or named entity recognizers are certainly useful features for name disambiguation, but unfortunately they cannot be used out of the box for other languages such as German or French. Creating such modules is often a laborious task, as it can require annotated training data or rely on language specific characteristics, such as capitalization and word order. For example, the German language has a free word order and a different capitalization scheme, thus, both entity recognizers as well

as relation extraction methods can be harder to construct. Since we wanted to apply our approach simultaneously to different languages without extensive manual annotation, we hence did not use these features.

To describe an entity [19] use a bag of features similar to [6]. Additionally, the authors employ entity profiles, which combine attributes of the entity (links from the entity to a value), relations (to or from another entity), and events that this entity is involved in. As a third feature the authors estimate a topic model with 50 topics and determined the top 10 words with highest probability for the document according to the topic model. Disambiguation is then performed via single-link hierarchical agglomerative clustering. Experiments show an improvement over the results of [6]. [19] generate words according to a topic model and then add these to the respective feature vector to overcome the synonymity problem. In contrast, our approach relies on the overall topic probability distribution of a document, thus using a completely different feature vector representation based on topic clusters instead of words.

Other approaches use relation evidence for entity resolution. [2] resolve ambiguity in the context of citations, considering the mutual relations between authors, paper titles, paper categories, and conference venues. For example, we may conclude that "R. Srikant" and "Ramakrishnan Srikant" designate the same author, since both are coauthors of another author. They argue that the joint resolution of the identity of authors, papers, etc. leads to a better result than considering each type alone.

Similarly, [9] disambiguate researcher names in citations by exploiting relational information contained in an ontology derived from the DBLP database. Attributes such as affiliations, topics of interests, or collaborators are extracted from the ontology and matched against the text surrounding a name occurrence. The results of the match are then combined in a linear scoring function that ranks all possible senses of that name. This scoring function is trained on a set of disambiguated name queries that are automatically extracted from Wikipedia articles. The method is also able to detect when a name denotes an entity that is not covered in Wikipedia.

Both of these approaches require relational information, for example provided by a relation extraction system or a well-maintained ontology. However this relational information is currently not available for arbitrary persons and in multiple languages. Therefore we did not include these features into our analysis.

[7] present a large-scale system for the recognition and semantic disambiguation of named entities based on information extracted from Wikipedia and Web search results. The system uses co-reference analysis to associate different surface forms of a name in a text, e.g. "George W. Bush" and "Bush". Again, context words are combined with Wikipedia categories to describe entities. The proposed method then assigns entities by maximizing the non-normalized scalar products for the contexts of entities and name phrases.

[5] disambiguate entities using a Ranking SVM [12], which generates a ranked list of plausible entities for a given context $T(m)$ of the target name mention m. Features are words in a window around m in combination with Wikipedia categories of the articles describing the corresponding real-world entities. To apply this scheme for uncategorized knowledge bases, one has to estimate classifiers to predict the Wikipedia categories for the text $T(m)$, which requires a considerable effort, especially if needed for several languages.

As we find the problem formulation in [5] most similar to ours, we re-implemented this approach (outlined in 4.2.2) and use it as a baseline for comparison with our approach. Similar to our work, the reference corpus is extracted from Wikipedia (for details, see 3.1).

3. WIKIPEDIA AS A KNOWLEDGE RESOURCE

In the last years, Wikipedia[1] has received much attention in the scientific community. Many projects and scientific studies build upon this free repository of world knowledge, which is constantly updated. Due to its sheer size, however, handling this structured knowledge resource for data mining or information retrieval purposes is a challenge.

Most of the textual content in Wikipedia is stored in articles. The English, German and French versions currently contain about 3.6 million, 1.2 million, and 1.0 million articles respectively. About 15% of its articles describe biographies and persons and 14% of the articles cover geography and places [13].

Each article in Wikipedia is uniquely identified by the most common name of the described subject. Ambiguous names are further qualified with additional terms such as affiliations or professions, e.g. *Jason Taylor (athlete)*, *Jason Taylor (jazz)*, or *Jason Taylor (racing driver)*. Articles hold information focused on one specific concept or entity, describing it in a concise but comprehensive way. The information is substantiated and enhanced by many links to other articles and citations using external documents. In the following we consider the article as the definition of the real-world entity it refers to.

Additionally, each Wikipedia article is labeled with one or more *categories*. These can represent content topics but also general attributes such as gender in the case of persons or the founding year in case of organizations. Aside from very general categories there exist very specific categories that apply to only very few articles. In our version of the English Wikipedia, which was downloaded on 15th January 2011, we found 600k different categories.

Relations between articles (entities) are expressed by links. When referring to an entity or concept with an existing article page, contributing authors are supposed to link at least the first mention of the related entity to its corresponding article. Inspired by [5] we exploit Wikipedia's internal link structure to generate an automatically disambiguated dataset of entity mentions, which is described in 3.1.

3.1 Generation of a disambiguated dataset from Wikipedia

To create a disambiguated training dataset, we assume the correctness of links in Wikipedia, and extract all articles referencing an entity of interest. Since the link provides the true entity, this results in a set of disambiguated entity mentions.

Note that this is not really a gold-standard dataset, as the linking is done by different Wikipedia contributors and thus depends on their individual taste. Sometimes links are rather conceptional links, i.e. link to a thematically related article, and do not mean identity. For example the term

[1] `http://www.wikipedia.org`

client can be linked to the article *Lawyer*. Unfortunately, we are not aware of other publicly available benchmark datasets (with links to Wikipedia), especially for German and French. Still, we assume that the model evaluated on this dataset can generalize to other corpora.

In the following, we use the definitions of section 2.1. Let $L(e_j)$ be the set of Wikipedia articles containing a mention m which is linked to a different article describing e_j. The articles $T(m)$ in $L(e_j)$ are also called *query documents*. Note that a pair $(T(m), T(e_j))$ with $T(m) \in L(e_j)$ can be used to train the relation between the entity e_j and its mention m. As we observed rather few articles without ingoing links we get positive examples for nearly all entities e_j in Wikipedia. Negative examples may be constructed by relating an entity e_j to a mention m not referring to it, i.e. $e(m) \neq e_j$.

Following [5] this set of disambiguated queries can be used for the training as well as the evaluation of a disambiguation model.

To construct such a disambiguated training data set, we first selected a set of entities e_j for which referencing documents (e.g. links in other articles) are to be found. Here, we consider persons and other entities with ambiguous names. As we did not focus on Wikipedia's category hierarchy, we use YAGO [20] to determine articles referencing persons. YAGO is build jointly over Wikipedia and WordNet and endows Wikipedia articles with attributes such as person, group, location, etc.. Each entity article, which is ambiguous, i.e. whose surface name corresponds to more than one Wikipedia article, is added to the entity dataset A.

Next we define a dataset Q containing the mentions corresponding to A. Note that there are articles with more than 1000 ingoing links while other articles have only one ingoing link. To balance this inequality we added at most 10 randomly selected referencing documents from $L(e_j)$ as examples to Q, also storing a reference to the the linked entity e_j. This restriction allows a better balanced model over all entities in A. Otherwise, we would run in danger to introduce a bias towards frequently referenced entities. We ignored references outgoing from documents in A, such that $Q \cap A = \emptyset$.

As the text $T(m)$ corresponding to a mention m we used the complete Wikipedia article containing m as this yields better results than just using a small window around m. We experimented with different context width configurations and found that an additional boost on the local context can improve disambiguation performance (see 5.3.1).

We apply the same configuration for English, German and French, remove stop words from respective lists and use the stemmed word forms obtained by the Snowball algorithm [18] (again for the respective language).

4. NAME DISAMBIGUATION AS A CLASSIFICATION PROBLEM

Ignoring the possibility of different spellings of the name of an entity, we consider an entity e_j only as a candidate for some name mention m, if its title matches the name mention in the query document. The assignment to one of the candidate entities in $\mathcal{E}(m) = \{e_1, \ldots, e_{|\mathcal{E}|}\}$ can be considered as a multi-class classification problem, with each e_j one distinctive class. Note that with increasing number of entities this classification problem rapidly becomes intractable, if we would want to learn one model per name (or per entity).

As an alternative, we can formulate this task as a binary classification problem, deciding if $e(m) = e$ or $e(m) \neq e$. Following [12], we create the classifier input using groups of feature vectors $\{\vec{\phi}(m, e_j)\}_{e_j \in \mathcal{E}(m)}$, where each $\vec{\phi}(m, e_j)$ is a feature vector describing the pair (m, e_j). To each $\vec{\phi}(m, e_j)$ we assign a label y

$$y(\vec{\phi}(m, e_j)) = \begin{cases} +1, & \text{if } e(m) = e_j \\ -1, & \text{else.} \end{cases} \quad (1)$$

For at most one e_j we get a positive label. In the special case that $e(m)$ is an uncovered entity and $e(m) \notin \mathcal{E}(m)$, none of the describing feature vectors $\vec{\phi}(m, e_j)$ receives a positive label.

This binary classification problem can be tackled with many different classifiers. We use the SVM^{Light}[2] implementation by Thorsten Joachims [11] as learning framework employing different kernels. This classifier has shown excellent performance in many text mining problems. It is trained with the feature vectors $\vec{\phi}(m, e_j)$ constructed from Wikipedia.

For a new mention $\tilde{m}$ we can construct new feature vectors $\vec{\phi}(\tilde{m}, e_j)$ for $e_j \in \mathcal{E}(\tilde{m})$. Subsequently, the estimated SVM is applied to all $\vec{\phi}(\tilde{m}, e_j)$. Then the entity which receives the highest score is selected as the best match. Note that it is possible that two entities come from a very similar field (i.e. two politicians from the same party). Then it is likely that both entities receive high scores for a given name mention m. The difference between the scores of the best and the second best match can be used as an indicator of the uncertainty of the decision.

4.1 Revealing uncovered Entities

As already discussed, there are millions of entities not present in Wikipedia, many of them appearing for example in the local parts of a news paper. To account for this, we simulate non-covered entities similar to [5]. In particular, this is done by removing the Wikipedia article and consequently all provided information for a fixed fraction of entities e_j. Consequently all mentions m linked to e_j now correspond to an entity not covered in Wikipedia and all training examples get assigned to class $y = -1$. During the application of the estimated SVM to new data, we consider a mention m as "uncovered", if the model predicts no score higher than the threshold 0 for any of the candidates $\vec{\phi}(\tilde{m}, e_j)$:

$$\forall_{e_j \in \mathcal{E}(\tilde{m})} \hat{y}(\vec{\phi}(\tilde{m}, e_j)) \leq 0. \quad (2)$$

Note that for future work, it would also be possible to fine tune this threshold on a validation set. Alternatively, [5] have shown that using a linear kernel, this threshold can be learned automatically from the weight of an indicative feature.

4.2 Baseline attributes for name disambiguation

We compare our proposed method to previous approaches to the task of name disambiguation and describe these attributes in the following.

[2]The SVM^{Light} package is available at `http://svmlight.joachims.org/`

4.2.1 Cosine Similarity

Cosine similarity was used by [1] as one of the first approaches to assess the similarity between two documents. [5] and [7] both evaluated experimentally a ranking function based on the cosine similarity between the text of the query document $T(m)$ and the text of the entity's article $T(e)$. Specifically, this denotes cosine similarity as

$$\phi_{cos} = \cos(T(m), T(e)) = \frac{V(T(m))}{||V(T(m))||} \frac{V(T(e))}{||V(T(e))||}. \quad (3)$$

Here $V(T(m))$ is the standard vector space model of $T(m)$, where each component corresponds to a term in the vocabulary and each entry to the word count in the respective context.

In the formulation in eq. 3, cosine similarity is an unweighted but normalized summation over common words (terms appearing both in the query document and the article text). The larger this number the more similar are $T(m)$ and $T(e)$ and hence the more similar are the entities denoted.

Measuring the similarity between contexts in this way has one major drawback: if alternative terms for one concept are used in $T(m)$ and $T(e)$, the similarity will be low even if the contexts denote the same entity. Additionally, entities in similar context will be difficult to distinguish using such an aggregated measure.

We use the feature ϕ_{cos} as a baseline feature in the feature vectors $\vec{\phi}(m, e_j)$ in all of the following experiments, as it evaluates directly matching words in the contexts of e_j and m.

4.2.2 Word-Category Pairs

[5] used the categories of Wikipedia articles as particularly indicative features, to learn the magnitude of semantic correlations between words and categories. In particular, [5] assume that common words are indicative, and create the following feature vector representation:

$$\phi_{wc}(m, e) = \begin{cases} 1, & \text{if } w \in T(m) \cap T(e) \text{ and } c \in c(e) \\ 0, & \text{else.} \end{cases} \quad (4)$$

In this approach, $T(m)$ is reduced to a context window of width 25 around the entity mention.

Here, the maximal dimension of $\vec{\phi_{wc}}$ is restricted by $|W| \times |C|$, where $|W|$ is the number of all possible words and $|C|$ the number of all possible categories.

We re-implement this approach using the Ranking SVM included in SVM^{Light} (and denote it as $Bun06$ in the following), with the slight modification, that we use only directly assigned categories, instead of analyzing the category hierarchy to extract top-level categories. In this paper we want to show, that the same approach is applicable for different languages and analyzing Wikipedia's category hierarchy is not a trivial task, as we can encounter loops and other pitfalls.

5. THEMATIC CONTEXT DISTANCE FOR NAME DISAMBIGUATION

5.1 Latent Dirichlet Allocation

LDA recently achieved very much attention in a growing number of research fields, ranging from text analysis to computer vision. One very attractive point of LDA is, that it effectively generates low-dimensional representations from sparse high-dimensional data, representing documents by a low-dimensional vector of "topics". In this section, we first briefly review the most important aspects of Latent Dirichlet Allocation, for more details the interested reader is referred to [4].

LDA, as introduced by [4], is a Bayesian probabilistic model, that describes document corpora in a fully generatively way. It assumes a fixed number K of underlying topics in a document collection D, where each document d is a mixture of topics t_i. According to LDA, the observable variables, i.e. the words in a document, are generated as follows: First, for each document d, document-specific topic proportions θ_d are drawn according to $\theta_d \sim Dir(\alpha)$. The parameter α is the concentration parameter of the Dirichlet prior, which is a convenient conjugate to the multinomial distribution. Then for each word i in d a topic z_{di} is randomly chosen according to $z_{di} \sim Mult(\theta_d)$. Finally the observed word w_{di} is randomly drawn from the distribution of the selected topic, $w_{di} \sim Mult(\beta_{z_{di}})$. This overall topic distribution is assumed to be drawn randomly beforehand from a Dirichlet distribution $\beta_k \sim Dir(\eta)$, where *beta* is the prior vector on the per-topic word distribution.

The nature of the priors α and β has been studied for example in [22]. We use the Mallet implementation [16], which automatically optimizes α according to the underlying collection.

The resulting word distributions $p(w_n|z_n, \beta)$ for each topic have high probabilities for words that often co-occur in documents. Topics alleviate two main problems arising in natural language processing: synonymy and polysemy. Synonymy refers to the case where two different words (say car and automobile) have the same meaning. These synonyms usually will co-occur in the same topics. Polysemy on the other hand refers to the case where a term such as plant has multiple meanings (industrial plant, biological plant). Depending on the context (industry or biology) different topics will be assigned to the word plant.

5.2 Topic based entity representation

We trained topic models on 100000 random documents extracted from the respective versions of the English, French and German Wikipedia, most of these documents describing persons. We use the trained models to infer the probability of a topic t_i for each word w in an article d. The average probability of topic t_i for document d is the average of the probabilities of topic t_i for each word w in d. These yield distributions both for the query (m) as well as the candidate articles (e) and define new attributes for both documents:

- $\mathbf{P}(e) = (p_1(e), \ldots, p_K(e))$: the probability distribution of K topics in the article text $T(e)$
- $\mathbf{P}(m) = (p_1(m), \ldots, p_K(m))$: the probability distribution of K topics in the query text $T(m)$.

The distributions $\mathbf{P}(e)$ and $\mathbf{P}(m)$ and functions thereof may be used as features in $\vec{\phi}(m, e_j)$, as they are able to represent the proportions of semantically similar words. Table 1 shows exemplary topics for Wikipedia articles with name *John Taylor* together with the titles (stemmed terms) for these topics generated by Mallet[3]. For each topic t_i, the associated probability $p(t_i)$ is the probability of the given

[3]The topics are taken from a model over a subset of Wikipedia with $K = 200$, $|D| = 100k$.

topic for the respective article text. For example, the most prominent topic (t_{80}) derived for the athlete *John Taylor* describes his sportive success in the Olympic Games. The topic with lower probability ($p(t_{135})$ is only about 10%) can be interpreted as an indicator for his nationality. We see that most articles can be described sufficiently by one or two topic clusters which is a good example for the dimensionality reduction provided by LDA. One phenomenon in LDA is that depending on the document length the mixture of topics varies. While in very short documents we often find one very prominent topic (for example the racing driver), longer documents have a higher variety (e.g. the article describing the bass guitarist is the longest one in this example).

5.3 Comparing Thematic Distances

For each pair (m, e_j) we compare the topic distribution $\mathbf{P}(m)$ of the query document containing m with the distributions $\mathbf{P}(e_j)$ for the candidates $e_j \in \mathcal{E}(m)$ computed over the corresponding article texts $T(e_j)$.

One of the simplest ways to compare the topic probabilities is a concatenation of the respective probabilities:

$$\phi_{TC} = \{p_1(m), ..., p_K(m), p_1(e), ..., p_K(e)\} \in R^{2K}. \quad (5)$$

Hence each of the probabilities is used as a separate feature and the classifier has the task to evaluate the differences.

Alternatively we may compute distance terms between corresponding topic probabilities directly. There are a number of distance measures for probability distributions, which propose different weighting factors. Instead of just summing these differences to a scalar difference value we use the difference terms separately as features. Then the classifier can evaluate correlations between these terms to improve performance. The symmetric Kullback-Leibler divergence[14] or relative entropy is very popular in the topic model literature:

$$\phi_{sKLD}(m,e)_i = \frac{1}{2}\left(p_i(m)\log\frac{p_i(m)}{p_i(e)} + p_i(e)\log\frac{p_i(e)}{p_i(m)}\right). \quad (6)$$

Here we use the symmetric version as the $T(e)$ and $T(m)$ are interchangeable with respect to similarity. Each term $\phi_{sKLD}(m,e)_i$ can be used as a scalar feature in $\vec{\phi}(m, e_j)$.

The Jensen-Shannon distance [15], is derived from the KL-distance and adds an additional factor $r = 0.5(p_i(m)+p_i(e))$ to (6):

$$\phi_{JSD}(m,e)_i = \frac{1}{2}\left(p_i(m)\log\frac{p_i(m)}{r} + p_i(e)\log\frac{p_i(e)}{r}\right). \quad (7)$$

The Hellinger distance [3] is an alternative measure for the similarity of probability distributions:

$$\phi_{HD}(m,e)_i = \left(\sqrt{p_i(m)} - \sqrt{p_i(e)}\right)^2. \quad (8)$$

The maximum distance 1 is achieved when $p_i(m)$ assigns probability zero to every set to which $p_i(e)$ assigns a positive probability, and vice versa.

Except for ϕ_{TC} all feature representations yield a maximum dimension of $\phi_{(\cdot)} \in R^K$, with K the number of topics.

We ignore topic indices if both $p_i(m)$ and $p_i(e)$ are less than 0.01. This is based on the assumption, that we don't need to spend modeling effort for uninfluential topics. Note that this has the side effect that the overall number of non-sparse features will be rather low, which speeds up the kernel computation.

Table 2: F1 (micro) performance for name disambiguation using different topic distance representations and kernel types

distance representation	kernel type linear	quadratic	RBF
Hellinger ϕ_{HD}	0.9248	0.9286	0.9256
Jensen-Shannon ϕ_{JSD}	0.9236	0.9249	0.9231
sym. Kullback-Leibler ϕ_{sKLD}	0.9278	**0.9328**	0.9096
topic concatenation ϕ_{TC}	0.8899	0.9107	0.9109

We evaluated all of the above distances using five-fold cross-validation on a small dataset (similar to $Q1$, see table 3) with different kernels using SVM^{Light} standard parameters to find the most appropriate distance and kernel for the task at hand.

As depicted in table 2, using ϕ_{TC} yields the weakest results with a linear kernel. This is because a linear kernel can not model the interactions between the topics for e and m. Better results can be achieved using a quadratic or an RBF kernel. The Hellinger distance also improves results over the Jensen-Shannon distance (significantly with $p<0.05$ only for the quadratic kernel). To summarize, we find that basically all of the above distances yield similar results, which we assume is based on their similar origin. Still, even though the superiority of the symmetric Kullback-Leibler distance is not striking, we found it to be significant with $p < 0.05$ for the linear kernel and $p<0.06$ for the quadratic kernel. Thus, we used this distance for all the experiments described in section 6.

5.3.1 Influence of the context width

[1] and [5] observed, that the direct context around an entity mention often contains most disambiguating information. Thus, we performed some initial experiments to evaluate the optimal context width. We found that reducing the context window to the 25 left and 25 right tokens around the entity mention (as done by [5], and also reported for the corresponding experiments here), yields a slight decrease in predictive performance for our approach. We assume that this is because the inferred topic distributions are less smooth and thus also less reliable.

Thus, we propose to use the complete document of the entity mention m with an additional boost on the local context. We found that a context window of [10,10] around the mention yields the best result. These terms are added five times to the overall tokens of the query document m. We found that this increases the performance significantly ($p<0.05$) in comparison to the unboosted version, and use this setting for all of the following experiments based on topic information.

5.3.2 LDA parameters

In preliminary experiments, we also evaluated different topic models for the task of entity disambiguation. That is, we varied the number of topics from 50 to 500 and found no major difference in the predictive performance when increasing the number of topics above 200. Note that the Mallet implementation of LDA automatically optimizes the α parameter, such that topic models with the same number of topics but different initial values for α yield the same

Table 1: Topics for entities with name *John Taylor* (excerpt) with associated probability value

disambiguation term	i	p(t_i)	Important words (titles) of the topics
South Carolina governor	109	0.3805	unit state, state senat, lieuten governor, hous repres, elect governor, ...
	120	0.2477	north carolina, south carolina, unit state, west virginia, civil war, ...
athlete	80	0.4190	summer olymp, gold medal, world record, silver medal, world championship, ...
	135	0.1047	unit state, rhode island, baltimor maryland, new hampshir, georg washington, ...
racing driver	129	0.7407	grand prix, race driver, motor race, formula, race team, sport car, ...
jazz	141	0.5781	jazz musician, big band, new york, duke ellington, jazz band, ...
bass guitarist	18	0.2964	rock band, solo album, play guitar, band member, rock roll, ...
	70	0.1594	album releas, studio album, debut album, record label, music video, ...

Table 3: Statistics on the disambiguation datasets for different languages. $|A|$ is the size of the reference data set, $|Q|$ the size of the query dataset.

	$\|A\|$	$\|Q\|$	avg. ambiguity level
English Q_1	6213	16582	2.06
English Q_2	10734	15410	26.76
German	22211	4442	2.91
French	7201	1440	1.88

(or very similar) performance. For all topic models we use $\beta = 0.01$ as prior on the word-topic distribution.

6. EXPERIMENTAL EVALUATION

The approach proposed in this paper using symmetric Kullback-Leibler distance over topic distributions, is denoted as *sKLD*. We compare our approach to that of [5] on the English version of Wikipedia. In the respective tables, we refer to this method as *Bun06*.

We first describe the employed datasets, the properties of the different data sets are summarized in table 3.

6.1 Datasets

6.1.1 English Wikipedia

The first dataset, Q_1, constitutes of references for a random selection of persons with an ambiguous name (the information, that an article refers to a person is extracted from YAGO). The reference dataset A contains 6213 different entities, from which we randomly selected each fifth entity as an uncovered entity. After the removal of these entities, we have a ratio of 1242 uncovered vs. 4971 covered entities. We then extracted 16582 queries with 2.06 candidates each (e.g. we build 34197 feature vectors).

In application data such as news articles, entities are often referenced merely by the surname, which makes their distinction even more difficult. To adapt to this, we create an additional dataset Q_2, in which we allow candidates on partial name matches as well. In contrast to the other datasets, here, a candidate is selected if the surface name is contained in the candidates title (without disambiguation term), e.g. the surface name *Jones* can match *Bruce Jones*, *Adam Jones*, *Catherine Zeta-Jones* but also *Jones Soda* or *Jones, Oklahoma*. This way we get more than 26 candidates per query mention and thus a highly ambiguous data set with more than 400k feature vectors representing all possible (name, entity)-pairs. Uncovered entities are modeled as above.

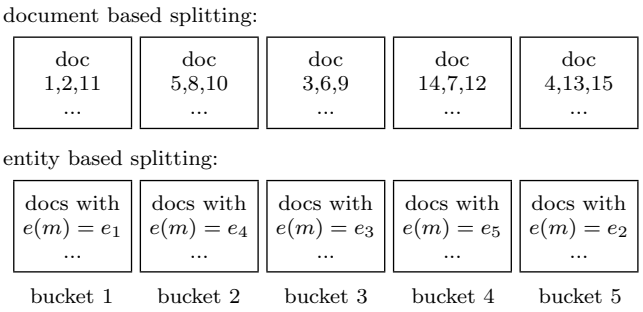

Figure 2: Exemplary document and entity based splitting for cross-validation

6.1.2 German and French Wikipedia

To show that our approach is in general language independent, we also report results for disambiguation using the German and the French Wikipedia. For the German version, we extracted 22211 articles (again persons with ambiguous names), and 44332 query documents for these. 4442 of these entities were modeled as default entities. Here, we had in average 2.91 candidates per query, resulting in 129091 feature vector instances.

For the French version, we extracted 7201 articles (again persons with ambiguous names[4], and 15149 query documents for these. 1440 of these entities were modeled as default entities. Here, we had in average 1.88 candidates per query, resulting in 28430 instances.

For both datasets, we trained a topic model with 200 topics on a random selection of articles from the respective version. We used the same preprocessing techniques as for the English version, adapting only the stop word lists and the language of the employed stemmer.

6.2 Results

We report results obtained on five-fold cross-validations for each of the datasets. Additionally, we propose two splitting strategies for cross-validation: one based on documents, one based on entities. The two splitting policies are depicted in figure 2 for an exemplary set of documents and entities. The document based splitting (upper part of figure 2) is analogous to the standard procedure for cross-validation,

[4]We used language links from the English version to extract persons in other languages.

documents (examples) are distributed i.i.d. over the buckets. The entity based splitting (lower part of figure 2), in contrast, considers the true entities denoted in the query documents, and creates a splitting over the overall entity set. Even though this can result in unbalanced bucket sizes (note that the number of query documents per entity varies and uncovered entities all fall into one bucket), this strategy allows for additional interpretation of the model's ability to generalize, as the testing instances only contain previously unseen entities (together with context). The intuition behind the entity based splitting is, that with new documents new entities described by previously unseen terms may appear, introducing new features that can not be considered by a trained SVM model based only on words.

We use several performance indicators for evaluation that are commonly used in classification scenarios. We report on micro and macro performance (for more details see [23]), to asses both document and class (entity) based performance. Macro measures constitute the averaged Precision (P), Recall (R) and F1-measure (F1) per entity. This measurement is often used when the number of instances is unbalanced over the classes. In contrast, the formulation of micro performance uses the averaged precision etc per document.

6.2.1 Results on dataset Q1

In table 4 we report results for Q_1 using a linear as well as quadratic kernel for the topic based approach. As shown in table 2 the RBF-kernel did not perform better, so we omitted these experiments. To emphasize the performance for uncovered entities, we report the accuracy for these mentions separately.

In the entity based splitting, the baseline *Bun*06 approach completely failed to predict uncovered entities. The reason for this is that in this scenario, *Bun06* never got the chance to learn the appropriate decision threshold, as uncovered entities are observed only in one fold. This results in a rather low recall in micro performance, while in the macro performance the correct predictions for covered entities compensate this. In contrast, in the document based splitting (where a threshold could be learned), the method achieved an accuracy of 70.74% for uncovered names.

We observe that using thematic context distance measured by symmetric Kullback-Leibler distance (*sKLD*) yields better results in the prediction of uncovered entities for both splitting policies, even though we did not learn an adapted threshold $\neq 0$. Considering macro performance, both *sKLD* approaches using a quadratic or linear kernel show significantly better results *($p<0.001$)* compared to the baseline approach. Thus we can increase recall and precision simultaneously in the disambiguation across all entities. While the increase in recall is significant across all experiments (document/entity split), we found that even though we can get better results for precision, the improvement is not always significant in the entity based splitting. This is due to the higher variance in this setting, which is both an effect of the differing bucket sizes as well as the distribution of uncovered entities.

Even though the higher recall achieved by *sKLD* justifies the assumption that word-based approaches often result in lower recall, we find that there is still potential and the challenge to increase this measure.

Note that although the entity based splitting seems difficult, the precision of all approaches is higher compared to the document based splitting. This is because the document based splitting has to treat the difficult uncovered entities in each fold, resulting in a lower precision but also a lower variance compared to the entity splitting.

Table 5: Results for name disambiguation on the dataset Q2

method	splitting type	perf.	F1	P	R
Bun06	by entity	micro	0.2299	0.9454	0.1722
		macro	0.2083	0.2612	0.1906
	by document	micro	0.1595	0.1600	0.1589
		macro	0.0661	0.0737	0.0629
sKLD	by entity	micro	**0.8752**	0.9733	0.7965
		macro	0.8557	0.9051	0.8346
	by document	micro	**0.8354**	0.8964	0.7822
		macro	0.8315	0.8414	0.8346

Table 4 also shows the averaged SVM's computation time (cpu-sec). These learning times are relative fractions, where 1 means the shortest time and all others are multiples of this. For example, a topic model with 200 topics and a quadratic kernel (second third of table 4) requires a more than 10 times longer computation time than a model with the same number of topics but a linear kernel (last third of table 4).

Comparing the explicit increase in learning time to the rather low increase in performance when using a quadratic kernel instead of a linear one (92.33% vs. 91.90% F1, resp. 90.44% vs. 88.28%), we chose to use the linear kernel in the remainder of the experiments. Note that *sKLD* using a linear kernel is also about for times faster than *Bun06*.

6.2.2 Results on dataset Q2

Especially for the highly ambiguous dataset Q_2 an SVM training run using a quadratic kernel consumes very much time. Thus, table 5 shows the results for this dataset using a linear kernel. Even though we increased the SVM's cost ratio for false negative predictions (as we have many more negative than positive examples), *Bun06* does not achieve satisfying results for this dataset as the F-measure is always below 22%. We assume, that this is because the approach is more focussed on the disambiguation of persons, while in this dataset many other entities (or entity types) appear. In contrast, our approach keeps up the good performance reported for the smaller corpus. Even though precision and recall are notably lower, we conclude that the proposed thematic context distance is a very good measure for the disambiguation of name phrases. We assume that for larger datasets a more sensitive topic model (i.e. with more topics over more documents) might be needed. Recent approaches to speed up LDA on very large corpora (for example [10] or [21]) provide a useful tool to be explored.

6.2.3 Results on German and French datasets

Tables 6 and 7 show results for the disambiguation model using the German and the French dataset. In either scenario, the obtained F1-measure is well above 80%, with low derivation, which is a promising result. We find that although we did not spend additional efforts on the specific characteristics of these languages, we can very accurately assign name phrases to the corresponding articles.

Table 4: Results for name disambiguation on the dataset Q1

		entity based splitting						document based splitting					
method		micro perf.	*p-val.*	macro perf.	*p-val.*	Accuracy uncov.	cpu sec.	micro perf.	*p-val.*	macro perf.	*p-val.*	Accuracy uncov.	cpu sec.
Bun06[5]	F1	0.7990		0.8286		0.0000	3.91	0.8730		0.8658		0.7074	4.89
	P	0.9563		0.9043				0.9056		0.8800			
	R	0.6937		0.8027				0.8427		0.8704			
sKLD	F1	0.9233	0.06	0.9152	0.00	0.6926	15.21	0.9044	0.00	0.9061	0.00	0.787	13.77
quadratic	P	0.9813	0.31	0.9516	0.00			0.9238	0.00	0.9177	0.00		
kernel	R	0.8720	0.03	0.8995	0.00			0.8859	0.00	0.9094	0.00		
sKLD	F1	0.9190	0.09	0.9079	0.00	0.7455	1.00	0.8828	0.03	0.8856	0.00	0.7862	1.00
linear	P	0.9821	0.28	0.9517	0.00			0.9012	0.13	0.9022	0.00		
kernel	R	0.8636	0.05	0.8874	0.00			0.8651	0.01	0.8862	0.00		

Table 6: Name disambiguation using thematic context distance *sKLD* on the German dataset

splitting type	perf.	F1	P	R	std. F1
by entity	micro	0.8796	0.9748	0.8015	0.0462
	macro	0.8491	0.9022	0.8279	0.0055
by document	micro	0.8292	0.8681	0.7937	0.0049
	macro	0.8275	0.8442	0.8288	0.0053

Table 7: Name disambiguation using thematic context distance *sKLD* on the French dataset

splitting type	perf.	F1	P	R	std. F1
by entity	micro	0.8801	0.9748	0.8025	0.0449
	macro	0.8486	0.9034	0.8268	0.0087
by document	micro	0.8378	0.8755	0.8032	0.0045
	macro	0.8314	0.8493	0.8348	0.0064

On the relatively small French dataset, we also evaluated the quadratic kernel and found that performance could be increased mostly in the document splitting setting, with up to 3% F-measure.

As LDA is a language independent method, these results are not surprising but nevertheless new: we have shown that the same approach to measure thematic context distance works for various source languages and yields very good results. Note that apart form training the LDA model, which is unsupervised, no other language specific adaptations needed to be made.

7. CONCLUSION

In this paper, we approach the problem of name ambiguity using thematic distances over describing documents. We compare our approach to a state of the art method that exploits word-category information extracted from Wikipedia. Our approach relies on semantic topics provided by LDA and thus is able to exploit more information than other, rather restrictive word-matching methods that compare common terms between two documents. Based on this generalized comparison, we significantly improve the assignment of name mentions to the underlying articles in Wikipedia. We also treat names that are not covered by an article in Wikipedia and show that our method can handle this problem very accurately, superior to the baseline approach. This is a crucial aspect, since when we retrieve information for a known entity, we don't want to assign false facts to it.

In future work we will apply the proposed method to the automatic updating of knowledge bases, which is a very interesting line of research. It opens the possibility to automatically generate articles in Wikipedia for entities, that are currently not covered.

Acknowledgements

The work presented here was funded by the German Federal Ministry of Economy and Technology (BMWi) under the THESEUS project. We would like to thank the reviewers for their constructive comments and suggestions on this paper.

8. REFERENCES

[1] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proc. of COLING-ACL*, pages 79–85, San Francisco, California, 1998.

[2] I. Bhattacharya and L. Getoor. Relational clustering for multi-type entity resolution. In *Proceedings of the 4th international workshop on Multi-relational mining*, MRDM '05, pages 3–12, 2005.

[3] D. Blei and J. Lafferty. Topic models. In A. Srivastava and M. Saham, editors, *Text Mining: Theory and Applications*. Taylor and Francis, 2009.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] R. C. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*, pages 9–16, 2006.

[6] Y. Chen and J. Martin. Towards robust unsupervised personal name disambiguation. In *Proc. EMNLP-CoNLL*, pages 190–198, 2007.

[7] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the Joint Conference on EMNLP and CoNLL*, pages 708–716, 2007.

[8] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, and J. Y. Zien. Semtag and seeker: bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th*

international conference on World Wide Web, WWW '03, pages 178–186. ACM, 2003.

[9] J. Hassel, B. Aleman-Meza, and I. B. Arpinar. Ontology-driven automatic entity disambiguation in unstructured text. *Lecture Notes in Computer Science (LNCS)*, pages 44–57, 2006.

[10] M. Hoffman, D. M. Blei, and F. Bach. Online learning for latent dirichlet allocation. In *Proceedings of NIPS*, 2010.

[11] T. Joachims. *Learning to Classify Text Using Support Vector Machines – Methods, Theory and Algorithms.* Kluwer/Springer, 2002.

[12] T. Joachims. Optimizing search engines using clickthrough data. In *Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, 2002.

[13] A. Kittur, E. H. Chi, and B. Suh. What's in wikipedia? mapping topics and conflict using socially annotated category structure. In *Proceedings of CHI '09*, pages 1509–1512, 2009.

[14] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):49–86, 1951.

[15] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

[16] A. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[17] G. Miller and W. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.

[18] M. F. Porter. Snowball: A language for stemming algorithms. Published online, October 2001.

[19] H. Srinivasan, J. Chen, and R. Srihari. Cross document person name disambiguation using entity profiles. In *Proceedings of the Text Analysis Conference (TAC) Workshop*, 2009.

[20] F. Suchanek, G. Kasneci, and G. Weikum. Yago - a large ontology from wikipedia and wordnet. *Elsevier Journal of Web Semantics*, 6(3):203–217, 2008.

[21] M. Wahabzada, K. Kersting, C. Bauckhage, and A. Pilz. More influence means less work: Fast latent dirichlet allocation by influence scheduling. Poster paper in CIKM, 2011.

[22] H. Wallach, D. Mimno, and A. McCallum. Rethinking lda: Why priors matter. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, 2009.

[23] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1–2):69–90, 1999.

[24] N. Yuhanna. Today's challenge in government: What to do with unstructured information and why doing nothing isn't an option. Technical report, Forrester Research, 2010.