Entity Disambiguation using Link based Relations extracted from Wikipedia

Anja Pilz ANJA.PILZ@IAIS.FRAUNHOFER.DE
Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS, Schloss Birlinghoven, Sankt Augustin, Germany

Abstract

We present an approach for the disambiguation of textual mentions of ambiguous names: disambiguation means here the identification of the true entity denoted by a name phrase appearing in a query context through its assignment to the corresponding Wikipedia article. If this article does not exist, we assign this query to a default entity. Ambiguity of names is a major problem in information retrieval and causes uncertainty in the assignment of name phrases to existing knowledge base entries. We propose a kernel classifier to approach this problem and compare two Wikipedia structures to construct a rich feature space. The first approach relies on Wikipedia categories, the second on relations constructed from Wikipedia's hyper link structure.

We evaluate both approaches on the German version of Wikipedia and show that both outperform a baseline approach using simple cosine similarity.

1. Introduction

Still, unstructured text is the most common container of information. The construction or 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. Most of these tasks are well studied for English, but less sufficiently for other languages such as German.

Entity recognition determines a name phrase in a context, that is to be endowed with a special entity type label such as person or location. The task of entity disambiguation is then to identify the entity mentioned in the given context, for example its assignment to a unique entity definition.

To illustrate the necessity of entity disambiguation, we describe some of the characteristics and effects of name ambiguity.

Name ambiguity means that entities often share the same name. For example, there is a *Michael Müller*, who is a federal minister in the federal state of Berlin, another who was federal minister in the federal state of North Rhine-Westphalia, a comedian, a handball player and more than 3000 others that are listed in the German phone book. Note that the first two entities also belong to the same political party which makes their distinction even more difficult.

Additionally, polysemy of names spans across entity types. The term *Napoli* may be used to refer to the Italian city *Naples* (there are more than ten municipalities of the same name in the United States), organizations (an Austrian confectionery manufacturer) and persons (there are three soccer players called *di Napoli* listed in the German Wikipedia).

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[1] search for the name *Peter Müller* mention seven different people. While it may be clear to a human that the Prime Minister from Saarland, the boxer from Cologne and the professor of mathematics are not the same person, it is difficult for a computer program to make this distinction. In fact, a human may have a hard time retrieving all the material relevant to the particular entity of interest in without being swamped by information on namesakes. Whereas pure word based context information can of-

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Haifa, Israel, 2010.

[1] http://www.google.de

ten be insufficient, the affiliations of an entity, which constitute relations to other entities, are a very good indicator of the entity's identity.

Clearly, Web searches could be much more effective, if the retrieved documents were in advance analyzed concerning the identity of the mentioned entity. This could then provide a grouped result which facilitates information extraction. The same is true for searches in bibliographic archives, publication databases etc.

In the next section we describe related work to entity resolution. Then we outline the properties of Wikipedia as a reference for unique named entities and the construction of a disambiguated dataset from it. Subsequently we describe the kernel approach used in this paper. Then we present the experimental setup and describe the obtained results.

2. Related Work

In the following, we describe related work in the field of entity disambiguation. Due to the absence of a public benchmark set, especially for German, we can not compare the presented results directly and hence outline only the basic ideas.

Named entity disambiguation is closely related to the task of word sense disambiguation which aims at resolving the ambiguity of common words in a text. Both tasks have in common that the meaning of a terms mention is strongly dependent on the context it appears in. (Miller & Charles, 1991) hypothesize that words with similar meanings are often used in similar contexts. Later studies translate this assumption to the disambiguation of proper names, assuming that a particular entity will likely be mentioned in certain contexts.

One of the first studies in the field of entity resolution is (Bagga & Baldwin, 1998). They propose to create context vectors for each occurrence of a name, 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, similarity among these vectors is measured using the cosine measure (see 4.1.1).

(Bhattacharya & Getoor, 2005) extensively use relational evidence for entity resolution. In the context of citations we may conclude that "R. Srikant" and "Ramakrishnan Srikant" are the same author, since both are coauthors of another author. They consider the mutual relations between authors, paper titles, paper categories, and conference venues. They argue that if they jointly resolve the identity of authors, papers, etc. that this leads to a better result than considering each type alone. They construct probabilistic networks capitalizing on two main ideas: First they use tied parameters for repeating features over pairs of references and their resolution decisions. Second they exploit the overlap between decisions, as two different decisions are dependent. They use similarity measures to resolve entities by clustering them taking into account the relations between objects, and get encouraging results.

(Hassel et al., 2006) 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.

(Cucerzan, 2007) 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.

(Bunescu & Pasca, 2006) resolve entities using a Ranking SVM (Joachims, 2002a), which generates a ranked list of plausible entities for a given query context. Features are words in a window around the name phrase as well as the Wikipedia categories. We re-implemented a slightly altered version of this approach (outlined in 4.1.2) and compare it to our approach using links instead of categories (see, 4.1.3). Similar to our work, the reference corpus is extracted from Wikipedia (see 3.2).

In this study we are searching for a domain independent method, where candidates for descriptive relations are not necessarily known in advance. Thus we combine the ideas presented in (Hassel et al., 2006) and (Bunescu & Pasca, 2006) to study the potential of Wikipedia's link structure as a relation indicator.

3. Wikipedia as Knowledge Resource

Wikipedia[2] is a web-based, free content encyclopedia project, written collaboratively by volunteers using a wiki software that allows almost anyone to add and change articles. Since its creation in 2001 it has become the largest organized knowledge repository on the Web. Here, we are concerned with the German version, that holds about 850k articles with an average length of 3500 bytes and 20.8 million internal links.

In this work, we study the task of entity disambiguation using Wikipedia as knowledge base. Given a context mentioning a name phrase, we want to assign this name and and the associated context information, e.g. the surrounding words, to one of the candidate entities $\mathcal{E}_n = \{e_1, \ldots, e_m, e_{out}\}$ extracted from Wikipedia, where each e_i corresponds to a "true" underlying entity. In this notation, entities $e_1, \ldots, e_m$ denote specific entities covered in Wikipedia. We augment the set of Wikipedia entities with a default entity e_{out} denoting entities not covered in Wikipedia to account for contexts mentioning entities not represented by an article in Wikipedia. As previously stated, there are more than 3000 persons with the name *Michael Müller* listed in the German telephone directory, but only seven articles exist in Wikipedia. We can assume that an arbitrary text to be analyzed, such as a news article, is likely to mention one of the unrepresented entities. Hence, to avoid an incorrect assignment to an article in Wikipedia, we use the formal assignment to e_{out}.

In the following, we describe some of Wikipedia's main characteristics that we also utilize in this study.

3.1. Attributes of Wikipedia articles

Most of the information contained in Wikipedia is stored in articles. An article is uniquely identified by the most common name of the subject described in the article. Ambiguous names are further qualified with additional information placed in parentheses. For instance, entities are distinguished by their affiliations, occupations or associated locations, e.g. *Michael Müller (Comedian)*, *Michael Müller (Handballspieler)*, *Michael Müller (SPD, NRW)* and *Michael Müller (SPD, Berlin)*.

Articles are supposed to hold information focused on one specific entity or concept, with varying wealth of detail. In the following we consider the article as the definition of the entity it refers to.

Additionally, each article is endowed with one or more categories. These can represent topics applying to the article but also general attributes such as gender or year of birth in the case of persons or the founding year in case of organizations. Next to these general categories exist also very specific categories that apply to only very few entities.

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. This link structure can be used as a first approach to model a network of relations in Wikipedia. Note that these links are not qualified as in relation extraction approaches (i.e. *memberOf* or *bornIn*), but the type of the relation should implicitly emerge from the context it is mentioned in.

We exploit Wikipedia's internal link structure in two ways: First we generate an automatically disambiguated dataset, which is described in 3.2, and second we compute entity attributes based on their links to other articles, as described in 4.1.3.

In the remainder of this article, we use the following notation:

- $e.T = \{w_1, ..., w_i\}$: the article text of entity e containing words w_i,
- $e.C = \{c_1, ..., c_j\}$: the set of categories assigned to e,
- $e.L = \{e_{rel_1}, ..., e_{rel_m}\}$: the set of related articles e links to.

3.2. Generation of a disambiguated dataset from Wikipedia

Assuming the correct placement and direction of links in Wikipedia, we can extract all articles referencing an entity of interest. Since the link provides the true entity, this results in a set of disambiguated documents. In the following, we refer to these documents as *query documents* q and use the notation:

- $q.T$ to denote the text in document q,
- $q.e$ to refer to the true entity mentioned in q,
- $q.n$ is the name phrase in q used to refer to $q.e$.

For each entity e_i in Wikipedia, we receive positive examples, i.e. those documents in which it is the true underlying entity ($q.e = e_i$), as well as negative examples (query documents with $q.e \neq e_i$). This set of

[2] http://www.wikipedia.org

disambiguated queries can be used for the training as well as the evaluation of a disambiguation model.

To construct such a data set, we first select a set of entities for which referencing documents are to be found. This is done by considering the seven most common German surnames (i.e. *Müller*, *Schmidt*, *Schneider*, *Fischer*, *Weber*, *Meyer* and *Wagner*) and extracting all names found in Wikipedia that contain one or more of these strings but refer to more than one entity. This way, we gained a set of 433 ambiguous names, collectively relating to 1333 different entities (which is not restricted to entity types). We determined the average number of candidates per name phrase to be 4.72. Hence, an uninformed baseline approach could achieve about 25% accuracy by simple guessing.

In the next step, we randomly chose a subset of the referencing articles for each of the above mentioned entities, resulting in 8155 query documents. Since the entity's anchor is linked to the correct entity (assuming the link was placed correctly by the Wikipedian) this constitutes then a disambiguated example context for the current entity. Additionally, we restrict the context to a window of 25 words left and 25 words right around the entity mention. We ignore stop words and use the stemmed word forms (obtained via the Snowball algorithm for German (Porter, 2001)).

4. Entity Disambiguation using Wikipedia and SVMs

4.1. Attributes for entity disambiguation

In this study we adapt previous approaches to entity resolution and compare them to an approach that uses links to other articles to model relations between Wikipedia entities.

4.1.1. Context-Article Similarity

The first approach to assess similarity between a query document and a Wikipedia article is cosine similarity. This means a simple summation over common words, based on the idea that the larger this number the more similar the context and hence the more similar the entities denoted. (Bunescu & Pasca, 2006) and (Cucerzan, 2007) both evaluated experimentally a ranking function based on the cosine similarity between the text of the query $q.T$ and the text of the entity's article $e.T$:

$$\phi_{cos} = \cos(q.T, e.T) = \frac{q.T}{||q.T||} \frac{e.T}{||e.T||}.$$

The factors $q.T$ and $e.T$ are represented in the standard vector space model, where each component corresponds to a term in the vocabulary.

Measuring the similarity between contexts in this way has one major drawback: if alternative terms for one meaning are used, 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.

4.1.2. Word-Category Pairs

The following approach is based on the ideas presented in (Bunescu & Pasca, 2006) and (Cucerzan, 2007). The main idea is to use the categories assigned to an entity as particularly indicative features. Consider the

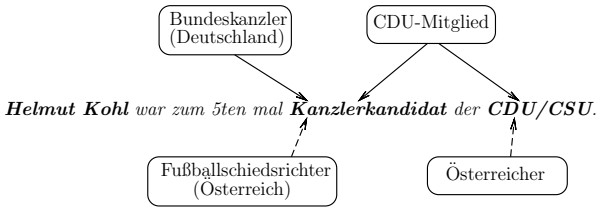

Figure 1. Example of a query document with associated candidate entity categories. Dashed arrows indicate low and solid high correlation of context term and category.

example depicted in fig. 1. The two candidate entities for the query name $q.n$=*Helmut Kohl* are *Helmut Kohl* (e_1) and *Helmut Kohl (Schiedsrichter)* (e_2), where *Helmut Kohl* is the true entity, i.e $q.e = e_1$. The categories depicted in the upper part of the figure are a selection of those belonging to the true entity $q.e$, the categories in the lower part are a selection of those belonging to the other candidate entity e_2. The query document (which is here only one sentence) contains some indicative words such as *Kanzlerkandidat* and *CDU* that are highly correlated with categories such as *Bundeskanzler* or categories indicating party membership (i.e. *CDU-Mitglied*). On the other hand, their semantic correlation with other categories such as *Fußballschiedsrichter* is considerably lower.

Consequently, it is possible to design a disambiguation model that learns the magnitude of semantic correlations between words and categories and exploits these correlations in the computation of feature weights.

Assuming that only common words (terms appearing both in the query document and the article text) are indicative, the feature vector representation for this approach is

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

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

These word-category features have binary values that depend on the query text $q.T$ in conjunction with the article text $e.T$ (which is $q.T \cap e.T$) and the categories to which the entity e belongs (the set $e.C$). Based on the example in figure 1, we can create two feature vectors $\phi(q, e_1)$ and $\phi(q, e_2)$ that contain a binary feature for every possible pair (w, c) of words $w \in q.T \cap e.T$ and categories c. The feature vector describing candidate e_1 for the query q is then composed of:

$$\begin{aligned} &\phi_{w,c}(q, e_1) = 1 \\ &\text{for } (w, c) \in \{(w_1, c_1), (w_1, c_2), ..., (w_2, c_1), (w_2, c_2), ...\} \\ &\phi_{w,c}(q, e_1) = 0 \\ &\text{for } (w, c) \in \{(w_1, c_3), (w_1, c_4), ..., (w_2, c_3), (w_2, c_4), ...\} \\ &\qquad \text{iff } w \in q.T \cap e.T, \end{aligned}$$

where w_1 and w_2 denote the terms *Kanzlerkandidat* resp. *CDU* and c_1 to c_4 denote the categories *Bundeskanzler (Deutschland)*, *CDU-Mitglied*,*Österreicher* and *Fußballschiedsrichter (Österreich)*.

The representing vector for the candidate e_2 is build analogously.

In contrast to (Bunescu & Pasca, 2006), we use only directly assigned categories and instead of analyzing the category hierarchy to extract top-level categories.

4.1.3. Modeling Relations in Wikipedia

The usage of relations between entities for entity disambiguation is motivated by multiple reasons. First, we assume that candidates for an entity mention often share the same categories (a simple example would be the gender category). Furthermore, categories are supposed to hold a reasonable set of articles, which makes them sometimes unspecific. In contrast, there are no restrictions for links apart from the requirement that there is some thematic relatedness. Hence, the number of relations (links to other articles) of a Wikipedia article is in general much lager than the number of its categories.

Since the construction of a named entity recognizer for German is a difficult task, but essential for a relation extraction model, we try to find here a different approach to model relations. We assume that links in Wikipedia are a first step to do this. As already outlined, most articles in Wikipedia have links to related subjects. Here, we use their article titles as attribute.

Let $e.L$ denote the set of links from e to other articles. Each linked article $e_{rel} \in e.L$ has a unique identifier l, in our case the article's headline, which is in the following used to refer to e_{rel}. Assuming that again only common words are indicative, the feature vector representation is then

$$\phi_{w,L}(q, e) = \begin{cases} 1, & \text{if } w \in q.T \cap e.T \text{ and } l \in e.L \\ 0, & \text{else.} \end{cases}$$

Hence, feature vectors are build analogical to the example in 4.1.2 simply by replacing each category c by a link l. In figure 1, we would replace the category names with the link titles.

Note that the binary representation above does not model the link's influence. Obviously not all links will have a distinctive character, as for example a link to an article listing events in the year 1980 is not very informative. Thus, we formulate a disagreement measure representing the relatedness between two articles to approximate the weight of links. Inspired by the link-based relatedness presented in (Milne & Witten, 2008), we define the disagreement measure between two articles of interest a and b to be

$$d(a, b) = \frac{\log(\max(|A|, |B|)) - \log(|A \cap B|)}{\log(|W|) - \log(\min(|A|, |B|))}$$

where $|A|$ is the number of articles a links to, $|B|$ is the number of articles b links to and $|A \cap B|$ is the number of articles both a and b link to. We define the disagreement to be one, if the set of common links $A \cap B$ is empty.

This measure is employed to add a weight to the binary representation above:

$$\phi_{w,d(e,l)}(q, e) = \begin{cases} d(e, l), & \text{if } w \in q.T \cap e.T \text{ and } l \in e.L \\ 0, & \text{else.} \end{cases}$$

If an entity e holds many references to other articles, that are not also referenced by the article l, the disagreement $d(e, l)$ is high. If on the other side, the set of common links is large, the disagreement is lower. It follows that the more entities referenced both by e and l, the closer their relationship. As experiments show (see 5), this simple weighting scheme results in a more informative link representation.

4.1.4. Detecting non-listed Entities

As already mentioned, there are millions of entities not present in Wikipedia. To account for this in the model's design we simulate non-listed entities. This is done by removing the article and consequently all provided information for a fixed fraction of entities. Additionally, we add a new feature

$$\phi_{out} = \mathbb{1}(e, e_{out}).$$

to mark non-listed entities/candidates. Hence, the feature vector representing a non-listed entity contains only this attribute, while the attribute is zero for all candidates listed in Wikipedia.

4.2. Support Vector Machines for Entity Disambiguation

We formulate the task of entity disambiguation as a binary classification problem. For each name phrase exists only one true entity that it should be linked to. This is the entity represented by an article in Wikipedia, which best fits to the query context, or a default entity if no such article exists. We compute feature vectors $\phi(q, e_i)$ for each candidate-context pair and assign labels as in the following example. If two candidates can be determined, than the models input consists of three vectors $\phi(q, e_i)$, $i = 1, 2, 3$ (candidates from Wikipedia plus the default entity) with according labels

$$y(\phi(q, e_i)) = \begin{cases} 1, & \text{if } q.e = e_i \\ -1, & \text{else.} \end{cases}$$

To solve this classification problem, we employ a standard SVM algorithm after (Vapnik, 1998) with a linear kernel. We use the SVM^{Light} implementation[3] by Thorsten Joachims, details are described in (Joachims, 2002b). Since it is possible that two entities come from a very similar field (i.e. the two politicians mentioned in the introduction), it is likely that both entities receive a high score for a given context. Since we want to assign the context to uniquely one entity, we use the entity with the highest score, which is here the entity represented by the feature vector with the maximal offset from the hyperplane. (Bunescu & Pasca, 2006) used an alternative formulation, a Ranking SVM as described in (Joachims, 2002a), which is similar to a structured output SVM. Since the rankings are here equivalent to positive/negative labels and hence the alternative optimization constrained in the Ranking SVM is simplified, we assume that a binary classification SVM should be fitted as well for the task at hand.

5. Experimental Results

We use the data set depicted in 3.2 with 8155 query documents. We perform all experiments in five fold cross validation scenarios, where we randomly split the queries into training (80%) and test (20%) sets for each fold. The splitting is not entity-based (i.e. dependent on the entity mentioned in the query), but document-based. Hence we have to take into account that potentially many entities are to appear in the test sets that were not seen in training and thus potentially many new features can not be considered.

We chose this splitting-policy, since this scenario is more likely to be the case when applying the model to news data, that contain much more entities than Wikipedia. Hence, the goal is not to learn models for entities (i.e. one model for each entity) but a more general model.

Note that experiments using an entity based splitting that results in a test set containing only references to previously unseen entities can give even deeper insight to the models generalization properties.

To account for entities not contained in Wikipedia, each 20th of the previously mentioned 1333 entities is modeled as non-listed.

For evaluation we use micro and macro performance (see (Yang, 1999)), to asses both document and class (entity) based performance. Macro measures constitute the averaged precision, recall and f-measure per entity. The formulation of micro performance uses the averaged precision etc per document. Since the number of false positives is then equal to the number of false negatives, precision and recall are the same and hence we report here only the f-measure.

We first evaluated the approach using simple cosine similarity. As depicted in table 1, the model's performance is well above the uninformed baseline (which is about 25% accuracy), but not satisfyingly good. Additionally, the standard deviation is very high which decreases reliability. In this scenario exist only two features (ϕ_{cos} and ϕ_{out}) and since we used a linear kernel, there are only two possible weights for adaption, which clearly is not sufficient. As micro performance is lower than the macro performance, we deduce that this approach has a bias for certain entities and hence assigns them very often. This can still result in high macro precision but yields lower micro precision.

Table 1. Cross validation results using cosine similarity.

	Avg. over folds	σ
F_{micro}	60.25%	0.64273
F_{macro}	65.81%	0.74606
P_{macro}	68.03%	0.77847
R_{macro}	67.31%	0.76722

As table 2 shows, results can be improved using a richer feature space. Here the maximal dimension of the attribute vectors is much higher, i.e. $|W| \times |C|$. We determined that there are in total 2017 different categories in the considered data set (hence $|C|$

[3] http://svmlight.joachims.org

= 2017). The results show that both micro and macro performance can be increased by more than 12% points which means a high error reduction. Next, we evaluate the model using relations derived from outgoing links. Obviously, this approach does not perform better than the category based approach. While micro performance is comparable to the category-based approach, macro measures clearly show a less good performance. Since most entities are very well described by their categories, a distinction by links to related entities is obviously not equally appropriate for all entities. This results in a lower macro performance that captures entity-related errors better than micro performance.

Table 2. Cross validation results using word-category pairs.

	Avg. over folds	σ
F_{micro}	89.85%	0.01411
P_{micro}	91.13%	0.01171
R_{micro}	88.61%	0.01820
F_{macro}	81.75%	0.18403
P_{macro}	81.86%	0.20991
R_{macro}	83.94%	0.10837

Table 3. Cross validation results using binary word-link pairs.

	Avg. over folds	σ
F_{micro}	87.07%	0.00920
P_{micro}	88.85%	0.00828
R_{micro}	85.37%	0.01446
F_{macro}	74.42%	0.28446
P_{macro}	74.15%	0.33004
R_{macro}	77.63%	0.17518

We next performed experiments using links qualified by similarity. As the results in table 4 show, there is a significant enhancement in performance as compared to the binary representation. While the increase in micro performance is not very high, there is a clear increase in macro performance. This shows that an approach using a qualified representation of links can provide more distinguished attributes for the entities. A comparison to the category based approach shows that there is no significant difference. We argue that although outgoing links provide valuable information for entity distinction, there should be more investigation concerning their quality respectively their informative value. This is encouraged by experiments showing that performance can be increased, when links are qualified by our simple agreement measure.

Wikipedia categories constitute considerably more restrictive attributes then links to other articles. Although we could show that weighted links bear a similarly high information content, we assume that an evaluation of links will enhance results. Automatic link detection methods, as for example proposed by (Milne & Witten, 2008) or (Gardner & Xiong, 2009), can be employed to estimate the relevance of a link or to find additional, more informative links. In both scenarios using outgoing links, the maximal dimension of feature vectors is much higher as compared to the previous approaches (we determined the total number of different links to be 13489). Again, this encourages a more selective choice of relevant links.

Table 4. Cross validation results using weighted word-link pairs.

	Avg. over folds	σ
F_{micro}	89.79%	0.01841
P_{micro}	91.31%	0.01432
R_{micro}	88.33%	0.02300
F_{macro}	81.32%	0.19129
P_{macro}	81.47%	0.22194
R_{macro}	83.51%	0.11060

In our experiments, we modeled each 20th entity as non-listed, resulting in only 66 entities modeled as not contained in Wikipedia. Hence, there are also very few examples for these entities both in training and test sets, and thus accuracy for these entities has low weight in the overall performance. We consequently analyzed the accuracy for them separately and found that highest average accuracy is achieved by the approach using outgoing links as relations (in average 42%). We see that this is a point for further investigations and assume that probably a higher ratio of non-listed entities allows increase in accuracy.

Note that due to the document based splitting policy, the model observes many test queries on entities unseen in training. We determined that over all cross validation folds, there are at least 10% of the entities to be resolved unknown to the model. Since the models performance is still very accurate we can argue that we build a model that can transport information among entities and is hence able to generalize. Hence we can suppose a more successfull application to new contexts containing unseen entities.

6. Summary

This article compares two approaches to entity disambiguation, with one approach utilizing Wikipedia categories and the other Wikipedia's link structure. While the category based approach results in a very

distinctive feature set, the link based approach produces a much larger but also more general feature set that requires a more sophisticated selection method.

Both approaches very accurately assign textual mentions of entities in German text to their representation in an external knowledge resource, e.g. Wikipedia, and are also able to detect entity mentions without reference in Wikipedia.

We evaluated a SVM approach with varying richness in features such as simple cosine similarity, word-category pairs and word-relation pairs.

We have shown that it is not necessary to rely on categories manually assigned to the Wikipedia articles and that links to other articles are high-performance attributes for entity disambiguation as well. Since the difference in performance between the two approaches is not significant, we assume that the usage of statistical relational learning methods, i.e. a graph approach, might be a better method to grasp the information contained in the Wikipedia link network.

A challenging question for future work is also, how the disambiguation of one entity affects the disambiguation of other entities or related concepts mentioned in the same document. This could be investigated using for example a cascade of communicating disambiguation models or collective inference.

Acknowledgments

The work presented here was funded by the German Federal Ministry of Economy and Technology (BMWi) under the THESEUS project.

References

Bagga, Amit and Baldwin, Breck. Entity-based cross-document coreferencing using the vector space model. In *Proc. of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pp. 79–85, San Francisco, California, 1998.

Bhattacharya, Indrajit and Getoor, Lise. Relational clustering for multitype entity resolution. In *Proc. Fourth International Workshop on MultiRelational Data Mining (MRDM2005)*, 2005.

Bunescu, Razvan C. and Pasca, Marius. Using encyclopedic knowledge for named entity disambiguation. In *Proc. of EACL*, pp. 9–16, 2006.

Cucerzan, Silviu. Large-scale named entity disambiguation based on wikipedia data. In *Proc. 2007 Joint Conference on EMNLP and CNLL*, pp. 708–716, 2007.

Gardner, James J. and Xiong, Li. Automatic link detection: a sequence labeling approach. In *CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management*, pp. 1701–1704. ACM, 2009.

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

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

Joachims, Thorsten. *Learning to Classify Text Using Support Vector Machines – Methods, Theory and Algorithms*. Kluwer/Springer, 2002b.

Miller, G.A. and Charles, W.G. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):128, 1991.

Milne, David N. and Witten, Ian H. Learning to link with wikipedia. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM'2008)*, pp. 509–518, 2008.

Porter, Martin F. Snowball: A language for stemming algorithms. Published online, 2001. URL `http://snowball.tartarus.org`.

Vapnik, Vladimir N. *Statistical Learning Theory*. John Wiley & Sons., 1998.

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