Do We Need Entity-Centric Knowledge Bases for Entity Disambiguation?

Stefan Zwicklbauer
University of Passau
Passau, 94032 Germany
stefan.zwicklbauer@uni-passau.de

Christin Seifert
University of Passau
Passau, 94032 Germany
christin.seifert@uni-passau.de

Michael Granitzer
University of Passau
Passau, 94032 Germany
michael.granitzer@uni-passau.de

ABSTRACT

Entity Disambiguation has been studied extensively in the last 10 years with authors reporting increasingly well performing systems. However, most studies focused on general purpose knowledge bases like Wikipedia or DBPedia and left out the question how those results generalize to more specialized domains. This is especially important in the context of Linked Open Data which forms an enormous resource for disambiguation. However, the influence of domain heterogeneity, size and quality of the knowledge base remains largely unanswered. In this paper we present an extensive set of experiments on special purpose knowledge bases from the biomedical domain where we evaluate the disambiguation performance along four variables: (i) the representation of the knowledge base as being either entity-centric or document-centric, (ii) the size of the knowledge base in terms of entities covered, (iii) the semantic heterogeneity of a domain and (iv) the quality and completeness of a knowledge base. Our results show that for special purpose knowledge bases (i) document-centric disambiguation significantly outperforms entity-centric disambiguation, (ii) document-centric disambiguation does not depend on the size of the knowledge-base, while entity-centric approaches do, and (iii) disambiguation performance varies greatly across domains. These results suggest that domain-heterogeneity, size and knowledge base quality have to be carefully considered for the design of entity disambiguation systems and that for constructing knowledge bases user-annotated texts are preferable to carefully constructed knowledge bases.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval—*Search Process*; I.2.7 [**Computing Methodologies**]: Natural Language Processing—*Text analysis*

i-Know '13, September 04 - 06 2013, Graz, Austria

General Terms

Algorithms, Experimentation

Keywords

Entity Disambiguation, Text Annotation, Linked Data

1. INTRODUCTION

Semantically structured information like Linked Data exhibit huge potential for improving unstructured information management processes in different domains like the web, enterprises or in research. Particularly, textual information can be linked to concepts found in the Linked Data Cloud to improve retrieval, storage and analysis of large document repositories. Entity disambiguation algorithms establish such links by identifying the correct semantic meaning (represented as unique ID or URI) from a set of candidate meanings (referred to as the knowledge base) to a selected text fragment.Basically, in shallow text parsing entity disambiguation succeeds entity recognition, i.e., the identification of text chunks that belong to a certain class of concepts, like for example persons, locations etc. It is also related to Word Sense Disambiguation tasks. In the database community this process is also known as Record Linkage (see [3] for more details).

While entity disambiguation techniques have been studied extensively in the past 10 years, they have mostly been tested on rather general knowledge bases or for particular classes of entities. For example, DBPedia Spotlight covers named entity recognition and disambiguation tasks which are optimized for DBPedia [14]. DBPedia Spotlight shows high accuracy (around 80%) by using standard Vector Space Models. A lot of previous researchers utilized similar models for linking textual data to Wikipedia or other encyclopedic knowledge bases with similar accuracy [1,3,15,18]. Exploiting different forms of semantic relationships can improve disambiguation accuracy. For example, in [6] the authors show that social and semantic relatedness can improve disambiguation by 16.7%. Again, Wikipedia serves as underlying knowledge base. Related work also shows that accurate results can be achieved when focusing on particular classes of entities like for instance geographic entities [11, 16], authors of research papers [5,9] or companies [4]. Commercial services like Open Calais[1] or AlchemyAPI[2] already exploit

[1] `http://www.opencalais.com/documentation/calais-web-service-api/api-metadata/entity-disambiguation`
[2] `http://www.alchemyapi.com/`

these good results. However, given such good results on general purpose knowledge, the question remains whether these results also hold for more specialized domains and when the data is taken from open data sources like the Linked Data Cloud. Moreover the Linked Data Cloud covers a large number of different domains and the stability of the accuracy when multiple domains are contained in one knowledge base remains open.

The type of knowledge base also impacts disambiguation accuracy and stability. In [15] the authors compared disambiguation results achieved by using context information about already disambiguated entities in documents and encyclopedic definitions of entities. We refer to the former as document-centric and to the latter as entity-centric knowledge base. They showed that context information (i.e. the document-centric approach) significantly outperforms encyclopedic definitions of entities (the entity-centric approach). While the result applies to Wikipedia, it remains unresolved whether similar findings can be made on special purpose domains. Stability remains another open question. Which approach provides more stable results when increasing the size and/or heterogeneity of a knowledge base?

In this paper we present an empirical evaluation to tackle those loose ends and to investigate disambiguation performance across domains and for huge knowledge bases, as it is the case with using Linked Data Sources. More specifically, we ask the following four questions and answer them by providing an in-depth evaluation of standard disambiguation approaches:

- **Representation of a knowledge base:** What influence does the type of knowledge base, i.e. document-centric have on the disambiguation performance?
- **Size:** Does disambiguation accuracy remain stable with increasing knowledge base size or heterogeneity?
- **Domain heterogeneity:** How dependent is the accuracy of disambiguation systems on the domain heterogeneity of a knowledge base?
- **Quality:** How does the data quality of Linked Data resources influence disambiguation accuracy?

We evaluate these questions using standard Vector Space based approaches for the entity-centric and text classification based approaches (i.e. K-NN) on document-centric knowledge bases. Our results indicate that, under certain assumptions, document-centric knowledge bases outperform entity-centric ones while also being more stable. Moreover, the size of a document-centric knowledge base degenerates disambiguation accuracy less than in the case of an entity-centric knowledge base. Also, domain heterogeneity plays a crucial role similar to the quality of the Linked Data sources the knowledge base has been created from. Overall, the results suggest that textual documents annotated with Linked Data Resources provide a better source for disambiguation algorithms than encyclopedic, entity-centric sources.

The rest of the paper is organized as follows. In section 2, we explain two approaches for achieving knowledge bases for entity disambiguation and describe our algorithms. In section 3, we analyze the biomedical data set CALBC, which constitutes an optimal foundation for our experiments. Section 4 presents experiments with different knowledge bases and settings in form of an in-depth evaluation to answer our questions, and section 5 concludes our paper.

2. APPROACH

Ambiguity in a text or name can arise from variations in how an entity may be referenced, from the existence of several entities with the same name or even from spelling mistakes in the name. Our approach assumes a retrieval based approach for disambiguating entities. Given a textual representation of an entity e_i, denoted as t_{e_i}, we return a ranked list R_i of possible entity candidates, i.e.

$$R_i = ranking(Kb, t_{e_i}) \tag{1}$$

Kb denotes the knowledge base containing all available entity candidates. In an interactive setting, a user would choose the correct entity from a list while in a fully automated setting the first entity would be chosen. We differentiate knowledge bases along the kind of data used for disambiguation, namely an entity-centric knowledge base (e.g. databases, ontologies, Linked Data) or a document-centric knowledge base (annotated text corpora). Entity-centric knowledge bases describe every entity through a well-defined schema (e.g. database, ontology). Properties that could be exploited for disambiguation tasks are part of such schemes, like for example the name of an entity or a general, human readable description. Annotated documents form a usage centered kind of knowledge base, where entities are described by their usage in text. There is no explicit schema or ontological description besides the annotation that a particular textual representation belongs to an entity (t_{e_i} in our definition). In literature these textual representations are also denoted as surface forms of an entity. In the context of our work we assume that documents have been wealthy annotated with entities on the word or phrase level.

More formally, we define an entity-centric knowledge base as

$$Kb_{\text{ent}} = \{e_0, ..., e_n | e_i \in E, n \in \mathbb{N}\} \tag{2}$$

and a document-centric knowledge base as

$$Kb_{\text{doc}} = \{d_0, ..., d_n | d_i \in D, n \in \mathbb{N}\} \tag{3}$$

The sum of all entities available in Kb_{ent} is denoted as E and the sum of all documents in Kb_{doc} is denoted as D. The variables e_i and d_i represent a specific entity or document as knowledge base entry. Basically, an entity entry $e_i \in \text{Kb}_{\text{ent}}$ must contain a primary key ID which represents a unique identifier as well as some describing information, i.e. attributes, in a variable amount of k fields. Formally we denote such an entity as

$$e_{\text{i}} = (ID, Field_1, ..., Field_k) \tag{4}$$

A document entry d_i consists of its document content representing a text string and a list of annotations of surface forms, i.e. $t^l_{e_i}$, where l denotes the $l-th$ annotation in the document. Each surface form is described by its beginning and ending defining the position in the document and a list of entity references. Formally we denote an entry in the document-centric knowledge base as

$$d_{\text{i}} = (Content, \{(Beginning, Ending, \{ID\}), ...\}) \tag{5}$$

Clearly, we need different disambiguation algorithms to treat both knowledge bases. Our two algorithms that form the basis for the experiments will be described in the following sections.

2.1 Disambiguation with an Entity-Centric Knowledge Base

Entity disambiguation with an entity-centric knowledge base requires a database containing explicit entries for each entity. In our entity-centric knowledge base, fields (i.e. attributes) of an entity include title and description, a link which represents the Semantic Web URL of the entity as well as all known surface forms including synonyms. All known surface forms for an entity which are referenced to their corresponding entity and the respective amount of occurrences with this surface form are stored in *Occurrences* (cf. equation 6). We collect these information by analyzing documents which already offer disambiguated surface forms.

$$e_{\mathrm{i}} = (ID, Name, Description, Occurrences, Link) \quad (6)$$

Note that we do not consider the context of a surface form, i.e. the text surrounding an entity.

For disambiguation, we utilize an information retrieval based approach: Given a surface form t_{e_i} and a text of length W surrounding the surface form, denoted as a set of words $C = \{c_1, \ldots, c_W\}$, we deliver a ranked list of candidate entities. Hence, we define an attribute-based ranking function between a given text t and a knowledge base entity as $s(t, e_i^a)$. We denote the use of a particular field of an entity in this superscript, e.g. e_i^a defines attribute a of entity i. More specifically, our approach leaves it up to choose the Vector Space Model [20] with TF-IDF [19] weights or a probabilistic model like Okapi BM25 [7]. However, we compare surface form t_{e_i} with field *Name* (n) and surrounding text c with field *Description* (d) of entities in our knowledge base:

$$Score_{e_i} = s(t_{e_i}, e_i^n) + \sum_{j=0}^{W} s(c_j, e_i^d) \quad (7)$$

Entities whose titles do not match with the surface form do not appear in the result list. After scoring all entities the ranked Top-N candidates constitute the disambiguation result R.

2.2 Disambiguation with a Document-Centric Knowledge Base

Our document-centric knowledge base contains annotated documents. An annotation may consist of multiple references to entities from different domains or namespaces. We subdivide the content of a document d_i into the document title and the document title combined with the document text in order to increase the recall of the system. Furthermore, all disambiguated surface forms are stored in the field *Keywords*, while the field *Entities* contains all identifiers of referenced entities in a document. Additionally, *ID* depicts a unique document identifier. The structure of our document-centric knowledge base is denoted as follows:

$$d_i = (ID, Title, Titleandtext, Entities, Keyword) \quad (8)$$

The disambiguation algorithm is similar to a K-Nearest-Neighbor classification (K-NN) using majority voting. First, we select relevant documents and second, we count their containing entity references. The first task is similar to the approach using an entity-centric knowledge base. We query the T most relevant documents concerning the surface form and surrounding text (similar to equation 7, but querying the fields *Title* and *Titleandtext* instead). Again we require the existence of a similar surface form being present in a document to reduce noise in the retrieved results. The classification task entails counting the appearances of all referenced entities K in our document set T. The parameter T influences the overall results and must be determined empirically (cf. section 4.1). As in K-NN based classification results, high K, which result of a high amount of documents T, are more robust against outliers but are more sensitive to the entity balance while low K's are less robust against outliers and computationally more efficient [10]. Consequently, the result list R consists of the N most appearing entities in K. In contrast to disambiguation with an entity-centric knowledge base, the result quality strongly depends on the amount of annotated disambiguated surface forms. For that reason a bootstrapping process with an entity-centric knowledge base and an integration of a document-centric knowledge base afterwards might be worthwhile.

3. DATA SET

Most disambiguation approaches in literature have been evaluated on general purpose knowledge bases like Wikipedia. Wikipedia contains a large number of locations, persons and organizations. Due to its rich and diverse feature set it allows an investigation on a broad range of approaches. However, in order to investigate our questions introduced in the introduction, we intend to use a more specific purpose knowledge base. Therefore we choose the CALBC[3] (Collaborative Annotation of a Large Biomedical Corpus) which depicts a very large, community-wide shared text corpus annotated with biomedical entity references [8]. CALBC represents a silver standard corpus which results from the harmonization of automatically provided annotations. The data set was released in 3 differently sized corpora: small, big and pilot. We use the small (CALBCSmall) and big (CALBCBig) corpora which contain 174.999 and 714.282 Medline abstracts. The occurring entities in document titles and abstracts are annotated by using element <*e*> that encloses the surface form where at least one entity exists. All entities are identified using the id attribute in element <*e*>. Example 9 shows an annotation of the surface form "H1N1".

$$\text{<e id="UMLS:C1615607:T005:diso>H1N1</e>} \quad (9)$$

An entity identifier referencing the corresponding entity resource is composed of the namespace (UMLS) of the knowledge source, the identifier (C1615607) of the entity in its corresponding source, the semantic type (T005) as well as the semantic group (diso). The semantic group can be seen as a first level classification of an entity. According to [13] semantic types represent intensional, or definitional knowledge. Table 1 displays basic statistics about CALBCSmall and CALBCBig, whereby both corpora are disjunct in terms of their appearing documents. Although the amount of non-distinct entity annotations is more than two times higher than in CALBCSmall, CALBCBig provides less distinct entity references. Additionally, it is important to mention that in contrast to other disambiguation corpora like Wikipedia, an annotation in CALBC may comprise more than one entity annotation. A rich taxonomy and classification system is responsible for 9 entity annotations on average per surface form. It is not ensured that these entities can be connected

[3] `http://www.calbc.eu/`

Table 1: Statistics of CALBCSmall and CALBCBig

	CALBCSmall	CALBCBig
Documents	174.999	714.282
Surface Forms	2.548.900	10.304.172
Unique Surface Forms	50.725	101.439
Entities	37.309.221	96.526.575
Unique Entities	453.352	308.644
Used Unique Entities	265.532	228.744
Namespaces	14	16

via a "same as" relation. In our work we asses this behavior with the possibility of having more valid disambiguation solutions per surface form.

Because some namespaces are not publicly available, we did not consider those entities during the parsing process. Instead, we focus on the four major namespaces UMLS [4], Disease (is contained in UMLS), Uniprot [5] and EntrezGene [6] which constitute a majority of annotated entities in both CALBC data sets. The UMLS dataset is a combination of many health and biomedical vocabularies, whereas Uniprot provides high-quality resources of protein sequences and function information and EntrezGene exclusively comprises gene-specific information.

In order to estimate the relevant disambiguation properties of the CALBC, we analyzed the distribution of surface forms and their corresponding entities (see figure 1). The histogram axis showing the number of entities is truncated at 40 entities due to very few existing surface forms which contain a lot of different meanings (maximum 9895). We found that about half of all existing surface forms may attain between 2 and 7 different entities. The other half provides up to 9895 different entity meanings which makes disambiguation particularly difficult because a term may attain far more different meanings than the average of entity annotations per surface form. In summary, it can be stated that it is worth using CALBC for disambiguation purposes due to its huge amount of rich annotated surface forms, which partially have a wide range of different meanings. To populate our entity-centric knowledge base we crawled the knowledge sources providing the entity annotations. For all knowledge sources we downloaded the RDF dump and extracted the required information. Unfortunately, entities of the Uniprot namespace do not contain a general description. Instead, we use the functional principle as protein description which mostly offers detailed information about the entity. In order to exploit other domain-specific properties, domain experts are required.

4. EXPERIMENTS

Our disambiguation approaches are implemented in Java with all queries being executed with Apache Lucene 4.2[7]. Our experiments assess the disambiguation approaches with an entity-centric and document-centric knowledge base, quality aspects regarding the entity-centric knowledge base which consists of data from the Semantic Web as well as disambiguation accuracy after increasing the knowledge base size.

[4] http://www.nlm.nih.gov/research/umls/
[5] http://www.uniprot.org
[6] http://www.ncbi.nlm.nih.gov/gene
[7] http://lucene.apache.org/

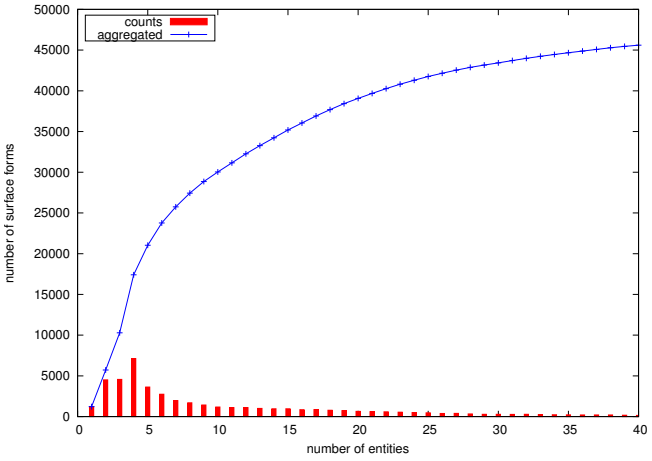

Figure 1: Distribution of surface forms and their corresponding entities

The CALBC data set provides a wealth of annotated entities that can be used to evaluate the approaches. We randomly selected 15000 entity samples and set them aside as test data. When using different filters for entity selection, for example a description filter which only selects entities with specific description properties, the candidate set varies respectively. Disambiguation queries can be subdivided into a surface form query and context query. In contrast to the context query, a document has to match with the surface form query to appear in the result list. Our single results are described by a set of comprehensive measures, including mean reciprocal rank (MRR), recall and mean average precision (MAP). Reciprocal rank is the multiplicative inverse of the rank of the first correct result in a result set. Average precision denotes the average of all precision values calculated at each correct hit in the result list [12]. Similar to search engines, correct evaluation results should appear at the top of the result list. For this very reason a high reciprocal rank in combination with a strong recall are desirable. On the other hand, we relinquish the usage of the precision measure because a fixed amount of results is returned by our disambiguation system. Instead, the MAP only computes the precision at each correct hit in the result list and is influenced in a sustained manner by the other two measures.

4.1 Parameters

Both disambiguation approaches offer various parameter settings to tweak the overall results. We focus on evaluating generic parameters instead of specific algorithm settings or thresholds to show correlation between the result set and disambiguation parameter settings in general. Table 2 shows on overview of our chosen parameters with their corresponding values. The adjustment *Context Length* affects the number of words in both directions, before and after the corresponding surface form. Our parameter *Query* selects the consideration of the context of the respective surface form to show its significant influence. In this context we investigate the difference between term and fuzzy queries using the surface form (*SF query*) and context (*Context query*). Fuzzy queries match terms with a maximum edit distance of size 2 and regard different spellings and possible typing errors which

might occur in all kind of documents. Additionally, the standard information retrieval measures TF-IDF and BM25 are compared (*Similarity*). It must be noted that Lucene's default TF-IDF score also takes internal parameters like term boosting and coordination factor into account, which may influence the result set slightly[8]. The option *Retrieved Results* influences the amount of retrieved results and *Top-T* denotes a major parameter during disambiguation with a document-centric knowledge base only.

Table 2: Settings of evaluation parameters

Parameter	Values
Context length	35, 70, 170, 350
Query	Surface Form, Surface Form & Description
SF query	Fuzzy Query, Term Query
Context query	Fuzzy Query, Term Query
Similarity	TF-IDF, BM25
Retrieved results	5, 10, 20, 50
Top-T (Kb_{doc} only)	20, 50, 100, 200

4.2 Entity-Centric vs. Document-Centric Knowledge Base Disambiguation

With the presence of a detailed parameter notation we are interested in the influence of parameters on the disambiguation results with an entity-centric and document-centric knowledge base whereby our focus was not to develop a new disambiguation algorithm. Due to an enormous amount of analyzed parameter combination (256 with an entity-centric and 1024 with an document-centric knowledge base) we refrain from discussing every single result, but emphasize the most important and noteworthy statistics. Additionally, we compare our results with a baseline which replaces missing disambiguation results using the CALBC dataset provided by other works. The results of document-centric knowledge base disambiguation are compared with outcomes resulting from a prior which estimates the probability of seeing an entity with a surface form [14]. On the other hand a term query of the surface form which has to match with the entity name serves as baseline in the entity-centric approach. Table 3 shows our findings in contrast to the baseline results. The column *Settings* consists of the following three printed parameters: *SF Query*, *ContextQuery* and *Similarity*. Due to an unoptimized disambiguation service, poor results are generally noticeable.

Parameter study entity-centric knowledge base

A comparison of TF-IDF and BM25 function indicates an increasing difference between them as soon as the overall results get better. Generally, TF-IDF shows better results in all experiments. The main difference is constituted by the distinction between term and fuzzy query. Combining a term query with the respective surface form results in very poor results considering all measures. Even the usage of stemming algorithms (i.e. Porter-Stemmer [17]) does not improve the results. After switching to fuzzy query the situation is different: Attaining better results in all measures means, that entity names mostly vary in their notation. However, changing query properties from term to fuzzy query when querying the context does not affect the result at all.

Parameter study document-centric knowledge base

The usage of an document-centric knowledge base which does not make use of information extracted from the Semantic Web obtains much better results. The amount of used documents to rank the entities (*Top-T*) is constrained to 100 due to decreasing accuracy if we increase the document count. Experiments show an average recall value in the range of 65 and 75 percent and mean average precision values between 50 and 60 percent reveal that a utilization of a document-centric knowledge base is worthwhile. In contrast to the other approach, BM25 similarity does not drop behind TF-IDF similarity. Instead, results do not show any significant differences. The combination term and fuzzy query with a surface form features a difference of 6 to 8 percent in all measures, but this time a term query provides better results. Term queries top fuzzy queries due to similar appearances of terms in the document-centric knowledge base. Again, the difference of term and fuzzy queries when querying the context is not noteworthy and negligible.

Baseline comparisons

Compared to our approaches both baselines show inferior results. Reviewing the measures of the entity-centric approach with its best parameter settings and the baseline shows a difference up to 21 percent. Comparing the baseline with the entity-centric approach using a term query when querying the surface form, the differences decrease significantly.
On closer consideration of the baseline concerning the approach with a document-centric knowledge base, we state comparatively good results which exceed all those of the entity-centric approach. Nevertheless, the baseline results are far weaker than the results attained with the document-centric knowledge base approach (difference up to 22 percent). Basically, both knowledge base approaches clearly outperform their corresponding baselines.

Parameters which were not taken into account in the sections, like context length, do not provide interesting information. A context length which exceeds 50 words on the left and right side of the surface form introduces additional noise in the result set in both approaches. The parameter *Query* is analyzed in detail in section 4.3. Considering the sensitiveness of the approaches to the selection of their values, both approaches do not suffer under poor parameter settings notably. Slight changes in parameter settings result in slightly decreasing results.

In summary, it can be stated that in our evaluation with the CALBC data set a document-centric knowledge base significantly outperforms an entity-centric knowledge base. Entity resources of the CALBC dataset do not provide additional useful information which could be exploited for disambiguation (i.e. interlinks between entity resources). Depending on the data set and its features, results with an entity-centric knowledge base may vary significantly. In the following, we investigate the quality of the knowledge base to asses the results of both approaches in more detail.

[8] `http://lucene.apache.org/core/4\_2\_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html`

Table 3: Disambiguation results with an entity-centric and document-centric knowledge base

Settings (SF Query/Context Query/Sim.)	MRR in %		Recall in %		MAP in %	
	Kb_{ent}	Kb_{doc}	Kb_{ent}	Kb_{doc}	Kb_{ent}	Kb_{doc}
Baseline	14.7	55.8	10.2	52.3	6.9	50.1
Term / Term / TF-IDF	18.4	77.3	12.0	73.9	8.6	58.1
Term / Term / BM-25	17.0	77.8	11.5	74.2	8.1	58.7
Term / Fuzzy / TF-IDF	18.2	77.4	12.1	73.1	8.6	57.8
Term / Fuzzy / BM-25	17.3	77.7	11.8	73.4	8.1	58.3
Fuzzy / Term / TF-IDF	35.4	71.7	31.1	68.9	20.9	53.9
Fuzzy / Term / BM-25	29.2	72.3	28.4	69.1	17.7	54.8
Fuzzy / Fuzzy / TF-IDF	35.4	68.8	31.8	66.4	21.0	50.9
Fuzzy / Fuzzy / BM-25	29.3	69.8	28.9	66.9	17.7	52.2

Table 4: Disambiguation results when using quality levels (i.e. description length)

Settings	MRR in %	Recall in %	MAP in %	Fraction in %
No Desc.	16.6	22.6	14.0	49.1
$Desc._{length > 0}$	37.5	37.4	25.6	50.9
$Desc._{length > 100}$	38.3	40.7	29.2	42.6
$Desc._{length > 200}$	40.0	44.5	32.3	28.9
$Desc._{length > 800}$	42.7	49.2	39.4	4.8

Table 5: Disambiguation results on UMLS, Uniprot, EntrezGene and Disease namespaces

Settings	MRR in %	Recall in %	MAP in %	Desc. in %	#Ent.
All	35.4	31.1	20.9	50.9	265532
UMLS	35.0	30.8	20.4	41.7	48774
Uniprot	7.7	6.7	4.1	83.1	126472
EntrezGene	15.0	20.9	13.6	7.3	83485
Disease	42.8	49.6	35.1	53.9	6801

4.3 Quality Criteria for Disambiguation

Entity disambiguation needs special quality aspects regarding well described entities in their corresponding knowledge base to deliver convincing results. In this context, we want to explore the quality of our entity data which is extracted from the Semantic Web and is integrated in our entity-centric knowledge base. Quantifying the quality of a knowledge base entry is not an easy task because, in addition to general properties like description length, the utilization of specific nouns and verbs also plays an important role.

However, our entities partially suffer from a lack of suitable descriptions. Table 4 illustrates the results when a candidate selection, whose filter selects those entities which feature special description properties, is integrated. In our case we distinguish between a set of entities which are only described by their names and those whose description contains more than 0, 100, 200 and 800 characters. The column *Fraction* represents the size of the candidate set in relation to our default knowledge base with 265532 entities. Only 50 percent of all entities that can be found in our index offer descriptions, which mainly leads to poor results. Even when exhibiting a description, one cannot automatically assume that results improve significantly. It is also important that the text consists of an adequate length. We measure the length by counting the amount of characters. But we cannot imply that entities containing long description are easy to disambiguate because the existence of specific keywords, which are especially useful for describing the entity, is also necessary. The fraction of our entities with a detailed characterization is rather low. This portends a low quality knowledge base. It is necessary to observe that all results in table 4 also depend on the quality of each namespace. Namespaces may feature different quality properties and may downgrade the results.

4.4 Quality of Namespaces

After considering disambiguation with entities showing specific properties, we investigate the quality of single namespaces. For this purpose we created four disjunct knowledge bases which only contain entities from one of the following namespaces: UMLS, Uniprot, EntrezGene and Disease. Table 5 shows the disambiguation results in combination with the fraction of entities providing a description (*Desc.*) and the amount of entities (*#Ent.*) in the corresponding knowledge base. Uniprot and EntrezGene namespaces constitute the majority of all entities, whereas EntrezGene entries suffer from a lack of representative descriptions. However, entities belonging to Uniprot namespace or to the minority group Disease, contain the most descriptions. It is noticeable that a high amount of available descriptions does not automatically provide convincing disambiguation results. Uniprot evaluation shows that offering a lot of descriptions does not automatically conduce the results. Instead, the wording plays an important role due to the fact that a description needs to match with the context of a surface form. Obviously, the wording of the Uniprot descriptions is not suitable for disambiguation. A lot of entities which are only described by their names are responsible for poor results in the EntrezGene evaluation. On the other hand the UMLS and especially the Disease namespace provide more satisfying results. All results can be improved further by specific algorithm adaptions.

As a summary, we can say that the quality of our disambiguation results is distinguished from the use of different namespaces and domains. Only the availability and high quality of entity resources ensuresensures convincing disambiguation results. Unfortunately, both aspects are often not sufficiently available when data is extracted from the Semantic Web.

Table 6: Results after increasing our knowledge base with different corpora

Experiment	Integrated Knowledge Bases	MRR in %	Recall in %	MAP in %	#Entities	Change in %
$Kb_{ent,\ intra}$	-	35.4	31.1	20.9	265532	-
$Kb_{ent,\ intra}$	UMLS	29.8	26.6	17.5	2098824	-15.6
$Kb_{ent,\ intra}$	UMLS, Uniprot	29.8	26.4	17.4	32407960	-15.9
$Kb_{ent,\ inter}$	Wikipedia	21.1	22.2	12.0	4643509	-37.2
$Kb_{ent,\ inter}$	UMLS, Uniprot, Wikipedia	18.1	19.9	10.4	36785937	-45.1
$Kb_{doc,\ intra}$	-	71.7	68.9	53.9	174999	-
$Kb_{doc,\ intra}$	CALBCBig	72.2	69.7	54.4	889282	+0.9

4.5 Influence of Knowledge Base Size

In this section, our main purpose is to investigate the development of results when increasing the size of our knowledge bases. Increasing the size can be done within the same domain or across domains. We refer to the former as intra-domain and to the latter as inter-domain. We talk about an intra-specific domain extension if the knowledge base is extended with entities or documents from the same domain (e.g. combining UMLS with Uniprot), otherwise it is an inter-specific domain extension (e.g. combining UMLS with Wikipedia). For this purpose we enrich our explicit knowledge base Kb_{ent}, consisting of entities which are annotated in CALBCSmall, with all entities from UMLS, Uniprot and the more general purpose knowledge base Wikipedia. All documents which are contained by CALBCBig are added to the index of the document-centric approach. Due to a lack of suitable document corpora, we do not evaluate an inter-specific domain extension when using a document-centric approach.

Table 6 displays the corresponding results and the amount of entities or documents which are stored in the respective knowledge base. The column *Change* expresses the changing of the result values. Therefor we average our measures MRR, Recall and MAP. Generally, an increase of the entity amount is responsible for worse results in an entity-centric knowledge base. Results are significantly weak if foreign domain entities from Wikipedia are appended which demonstrates that domain heterogeneity in a knowledge base plays an important role. However, it is noticeable that the adding of 30 million entities of the Uniprot corpus does not affect the results. This can be explained by the existence of unique entity names only in the Uniprot knowledge base. Only very few surface forms, which reference an entity belonging to the Uniprot namepsace, coincide with the corresponding entity names. The notation differs in the knowledge base because gene names underlie a complex naming with consecutive gene numbers. Due to a required surface form matching these specific names prevent Uniprot entities to appear as false positive disambiguation results.

Surprisingly, when using a document-centric knowledge base the results do not suffer from an addition of documents. Instead of a decreasing performance, our evaluation shows a slight improvement of all three measures. Although we made no inter-specific domain extension, the results show that this approach is robust against a high amount of documents in the data base.

5. DISCUSSION AND CONCLUSIONS

In this paper, we demonstrated that document-centric knowledge bases outperform laboriously constructed entity-centric knowledge bases if an adequate amount of annotations is available. Nevertheless, the disambiguation results, when using entity-centric knowledge bases, strongly depend on the underlying dataset and its exploitable features. Additionally, our experiments show that in contrast to entity-based knowledge bases the results attained with a document-centric knowledge base are robust against an increase of dataset entries. Also, it transpired that domain heterogeneity plays a crucial role. Domain-foreign data sets within a knowledge base degenerate disambiguation accuracy significantly. Finally, results strongly depend on the quality of the Linked Data sources the knowledge bases have been created from.

In terms of limitations, our results are constrained by the use of standard approaches that exclude the consideration of semantic relationships like taxonomies or part-of-relationships. Several approaches which are designed for entity-centric knowledge bases are reported to be able to output high quality results utilizing machine learning techniques like in the work of [11], [2] and [21]. We did not use machine-learning techniques in order to optimize the disambiguation process. However, as results from other approaches show, the Vector Space Model serves as a good baseline and hence we see our results as an accurate estimation.

For the future, we consider integrating machine learning algorithms to improve disambiguation accuracy significantly. In this context the integration of several knowledge bases, each containing domain-specific data, might be worthwhile. Weighting the results of each knowledge base query by machine learning algorithms could circumvent the decrease of accuracy if domain heterogeneity occurs in knowledge bases. A first integration of learn to rank shows very satisfying results in document-centric and entity-centric approaches. Additionally, it is worth considering integrating our approach in a user interface where users are able to disambiguate arbitrary terms with an automatic user correction subsequently.

6. ACKNOWLEDGMENTS

The presented work was developed within the CODE project funded by the EU Seventh Framework Programme, grant agreement number 296150.

7. REFERENCES

[1] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*, volume 6, pages 9–16, 2006.

[2] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 708–716, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[3] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 277–285. Association for Computational Linguistics, 2010.

[4] A. L. Gentile, Z. Zhang, L. Xia, and J. Iria. Graph-based Semantic Relatedness for Named Entity Disambiguation. In *1st International Conference on Software, Services and Semantic Technologies*, 2009.

[5] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '04, pages 296–305, New York, NY, USA, 2004. ACM.

[6] X. Han and J. Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 215–224, New York, NY, USA, 2009. ACM.

[7] K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Inf. Process. Manage.*, 36(6), 2000.

[8] S. Kafkas, I. Lewin, D. Milward, E. van Mulligen, J. Kors, U. Hahn, and D. Rebholz-Schuhmann. Calbc: Releasing the final corpora. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, May 2012.

[9] R. Kern, M. Zechner, and M. Granitzer. Model selection strategies for author disambiguation. In *Database and Expert Systems Applications (DEXA), 2011 22nd International Workshop on*, pages 155–159. IEEE, 2011.

[10] T.-Y. Liu. *Learning to Rank for Information Retrieval*. Springer, 2011.

[11] A. Luberg, M. Granitzer, H. Wu, P. Järv, and T. Tammet. Information retrieval and deduplication for tourism recommender sightsplanner. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, page 50. ACM, 2012.

[12] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[13] A. McCray, A. Burgun, and O. Bodenreider. Aggregating umls semantic types for reducing conceptual complexity. *Proceedings of Medinfo*, 10(pt 1):216–20, 2001.

[14] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 1–8, New York, NY, USA, 2011. ACM.

[15] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.

[16] Y. Peng, D. He, and M. Mao. Geographic named entity disambiguation with automatic profile generation. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '06, pages 522–525, Washington, DC, USA, 2006. IEEE Computer Society.

[17] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[18] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, 2011.

[19] G. Salton. *Automatic Information Organization and Retrieval*. McGraw Hill Text, 1968.

[20] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, Nov. 1975.

[21] J. Wang, G. Li, J. X. Yu, and J. Feng. Entity matching: How similar is similar. *PVLDB*, 4(10):622–633, 2011.