Named Entity Disambiguation for little known referents: a topic-based approach

Andrea Glaser and Jonas Kuhn
Institute for Natural Language Processing
University of Stuttgart, Germany
andrea.glaser@ims.uni-stuttgart.de

Abstract

We propose an approach to Named Entity Disambiguation that avoids a problem of standard work on the task (likewise affecting fully supervised, weakly supervised, or distantly supervised machine learning techniques): the treatment of name mentions referring to people with no (or very little) coverage in the textual training data is systematically incorrect. We propose to indirectly take into account the property information for the "non-prominent" name bearers, such as nationality and profession (e.g., for a Canadian law professor named *Michael Jackson*, with no Wikipedia article, it is very hard to obtain reliable textual training data). The target property information for the entities is directly available from name authority files, or inferrable, e.g., from listings of sportspeople etc. Our proposed approach employs topic modeling to exploit textual training data based on entities sharing the relevant properties. In experiments with a pilot implementation of the general approach, we show that the approach does indeed work well for name/referent pairs with limited textual coverage in the training data.

1 Introduction

A central subtask for complex information retrieval and natural language understanding problems lies in the determination of what are the real-world entities that the proper names in a text refer to. While for some proper names (e.g., *Henry VIII of England*), the correct referent can be uniquely determined, the great majority of name mentions requires disambiguation. For instance, the name *Michael Jackson* can refer to the famous American singer, a British writer and beer expert, a Canadian actor, and many other people.

The corresponding technical Natural Language Processing (NLP) Task, which is known by various names – Named Entity Disambiguation (e.g., Hoffart et al. (2011)), Entity Linking (e.g., Han et al. (2011)), or Wikification (Mihalcea and Csomai, 2007) – is typically construed as determining for each textual mention of a proper name, which of (typically) several entries in a knowledge base (such as DBpedia or Wikipedia) representing unique referents is the correct one in the given context.

The standard approach to this task is to view it as a supervised classification problem, i.e., training data of textual mentions labeled with the correct disambiguation target are used to induce knowledge about indicative contextual clues for each candidate. A considerable amount of research has gone into the development of effective models (Mann and Yarowsky, 2003; Malin, 2005; Bollegala et al., 2006; Chen and Martin, 2007), and particularly into weakly supervised or distant supervision techniques, i.e., finding ways of exploiting explicit or implicit indications for the correct name references in real-life data (Cohen, 2005; Bunescu and Pasca, 2006; Cucerzan, 2007; Mihalcea and Csomai, 2007; Han et al., 2011; Hoffart et al., 2011). Indeed, for medium-to-high frequency name/referent combinations, it is not hard to harvest the web for suitable training material.

The contribution that we present in this paper is motivated by a type of name/referent pairs that falls outside of the standard training scenario and has so far received little attention from the research com-

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 1481–1492, Osaka, Japan, December 11-17 2016.

Entity	Wikipedia
American singer	✓
British writer	✓
Canadian actor	✓
...	
Canadian law professor	×

Table 1: Examples for *Michael Jackson*.

munity[1]: besides the "prominent" bearers of a name, there are almost always others for whom little or no training material can be found by web harvesting, and who will often not even appear in the major reference knowledge bases (DBpedia, Wikipedia etc.). Table 1 shows examples of entities with the proper name *Michael Jackson* and whether or not they have a Wikipedia article. Persons need to be "worthy of notice" to be included in Wikipedia. There is a *Michael Jackson* who is a Canadian law professor, but is not among the 35 or so Michael Jacksons with their own Wikipedia articles. Still, the reference for many of these people *could* be uniquely identified by the name authority files curated by national libraries (e.g., the German Integrated Authority File; "Gemeinsame Normdatei") or other lists that provide unique identification from a specific application perspective, such as staff lists on institutional websites or listings of sportspeople.

Under the established approaches, a trained Named Entity Disambiguation system will incorrectly map those mentions which actually refer to the less known name bearers to the contextually most similar prominent person. In a standard evaluation, these false positives are often negligible since the "non-prominent" mentions are orders of magnitude less frequent. In practice however, any system that is systematically missing out on theoretically identifiable people is problematic (e.g., search engines and Question Answering (QA) systems would return more unwanted or incorrect results, or not find results about the person at all because of an incorrect mapping to a similar, more well-known person).

The purpose of this paper is (i) to propose a general approach for dealing with these cases systematically, and (ii) to present a pilot implementation of this idea using topic modeling. For evaluation, we "simulate" the situation of non-prominent name bearers by leaving documents about them out of the actual training data; this allows us to perform comparative experiments on a manageable collection. As test data we actually use texts containing explicit links to Wikipedia entries, which allows us to circumvent costly manual annotation of the name disambiguation (we call this resource our "silver standard" corpus).

The approach rests on the idea that for non-prominent name bearers, for which no or very little training material containing real mentions is available, we will nevertheless know some characteristic properties – namely, the ones mentioned in a name authority file (typically profession and nationality, as well as age and place of birth, plus possibly institutional affiliation), or similarly properties derivable from other listings. So, while we have no textual training data for the specific person (say, the Canadian law professor *Michael Jackson*), we can use some aggregate of the textual material about different people with the same properties as a proxy (i.e., mentions of all Canadians, of all law scholars, professors etc.).

For our pilot implementation, we conducted several experiments with different parameters (context size around the proper name, number of topics, different corpora) to analyze and evaluate how to the prediction quality is affected. Our results indicate that our approach is indeed very helpful for disambiguating (i) entities for which not much training material is available, and (ii) also for entities with little surrounding context, both of which are useful for many applications.

In Section 2 we discuss related work. In Section 3 we describe our approach and how we extract properties and apply them to new unknown proper names in a document. Section 4 presents the data we use and describes how we create a silver standard corpus for evaluating our system. In Section 5 we describe our experiments and discuss our experimental results and the effects of different parameters. We give our conclusions and future work plans in Section 6.

[1]Sarmento et al. (2009) address the high skewness of distribution of mentions by developing a scalable approach that can cluster a billion mentions.

2 Related Work

Topic models have been used for named entity disambiguation in previous work. Bhattacharya and Getoor (2006) look at data with authors and try to determine the number of individual authors and link them to their corresponding entities. They do not make pairwise decisions of whether two names refer to the same entity, but look at all names in the collection and make a collective decision. For this they use a Latent Dirichlet Allocation (LDA) Model and introduce a hidden variable that captures relationships between entities. With their work they extend work done by Rosen-Zvi et al. (2004) and propose a solution to the duplicate authors problem. Shu et al. (2009) extend the LDA model to include global information from documents to identify authors. All of these approaches look at author data only, they cannot be applied to people in general which we do in our work.

Some work has been done using topic models on knowledge bases. Zhang et al. (2011) train a topic model on a knowledge base, then incorporate the information as a semantic feature by mapping documents with the name mention to the hidden topic space. They use Wikipedia pages to train their model on the first parts of the pages and to test it on the second parts of the pages. Sen (2012) use topic models to learn context information and relations between co-occurring entities. They train their model on only those Wikipedia articles which describe the entities they are dealing with. Yet, to use their model for other entities they would have to use an expanded version of the knowledge base. This would require the knowledge base to contain information about these entities. Han and Sun (2012) combine context compatibility of a referent entity and its context and topic coherence of the entity and the document's main topics. Kataria et al. (2011) use a hierarchical variant of LDA that incorporates information from Wikipedia (words, annotations, and category information) and have a separate topic for each Wikipedia entry in their model. They report results on precision while we also look at the recall and F_1 score, and compare different parameters.

All of these approaches are limited to the entities that occur in the knowledge bases. To apply the approaches to other entities, the knowledge base would have to be extended or data about these entities would have to be collected otherwise. While our approach also uses Wikipedia as a collection to train some of our models, the information we obtain is independent from specific entities and can be applied to any new entity, even if there is no information about them available in Wikipedia.

Li et al. (2013) use information from Wikipedia and an external source (websites referring to entities in Wikipedia obtained by crawling the web). They conduct experiments on two datasets, TAC-KBP 2009 and twitter data about 25 ambiguous and randomly picked entities, however they filter this data to only include entities that occur in Wikipedia, because their approach is also limited to entities in Wikipedia.

Bamman et al. (2013) extend a topic model to learn character types (e.g., {*dark, major, henchman*} or {*shoot, aim, overpower*}) in movies. In subsequent work, Bamman et al. (2014) apply an extended topic model to learn character types in English novels of the 18th and 19th century. They do not identify and link the individual names to their real world entities.

Topic models have also been used in named entity recognition (NER). Guo et al. (2009) use LDA for NER in query. Ritter et al. (2011) extend LDA and take information from Freebase for NER on twitter data. They have a similar problem with ambiguous expressions which they need to solve to determine the correct class (e.g., *China* can belong to several classes such as LOCATION or PERSON). However, they do not identify the actual real world entity of the expression (e.g., there are several cities called *China* in the US and other countries, but they all belong to the class LOCATION).

3 Approach

Our system does not rely on having textual training data for a specific person (e.g., the Canadian law professor *Michael Jackson*). Instead, it uses some aggregate of the textual material about different people that have the same properties (i.e., mentions of all Canadians, of all law scholars, and all professors etc.). This means that our system is independent from existing training data or obtaining training data through other means (e.g., web scraping), and can be applied to any text without the preliminary step of extracting specific information about the entities in the text.

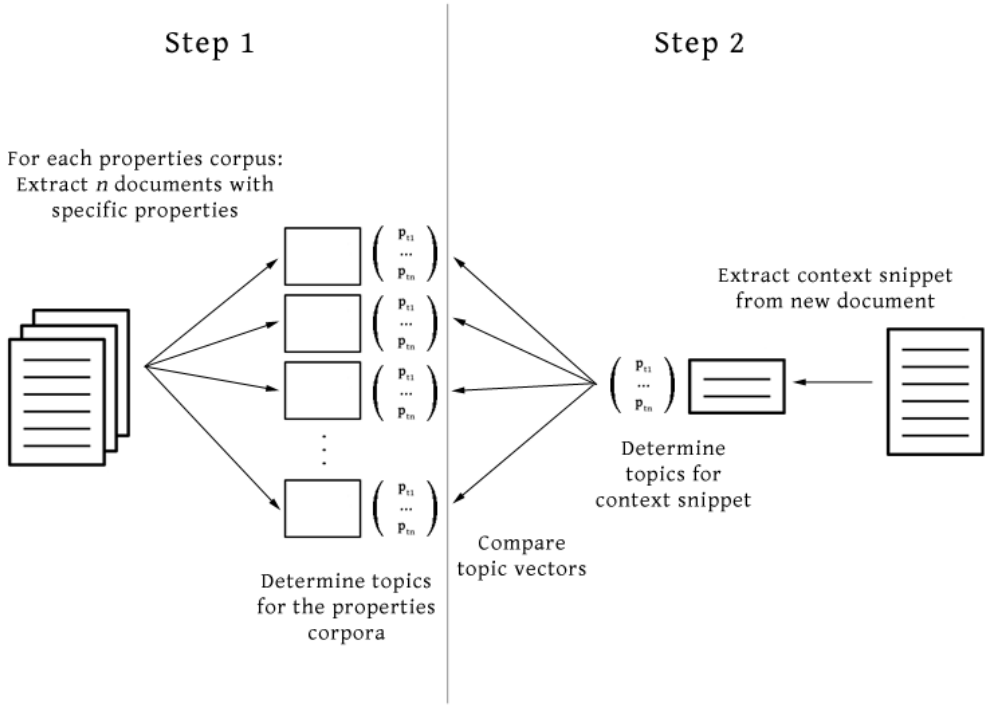


Figure 1: System that learns characteristics of people and uses them to disambiguate unknown people.

Figure 1 shows our system, which is divided into two steps. In the first step, we take a collection of documents and extract documents with specific properties, e.g., documents about singers, authors, and other professions, and documents about Americans, Canadians, and other nationalities. We then concatenate these extracted documents to individual corpora (e.g., a *singer corpora*), which we call "properties corpora". After this, the topics for these properties corpora are determined by using a topic model. In the second step, our system is applied to new unknown proper names. This is done by determining the topics for the proper name based on the chosen context and then comparing these topics with the topics of our properties corpora to find the ones with the most similarity. We describe both steps in more detail in the next two subsections.

3.1 Learning Properties of Persons

In the first step (left side of Figure 1), our system learns characteristic properties that people can have. Properties that can be learned are for example:

- Professions (singer, author, president, tennis player, ...)
- Nationalities (American, Canadian, German, Irish, Japanese, ...)
- Affiliations (university, company, ...)
- Engagements (organization, charity, ...)

In our pilot implementation of the approach we investigate the usefulness of professions and nationalities as properties. We chose these properties because they are the most prominent properties people have and can be obtained without much effort. Our system can be easily expanded with other properties in the future. Helpful properties are ones that are mentioned with the entity, for example, affiliations or engagements.

We first extract documents from a collection (e.g., Wikipedia) with the specific properties. For example, we extract documents about singers, authors, and other professions, and documents about people whose nationality is American, Canadian etc. We then concatenate randomly n of the extracted documents of one category into one corpus. After this we have many small corpora consisting of documents

with certain properties, e.g., a *singer corpus* and an *American corpus*. We call these concatenated documents "properties corpora". These properties corpora are the basis for disambiguating new entities. For example, *Michael Jackson, the American singer* has properties of the corpora *singer* and *American*.

The properties corpora are not very helpful in this form yet because the extracted documents contain a lot of information about many specific people which might not be helpful for disambiguating a new person. To obtain the relevant information from these corpora we use topic models. In natural language processing, topic models can be used to extract and explore topics from a collection of documents. Every topic consists of certain words that characterize this topic. In our work we use Latent Dirichlet Allocation (LDA) (Blei et al., 2003).

Using the topic model consists of two steps. First, we need to train the topic model. For this we use different training collections which we describe in Section 4.1. After the topic model is trained, we apply it to our properties corpora to obtain topic information from them. For example, the singer corpus will have a topic with words related to a singer, e.g., album, music, concert etc. with a high probability given the corpus.

3.2 Using Properties to Disambiguate Persons

In the second step (right side of Figure 1), our system is applied to new unknown proper names. To disambiguate a new person, we need to extract some context around the person (e.g., a sentence or a paragraph). We call the extracted text "context snippet". We use the same trained topic models as in Section 3.1 and apply them to the context snippet to obtain topic information from it.

The topic information we obtain from a document (i.e., a properties corpus or a context snippet) can be represented as a vector of length n, where n is the number of topics used by the topic model. Each entry p_{t_i} in the vector corresponds to the probability of the topic t_i given the document.

We then compare the topic vector of the new person with the topic vectors of each properties corpora to find the corpus that is most similar to the context the person occurs in. For this comparison we use cosine similarity. Let $\mathbf{x}$ be the vector of the new person and $\mathbf{y}_c$ be the vector of the properties corpus c. The cosine similarity between these two vectors is defined as:

$$\cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}_c}{||\mathbf{x}|| \cdot ||\mathbf{y}_c||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \qquad \text{with } \mathbf{x} = \begin{pmatrix} p_{t_1} \\ \vdots \\ p_{t_n} \end{pmatrix} \text{ and } \mathbf{y}_c = \begin{pmatrix} p_{t_1} \\ \vdots \\ p_{t_n} \end{pmatrix}$$

where p_{t_i} is the probability of topic t_i given the document.

4 Data

We use two collections in this work: (i) the English Wikipedia[2] (February 2014 version) and (ii) the English Gigaword corpus (Graff and Cieri, 2003).

4.1 Corpora for Training Topic Models

We have trained different topic models on different collections and parts of collections to study the effects of different sources.

- **Wikipedia - all ($\mathbf{W}_{all}$):** Wikipedia is an internet encyclopedia which provides a great variety of articles. For this model we used all articles without any restrictions.
- **Wikipedia - living people ($\mathbf{W}_{lp}$):** Wikipedia articles are classified into different categories. We built a model that only uses articles from the category *living people* to see if the properties we can learn from articles about people are more helpful for identifying new people than the properties we can learn from the entire Wikipedia.

[2]wiki dump enwiki-20140203

- **Wikipedia - living people - individual sections (W_{lps}):** Many Wikipedia articles are very long and separated into several sections that are often very different with respect to their topics (e.g., *early life, career*). We want to investigate if we can obtain more helpful topics by using individual sections that are about specific topics as opposed to taking entire articles that consist of many different topics.
- **English Gigaword - nyt (G_{nyt}):** The English Gigaword corpus consists of newswire text data in English. With this model we want to analyze if a newswire corpus provides different topics than an encyclopedia. We only use one part of the English Gigaword corpus for this model, newswire data from the source *The New York Times Newswire Service* (nyt).

		Topics					
Corpus	Docs	100	1000	2500	5000	7500	10000
W_{all}	4.3m	✓	✓	×	×	×	×
W_{lp}	650k	✓	✓	✓	✓	✓	✓
W_{lps}	1.7m	✓	✓	×	×	×	×
G_{nyt}	1.3m	✓	✓	✓	×	×	×

Table 2: Used models.

We experiment with different numbers of topics for each of these collections, ranging from 100 (more coarse-grained topics) to 10000 (more fine-grained topics). Table 2 shows the numbers of topics we used for the different collections. We restrict the larger collections to a maximum of 1000 topics (Wikipedia) and 2500 topics (English Gigaword) due to efficiency reasons. The approximate number of used documents for each collection is listed in the second column of Table 2.

By experimenting with different numbers of topics we want to investigate if more fine-grained topics will give us better results when disambiguating new people, or if a smaller number of topics is enough for the task.

4.2 Properties Corpora

We created properties corpora by extracting Wikipedia articles about people that share these properties. To determine whether a person has certain properties we used metadata in Wikipedia. For our purpose the `short description` field in `Persondata` provides the information we need about nationalities and professions, such as *American singer*. For example, we created a corpus with the property *American* by extracting articles that contain the word *American* in this field, and created a corpus with the property *singer* by extracting articles that contain *singer* in this field.

For each properties corpus we concatenated n random articles with this property. We chose $n = 500$. Some properties are rare in Wikipedia and we extracted less than 500 articles. In these cases we concatenated all articles we could find.

For the experiments in our pilot implementation we created a total of 15 nationalities corpora and 83 professions corpora. Choosing which nationalities and professions to use was done manually, extracting and concatenating documents to the properties corpora was done automatically. To apply the system to entities that are not included in our test system, more properties corpora need to be created. This can be done automatically, for example, by using lists of nationalities and professions.

4.3 Silver Standard Test Corpus

For our experiments we need data about persons that share the same name. Since collecting and manually annotating data is expensive, we automatically created a dataset by exploiting the link structure in Wikipedia.

We chose 14 people with the same names for which we show statistics in Table 3. All names refer to different people found in Wikipedia, with the number of different real world entities for each name ranging from 2 to 38, which can be seen in the second column (Ent).

We then went through all Wikipedia articles and every time we found one of these names linked to another article, we extracted the name, the link, and the context around the name. We experimented

Proper name	Ent	All	Max	Avg
David Mitchell	9	201	111	22.56
David Thomas	9	85	47	9.44
Jack Johnson	8	427	230	53.38
John Edwards	7	418	400	59.71
John Smith	38	466	107	12.11
John Williams	30	852	650	28.37
Michael Collins	7	443	364	63.29

Proper name	Ent	All	Max	Avg
Michael Jackson	12	3968	3899	330.67
Michael Moore	9	566	532	62.89
Paul Williams	13	385	149	29.62
Peter Müller	2	15	10	7.50
Richard Burton	4	606	592	151.50
Roger Taylor	3	79	39	26.33
Tony Martin	13	275	91	21.15

Table 3: Statistics for extracted entities. Numbers are counts.

with three different context sizes: (i) the sentence around the entity, (ii) the paragraph around the entity, (iii) the section around the entity. The third column in Table 3 (All) shows the overall numbers of context snippets we extracted for each name, the fourth column (Max) shows the maximum numbers of snippets for one entity with the name, i.e., the numbers of snippets used for the majority class baseline. For example, we extracted a total of 3968 snippets for the name *Michael Jackson* (All). 3899 of these snippets (Max) belong to the entity *Michael Jackson, the American singer*. The remaining 69 snippets belong to the other 11 entities with the same name. We have some extreme cases like this one in our dataset, with one entity having many more snippets than the others because it is a very famous person. In other cases the number of snippets is more evenly distributed over all entities with the same name. The last column in Table 3 (Avg) lists the average numbers of snippets that were extracted for each entity.

We use the extracted link information as gold labels to disambiguate the person. For example, if we find the link *Michael_Jackson_(English_singer)* in an article, we know that the extracted name and context around it refers to *Michael Jackson, the English singer*. By using this link information we do not have to annotate a dataset manually. Similar datasets have been created before (Nothman et al., 2008; Nothman et al., 2013; Hahm et al., 2014).

5 Experiments and Discussion

For obtaining topic information we use the MALLET toolkit (McCallum, 2002) which contains several machine learning applications, for example, document classification, clustering, and information extraction. For topic modeling it provides implementations of Latent Dirichlet Allocation (LDA), Pachinko Allocation, and Hierarchical LDA. We use the ParallelTopicModel class which is a simple parallel threaded implementation of LDA based on Newman et al. (2009) and Yao et al. (2009). We trained different topic models using the corpora and topic parameters we described in Section 4.1.

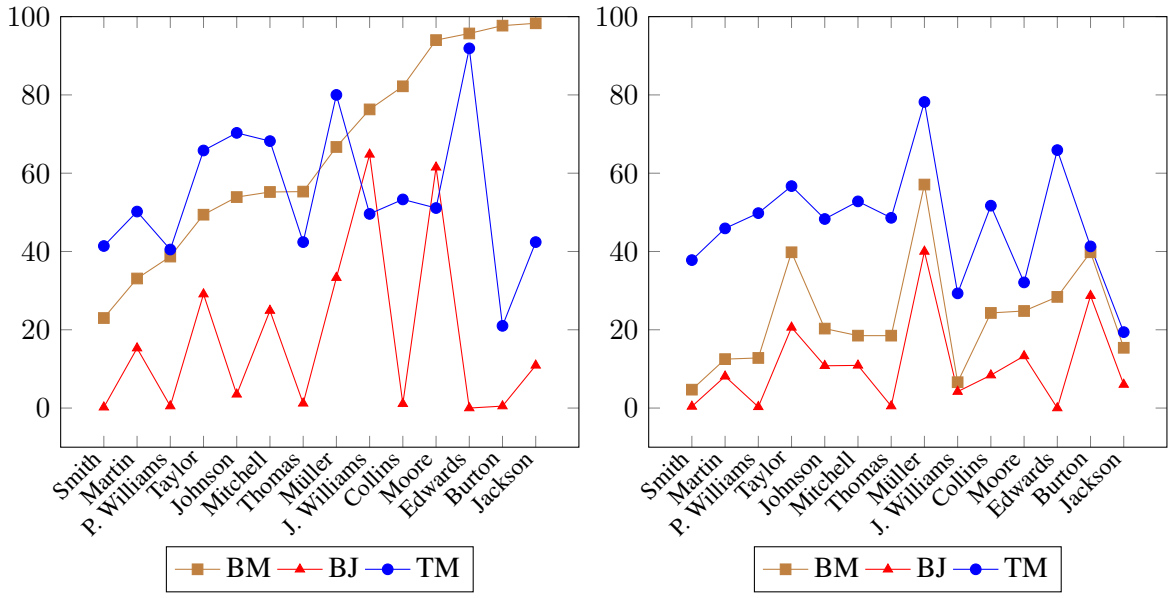
For every entity in our test set we created a corpus consisting of the properties of this entity (e.g., for *Michael Jackson, the American singer* we created a corpus with the properties *American* and *singer*) as described in Section 4.2. We then apply our trained topic models to obtain topic information from these corpora, as well as from the test snippets of our silver standard corpus. The vector representing the topic information from each new entity is then compared to each vector consisting of topic information from the properties corpora as we showed in Section 3.2.

We use two baselines. The **Baseline Majority (BM)** simply predicts the majority class. The **Baseline Jaccard (BJ)** uses the Jaccard index to compare the similarity between the context of the new unknown entity and the corpora we created. The Jaccard index is defined as the size of the intersection divided by the size of the union of two sample sets A and B:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

5.1 Results and Discussion

We conducted a total of 546 experiments with different parameters (context size, number of topics, corpus used to train the topic model). Due to limited space we show results only for one setting of parameters. We chose parameters which had an average performance in the experiments to give an estimate of how the approach works. Using other parameters can improve the results.



(a) micro-averaged F_1

(b) macro-averaged F_1

Figure 2: F_1 score of BM, BJ, and TM with parameters: context size=paragraphs, topics=1000, corpus=W_{all}.

Figure 2 shows the F_1 score for all experiments with the following parameters: context size = paragraphs, topics = 1000, corpus = W_{all}. The names are ordered by the micro-averaged F_1 score of the Baseline Majority (BM). In subsequent graphs we use the same ordering for better comparison. We present both the micro-averaged F_1 score (2a) and the macro-averaged F_1 score (2b), which give quite different results. Micro-averaged F_1 score weights each classification decision equally, i.e., it favors large classes, while macro-averaged F_1 score weights each class equally, i.e., it shows the effectiveness of small classes better (Manning et al., 2008).

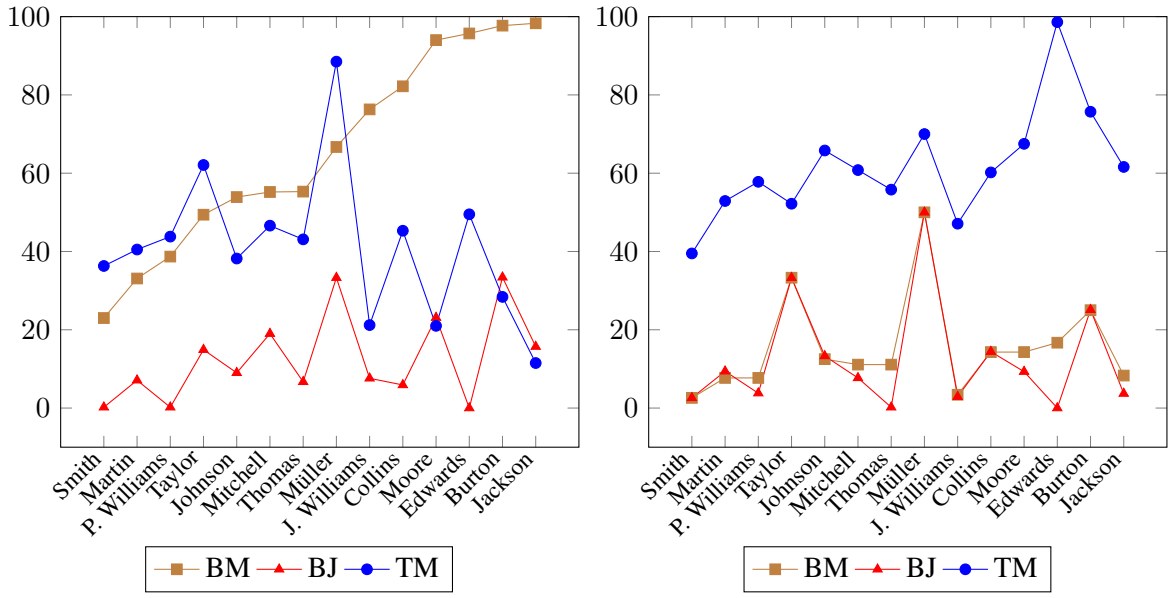
Figure 2a shows that when using micro-averaging, in some cases the majority baseline is better than our topic model approach. This is the case when there is one famous name-bearer who has many more examples than the other persons with the same name in the test set, e.g., *Edwards, Burton, Jackson*, which results in one class that is much larger than the others. For example, our test set contains 3968 snippets for *Michael Jackson*, with 3899 of the snippets belonging to the majority class (*Michael Jackson, the American singer*), which results in a micro-averaged F_1 score of 98.26% for BM.

The macro-averaged results in Figure 2b give a better sense of how well our approach works on smaller classes. It can be seen that our approach performs better than both baselines in all cases. Since we are interested in the not well-known entities, which have smaller classes in our test set, the macro-averaged results show that our approach works well for these entities.

The Baseline Jaccard (BJ) stays even below the Baseline Majority in most cases. It generally does not work well for persons that are rather unknown. One reason is that the contexts extracted for these entities are often smaller and do not provide as much information as is needed for this baseline approach. The main advantage of our TM approach is that it works well for unknown entities which usually have little or no available training material, and for entities which have little surrounding context.

In Figure 3 we give more insight into the macro-averaged results of Figure 2b and show the macro-averaged precision (3a) and macro-averaged recall (3b) with the same parameters as before. Precision-wise, the Baseline Majority does better in cases with large classes. This is expected because if most examples in the test set belong to the majority class, the number of false positives is small, which leads to higher precision.

Figure 3b shows that the recall for our approach outperforms both baselines by far in all cases. In one case (*Edwards*) we even achieve a recall of 98.6%. The reason for this is that the baseline approaches do not work well for the not well-known people with small classes in our test set. BM does not work well



(a) macro-averaged precision

(b) macro-averaged recall

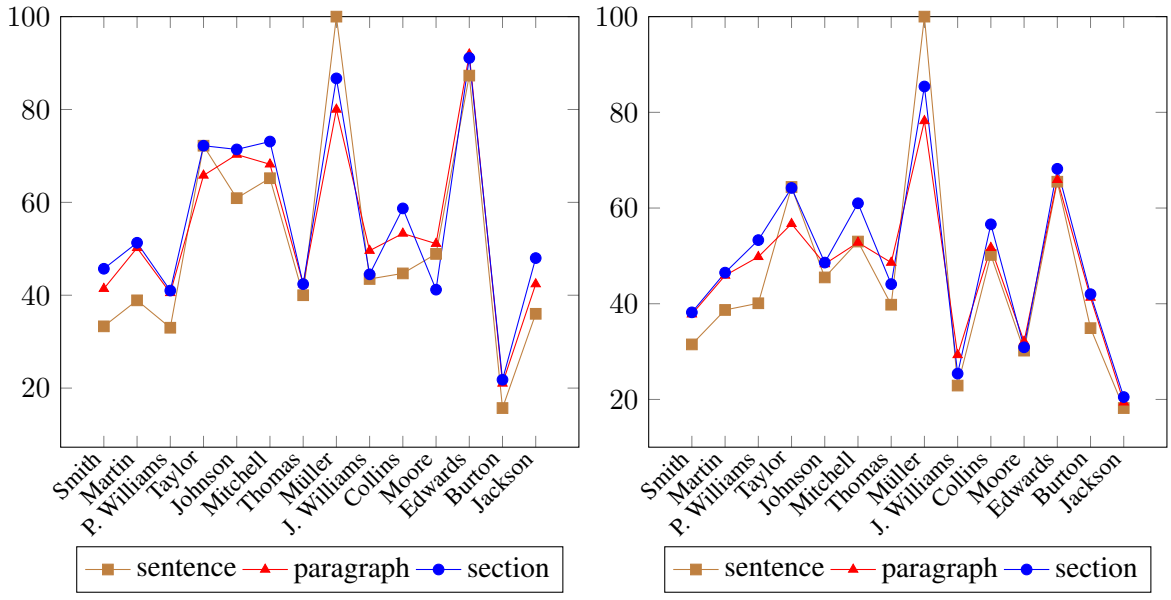
Figure 3: Precision and Recall of BM, BJ, and TM with parameters: context size=paragraphs, topics=1000, corpus=W$_{all}$.

because it incorrectly tags the small classes with the majority class and BJ does not work well because the information in the context snippets is not sufficient for this approach. Measuring the similarity between words, n-grams, or similar approaches suffers from sparsity problems. Our TM approach makes better use of smaller context snippets because it extracts and uses relevant information (i.e., the topic information) about a person more effectively. This leads to higher recall results in most cases. In some cases with different parameters, we even achieve a recall of 100% for some entities (not shown in Figure 3). Obtaining a high recall is important in applications that aim for high recall results (e.g., in search engines or QA systems; the system returns more correct results instead of returning results of similar persons which are more well-known).

We have investigated how the context size, number of topics, and the corpus used for training the topic model influences the results of our TM approach. Figure 4 shows results for the different context sizes (sentence, paragraph, section), when using the same parameters for the topic model as in the previous figures (topics = 1000, corpus = W$_{all}$). In most cases, taking more context gives better results (i.e., paragraphs are better than sentences and sections are better than paragraphs). In some cases, a larger context introduces more noise which leads to worse results than when taking a smaller context (e.g., *Moore* in Figure 4a). Generally, taking a smaller context does not worsen the results a lot and in some cases gives nearly the same results as when taking a larger context. This shows that our TM approach works well if there is only little context available for a person. This is important because often the context for a person is rather limited, for example, when a document is mainly about another person there might be only one relevant sentence as context, or when the document is very short.

When we investigated the usefulness of the number of topics, we found that taking a small number of topics (like 100) works best. Using a higher number results in more fine-grained topics, which makes finding the most similar corpus harder. An advantage of using fewer topics is that it takes less time for the topic model to determine the topics of a new document. In a real application it is important to reduce the time needed as much as possible because a user of the application does not want to wait for several minutes or even hours for results.

There is no clear preference for which corpus is the most useful one for the task and it also depends on the context size. Overall, using the entire Wikipedia seems to be the least helpful and it is better to use a more specific collection which produces better topics for the task. We also found that when using different collections, the results can change a lot. While the results for using different parts of the



(a) micro-averaged F_1

(b) macro-averaged F_1

Figure 4: Different context sizes of TM with parameters: topics=1000, corpus=W_{all}.

Wikipedia collection all followed the same trend, the results for using the Gigaword corpus were very different and often worse than Wikipedia's results.

6 Conclusion and Future Work

We presented work on a new system for Named Entity Disambiguation which does not need specific textual training data to disambiguate unknown persons. Instead, it uses some aggregate of the textual material about different people that share the same properties (e.g., nationalities and professions). The system learns which properties people can have, then uses these properties to disambiguate new proper names in documents.

To learn properties we extracted documents about people sharing the same properties from a collection, then applied topic models on the data to obtain topic information. The topic models were trained on different collections and with different numbers of topics to investigate which parameters are most useful. To disambiguate a new unknown person in a text document, we obtained topic information from the context of the person, then compared this topic information with the information from the extracted material to find out which properties are closest to the person.

In our pilot implementation of the approach, we conducted 546 experiments on 14 ambiguous names using different parameters (context size, number of topics, corpus used for training the topic model). For evaluating our system we created a silver standard corpus that does not need any manual annotation, and compared our approach to two baselines. We showed that our system outperforms the baselines in many cases, especially for (i) entities for which not much training material is available, and (ii) entities with little surrounding context. We also showed that with our approach we can achieve high recall results, which is important for many applications, e.g., search engines and QA systems.

The approach can be expanded with more properties in the future. For example, it could include properties like age, place of birth, and affiliation (university, company etc.) and properties about what people are doing besides their profession (e.g., working in organizations or for charity).

Acknowledgements

The work reported in this paper was supported by a Nuance Foundation Grant.

References

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361, Sofia, Bulgaria, August. Association for Computational Linguistics.

David Bamman, Ted Underwood, and Noah A. Smith. 2014. A bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 370–379, Baltimore, Maryland, June. Association for Computational Linguistics.

Indrajit Bhattacharya and Lise Getoor. 2006. A latent dirichlet model for unsupervised entity resolution. In *SIAM Conference on Data Mining (SDM)*, April. Winner of the Best Paper Award.

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

Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2006. Extracting key phrases to disambiguate personal name queries in web search. In *Proceedings of the Workshop on How Can Computational Linguistics Improve Information Retrieval?*, CLIIR '06, pages 17–24, Stroudsburg, PA, USA. Association for Computational Linguistics.

Razvan Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceesings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, pages 9–16, Trento, Italy.

Ying Chen and James Martin. 2007. Towards robust unsupervised personal name disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 190–198, Prague, Czech Republic, June. Association for Computational Linguistics.

Aaron M. Cohen. 2005. Unsupervised gene/protein named entity normalization using automatically extracted dictionaries. In *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, ISMB '05, pages 17–24, Stroudsburg, PA, USA. Association for Computational Linguistics.

Silviu Cucerzan. 2007. 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. Association for Computational Linguistics.

David Graff and Christopher Cieri. 2003. English Gigaword LDC2003T05. Web Download. Philadelphia: Linguistic Data Consortium.

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 267–274, New York, NY, USA. ACM.

Younggyun Hahm, Jungyeul Park, Kyungtae Lim, Youngsik Kim, Dosam Hwang, and Key-Sun Choi. 2014. Named entity corpus construction using wikipedia and dbpedia ontology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2565–2569, Reykjavik, Iceland, May.

Xianpei Han and Le Sun. 2012. An entity-topic model for entity linking. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 105–115, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: A graph-based method. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 765–774, New York, NY, USA. ACM.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 782–792, Stroudsburg, PA, USA. Association for Computational Linguistics.

Saurabh S. Kataria, Krishnan S. Kumar, Rajeev R. Rastogi, Prithviraj Sen, and Srinivasan H. Sengamedu. 2011. Entity disambiguation with hierarchical topic models. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1037–1045, New York, NY, USA. ACM.

Yang Li, Chi Wang, Fangqiu Han, Jiawei Han, Dan Roth, and Xifeng Yan. 2013. Mining evidences for named entity disambiguation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1070–1078, New York, NY, USA. ACM.

Bradley Malin. 2005. Unsupervised name disambiguation via social network similarity. In *In Proceedings of the SIAM Workshop on Link Analysis, Counterterrorism, and Security*, pages 93–102.

Gideon S. Mann and David Yarowsky. 2003. Unsupervised personal name disambiguation. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 33–40, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, pages 233–242, New York, NY, USA. ACM.

David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2009. Distributed algorithms for topic models. *J. Mach. Learn. Res.*, 10:1801–1828, December.

Joel Nothman, James R. Curran, and Tara Murphy. 2008. Transforming wikipedia into named entity training data. In *In Proceedings of the Australasian Language Technology Association Workshop 2008*, pages 124–132.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R. Curran. 2013. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194(0):151–175. Artificial Intelligence, Wikipedia and Semi-Structured Resources.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1524–1534, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States. AUAI Press.

Lus Sarmento, Alexander Kehlenbeck, Eugnio C. Oliveira, and Lyle H. Ungar. 2009. An approach to web-scale named-entity disambiguation. In Petra Perner, editor, *MLDM*, volume 5632 of *Lecture Notes in Computer Science*, pages 689–703. Springer.

Prithviraj Sen. 2012. Collective context-aware topic models for entity disambiguation. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 729–738, New York, NY, USA. ACM.

Liangcai Shu, Bo Long, and Weiyi Meng. 2009. A latent topic model for complete entity resolution. In *Proceedings of the 2009 IEEE International Conference on Data Engineering*, ICDE '09, pages 880–891, Washington, DC, USA. IEEE Computer Society.

Limin Yao, David Mimno, and Andrew McCallum. 2009. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 937–946, New York, NY, USA. ACM.

Wei Zhang, Yan Chuan Sim, Jian Su, and Chew Lim Tan. 2011. Entity linking with effective acronym expansion, instance selection and topic modeling. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 1909–1914. AAAI Press.