Isaac Bloomberg Meets Michael Bloomberg: Better Entity Disambiguation for the News

Luka Bradesko
Bloomberg L.P., USA
lbradesko1@bloomberg.net
Institut "Jožef Stefan", Slovenia
luka.bradesko@ijs.si

Janez Starc
Bloomberg L.P., USA
jstarc1@bloomberg.net
Institut "Jožef Stefan", Slovenia
janez.starc@ijs.si

Stefano Pacifico
Bloomberg L.P., USA
spacifico1@bloomberg.net

ABSTRACT

This paper shows the implementation and evaluation of the Entity Linking or Named Entity Disambiguation system used and developed at Bloomberg. In particular, we present and evaluate a methodology and a system that do not require the use of Wikipedia as a knowledge base or training corpus. We present how we built features for disambiguation algorithms from the Bloomberg News corpus, and how we employed them for both single-entity and joint-entity disambiguation into a Bloomberg proprietary knowledge base of people and companies. Experimental results show high quality in the disambiguation of the available annotated corpus.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Algorithms, Theory

Keywords

Bloomberg, Named Entity Disambiguation, Entity Linking, Text Mining

1. INTRODUCTION

The task of Named Entity Disambiguation (*NED*), or Entity Linking is the task that associates unambiguous entities, from a Knowledge Base, to the named-entity mentions (surface forms) within an input text. In the past, except for a few exceptions [14], the majority of the research focused around disambiguating named entities to Wikipedia pages, as entities of a Wikipedia Knowledge Base. The reasons for the popularity of this approach include the breadth of coverage of Wikipedia; the existence of formal Knowledge Bases like Yago, DBpedia, and Freebase; and the presence of many Wikipedia links that act as annotated data to train or validate NED systems. While disambiguating to Wikipedia entities may prove useful for a generic Natural Language Processing application, specific requirements may include the capability to disambiguate named entities into a knowledge base different than Wikipedia. Such is the case for Bloomberg.

Most Bloomberg News articles contain manually curated annotations of all kinds, ranging from named entities to important article keywords and topics. In particular, we are interested in the annotations of *people* and *companies* only, categories for which Bloomberg has an extensive proprietary knowledge base. The knowledge base used for this paper contains more than 5 million entities between people and companies. In this paper, we present the Bloomberg Named Entity Disambiguation system, a system that disambiguates named entities to Bloomberg entities.

In Section 2, we show the process for creating a NED system that is not based on Wikipedia. Also, in Section 2.3 we illustrate the features that we have selected and how they participate to the overall performance. Further, in Section 2.4 we show the performance of the Bloomberg NED system on the Bloomberg News proprietary corpus and discuss the meaning of the results obtained. Finally, in Section 3 we discuss the overall contributions of the paper and future work.

1.1 Related Work

The initial attempts at Wikipedia-based disambiguation, [1] [2] [7] [11] mainly involved defining textual similarity and semantic relatedness measures, for disambiguating *single* mentions, one at a time, and ignoring the inter-dependency that may exist among multiple entities in the same document. In order to leverage the information content of all the entities in a document, [9] [8] proposes models that attempt the disambiguation of the entities in the document, all at once. Even though these approaches can offer good precision, the combinatorial nature of the problem typically results in intractability (*NP-Hard*), and approximate approaches ensue [8].

Systems like Aida-light [12] and DBpedia Spotlight [10][3] propose faster disambiguation systems. The Spotlight [10] system uses a statically-based approach for multilingual NED, disambiguating text-mentions one at a time. With Aida-light [12] the authors describe an iterative, *two-phase* joint

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2741711.

disambiguation approach: first the *easier* mentions are jointly disambiguated, then categorical domains are selected for the entities already disambiguated, and the entity-relatedness scores are calculated again. Both make use of a Named Entity Recognition pre-processing step (*NER*), to spot the mentions and context similarity during the disambiguation phase.

Other systems do not use Wikipedia to disambiguate. For example, in [14], a naive-Bayes probabilistic clustering approach is used to attribute categories to entities. The categories are taken from the knowledge base *Probase*[1].

2. BUILDING BLOOMBERG NED

2.1 Proposed Approach

We start by building a dictionary and calculating features from a structured knowledge base containing data about Bloomberg entities and from the documents presented in Section 2.2. The dictionary contains sets of mentions m associated with each entity e. For example the mentions *I.B.M.*, *International Business Machines*, and *Big Blue* are associated with the company-entity *I.B.M.*. At this time the only type of entities disambiguated by the Bloomberg NED system are people and companies.

Then, we define, and calculate for each entity in the dictionary, the following quantities:

- *Prior probability*: for each entity e, the probability $p(e|m)$ that represents the probability that the mention m refers to the entity e without using any other contextual information. For example, if the mention is *Bloomberg* and it will more likely refer to New-York based Bloomberg L.P. than to Isaac P. Bloomberg Limited, registered in Essex, U.K. In this work, the prior probability for an entity e is calculated from the number of times that a mention m appears in the training corpus, and the number of times it is annotated with e.

- *Popularity*: in the knowledge base used at Bloomberg, each entity e has a popularity score described as a probability function of the mention e.

$$0 \leq popularity(e) \leq 1 \tag{1}$$

- *Entity Context*: for each entity e we define the context of the entity, $C(e)$ as a collection of words that describes e. In this work, we build the entity contexts by merging the corresponding properties available in the knowledge base (including textual descriptions).

For each mention m in the text, a context $T_k(m)$ is produced by selecting k sentences, centered around the mention. We then define:

- *Mention Entity Context Similarity*: similarity score between a mention m and an entity e, as the number of words that overlap between the context of m and the context of e:

$$mec\_sim(m, e) = |T(m) \cup C(e)| \tag{2}$$

- *Mention Dictionary*: for a mention m, the set of entities that have m as a surface form.

$$dict(m) = e|m \text{ surface form for } e \tag{3}$$

For each entity pair e_1 and e_2, we define the *Entity Context Similarity*: as the number of words of overlap between the contexts of e_1 and e_2:

$$eec\_sim(e_1, e_2) = |C(e_1) \cup C(e_2)| \tag{4}$$

Also, we define entity co-occurrence as the probability that e_1 and e_2 appear together in the same document D:

$$ee\_occ(e_1, e_2) = p(e_1, e_2|D) \tag{5}$$

In the case of single-entity iterative disambiguation, given a mention m and a set of q entities $E = \{e_1, , e_q\}$, we define the named-entity disambiguation $NED(m)$ as follows:

$$NED(m) = e^* = \arg \max_{e_i \in \{e_1, , e_q\}} [\alpha \cdot mec\_sim(m, e_i) +$$

$$+\beta \cdot popularity(e_i) + \gamma \cdot p(e_i|m)]$$

In the case of joint-entity disambiguation, given the sequence of k mentions $M = t\langle m_1 \dots m_k \rangle$, and given the set of q candidate entities $E = e_1 \dots e_q$, we define the named-entity disambiguation $NED(M)$ as:

$$NED(M) = NED(\langle m_1, \dots, m_k \rangle) := \langle e_1^*, \dots e_k^* \rangle$$

where:

$$\langle e_1^*, \dots e_k^* \rangle =$$

$$\arg \max_{e_i \in \{e_1, , e_q\}} [\alpha \cdot \sum_{i,j \in 1 \dots k} ee\_occ(e_i, e_j) + \beta \cdot \sum_{i=1 \dots k} mec\_sim(m_i, e_i)$$

$$+\gamma \cdot \sum_{i=1 \dots k} popularity(e_i) + \delta \cdot \sum_{i=1 \dots k} p(e_i|m_i)]$$

where $\alpha + \beta + \gamma + \delta = 1$.

In both cases, the weights α, β, γ, and δ are determined experimentally, using a brute-force approach, but could be obtained with optimization or machine learning techniques. In [8] for example, an *SVM* classifier is used to determine them. The algorithms for disambiguation are partially based on the Aida-light project [12].

2.2 News Data

Bloomberg News articles are manually annotated and contain entities, among other things, of *people* and *companies*. It's important to highlight that this data set contains annotations only for the first occurrence of an entity in the text, and that there are some articles that do not have annotations at all. The data used for this paper is a subset of the Bloomberg News corpus, and is composed of two sets: training data, and evaluation data.

Initially, we considered all Bloomberg articles from the last 10 years (in the hundreds of millions - see [4] for more information on Bloomberg News data). Then, we created two sets: a training set with 100,000 articles selected uniformly at random, and a test set, generated by taking 100 articles per month (12,000) and then 2,000 selected randomly. Of these, 1005 had no annotations, leaving the evaluation set to 995 articles. The evaluation data set includes 6,244 annotated mentions, 5,316 (or about 85%) of which are mentions of people, and 928 (or about 15%) are companies, for a total of 3,842 unique entities. Of the 6,244 annotated mentions around 62% were aliases that referred to one entity only, and can be disambiguated directly with string matching. Note

[1]http://research.microsoft.com/en-us/projects/probase/

that, compared to the general case, the set used for evaluation contains *easier* mentions regarding the disambiguation: The annotated mentions typically connotate the first occurrence of entities in the article, and as such often appear in a canonical form that is easier to disambiguate. Therefore, no anaphora resolution is necessary. Finally, we highlight that 3.3% of the mentions in the evaluation corpus are not an alias to any of the entities in our knowledge base. The system will disambiguate them to a *null* entity.

2.3 Experiments

In this section, we present the experiments that we have run. Broadly speaking, each experiment looks at the performance of a subset of the features illustrated, trying to elicit what is the impact of removing or adding any feature. In all the experiments, the candidate entities for disambiguating a textual mention are selected among the entities whose aliases match the mention exactly. We did not use any fuzzy matching, like Locality Sensitive Hashing [6] or SimString [13], which would improve that 3.3% of the entities for which we do not have an alias, as mentioned in Section 2.2.

The experiments encompass the two strategies that we have discussed in Section 1.1: single- and joint-entity disambiguation.

In the case of single-entity disambiguation, we calculate a score for each mention-entity pair. The score is the affine combination of the entity, and mention-entity features (*prior*, *popularity*, *mention-entity context similarity*). Then, we rank the candidate entities for each mention by the calculated score. On the other hand, in the case of joint-entity disambiguation, we calculate the entity-entity features (*entity context similarity*, and *entity co-occurrence*). Then, we use a graph-based approach described in [12] and in Section 2.1. Because of the poor quality of the entity context similarity feature we decided to drop it, and use only the entity co-occurrence feature. In the figures and tables, we will refer to these features as **Prior**, **Pop**, **Ctx**, and with **Graph** for the joint-entity features as a whole.

In this set of experiments, we use as baseline (**Random**), an algorithm that disambiguates candidate entities, by randomly (uniform) selecting one. Also, we evaluate the disambiguation part of the NED algorithm only, without evaluating the NER step. While in the normal processing of the system we use the Stanford Named Entity Recognizer [5] for finding the mentions, in the experiments we use pre-annotated mentions, assuming a perfect NER system, similarly to others. [8] [12]

2.4 Results

The results are separated into three sections. The overall NED performance (Table 1), which is then separately assessed for entities of people (Table 2) and companies (Table 3).

All of the tables are presenting the evaluation with the baseline (Random) and various other feature combinations. Each of the feature combinations is then additionally evaluated as well with and without the joint-disambiguation algorithm (Graph). All of the results are graphically summarized on the Graph 1. The features and their abbreviations that are used here, are explained in Section 2.3.

We can notice that in the baseline (`Random`) and also some of the other experiments, companies are better disambiguated than people, but this changes when we start to use more of the features. This is due to the fact that there are more people than companies (85% vs. 15%) in the corpus at

Table 1: Overall results by feature

	Precision	Recall	F1
Random	73.99%	71.52%	72.73%
Graph	77.90%	75.30%	76.58%
Ctx	82.62%	79.87%	81.22%
Prior	95.94%	92.75%	94.32%
Pop	81.30%	78.59%	79.92%
Pop+Graph	89.31%	86.34%	87.80%
Ctx+Graph	89.28%	86.31%	87.77%
Prior+Graph	96.27%	93.07%	94.64%
Prior+Ctx	96.85%	93.63%	95.21%
Prior+Pop	96.92%	93.69%	95.28%
Ctx+Pop	83.57%	80.78%	82.15%
Pior+Pop+Ctx	97.05%	93.82%	95.41%
Prior+Ctx+Graph	97.05%	93.82%	95.41%
Prior+Pop+Graph	97.02%	93.79%	95.38%
Ctx+Pop+Graph	89.73%	86.74%	88.21%
All	**97.15%**	**93.91%**	**95.50%**

Table 2: People results by feature

	Precision	Recall	F1
Random	73.36%	71.86%	72.60%
Graph	77.35%	75.76%	76.55%
Ctx	82.29%	80.60%	81.44%
Prior	96.23%	94.26%	95.23%
Pop	79.93%	78.30%	79.11%
Pop+Graph	89.07%	87.24%	88.15%
Ctx+Graph	89.20%	87.38%	88.28%
Prior+Graph	96.48%	94.50%	95.48%
Prior+Ctx	97.22%	95.24%	96.22%
Prior+Pop	97.18%	95.20%	96.18%
Ctx+Pop	82.61%	80.92%	81.76%
Pior+Pop+Ctx	97.43%	95.44%	96.42%
Prior+Ctx+Graph	97.40%	95.40%	96.39%
Prior+Pop+Graph	97.26%	95.27%	96.25%
Ctx+Pop+Graph	89.43%	87.60%	88.51%
All	**97.51%**	**95.52%**	**96.50%**

Table 3: Company results by feature

	Precision	Recall	F1
Random	78.01%	69.55%	73.54%
Graph	81.45%	72.62%	76.78%
Ctx	84.77%	75.58%	79.91%
Prior	94.10%	83.90%	88.71%
Pop	90.05%	80.28%	84.88%
Pop+Graph	90.91%	81.05%	85.70%
Ctx+Graph	89.80%	80.07%	84.66%
Prior+Graph	94.96%	84.67%	89.52%
Prior+Ctx	94.47%	84.23%	89.06%
Prior+Pop	95.21%	84.88%	89.75%
Ctx+Pop	89.68%	79.96%	84.54%
Pior+Pop+Ctx	94.59%	84.34%	89.17%
Prior+Ctx+Graph	94.84%	84.56%	89.41%
Prior+Pop+Graph	95.45%	85.10%	89.98%
Ctx+Pop+Graph	91.65%	81.71%	86.40%
All	**94.84%**	**84.56%**	**89.41%**

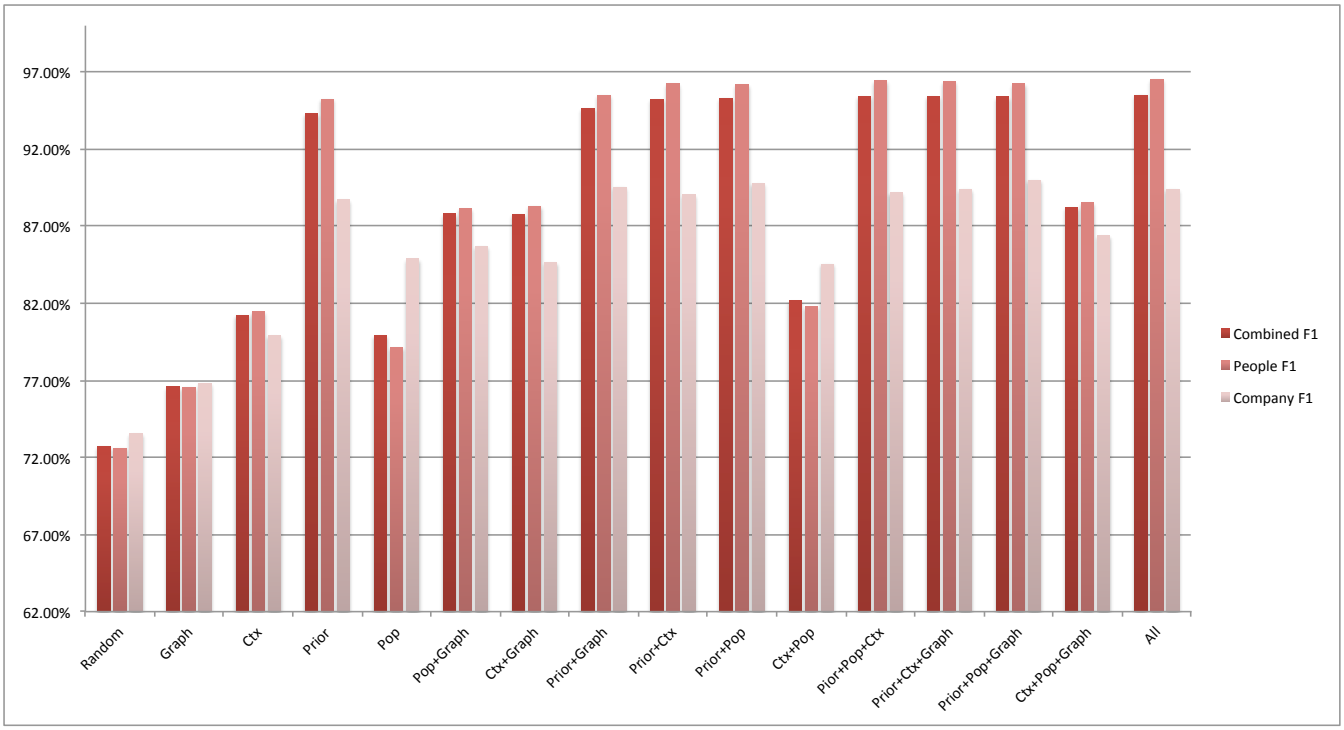

Figure 1: F1 scores by feature

hand. This makes the randomized disambiguation less effective for people, but later enables us to retrieve more feature related data, which helps the algorithm disambiguate them better than companies.

Also, we observe that the *prior probability* feature seems to have the largest impact of all the features. This is explained by considering that both the training and evaluation corpora are news articles from the same news source, which really helps with the disambiguation on the similar types of documents.

Further, it is worth noting, that all of the features presented in Section 2.4, improve both, precision and recall in any combination of usage. This proves that each of the features is useful and it's contribution to overall NED not just a random effect. The features that we experimented with, but had a negative impact when added, were *entity entity context similarity*, and trying larger radii for extracting the context around the mentions.

In addition, we observe that, as mentioned in Section 2.2, 62% of the mentions in the evaluation corpus are not ambiguous, and 3.3% of the mentions are not aliases to any of the entities with the respect to the dictionary. Therefore, we calculate an *effective accuracy* of **92.6%**. This measure is the fraction of correctly disambiguated mentions that have non-trivial disambiguation.

3. CONCLUSIONS

We have constructed and evaluated a NED system, which disambiguates to a Bloomberg-specific set of entities, illustrating a methodology for creating NED systems around arbitrary knowledge bases, different from Wikipedia. Also, we illustrate how to build features featuers for a NED system. Further, we illustrate experiments that show how all the features contribute to the overall performance of the system. Among other things, the results show that the features extracted from training corpus, such prior probability, have a greater impact in our experiments.

3.1 Future work

We identified the following threads for the future work on the real-time NED on Bloomberg news:

- Studying new features to include in the model to make the disambiguation faster and more robust.

- Active learning definition of the problem for effective semi-supervision in model improvement: reducing the human effort spent by correcting erroneous predictions of the disambiguation system, for improving the quality of its models by ranking the examples that need review from a human being.

- Validation on external data: in order to improve the Bloomberg NED system it is necessary to evaluate it on data sets different from Bloomberg (see next point).

- Data annotation: while working on the Bloomberg NED system, we have run into many obstacles deriving from the absence of a large, up-to-date annotated data sets for disambiguation. In particular, the situation is quite problematic for arbitrary knowledge bases. Studying effective ways of producing golden data sets is important.

4. ACKNOWLEDGMENTS

We would like to thank James Hodson and Anastassia Fedyk for the insightful and precious feedback, and the time spent helping us improve this paper.

5. REFERENCES

[1] R. Bunescu and M. Pasca. 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, 2006.

[2] S. Cucerzan. Large-scale Named Entity Disambiguation Based on Wikipedia Data. In *In Proc. 2007 Joint Conference on EMNLP and CNLL*, pages 708–716, 2007.

[3] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes. Improving Efficiency and Accuracy in Multilingual Entity Extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013.

[4] A. Fedyk and J. Hodson. Aggregation Effect in Stale News. *Available at SSRN 2433234*, 2014.

[5] J. R. Finkel, T. Grenager, and C. Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370. ACL, 2005.

[6] A. Gionis, P. Indyk, R. Motwani, et al. Similarity Search in High Dimensions via Hashing. In *VLDB*, volume 99, pages 518–529, 1999.

[7] 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 2009, Hong Kong, China, November 2-6, 2009*, pages 215–224, 2009.

[8] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust Disambiguation of Named Entities in Text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, Edinburgh, Scotland*, pages 782–792, 2011.

[9] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective Annotation of Wikipedia Entities in Web Text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 457–466, New York, NY, USA, 2009. ACM.

[10] P. N. Mendes, M. Jakob, A. Garcia-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)*, 2011.

[11] D. Milne and I. H. Witten. Learning to Link with Wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM, 2008.

[12] Nguyen, Dat Ba and Hoffart, Johannes and Theobald, Martin and Weikum, Gerhard. AIDA-light: High-Throughput Named-Entity Disambiguation. In *Linked Data on the Web at WWW2014*, 2014.

[13] N. Okazaki and J. Tsujii. Simple and Efficient Algorithm for Approximate Dictionary Matching. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 851–859, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[14] M. Shirakawa, H. Wang, Y. Song, Z. wang, K. Nakayama, and T. Hara. Entity Disambiguation based on a Probabilistic Taxonomy. Technical Report MSR-TR-2011-125, Technical Report MSR-TR-2011-125, Microsoft Research, November 2011.