Graph Ranking for Collective Named Entity Disambiguation

Ayman Alhelbawy[1,2] and **Robert Gaizauskas**[1]
[1]The University of Sheffield, Regent Court, 211 Portobello Street, Sheffield, S1 4DP, U.K
[2]Faculty of Computers and Information, Fayoum University, Fayoum, Egypt
ayman,R.Gaizauskas@dcs.shef.ac.uk

Abstract

Named Entity Disambiguation (NED) refers to the task of mapping different named entity mentions in running text to their correct interpretations in a specific knowledge base (KB). This paper presents a collective disambiguation approach using a graph model. All possible NE candidates are represented as nodes in the graph and associations between different candidates are represented by edges between the nodes. Each node has an initial confidence score, e.g. entity popularity. Page-Rank is used to rank nodes and the final rank is combined with the initial confidence for candidate selection. Experiments on 27,819 NE textual mentions show the effectiveness of using Page-Rank in conjunction with initial confidence: 87% accuracy is achieved, outperforming both baseline and state-of-the-art approaches.

1 Introduction

Named entities (NEs) have received much attention over the last two decades (Nadeau and Sekine, 2007), mostly focused on recognizing the boundaries of textual NE mentions and classifying them as, e.g., Person, Organization or Location. However, references to entities in the real world are often ambiguous: there is a many-to-many relation between NE mentions and the entities they denote in the real world. For example, *Norfolk* may refer to a person, "Peter Norfolk, a wheelchair tennis player", a place in the UK, "Norfolk County", or in the US, "Norfolk, Massachusetts"; conversely, one entity may be known by many names, such as "Cat Stevens", "Yusuf Islam" and "Steven Georgiou". The NED task is to establish a correct mapping between each NE mention in a document and the real world entity it denotes. Following most researchers in this area, we treat entries in a large knowledge base (KB) as surrogates for real world entities when carrying out NED and, in particular, use Wikipedia as the reference KB for disambiguating NE mentions. NED is important for tasks like KB population, where we want to extract new information from text about an entity and add this to a pre-existing entry in a KB; or for information retrieval, where we may want to cluster or filter results for different entities with the same textual mentions.

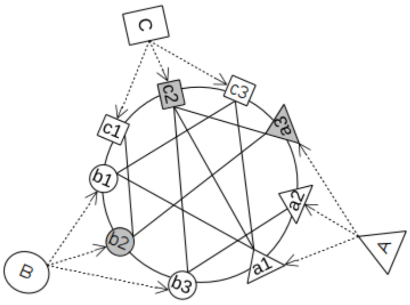

Figure 1: Example of solution graph

The main hypothesis in this work is that different NEs in a document help to disambiguate each other. The problem is that other textual mentions in the document are also ambiguous. So, what is needed is a *collective disambiguation* approach that jointly disambiguates all NE textual mentions.

In our approach we model each possible candidate for every NE mention in a document as a distinct node in a graph and model candidate coherence by links between the nodes. We call such graphs *solution graphs*. Figure 1 shows an example of the solution graph for three mentions "A", "B", and "C" found in a document, where the candidate entities for each mention are referred to using the lower case form of the mention's letter together with a distinguishing subscript. The goal of disambiguation is to find a set of nodes where only one candidate is selected from the set of entities associated with each mention, e.g. a_3, b_2, c_2.

Our approach first ranks all nodes in the solution graph using the Page-Rank algorithm, then re-

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 75–80,
Baltimore, Maryland, USA, June 23-25 2014. ©2014 Association for Computational Linguistics

ranks all nodes by combining the initial confidence and graph ranking scores. We consider several different measures for computing the initial confidence assigned to each node and several measures for determining and weighting the graph edges. Node linking relies on the fact that the textual portion of KB entries typically contains mentions of other NEs. When these mentions are hyper-linked to KB entries, we can infer that there is some relation between the real world entities corresponding to the KB entries, i.e. that they should be linked in our solution graph. These links also allow us to build up statistical co-occurrence counts between entities that occur in the same context which may be used to weight links in our graph.

We evaluate our approach on the AIDA dataset (Hoffart et al., 2011). Comparison with the baseline approach and some state-of-the-art approaches shows our approach offers substantial improvements in disambiguation accuracy.

2 Related Work

In 2009, NIST proposed the shared task challenge of Entity Linking (EL) (McNamee and Dang, 2009). EL is a similar but broader task than NED because NED is concerned with disambiguating a textual NE mention where the correct entity is known to be one of the KB entries, while EL also requires systems to deal with the case where there is no entry for the NE in the reference KB. Ji et al. (2011) group and summarise the different approaches to EL taken by participating systems.

In general, there are two main lines of approach to the NED problem. *Single entity disambiguation approaches (SNED)*, disambiguate one entity at a time without considering the effect of other NEs. These approaches use local context textual features of the mention and compare them to the textual features of NE candidate documents in the KB, and link to the most similar. The first approach in this line was Bunescu and Pasca (2006), who measure similarity between the textual context of the NE mention and the Wikipedia categories of the candidate. More similarity features were added by Cucerzan (2007) who realized that topical coherence between a candidate entity and other entities in the context will improve NED accuracy and by Milne and Witten (2008) who built on Cucerzan's work. Han and Sun (2011) combine different forms of disambiguation knowledge using evidence from mention-entity associations and entity popularity in the KB, and context similarity.

The second line of approach is *collective named entity disambiguation (CNED)*, where all mentions of entities in the document are disambiguated jointly. These approaches try to model the interdependence between the different candidate entities for different NE mentions in the query document, and reformulate the problem of NED as a global optimization problem whose aim is to find the best set of entities. As this new formulation is NP-hard, many approximations have been proposed. Alhelbawy and Gaizauskas (2013) proposed a sequence dependency model using HMMs to model NE interdependency. Another approximation uses a mixture of local and global features to train the coefficients of a linear ranking SVM to rank different NE candidates (Ratinov et al., 2011). Shirakawa et al. (2011) cluster related textual mentions and assign a concept to each cluster using a probabilistic taxonomy. The concept associated with a mention is used in selecting the correct entity from the Freebase KB.

Graph models are widely used in collective approaches[1]. All these approaches model NE interdependencies, while different methods may be used for disambiguation. Han (2011) uses local dependency between NE mention and the candidate entity, and semantic relatedness between candidate entities to construct a referent graph, proposing a collective inference algorithm to infer the correct reference node in the graph. Hoffert (2011) poses the problem as one of finding a dense sub-graph, which is infeasible in a huge graph. So, an algorithm originally used to find strongly interconnected, size-limited groups in social media is adopted to prune the graph, and then a greedy algorithm is used to find the densest graph.

Our proposed model uses the Page-Rank (PR) algorithm (Page et al., 1999), which to our knowledge has not previously been applied to NED. Xing and Ghorbani (2004) adopted PR to consider the weights of links and the nodes' importance. PR and Personalized PR algorithms have been used successfully in WSD (Sinha and Mihalcea, 2007; Agirre and Soroa, 2009).

3 Solution Graph

In this section we discuss the construction of a graph representation that we call the *solution*

[1] Graph models are also widely used in Word Sense Disambiguation (WSD), which has lots of similarities to NED (Gutiérrez et al., 2011; Gutiérrez et al., 2012).

graph. The input is a document containing pre-tagged NE textual mentions. The solution graph is an undirected graph $G = (V, D)$ where V is the node set of all possible NE candidates for different textual mentions in the input document and D is the set of edges between nodes. Edges are not drawn between different nodes for the same mention. They are drawn between two entities when there is a relation between them, as described below. Each candidate has associated with it an initial confidence score, also detailed below.

Assume the input document D has a set of mentions $M = \{m_1, m_2, m_3, ..., m_k\}$. For each $m_i \in M$, we rank each candidate entity, where the list of candidates for a mention m_i is $E_i = \{e_{i,1}, e_{i,2}, ..., e_{i,j}\}$. The graph nodes are formulated as a set $V = \{(m_i, e_{i,j}) \mid \forall e_{i,j} \in E_i, \forall m_i \in M\}$. Nodes are represented as ordered pairs of textual mentions and candidate entities, since the same entity may be found multiple times as a candidate for different textual mentions and each occurrence must be evaluated independently.

3.1 NE Candidate Generation

The first step in constructing a solution graph is to find all possible candidates for each NE mention in the query document. For each such mention the KB entry titles are searched to find all entries to which the mention could refer. This includes entries with titles that fully or partially contain the query mention and those that could be an acronym of the query mention. These candidate entries are paired with their textual mentions in the document to become nodes in the solution graph.

3.2 Initial Confidence

Initial confidence $IConf(e_{i,j})$ is an independent feature of the NE candidate regardless of other candidates in the document. This confidence may be calculated locally using the local mention context, or globally using, e.g., the Freebase popularity score for the KB entry (Bollacker et al., 2008).

Local NE Candidate Confidence: The local confidence is computed by a similarity measure between the NE mention in the query document and the KB entry of the candidate entity. We propose four different measures to be used in the disambiguation phase.

cos: The cosine similarity between the named entity textual mention and the KB entry title.

jwSim: While the cosine similarity between a textual mention in the document and the candidate NE title in the KB is widely used in NED, this similarity is a misleading feature. For example, the textual mention “Essex” may refer to either of the following candidates “Essex County Cricket Club” or “Danbury, Essex”, both of which are returned by the candidate generation process. The cosine similarity between “Essex” and “Danbury, Essex” is higher than that between “Essex” and “Essex County Cricket Club”, which is not helpful in the NED setting. We adopted a new mention-candidate similarity function, $jwSim$, which uses Jaro-Winkler similarity as a first estimate of the initial confidence value for each candidate. This function considers all terms found in the candidate entity KB entry title, but not in the textual mention as disambiguation terms. The percentage of disambiguation terms found in the query document is used to boost in the initial $jwSim$ value, in addition to an acronym check (whether the NE textual mention could be an acronym for a specific candidate entity title). Experiments show that $jwSim$ performs much better than cos.

ctxt: The cosine similarity between the sentence containing the NE mention in the query document and the textual description of the candidate NE in the KB (we use the first section of the Wikipedia article as the candidate entity description).

Global NE Candidate Confidence: Global confidence is a measure of the global importance of the candidate entity. Entity popularity has been used successfully as a discriminative feature for NED (Nebhi, 2013). Freebase provides an API to get an entity’s popularity score (**FB**), which is computed during Freebase indexing. This score is a function of the entity’s inbound and outbound link counts in Freebase and Wikipedia[2]. The initial confidence is not normalized across all NEs because each score is calculated independently. Initial confidence scores of all candidates for a single NE mention are normalized to sum to 1.

3.3 Entity Coherence

Entity coherence refers to the real world relatedness of different entities which are candidate interpretations of different textual mentions in the document. It is not based on context, so it is always the same regardless of the query document. Coherence is represented as an edge between nodes in the solution graph. We used two measures for coherence, described as follows:

[2] https://developers.google.com/freebase/v1/search

Ref: Uses the Wikipedia documents for both entity candidates to check if either document has a link to the other. This relation is directed, but we assume an inverse relation also exists; so this relation is represented as undirected.

$$\text{Ref}(e_i, e_j) = \begin{cases} 1, & \text{if } e_i \text{ or } e_j \text{ refers to the other} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

JProb: An estimate of the probability of both entities appearing in the same sentence. Wikipedia documents are used to estimate this probability, as shown in (2), where $S(e)$ is the set of all sentences that contain the entity e and S the set of sentences containing any entity references.

$$\text{JProb}(e_i, e_j) = \frac{|S(e_i) \bigcap S(e_j)|}{|S|} \quad (2)$$

4 Disambiguation

The solution graph contains all possible candidates for each NE mention in the document. Each candidate has an initial confidence, with some connected by association relations. The disambiguation phase ranks all nodes in the solution graph and selects the best from the candidate list for each NE textual mention. The process of disambiguation consists of three steps. The first step is initial graph ranking, where all nodes are ranked according to the link structure. The second step is to re-rank the nodes by combining the graph rank with the initial confidence. The highest rank is not always correct, so in the third step a selection algorithm is used to choose the best candidate.

Graph Ranking: The links between different candidates in the solution graph represent real world relations. These relations may be used to reliably boost relevant candidates. All nodes in the graph are ranked according to these relations using PR. Initial confidence is used as an initial rank for the graph nodes, while entities' coherence measures are used as link weights which play a role in distributing a node's rank over its outgoing nodes.

Candidate Re-ranking: A problem with PageRank for our purposes is the dissipation of initial node weight (confidence) over all outgoing nodes. The final rank of a node is based solely on the importance of incoming nodes and the initial confidence play no further role. In our case this is not appropriate, so the final rank for each mention is determined after graph ranking, by combining the graph rank with the initial confidence.

Let us refer to the graph rank of a candidate as $PR(e_i)$. Two combination schemes are used:

$$R_s(e_{i,j}) = IConf(e_{i,j}) + PR(e_{i,j}) \quad (3)$$

$$R_m(e_{i,j}) = IConf(e_{i,j}) \times PR(e_{i,j}) \quad (4)$$

Named Entity Selection: The simplest approach is to select the highest ranked entity in the list for each mention m_i according to equation 5, where R could refer to R_m or R_s. However, we found that a dynamic choice between the re-ranking schemes, based on the difference between the top two candidates, as described in algorithm 1 and indicated by e^g,works best. The underlying intuition of this algorithm is that a greater difference between the top ranks reflects more confident discrimination between candidates. So, the two combination schemes assign different ranks to the candidates and the algorithm selects the scheme which appears more discriminative.

$$\hat{e}_i = \underset{e_{i,j}}{\text{argmax}}\ R(e_{i,j}) \quad (5)$$

```
Data: Two lists, R1 and R2, of candidates $E_i$, where R1
      is ranked using $R_s$, and R2 is ranked using $R_m$
Result: One NE $e_i^g$
Sort R1 and R2 in descending order;
R1diff = R1[0]-R1[1];
R2diff = R2[0]-R2[1];
if R1diff > R2diff then
    return highest rank scored entity of R1
else
    return highest rank scored entity of R2
end
```

Algorithm 1: Selection Algorithm

5 Experiments and Results

We used AIDA dataset[3], which is based on the CoNLL 2003 data for NER tagging. All mentions are manually disambiguated against Wikipedia (Hoffart et al., 2011). This dataset contains 1393 documents and 34,965 annotated mentions. We only consider NE mentions with an entry in the Wikipedia KB, ignoring the 20% of query mentions (7136) without a link to the KB, as Hoffart did. Micro-averaged and macro-averaged accuracy are used for evaluation. In this context micro-averaged accuracy corresponds to the proportion of textual mentions correctly disambiguated while macro-averaged accuracy corresponds to the proportion of textual mentions correctly disambiguated per entity, averaged over all entities.

5.1 Results

Initially, we evaluated the performance of two baselines. One is a setup where a ranking based solely on different initial confidence scores is used

[3] http://www.mpi-inf.mpg.de/yago-naga/aida/

	$IConf$	PR_C	PR_I	PR_{IC}	Cucerzan	Kulkarni	Hoffart	Shirakawa	Alhelbawy
A_{macro}	78.09	80.98	84.19	82.80	43.74	76.74	81.91	83.02	74.18
A_{micro}	80.55	83.59	87.59	86.10	51.03	72.87	81.82	82.29	78.49

Table 1: Results comparison between Proposed Approach and State-of-the-art

	PR		e^g	
$IConf$	A_{micro}	A_{macro}	A_{micro}	A_{macro}
cos	70.6	60.83	78.41	72.35
jwSim	70.61	60.94	83.16	78.28
ctxt	70.61	60.83	75.45	65.22
freebase	71.78	81.07	87.59	84.19

Table 2: Results using initial confidence (PR_I)

	PR		e^g	
Edge Weight	A_{micro}	A_{macro}	A_{micro}	A_{macro}
$Jprob$	66.52	55.83	83.31	80.38
Ref	67.48	59.76	81.80	78.53
$prob + refs$	72.69	65.71	83.46	80.69

Table 3: Results using weighted edges (PR_C)

for candidate selection, i.e. without using PR. In this setup a ranking based on Freebase popularity does best, with micro- and macro-averaged accuracy scores of 80.55% and 78.09% respectively. This is a high baseline, close to the state-of-the-art. Our second baseline is the basic PR algorithm, where weights of nodes and edges are uniform (i.e. initial node and edge weights set to 1, edges being created wherever REF or JProb are not zero). Micro and macro accuracy scores of 70.60% and 60.91% were obtained with this baseline.

To study graph ranking using PR, and the contributions of the initial confidence and entity coherence, experiments were carried out using PR in different modes and with different selection techniques. In the first experiment, referred to as PR_I, initial confidence is used as an initial node rank for PR and edge weights are uniform, edges, as in the PR baseline, being created wherever REF or JProb are not zero. Table 2 shows the results both before re-ranking, i.e. using only the PR score for ranking, and after re-ranking using the dynamic selection scheme e^g. When comparing these results to the PR baseline we notice a slight positive effect when using the initial confidence as an initial rank instead of uniform ranking. The major improvement comes from re-ranking nodes by combining initial confidence with PR score.

In our second experiment, PR_C, entity coherence features are tested by setting the edge weights to the coherence score and using uniform initial node weights. We compared JProb and Ref edge weighting approaches, where for each approach edges were created only where the coherence score according to the approach was non-zero. We also investigated a variant, called JProb + Ref, in which the Ref edge weights are normalized to sum to 1 over the whole graph and then added to the JProb edge weights (here edges result wherever JProb or Ref scores are non-zero). Results in Table 3 show the $JProb$ feature seems to be more discriminative than the Ref feature but the combined $Jprob + Ref$ feature performs better than each separately, just outperforming the baseline. We used the best initial confidence score (Freebase) for re-ranking. Again, combining the initial confidence with the PR score improves the results.

	e^g(jwSim)		e^g(freebase)	
Edge Weight	A_{micro}	A_{macro}	A_{micro}	A_{macro}
$Jprob$	82.56	76.16	86.29	82.77
Ref	78.61	71.12	83.16	80.01
$Jprob + Ref$	81.97	75.63	86.10	82.80

Table 4: Results using $IConf$ and weighted edges PR_{IC}

Finally, Table 4 shows the accuracy when using different combinations of initial confidence and entity coherence scores just in the case when reranking is applied. Here the $jprob + refs$ combination does not add any value over $jprob$ alone. Interestingly using initial confidence with differentially weighted edges does not show any benefit over using initial confidence and uniformly weighted edges (Table 2).

To compare our results with the state-of-the-art, we report Hoffart et al.'s (2011) results as they reimplemented two other systems and also ran them over the AIDA dataset. We also compare with Alhelbawy and Gaizauskas (2013) and Shirakawa et al. (2011) who carried out their experiments using the same dataset. Table 1 presents a comparison between our approach and the state-of-the-art and shows our approach exceeds the state-of-the-art. Futhermore our approach is very simple and direct to apply, unlike Hoffart et al.'s and Shirakawa et al.'s which are considerably more complex. Also, our approach does not need any kind of training, as does the Alhelbawy approach.

6 Conclusion

Our results show that Page-Rank in conjunction with re-ranking by initial confidence score can be used as an effective approach to collectively disambiguate named entity textual mentions in a document. Our proposed features are very simple and easy to extract, and work well when employed in PR. In future work we plan to explore enriching the edges between nodes by incorporating semantic relations extracted from an ontology.

References

Eneko Agirre and Aitor Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41. Association for Computational Linguistics.

Ayman Alhelbawy and Robert Gaizauskas. 2013. Named entity disambiguation using hmms. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, volume 3, pages 159–162.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250, New York, NY, USA. ACM.

Razvan C. Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *EACL*. The Association for Computer Linguistics.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL*, volume 6, pages 708–716.

Yoan Gutiérrez, Sonia Vázquez, and Andrés Montoyo. 2011. Word sense disambiguation: a graph-based approach using n-cliques partitioning technique. In *Natural Language Processing and Information Systems*, pages 112–124. Springer.

Yoan Gutiérrez, Sonia Vázquez, and Andrés Montoyo. 2012. A graph-based approach to wsd using relevant semantic trees and n-cliques model. In *Computational Linguistics and Intelligent Text Processing*, pages 225–237. Springer.

Xianpei Han and Le Sun. 2011. A generative entity-mention model for linking entities with knowledge base. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 945–954. 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*, pages 765–774. 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*, pages 782–792. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1148–1158. Association for Computational Linguistics.

Paul McNamee and Hoa Trang Dang. 2009. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, volume 17, pages 111–113.

David Milne and Ian H Witten. 2008. Learning to link with wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Kamel Nebhi. 2013. Named entity disambiguation using freebase and syntactic parsing. In CEUR-WS.org, editor, *Proceedings of the First International Workshop on Linked Data for Information Extraction (LD4IE 2013) co-located with the 12th International Semantic Web Conference (ISWC 2013)*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November. Previous number = SIDL-WP-1999-0120.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1375–1384. Association for Computational Linguistics.

Masumi Shirakawa, Haixun Wang, Yangqiu Song, Zhongyuan Wang, Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio. 2011. Entity disambiguation based on a. Technical report, Technical report, Technical Report MSR-TR-2011-125, Microsoft Research.

Ravi Sinha and Rada Mihalcea. 2007. Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In *Semantic Computing, 2007. ICSC 2007. International Conference on*, pages 363–369. IEEE.

Wenpu Xing and Ali Ghorbani. 2004. Weighted pagerank algorithm. In *Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on*, pages 305–314. IEEE.