Personalized Page Rank for Named Entity Disambiguation

Maria Pershina **Yifan He** **Ralph Grishman**
Computer Science Department
New York University
New York, NY 10003, USA
{pershina,yhe,grishman}@cs.nyu.edu

Abstract

The task of Named Entity Disambiguation is to map entity mentions in the document to their correct entries in some knowledge base. We present a novel graph-based disambiguation approach based on Personalized PageRank (PPR) that combines local and global evidence for disambiguation and effectively filters out noise introduced by incorrect candidates. Experiments show that our method outperforms state-of-the-art approaches by achieving 91.7% in micro- and 89.9% in macroaccuracy on a dataset of 27.8K named entity mentions.

1 Introduction

Name entity disambiguation (NED) is the task in which entity mentions in a document are mapped to real world entities. NED is both useful on its own, and serves as a valuable component in larger Knowledge Base Construction systems (Mayfield, 2014).

Since the surge of large, publicly available knowledge bases (KB) such as Wikipedia, the most popular approach has been linking text mentions to KB nodes (Bunescu and Paşca, 2006). In this paradigm, the NED system links text mentions to the KB, and quite naturally utilizes information in the KB to support the linking process. Recent NED systems (Cucerzan, 2007; Ratinov et al., 2011; Alhelbawy and Gaizauskas, 2014) usually exploit two types of KB information: *local* information, which measures the similarity between the text mention and the a candidate KB node; and *global* information, which measures how well the candidate entities in a document are connected to each other, with the assumption that entities appearing in the same document should be coherent. Both types of features have their strengths and drawbacks: local features better encode similarity between a candidate and a KB node, but overlook the coherence between entities; global features are able to exploit interlinking information between entities, but can be noisy if they are used by their own, without considering information from the text and the KB (cf. Section 4).

In this paper, we propose to disambiguate NEs using a Personalized PageRank (PPR)-based random walk algorithm. Given a document and a list of entity mentions within the document, we first construct a graph whose vertices are linking candidates and whose edges reflects links in Wikipedia. We run the PPR algorithm on this graph, with the constraint that we only allow the highest scored candidate for each entity to become the start point of a hop. As all candidates but the correct one are erronous and probably misleading, limiting the random walk to start from the most promising candidates effectively filters out potential noise in the Personalized PageRank process.

Our method has the following properties: 1) as our system is based on a random walk algorithm, it does not require training model parameters ; 2) unlike previous PageRank based approaches in NED (Alhelbawy and Gaizauskas, 2014) which mainly rely on global coherence, our method is able to better utilize the local similarity between a candidate and a KB node (Section 3); and 3) we tailor the Personalized PageRank algorithm to only focus on one high-confidence entity at a time to reduce noise (Section 4).

2 Related Work

Early attempts at the NED tasks use local and surface level information. Bunescu and Paşca

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 238–243,
Denver, Colorado, May 31 – June 5, 2015.

(2006) first utilize information in a knowledge base (Wikipedia) to disambiguate names, by calculating similarity between the context of a name mention and the taxonomy of a KB node.

Later research, such as Cucerzan (2007) and Milne and Witten (2008) extends this line by exploring richer feature sets, such as coherence features between entities. Global coherence features have therefore been widely used in NED research (see e.g. (Ratinov et al., 2011), (Hoffart et al., 2011), and (Cheng and Roth, 2013)) and have been applied successfully in TAC shared tasks (Cucerzan, 2011). These methods often involve optimizing an objective function that contains both local and global terms, and thus requires training on an annotated or distantly annotated dataset.

Our system performs collective NED using a random walk algorithm that does not require supervision. Random walk algorithms such as PageRank (Page et al., 1999) and Personalized PageRank (Jeh and Widom, 2003) have been successfully applied to NLP tasks, such as Word Sense Disambiguation (WSD: (Sinha and Mihalcea, 2007; Agirre and Soroa, 2009)).

Alhelbawy and Gaizauskas (2014) successfully apply the PageRank algorithm to the NED task. Their work is the closest in spirit to ours and performs well without supervision. We try to further improve their model by using a PPR model to better utilize local features, and by adding constraints to the random walk to reduce noise.

3 The Graph Model

We construct a graph representation $G(V, E)$ from the document D with pre-tagged named entity textual mentions $M = \{m_1, ..., m_k\}$. For each entity mention $m_i \in M$ there is a list of candidates in KB $C_i = \{c_1^i, ..., c_{n_i}^i\}$. Vertices V are defined as pairs

$$V = \{ \ (m_i, c_j^i) \mid m_i \in M, c_j^i \in C_i \ \},$$

corresponding to the set of all possible KB candidates for different mentions in M. Edges are undirected and exist between two vertices if the two candidates are directly linked in the knowledge base, but no edge is allowed between candidates for the same named entity. Every vertex (m, c) is associated with an initial similarity score between entity mention m and candidate c (Figure 1).

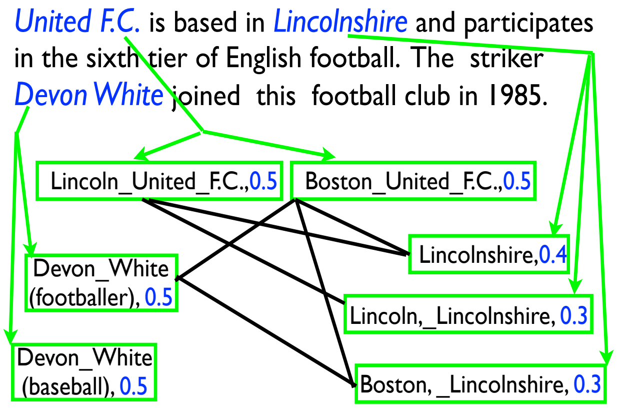
United F.C. is based in Lincolnshire and participates in the sixth tier of English football. The striker Devon White joined this football club in 1985.
Lincoln_United_F.C.,0.5
Boston_United_F.C.,0.5
Lincolnshire,0.4
Devon_White (footballer), 0.5
Lincoln,_Lincolnshire, 0.3
Devon_White (baseball), 0.5
Boston,_Lincolnshire, 0.3

Figure 1: A toy document graph for three entity mentions: *United F.C., Lincolnshire, Devon White*. Candidates and their initial similarity scores are generated for each entity mention.

3.1 Vertices

Candidates. Given named entity mentions M in the document, we need to generate all possible candidates for every mention $m \in M$. We first perform coreference resolution on the whole document and expand m to the longest mention in the coreference chain. We then add a Wikipedia entry c to the candidate set C_i for mention m_i if 1) the title of c is the same as the expanded form of m_i, or 2) string m_i redirects to page c, or 3) c appears in a disambiguation page with title m_i.

Initial Similarity. Initial similarity $iSim$ for vertex (m, c) describes how similar entity mention m to candidate c is. It is independent from other candidates in the graph G. We experiment with the local measure (localSim), based on the local information about the entity in the text, and the global measure (popSim), based on the global importance of the entity. Initial similarity scores of all candidates for a single named entity mention are normalized to sum to 1.

- **localSim**: The local similarity score is produced by a MaxEnt model trained on the TAC2014 EDL training data (LDC2014E15). MaxEnt features include string similarity between the title of the Wikipedia entry and the entity mention, such as edit distance, whether the text mention starts or ends with the Wikipedia title, etc; and whether they have the same type (e.g. person, organization, location, etc).

- **popSim**: We use the Freebase popularity as an alternative similarity measure. The Freebase popularity is a function of entity's incoming and outgoing link counts in Wikipedia and Freebase.[1]

3.2 Edges

Edges in our graph model represent relations between candidates. We insert an edge between two candidates if the Wikipedia entry corresponding to either of the two candidates contains a link to the other candidate. We assume that this relation is bidirectional and thus this edge is undirected.

There is a toy document graph in Figure 1 with three entity mentions and seven candidates: three candidates generated for *Lincolnshire*, and two candidates generated for *United F.C.* and *Devon White* each. Each graph node $e(m, c)$ is a pair of an entity mention m and a candidate c; every node is assigned an initial score, normalized across all candidates for the same entity. An edge is drawn between two candidates for different entities whenever there is a link from the Wikipedia page for one candidate to the Wikipedia page for another. There is no edge between candidates competing for the same entity.

4 The Challenge

A successful entity disambiguation algorithm would benefit from both the initial similarity between candidate and entity, as well as the coherence among entities in the same document. We assume that every entity can refer to at most one in the list of possible candidates, so all candidates except for the correct one for each entity are erroneous and will introduce noise into the document graph. Based on this observation, we contend that the typical random walk approach, which computes coherence of one candidate to the whole graph, is not suitable for our scenario. To address this problem, we propose to consider pairwise relations between every two nodes, given by PPR scores, compute the contribution of every node to the coherence of the other, and impose *aggregation constraints* to avoid redundant contributions.

4.1 Personalized PageRank

The PageRank algorithm considers random walk on a graph, where at each step with probability ϵ (teleport probability) we jump to a randomly selected node on a graph, and with probability $1 - \epsilon$ we follow a random outgoing edge of the current node. Stationary distribution of this walk gives PageRank weights associated with each node. Personalized PageRank is the same as PageRank, except that all teleports are made to the same source node, for which we are personalizing the PageRank.

4.2 Coherence and Constraints

The $coherence$ of the node e to the graph G quantifies how well node e "fits" into this graph. Intuitively, pairwise weights $PPR(s \rightarrow e)$ represent relationships between nodes in the graph: the higher the weight is, the more relevant endpoint e is for the source s. Candidate nodes in the graph have different quality, measured by their initial similarity $iSim$. Thus, coherence of the node e to the graph G due to the presence of node s is given by

$$coh_s(e) = PPR(s \rightarrow e) \cdot iSim(s), \qquad (1)$$

where relevance *e* for *s* is weighted by the $iSim(s)$, which is the similarity between entity e and candidate s. We experiment with a MaxEnt-trained local score and the Freebase popularity as the $iSim$ in Section 5.

We observe that summing the contributions $coh_s(e)$ for all nodes $s \in V$ would accumulate noise, and therefore impose two *aggregation constraints* to take into account this nature of document graph G. Namely, to compute coherence $coh(e)$ of the node $e(m, c)$, corresponding to the entity mention m and the candidate c, to the graph G we enforce:

(c1) ignore contributions from candidate nodes competing for an entity m;
(c2) take only one, highest contribution from candidate nodes, competing for an entity $m' \neq m$;

The first constraint **(c1)** means that alternative candidates $\bar{e}(m, \bar{c})$, generated for the same entity mention m, should not contribute to the coherence of $e(m, c)$, as only one candidate per entity can be correct. For the same reason the second constraint **(c2)** picks the single candidate node $s(m', c')$ for entity $m' \neq m$ with the highest contribution $coh_s(e)$ towards e. So these constraints guarantee that exactly one and the $most\ relevant$ candidate per entity will contribute

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

to the coherence of the node e. Thus, the set of contributors towards $coh(e)$ is defined as

$$\begin{aligned} &CONTR_{e(m,c)} = \\ &\{\ (m', \operatorname*{argmax}_{c} coh_{(m',c)}(e)\ ) \in V,\ m' \neq m\ \} \end{aligned} \tag{2}$$

Then coherence of the node e to graph G is given by

$$coh(e) = \sum_{s \in CONTR_{e(m,c)}} coh_s(e) \tag{3}$$

Consider the example in Figure 1, which has two connected components. Candidate Devon_White_(baseball) is disconnected from the rest of the graph and can neither contribute towards any other candidate nor get contributions from other nodes. So its coherence is zero. All other candidates are connected, i.e. belong to the same connected component. Thus, the random walker, started from any node in this component, will land at any other node in this component with some positive likelihood.

Let us consider the $CONTR_{e(m,c)}$ for entity mention *m = Lincolnshire* and candidate c = Lincolnshire, 0.4,. Without our constraints, nodes Devon_White_(footballer), 0.5, Lincoln_United_F.C., 0.5, Boston_United_F.C., 0.5, Lincoln_Lincolnshire, 0.3, Boston_Lincolnshire, 0.3 can all potentially contribute towards coherence of Lincolnshire, 0.4.

However, **(c1)** and **(c2)** will eliminate contribution from some of the candidates: Constraint **(c1)** does not allow Lincoln_Lincolnshire, 0.3 and Boston_Lincolnshire, 0.3 to contribute, because they compete for the same entity mention as candidate Lincolnshire, 0.4; constraint **(c2)** will allow only one contribution from either Lincoln_United_F.C., 0.5 or Boston_United_F.C., 0.5 whichever is bigger, since they compete for the same entity mention *United F.C.*. Therefore, set $CONTR_{e(m,c)}$ for entity mention *m = Lincolnshire* and candidate c = Lincolnshire, 0.4, will contain only two contributors: candidate Devon_White_(footballer), 0.5, for entity mention *Devon_White,* and exactly one of the candidates for entity mention *United F.C.*

4.3 PPRSim

Our goal is to find the best candidate for every entity given a candidate's coherence and its initial similarity to the entity. To combine the coherence score $coh(e)$ with $iSim(e)$, we weight the latter with an average value of PPR weights used in coherence computation (3) across all nodes in the document graph $G(V, E)$:

$$PPR_{avg} = \frac{\sum_{e \in V} \sum_{s \in CONTR_e} PPR(s \to e)}{|V|} \tag{4}$$

Thus, the final score for node e is a linear combination

$$score(e) = coh(e) + PPR_{avg} \cdot iSim(e) \tag{5}$$

If the document graph has no edges then PPR_{avg} is zero and for any node e its coherence $coh(e)$ is zero as well. In this case we set $score(e)$ to its initial similarity $iSim(e)$ for all nodes e in the graph G. Finally, PPRSim disambiguates entity mention m with the highest scored candidate $c \in C_m$:

$$disambiguate(m) = \operatorname*{argmax}_{c \in C_m}\ score(m, c) \tag{6}$$

To resolve ties in (6) we pick a candidate with the most incoming wikipedia links.

Thus, candidate Devon_White_(footballer), 0.5 in Figure 1 will get higher overall score than its competitor, Devon_White_(baseball), 0.5. Their initial scores are the same, 0.5, but the latter one is disconnected from other nodes in the graph and thus has a zero coherence. So, entity mention *Devon White* will be correctly disambiguated with the candidate Devon_White_(footballer), 0.5. This candidate is directly connected to Boston_United_F.C., 0.5 and has a shortest path of length 3 to Lincolnshire_United_F.C., 0.5, and therefore contributes more towards Boston_United_F.C., 0.5, and boosts its coherence to make it the correct disambiguation for *United F.C.* Similarly, *Lincolnshire* is correctly disambiguated with Boston_Lincolnshire_F.C., 0.3.

5 Experiments and Results.

Data. For our experiments we use dataset AIDA[2]. All textual entity mentions are manually disambiguated against Wikipedia links (Hoffart et al.,

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

Models	Cucerzan	Kulkarni	Hoffart	Shirakawa	Alhelbawy	iSim	PPR	PPRSim
Micro	51.03	72.87	81.82	82.29	87.59	62.61	85.56	91.77
Macro	43.74	76.74	81.91	83.02	84.19	72.21	85.86	89.89

Table 1: Performance of PPRSim compared to baselines and state-of-the-art models on AIDA dataset. Baselines iSim and PPR choose a candidate with the highest initial similarity or coherence correspondingly.

2011). There are 34,965 annotated mentions in 1393 documents. Only mentions with a valid entry in the Wikipedia KB are considered (Hoffart et al., 2011), resulting in a total of 27,816 mentions. We use a Wikipedia dump from June 14, 2014, as the reference KB. Our set of candidates is publicly available for experiments[3].

Evaluation. We use two evaluation metrics: (1) Microaccuracy is the fraction of correctly disambiguated entities; (2) Macroaccuracy is the proportion of textual mentions, correctly disambiguated per entity, averaged over all entities.

PPR. We adopt the Monte Carlo approach (Fogaras and Racz, 2004) for computing Personalized PageRank. It performs a number of independent random walks for every source node and takes an empirical distribution of ending nodes to obtain PPR weights with respect to the source. We initialized 2,000 random walks for every source node, performed 5 steps of PPR, and computed PPR weights from all iterations dropping walks from the first one. The teleport probability is set to 0.2.

Baselines. We performed a set of experiments using initial similarity and Personalized PageRank weights. Model iSim uses only Freebase scores and achieves microaccuracy of 62.61% (Table 1). PPR model picks a candidate with highest coherence, computed in (3), where no initial similarity is used ($iSim \equiv 1.0$) and no constraints are applied. It has microaccuracy of 85.56%. This is a strong baseline, proving that coherence (3), solely based on PPR weights, is very accurate. We also reimplemented the most recent state-of-the-art approach by Alhelbawy (2014) based on the PageRank. We ran it on our set of candidates with freebase scores and got 82.2% and 80.2% in micro- and macroaccuracy correspondingy.

PPRSim	Micro	Macro
$iSim \equiv 1.0$	85.56	85.86
$iSim =$ localSim	87.01	86.65
$iSim =$ popSim	90.26	88.98
+(c1)	90.52	89.21
+(c2)	91.68	89.78
+(c1),(c2)	91.77	89.89

Table 2: Performance of PPRSim with different initial similarities and constraints.

Results. We observe that PPR combined with global similarity popSim achieves a microaccuracy of 90.2% (Table 2). Adding constraints into the coherence computation further improves the performance to 91.7%. Interestingly, (c2) is more accurate than (c1). When put together, (c1)+(c2) performs better than each individual constraint (Table 2). Thus, combining coherence and initial similarity via (5) improves both micro- and macroaccuracy, outperforming state-of-the-art models (Table 1).

6 Conclusion and Future Work

In this paper we devise a new algorithm for collective named entity disambiguation based on Personalized PageRank. We show how to incorporate pairwise constraints between candidate entities by using PPR scores and propose a new robust scheme to compute coherence of a candidate entity to a document. Our approach outperforms state-of-the-art models and opens up many opportunities to employ pairwise information in NED. For future work, we plan to explore other strategies and constraints for noise reduction in the document graph.

[3] https://github.com/masha-p/PPRforNED

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 ACL*, pages 33–41, Athens, Greece.

Ayman Alhelbawy and Robert Gaizauskas. 2014. Graph Ranking for Collective Named Entity Disambiguation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 75–80.

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796, Seattle, WA.

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*, pages 708–716.

Silviu Cucerzan. 2011. Tac entity linking by performing full-document entity extraction and disambiguation. In *Proceedings of the 2011 TAC Workshop*, pages 708–716.

Fogaras and Racz. 2004. Towards scaling fully personalized page rank. In *Proceedings of the 3rd Workshop on Algorithms and Models for the Web-Graph (WAW)*, pages 105–117.

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.

Glen Jeh and Jennifer Widom. 2003. Scaling personalized web search. In *Proceedings of the 12th International Conference on World Wide Web*, pages 271–279.

James Mayfield. 2014. Cold start knowledge base population at tac 2014. In *Proceedings of the 2014 TAC Workshop*.

David Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 509–518.

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.

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*, pages 1375–1384, Portland, OR.

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