Entity Linking with a Paraphrase Flavor

Maria Pershina, Yifan He, Ralph Grishman
New York University
719 Broadway, 7th floor, New York, NY 10003, USA
{pershina,yhe,grishman}@cs.nyu.edu

Abstract

The task of Named Entity Linking is to link entity mentions in the document to their correct entries in a knowledge base and to cluster NIL mentions. Ambiguous, misspelled, and incomplete entity mention names are the main challenges in the linking process. We propose a novel approach that combines two state-of-the-art models — for entity disambiguation and for paraphrase detection — to overcome these challenges. We consider name variations as paraphrases of the same entity mention and adopt a paraphrase model for this task. Our approach utilizes a graph-based disambiguation model based on Personalized Page Rank, and then refines and clusters its output using the paraphrase similarity between entity mention strings. It achieves a competitive performance of 80.5% in $\mathbf{B}^3$+$\mathbf{F}$ clustering score on diagnostic TAC EDL 2014 data.

Keywords: disambiguation, linking, pagerank

1. Introduction

Entity Linking (EL), such as the EL track at NIST Text Analysis Conference Knowledge Base Population (TAC-KBP), aims to link a given named entity mention from a source document to an existing Knowledge Base (KB) (Ji et al., 2014). In TAC-KBP, all entity mentions linked to the same KB entry are considered to form a cluster, while unlinked entities (NILs) have to be clustered as well.

Linking raw entity mentions in a document to real world entities is useful on its own and serves as a valuable component in larger Knowledge Base Construction systems (Mayfield, 2014), e.g. the Cold Start track of TAC KBP program where the goal is to develop an automatic system to construct a KB from scratch. In the Wikification community (Bunescu and Paşca, 2006) text mentions are linked to Wikipedia, a large and publicly available knowledge base.

There are two paradigms to solve the EL problem: local, non-collective approaches for Entity Linking resolve one mention at a time relying on a context and local features, while collective approaches try to disambiguate the set of relevant mentions simultaneously assuming that entities appearing in the same document should be coherent (Cucerzan, 2007; Kulkarni et al., 2009; Ratinov et al., 2011; Hoffart et al., 2011; Alhelbawy and Gaizauskas, 2014; Pershina et al., 2015b). Our approach in this paper is based on PPRSim, a state-of-the-art *collective* model for named entity disambiguation, utilizing the Personalized PageRank (PPR) algorithm (Pershina et al., 2015b).

Nevertheless, we still try to capture the local similarity between the entity mention and its candidates in our model. To measure the local similarity, we propose to use an approach which was proven effective for paraphrase detection. For this purpose we adopt the state-of-the-art ASOBEK paraphrase model (Eyecioglu and Keller, 2015). It was developed for paraphrase identification in Twitter and was ranked first among 19 teams on the Paraphrase In Twitter (PIT) 2015 task. It uses six simple character and word features and trains an SVM. This universal system is trained on pairs of entity name variations, which we make publicly available, and provides an accurate similarity measure between entity mention strings.

We make the following contributions in this paper: 1) we propose to use the paraphrase model to measure the similarity between entity mention strings and provide publicly available training data for this model; 2) we efficiently incorporate this model into state-of-the-art entity disambiguation technique applied to the Entity Linking task and achieve the competitive result of 80.5% in $\mathbf{B}^3$+$\mathbf{F}$ score on the diagnostic TAC EDL 2014 dataset.

2. Document Graph

2.1. Candidates

Given a document with pre-tagged named entity textual mentions M, we generate all possible candidates for every entity mention $m \in M$. First, we 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 in one of three cases: 1) the title of c is the same as the expanded form of m_i; 2) string m_i redirects to page c; 3) c appears in a disambiguation page with title m_i.

2.2. Edges

To 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 information can flow in either direction and thus this edge is undirected.

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 KB candidates $C_i = \{c_1^i, ..., c_{n_i}^i\}$. Vertices V are defined as pairs

$$V = \{(m_i, c_j^i) | m_i \in M, c_j^i \in C_i\}, \quad (1)$$

corresponding to the set of all possible KB candidates for different mentions in M. Every vertex (m, c) has an initial similarity score $iSim(m, c)$ between m and c.

2.3. Initial Similarity

We split m and c into sets of tokens T_m and T_c and recognize two cases: 1) if T_m and T_c have any tokens in common

then their similarity is 1.0; 2) otherwise it is a reciprocal of the edit distance between m and c:

$$iSim(m,c) = \begin{cases} 1.0\,, & \text{if } T_m \cap T_c \neq \varnothing \\ \frac{1}{\text{edit}(m,c)}, & \text{otherwise} \end{cases} \quad (2)$$

Thus, the pairwise initial similarity for "*Buenos Aires*" vs "*Buenos Aires Wildlife Refugee*" and for "*Buenos Aires*" vs "*University of Buenos Aires*" equals to 1.0. This simple metric does not use any external resources and is applicable to all entity mentions even if they do not appear in a Freebase and Wikipedia, as opposed to the freebase popularity metric used in (Alhelbawy and Gaizauskas, 2014; Pershina et al., 2015b). We show in Section 4 that combining (2) with recent state-of-the-art NED technique (Pershina et al., 2015b) can efficiently utilize the document graph, that represents other entities, and perform competitively on the TAC EDL 2014 data.

3. Name Variations as Paraphrases

Depending on the text genre there can be different variations of the same named entity. Official sources such as newswire are more strict and more likely to use official titles to address people and organizations. The forum data, on the opposite, does not have such standards and may use interchangeably "Hillary Clinton" vs "Hitlery Clinton", "richardsdenni" vs "Rich Dennison", "mich state fair" vs "Michigan st Fair", "the blond demon" vs "le demon blond", etc. Edit distance is not a reliable clue to detect these kind of differences. For example, the above pairs have edit distance of 4, 12, 11, and 15 correspondingly.

One can view name variations as paraphrases of the same entity mention. There is no strict definition of a paraphrase (Bhagat and Hovy, 2013) and in linguistic literature paraphrases are most often characterized by an approximate equivalence of meanings across phrases. Thus, in a broad sense, detecting whether two phrases refer to the same entity mention is a particular case of the paraphrase problem. A growing body of research studied the problem of paraphrases in Twitter (Xu et al., 2015b; Guo et al., 2013; Guo and Diab, 2013; Socher et al., 2011), in bilingual data (Bannard and Callison-Burch, 2005), and even paraphrases between idioms (Pershina et al., 2015a). Finally, there was a new **P**araphrase **I**n **T**witter track (PIT) proposed in SemEval 2015 (Xu et al., 2015a). Most paraphrase models are tailored for a data set that they will be applied to. Thus, Twitter paraphrase models often make use of hashtags, timestamps, geotags, or require topic and anchor words (Xu et al., 2015b). None of this is applicable to named entity mentions.

Based on this observation, we focus on a holistic ASOBEK approach (Eyecioglu and Keller, 2015) for paraphrase identification in entity linking. The ASOBEK model uses simple character and word features and trains a linear SVM. This work is motivated by the set theory and every phrase is considered as a set of either character uni/bi-grams (C_1, C_2), or word uni/bi-grams (W_1, W_2). There are three types of features derived from these sets: 1) count of elements in a set, e.g. $|C_1|$ (length); 2) count of elements in the set overlap, e.g. $|C_1^{phrase_1} \cap C_1^{phrase_2}|$; 3) count of elements in the set union, e.g. $|C_1^{phrase_1} \cup C_1^{phrase_2}|$. (Eyecioglu and Keller, 2015) reported best performance using just six features:

$$\begin{gathered} |C_2^{phrase_1} \cap C_2^{phrase_2}|, \\ |C_2^{phrase_1} \cup C_2^{phrase_2}|, \\ |W_1^{phrase_1} \cap W_1^{phrase_2}|, \\ |W_1^{phrase_1} \cup W_1^{phrase_2}|, \\ |C_2^{phrase_1}|, \\ |C_2^{phrase_2}|. \end{gathered} \quad (3)$$

We adopt this model to our task for detecting name variations. Namely, we built our training data set of name variation pairs, extracted ASOBEK best features, and trained a linear SVM (Joachims, 2006)[1] on this data.

We tested the ASOBEK model for three different feature sets that were explored in original paper: 1) feature set that performed best (ASOBEK), six features (3) in total; 2) same as above plus length in words $|W_1^{phrase_1}|, |W_1^{phrase_2}|$, eight features in total; 3) same as above plus unigram features, twelve in total. We plot precision-recall curves for these three variations (Figure 2). First feature set performs slightly better confirming the result of (Eyecioglu and Keller, 2015); all three achieve maximal F-score around 92% with precision of 96% and recall 88%. For our experiments we use the first feature set that was proven to be the best in the original paper.

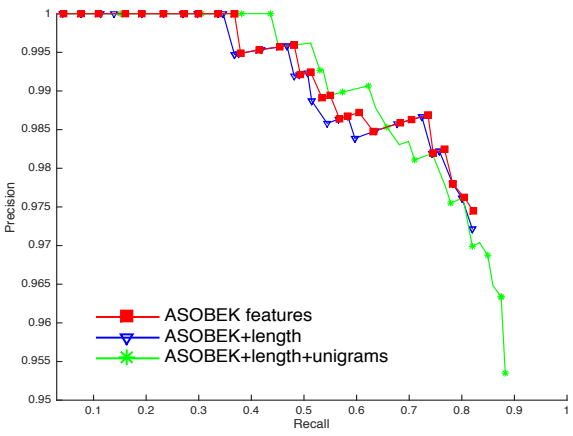

Figure 1: Performance of ASOBEK model with different feature sets applied to name variation task.

4. ParaLink

The most recent state-of-the-art entity disambiguation model PPRSim (Pershina et al., 2015b) runs Personalized PageRank (PPR) on the document graph and is based on intuition that pairwise weight $PPR(s \to e)$ measures how relevant endpoint e is for the source s. Then coherence of the node e to the graph G due to the presence of node s is computed as

$$coh_s(e) = PPR(s \to e) \cdot iSim(s) \quad (4)$$

[1] https://www.cs.cornell.edu/people/tj/svm_light/

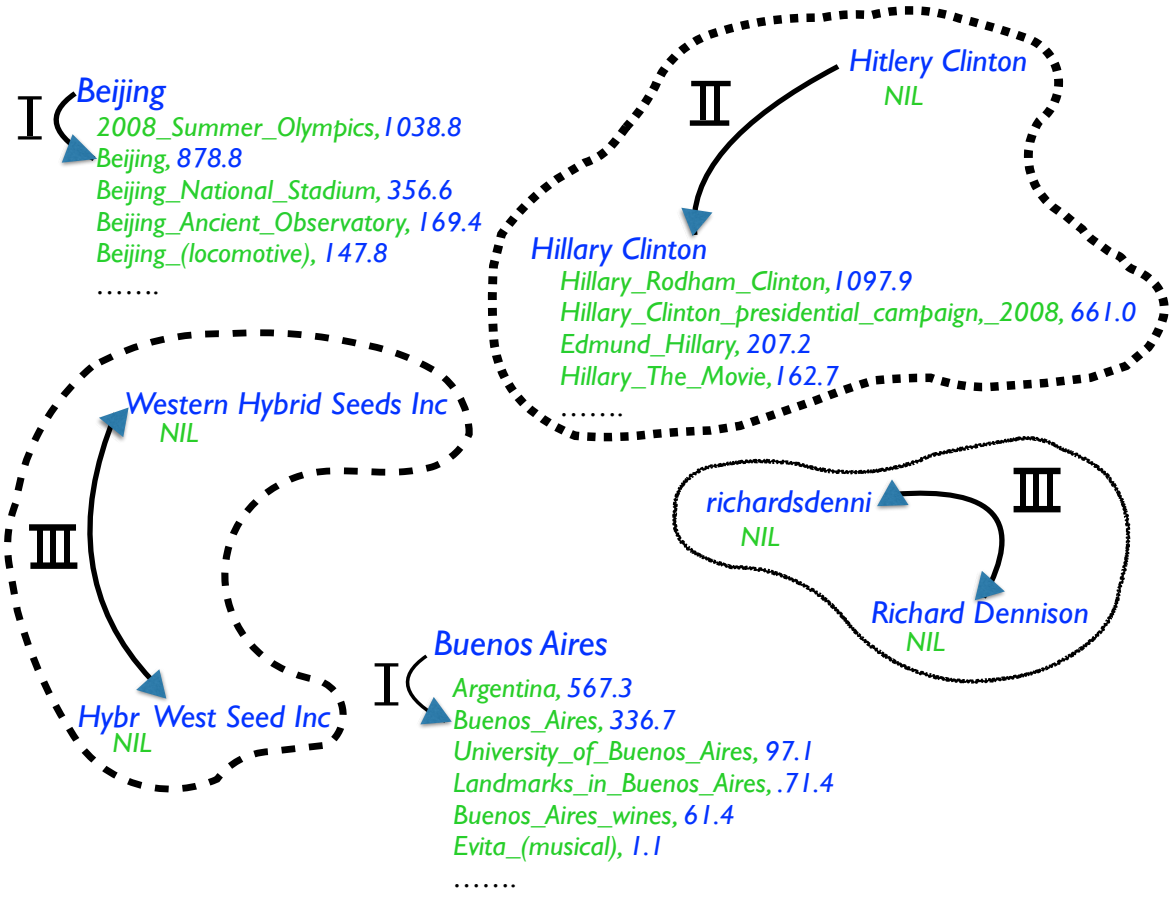

Figure 2: Examples of ParaLink refining and clustering steps I, II, III.

Since there can be only one correct candidate per entity, PPRSim imposes aggregation constraints to take only the highest contribution from candidate nodes, competing for the same entity. Finally, the total score for the node e is

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

where total coherence $coh(e)$ of node e to the graph is computed with respect to aggregation constraints, and initial similarity score $iSim(e)$ is weighted by an average value of PPR weights used in coherence computation.

However, this approach often ranks higher a popular candidate connected to many nodes in a graph over the correct but less popular one. In fact, running PPRSim on the AIDA dataset yields a precision of 91.7% while the correct disambiguation link is contained within the top three ranked candidates for more than 99% of entity mentions[2].

For example, the top candidate for mention *Buenos Aires* is the incorrect entity *Argentina*, generated from the disambiguation page. It is winning over the correct one *Buenos Aires*, ranked second, due to a larger amount of incoming links (56K vs 12K) and thus a better connected neighborhood in a document graph (34 vs 26 edges). These candidates are top ranked by PPRSim on a document graph. However, the second candidate is a perfect paraphrase of the textual entity mention, while the first one is not. Thus, using the similarity between the entity mention string and the KB entry title to select among the top-scoring candidates found by PPRSim can solve this problem (step I).

Entity disambiguation models usually assume that every entity mention has a valid KB entry and do not explicitly handle NIL entities. Thus NILs get clustered using the default one-name-per-cluster strategy. So, "Hitlery Clinton" will be clustered separately from "Hillary Clinton", "richardsdenni" will be separate from "Rich Dennison", etc. We propose to cluster every NIL candidate together with the most similar already linked entity mention if their paraphrase similarity is above a certain threshold obtained on a development dataset (step II).

Finally, NIL candidates, that were not assigned a link at the previous step, get clustered with the most similar NILs or constitute a singleton NIL cluster if no similar mentions can be found (step III). Thus ParaLink combines PPRSim with three additional refining steps based on paraphrase similarity between entity mention strings (Figure 2,3).

5. Experiments and Results.

5.1. Data

For our experiments we use the diagnostic TAC EDL 2014 dataset. Its training part consists of 158 documents with 5966 pretagged entity mentions; the test set contains 138 documents with 5234 pretagged entity mentions. All entity mentions are manually disambiguated against Wikipedia links, all NIL entities are clustered.

To train an ASOBEK model we extract name variations

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

Models	NYU(PR)	PPRSim	PPRSim + I	PPRSim + I + II	PPRSim + I + III	ParaLink	ParaLink*
Training data	76.2	78.4	79.1	79.5	79.2	79.7	79.8
Test data	76.3	78.9	80.0	80.3	80.2	80.5	80.7

Table 1: Performance of ParaLink in $\mathbf{B}^3$+**F** score compared to the baseline and state-of-the-art models on TAC EDL 2014 train/test datasets. NYU (PR): PageRank with one-name-per-cluster name clustering; PPRSim: Personalized PageRank as described in (Pershina et al. 2015); PPRSim+I/II/III: Combining PPRSim separately with steps in ParaLink; ParaLink: PPRSim with all steps I,II,III; ParaLink*: ParaLink scored on manually corrected TAC answer key.

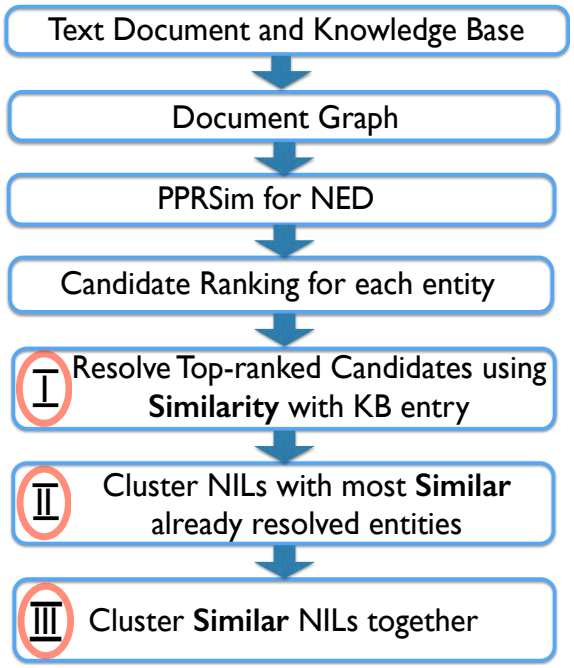

Figure 3: ParaLink diagram with refining and clustering steps I, II, III.

from the training data for TAC EDL 2014 task. Given entity clusters we pick pairs of entity mentions from the same cluster to create a set of name variations that refer to the same real world entity and we pair entity mentions from different clusters to have negative examples. Our training data consists of 1143 positive pairs and 1500 negative pairs, our test has 511 positive pairs and 1168 negative pairs. It is publicly available for future experiments.[3] We use TAC training data to tune for an optimal threshold for each step I, II, III.

5.2. Evaluation

We use the standard TAC EDL clustering metric $\mathbf{B}^3$+**F** to evaluate baseline and ParaLink models.

B-cubed cluster scoring compares clusters in the gold and response partition (Bagga and Baldwin, 1998). The B-cubed cluster precision is the weighted average of a per-element precision score. Precision of an element A is the following:

$$\mathbf{B}^3 Precision(A, goldPartition, resPartition) = \frac{|cluster(goldPartition, A) \cap cluster(resPartition, A)|}{|cluster(resPartition, A)|}$$

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

where $cluster(partition, A)$ is the cluster in the partition containing the element A; in other words, this is A's equivalence class and contains the set of all elements equivalent to A in the partition. Then each cluster in the gold partition is weighted equally, and each element is weighted equally within a cluster:

$$\mathbf{B}^3 ClusterPrecision(goldPartition, resPartition) = \sum_a \frac{\mathbf{B}^3 Precision(a, goldPartition, resPartition)}{|goldPartition| * |cluster(goldPartition, a)|}$$

Recall is defined dually by switching the roles of gold and response partitions, and the F1-measure is defined in the usual way.

A brief analysis of the answer key revealed some mistakes in the TAC annotation. By fixing the answer link for 6 mentions in the training data (from the total of 5966) and for 22 mentions in a test data (from the total of 5234) we improved $\mathbf{B}^3$+**F** by 0.1 and 0.2 correspondingly (Table 1). Our corrected answer keys are publicly available.

5.3. Baselines

We compare our model with several graph-based approaches. Our baseline is a faithful re-implementation of the NYU 2014 entity linking system based on PageRank (Alhelbawy and Gaizauskas, 2014; Nguyen et al., 2014; Heng et al., 2014). We compare it with the state-of-the-art PPRSim model for named entity disambiguation (Pershina et al., 2015b).

5.4. Results

We observe that the refined disambiguation process for PPRSim (step I) improves the performance from 78.4% to 79.1% on training, and from 78.9% to 80.0% on test datasets. Adding paraphrase clustering (step II and III) further improves the $\mathbf{B}^3$+**F** score to achieve 79.7% and 80.5% correspondingly. Thus we show that paraphrase similarity can be efficiently incorporated into the entity linking pipeline and improve the performance.

5.5. Discussion

Interestingly, performance of PageRank is about the same on both training and test data, while ParaLink achieves a better result on test dataset than on training one. The reason is that the fraction of discussion forum posts is slightly higher in test data than in training - about 20% vs 15%. ParaLink is particularly efficient for this type of data since it combines the power of disambiguation PPRSim model with ability to efficiently cluster misspelled and corrupted names, that are typical for forum posts. Thus it achieves a

better performance on a dataset with more informal documents.

6. Conclusion and Future Work

In this paper we discuss the problem of name variations for the entity linking task. We show how to adopt ASOBEK paraphrase model to solve this problem and how to incorporate it into the entity linking pipeline. Using paraphrase paradigm for the name variations problem opens new perspectives for future research in Information Extraction.

For the future work we will further explore the problem of name variations and will extend our graph-based approach for better NIL detection and clustering.

7. Bibliographical References

Alhelbawy, A. and Gaizauskas, R. (2014). Graph ranking for collective named entity disambiguation. In *Proceedings of the 52nd Meeting of Association for Computational Linguistics (ACL)*.

Bagga, A. and Baldwin, B. (1998). Algorithms for scoring coreference chains. In *Proceedings of the First International Conference on Language Resources and Evaluation Workshop on Linguistic Coreference (LREC)*.

Bannard, C. and Callison-Burch, C. (2005). Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Bhagat, R. and Hovy, E. (2013). What is a paraphrase? In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Bunescu, R. and Paşca, M. (2006). Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*.

Cucerzan, S. (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.

Eyecioglu, A. and Keller, B. (2015). Twitter paraphrase identification with simple overlap features and svms. In *Proceedings of the SemEval 2015*.

Guo, W. and Diab, M. (2013). Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Guo, W., Li, H., Ji, H., and Diab, M. (2013). Linking tweets to news: A framework to enrich short text data in social media. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Heng, J., Nothman, J., and Hachey, B. (2014). Overview of tac-kbp 2014 entity discovery and linking tasks. In *Proceedings of the TAC 2014*.

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S., and Weikum, G. (2011). Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792.

Joachims, T. (2006). Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 217–226.

Kulkarni, S., Singh, A., Ramakrishnan, G., and Chakrabart, S. (2009). Collective annotation of wikipedia entities in web text. In *Proceedings of the KDD 2009*.

Mayfield, J. (2014). Cold start knowledge base population at tac 2014. In *Proceedings of the 2014 TAC Workshop*.

Nguyen, T., He, Y., Pershina, M., Li, X., and Grishman, R. (2014). New york university 2014 knowledge base population systems. In *Proceedings of the Text Analysis Conference (TAC)*.

Pershina, M., He, Y., and Grishman, R. (2015a). Idiom paraphrases: Seventh heaven vs cloud nine. In *Proceedings of the EMNLP-LSDSem*, pages 76–82.

Pershina, M., He, Y., and Grishman, R. (2015b). Personalized page rank for named entity disambiguation. In *Proceedings of the NAACL-HLT*, pages 238–243.

Ratinov, L., Roth, D., Downey, D., and Anderson, M. (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.

Socher, R., Huang, E. H., Pennington, J., Ng, A. Y., and Manning, C. D. (2011). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*.

Xu, W., Callison-Burch, C., and Dolan, W. B. (2015a). Semeval-2015 task 1: Paraphrase and semantic similarity in twitter (pit). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*.

Xu, W., Ritter, A., Callison-Burch, C., Dolan, W. B., and Ji, Y. (2015b). Extracting lexically divergent paraphrases from twitter. In *Transactions of the Association for Computational Linguistics (TACL)*.