Local Named Entity Disambiguation with Neural Attention

Anonymous ACL submission

Abstract

Hello, My name is. what? my name is. who? My name is. what? my name is. who? My name is. what? my name is. who? My name is. what? my name is. who? My name is. what? my name is. who? My name is. what? my name is. who? My name is. what? my name is. who? My name is. what? my name is. who? My name is. what? my name is. who? My name is. what? my name is. who? My name is. what? my name is. who? My name is. what? my name is. who? My name is. what? my name is. who? My name is. what? my name is. who? My name is. what? my name is. who? Lorem ipsum.

1 Introduction

The intro is currently super-drafty (even after my pass). My comments are mainly intended to restructure the flow and point out our main arguments. We will refine it over and over again until we're sababa with it :)

Named Entity Disambiguation (NED) is the task of linking entity mentions within a fragment of text against a given knowledge base of entities, such as Freebase or Wikipedia. NED is a key component in Entity Linking (EL) systems, focusing on the disambiguation task itself independently from other tasks such as detecting mention bounds (Named Entity Recognition) and retrieving an high-recall set of candidate entities (Candidate Generation). Both NED and EL has been recognized as an important components in semantic parsing (**?**), as well as other NLP tasks.

These tasks don't really interest the NLP community these days... text categorization is considered virtually "solved", and IR is uncompetitive because Google, unlike academia, has the ability to A/B test its solutions.

Also, if we intend to perform some extrinsic evaluation (e.g. plug our NED into a QA system), we should name that task explicitly, and even cite the particular paper we intend to augment.

Question: are we dealing with finding the span as well, or given the span, are we just trying to link it to the right concept? The exact definition should appear here, and this distinction needs to be discussed in the background.

NED algorithms can broadly be divided into local and global approaches. Local algorithms disambiguate each mention independently using local context (e.g. the rest of the sentence), whereas global approaches assume some coherence among mentions within a single document, and try to disambiguate all mentions simultaneously. Global algorithms have significantly outperformed the local approach on standard datasets (Ratinov et al., 2011a; Guo and Barbosa, 2014; Pershina et al., 2015). However, most standard datasets are based on news corpora and Wikipedia, which are naturally coherent, well-structured, and rich in context. Other domains, such as web page fragments, social media, or questions, lack the sufficient coherence and context for global models to pay off.

add newer stuff

Potentially give citations for each domain.

Question: could we create a similar dataset of questions or tweets? I think this would provide a much more versatile benchmark, and perhaps allow more cases for your method to shine.

You start by telling the point about global vs local, and mentioning (too lightly) that current datasets favor global algorithms. Rather than delivering the punchline, you start talking about deep learning. I have restructured the intro to reflect the first point, and moved the DNN literature to the "background" section.

In this work, we investigate the task of NED in a setting where only local and noisy context is available. In particular, we create a dataset of $3.2M$ short fragments each containing a mention of a named entity, extracted from web pages. Our dataset contains $18K$ unique mentions linking to over $100K$ unique entities. This dataset is significantly larger than previously collected ones such as CoNLL, TAC and ACE(Hoffart et al., 2011; **?**; **?**), allowing us to train a deep neural network model.

We propose a novel neural network architecture based on Recurrent Neural Networks (RNNs) with an attention mechanism, where the RNN units model textual context as a sequence and the attention mechanism gives importance to contextual signals based on the specific candidate entity being evaluated. Our model differs from non-neural approaches by automatically learning feature representations for entity and context, allowing it to extract features from noisy and unexpected context patterns where it can be hard to manually design useful features. We differ from existing neural-based approaches by accounting for the sequential nature of textual context using RNNs and by devising an attention model that can reduce the impact of noise by assigning weights to different contextual signals based on the specific candidate entity being evaluated.

We also describe a novel method for initializing word and entity embeddings used in our model and demonstrate its importance for model performance and training efficiency.

YOTAM: training taking forever but approaching performance with initialization

This sentence, or the one/two that follows, needs to state the algorithmic innovation. By "innovation", I mean: what does it do differently from Globerson/Yamada?

Mention this as well: "We also describe a novel method for initializing word and entity embeddings used in our model and demonstrate its importance." What is its importance?

Incorporate the information from here into the new flow

We demonstrate our model greatly outperforms existing state-of-the-art NED algorithms on our web based dataset, showing that existing state-of-the-art methods are not optimal in such settings, and that our model can better model noisy and short context. In addition we evaluate our algorithm on CoNLL (Hoffart et al., 2011), a standard NED dataset, and show results comparable to other state-of-the-art local methods on a smaller and cleaner dataset. We conclude that RNNs are well suited for local disambiguation, but that there is still much room for improvement in real world scenarios where text is short, noisy and less coherent.

What are the main results? Conclusions? What have we learned from this paper? Why is it important that this paper is accepted?

2 Background

2.1 Related work

The first attempt to use Wikipedia as an knowledge base for local disambiguation was offered by Busco and Pasca in 2006 (Bunescu and Bunescu, 2006). Hoffart et al. suggested a collective global disambiguation graph-based framework, named AIDA, which employs a mention, context and coherency models. (Hoffart et al., 2011). Similar models were exploited in other global algorithms, such as the GLOW and the Relational Inference systems for the task of Wikification (Ratinov et al., 2011b; Cheng and Roth, 2013), when the latter extends the former by combing semantic knowledge with a classification model. Chisholm et al. incorporated web-links data from the Wikilinks dataset to push NED performance to a new level (Chisholm and Hachey, 2015a). More recently, a selective-context model, proposed by Lazic et al., has shown that only few context words are informative for the tasks of disambiguation (Lazic et al., 2015).

In the aforementioned studies two important observations stand out and influence our algorithmic approach. First, it has been shown that local disambiguation techniques produce a hard baseline to beat even by global NED extensions, which are not even engaged in most of the disambiguation task of traditional data sets (Ratinov et al., 2011b; **?**). This observation, which seems to be intuitive to the human reader, has led us to concentrate on generating a strong local context-based disambiguation solution. Moreover, we designed our model to have an attention mechanism, for reducing the effect of non-informative nearby words, following the results of Lazic et al.

survey CoNLL,ACE, TAC other tasks

Previous work of NED with NN

** He (2013)

** Sun (2015)

** francis (2016)

** yamada (2016)

This section provides readers who are less familiar with the literature the necessary information to understand your contribution. Things that need to appear in this section:
- Previous work on NED.
- An in-depth survey of the existing datasets and how they were built.
- Neural work on NED.
At the end of each paragraph/subsection, mention how this work improves upon / differs from what you just discussed.

I recommend restructuring the rest of the paper as follows:
Section 3: Dataset. Must begin with the rationale for why you're constructing it in this particular way. Should include some quantitative analysis of how this dataset differs from existing ones, e.g. number of examples, distribution of context size, distribution of possible candidates per mention, etc. You should convince that this is a fundamentally different dataset, and that it captures a very realistic scenario that is not captured in the current datasets.
Section 4: Algorithm / Model (Methodology is not the correct term).
Section 5: Evaluation. Should include error analysis as well (maybe as a different section). Should also discuss the qualitative observations, e.g. initializing embeddings helps, the dataset is much harder, etc.
Depending on how Section 5 turns out, we will restructure the remainder of the paper accordingly.

3 Web-Fragment based NED Dataset

We introduce a new large-scale NED dataset of web-fragments crawled from the web. Our dataset is derived from the Wikilinks dataset originally collected by Singh at el. (2012) for a cross-document co-reference task. cross-document co-reference entails clustering mentions referring to the same entity across a set of documents without consulting a predefined knowledge base of entities, and is many-a-time regarded as a downstream task for knowledge base population (KBP). Wikilinks was constructed by crawling the web and collecting hyperlinks linking to Wikipedia and the web context they appear in. The anchor texts act as mentions and the link targets in Wikipedia act as ground-truths. Wikilinks contains 40 million mentions covering 3 million entities and collected from over 10 million web pages.

Wikilinks can be seen as a large-scale, naturally-occurring and crowd-sourced dataset where thousands of human annotators provide ground-truths for mentions of interest. Its web sourcing entails every kind of noise expected from automatically gathered web content, including many faulty, misleading and peculiar ground truth labels on the one hand, and on the other hand noisy, malformed and incoherent textual context for mentions. While noise in crowd-sourced data is arguably a necessary trade-off for quantity, we believe the contextual noise in particular represents an interesting test-case that supplements existing standard datasets such as CoNLL (Hoffart et al., 2011), ACE and Wiki (Ratinov et al., 2011a) as these are all sourced from mostly coherent and well formed text such as news articles and Wikipedia pages. Wikilinks emphasizes utilizing strong and adaptive local disambiguation techniques, and marginalizes the utility of coherency based global approaches.

The original dataset exists in a number of formats, and we have chosen a version with only short local contexts[1] since it renders the size of the dataset a manageable 5Gb of compressed data (compared to 180Gb for the full texts). Following are the filtering and preprocessing steps used to create a NED evaluation dataset from Wikilinks:

- we resolved ground-truth links using a 7/4/2016 dump of the Wikipedia database[2]. The same dump was consistently used throughout this research. We used the *page* and *redirect* tables for resolution and kept the database *pageid* column as a unique identifier for Wikipedia pages (entities). To re-

[1] Available at http://www.iesl.cs.umass.edu/data/wiki-links

[2] Recent Wikipedia dumps are found at https://dumps.wikimedia.org/

duce loss of unresolved mentions due to malformed URLs we compared page names using a case-insensitive and normalized[3] title matching. We discarded mentions where the ground-truth could not be resolved, resulting in retention of 97% of the mentions.

- We collected all pairs of mention m and entity e appearing in the dataset and computed the following two statistics: how many times m refers to e: $\#\{e|m\}$ and the conditional probability of e given m: $p(e|m) = \#\{e|m\}/\sum_{e'}\#\{e'|m\}$. Examining these distributions revealed many mentions belong to two extremes: either they had very little ambiguity or had a number of candidate entities each appearing very few times. We have deemed the former to be unambiguous and not-interesting, and the latter to be suspected as noise with high probability. We therefore designed a procedure to filter both this cases: We retained only mentions for whom at least two ground-truth entities have $\#\{e|m\} \geq 10$ and $p(e|m) \geq 0.1$.
- Finally, We randomly permuted the order of mentions within the data and split it into train, evaluation and test set. We split the data 90%/10%/10% respectively. Since websites might include duplicate or closely related content we did not assign mentions into splits on an individual basis but rather collected all origin domains and assigned each domain along with all mentions collected from it into a split collectively.

This procedure aggressively filtered the dataset and we were left with $2.6M$ training, $300K$ test and $300K$ evaluation samples. We believe that doing so filters uninteresting cases while emitting a dataset that is large-scale yet manageable in size for research purposes. We note that we have considered filtering (m, e) pairs where $\#\{e|m\} \leq 3$ since these are suspected as additional noise however decided against this procedure as it might filter out long-tail entities, a case which was deemed interesting by the community.

4 Algorithm

Our DNN model is a discriminative model which takes a pair of local context and candidate entity, and outputs a likelihood for the candidate entity being correct. Both words and entities are represented using embedding dictionaries and we interpret local context as a window-of-words to the left and right of a mention. The left and right contexts are fed into a duo of Attentional RNNs (ARRN) components which process each side and produce a fixed length vector representation. The left context is fed in a forward manner while the right context is fed backwards into the model. Each Attentional RNN uses the candidate entity input to control its attention, allowing it to attend to the most discriminating parts of the context given the candidate at hand.

The output vectors generated by both Attentional RNNs and the embedding of the entity itself are then fed into a classifier network consisting of a hidden layer and an output layer with two output units and a softmax activation. The output units are trained to emit the likelihood of the candidate being a correct or corrupt assignment by optimizing a cross-entropy loss function.

We assume our model is only given examples of correct entity assignments during training and therefore automatically generate examples of corrupt assignments. For each $(context, entity)$ pair where $entity$ is a correct assignment for a given $contex$ we produce k corrupt examples with the same $context$ and a corrupt entity uniformly sampled from all entities in the dataset. Using the combined dataset of correct and corrupt examples our algorithm learns to separate correct assignments from the generated corrupt ones.

In our implementation we have set the hidden layer size to be 300 and used a ReLU non-linearity for this layer. Preliminary evaluations showed the width and depth of the classifier to be of little impact on performance, but using a ReLU nonlinearity was found to be important. We have also applied dropout with $p = 0.5$ to the hidden layer.

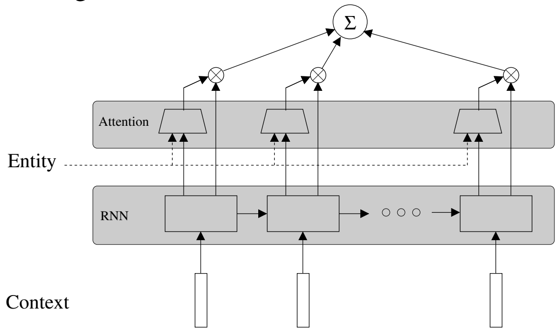
4.1 Attentional RNN component

Our Attentional RNN component is based on a general RNN unit fitted with an attention mechanism. The mechanics of the Attentional RNN component are depicted in Figure 1.

Equation 1 represents the general semantics of an RNN unit. An RNN reads a sequence of vectors $\{v_t\}$ and maintains a hidden state vector $\{h_t\}$. At each step a new hidden state is computed based on the previous hidden state and the next input vector

[3]Normalization was done using the unidecode python library

Figure 1: Attentional RNN Architecture



by a function f parametrized by Θ_1. The output at each step is computed from the hidden state using a function g parametrized by Θ_2. This allows the RNN to 'remember' important signals while scanning the context and to recognize signals spanning multiple words.

$$\begin{aligned} h_t &= f_{\Theta_1}(h_{t-1}, v_t) \\ o_t &= g_{\Theta_2}(h_t) \end{aligned} \tag{1}$$

In out implementation we have used a standard GRU unit (Cho et al., 2014), however any RNN can be a drop-in replacement. While an RNN unit can be used as-is in our model by feeding the last output vector o_t directly into the classifier network, we have implemented an attention mechanism that allows the model to be aware of the candidate entity it is evaluating when computing an output. Equation 2 details the equations governing the attention model.

$$\begin{aligned} &a_t \in \mathbb{R}; a_t = r_{\Theta_3}(o_t, v_{candidate}) \\ &a'_t = \frac{1}{\sum_{i=1}^{t} \exp\{a_i\}} \exp\{a_t\} \\ &o_{attn} = \sum_{i=1}^{t} a'_t o_t \end{aligned} \tag{2}$$

The main component in equation 2 is the function r, parametrized by Θ_3, which computes an attention value at each step using $v_{candidate}$, the candidate entity embedding, as a control signal. We use the softmax function to normalize the attention values such that $\sum_{i=1}^{t} a'_i = 1$ and compute the final output o_{attn} as a weighted sum of all the output vectors of the RNN. This allows the attention mechanism to decide on the importance of different context parts when examining a specific candidate. We parametrize our attention function r as a single layer NN as shown in equation 3 where A, B are the layer weights and b is a bias term.

$$r_{\Theta_3}(o_t, v_{candidate}) = Ao_t + Bv_{candidate} + b \tag{3}$$

4.2 Training initial word and entity embeddings

Training our model implicitly trains its dictionaries of both word and entity embedding by error back-propagation. However, as will be shown in section 5, we have found using pre-trained embeddings to significantly improve model performance/greatly reduce training time(??). To this end we have devised a Skip Gram with Negative Sampling (SGNS) (Mikolov et al., 2013) based training procedure that simultaneously trains both word and entity vectors in the same embedded space.

We use the word2vecf library[4] by Levy and Goldberg (2014a) that is adapted from word2vec code and allows to train on a dataset made of $(word, context)$ pairs rather then a textual corpus in string format, as is done in the original word2vec. We exploit this to redefine $context$ as a context entity rather then a contextual word.

We do this by considering each page in Wikipedia to represent a unique entity, enumerated by the $pageid$ identifier in Wikipedia database and having a textual description (the page itself). For each word $\{word_i\}$ in the page we add the pair $(word_i, pageid)$ to our dataset. We however limit our vocabularies by ignoring both rare words that appear less then 20 times and entities that have less then 20 words in their description.

As shown by Levy and Goldberg (2014b) training embeddings on this dataset using SGNS produces word and entity embedding that implicitly factorize the word-entity co-occurrence PPMI matrix. This matrix is closely related to the TFIDF word-entity matrix used by Gabrilovich and Markovitch (2007) in Explicit Semantic Analysis and found to be useful in a wide array of NLP tasks.

For our experiments we trained embeddings of length 300 for 10 iterations over the dataset. We used default values for all other parameters in word2vec.

-needs developing-

-show results of the analogies experiment we did indicating semantic structure for the WORD vectors-

[4] Available at https://bitbucket.org/yoavgo/word2vecf

5 Evaluation

In this section we describe the setup used when evaluating our model and present evaluation results for two datasets. We evaluate the effect of initializing word and entity embeddings on our model as well.

5.1 Wikilinks

Prior to evaluating our method on the Wikilinks dataset we have collected the following statistics from the Wikilinks training set: $P(e)$ is the prior probability of seeing entity e in the training set and $P(e|m)$ is the probability of seeing entity e as a ground-truth for mention m.

When evaluating a NED system it is required to use some method for generating candidate entities first. We use a simple method where given mention m, we considered all candidates for whom $P(e|m) > 0$ as candidates. This simple method gives 97% ground-truth recall on the test set. We used GRUs as our RNN unit and used fixed size left and right contexts, using a 20 word window to each side of the mention. In cases were the context was shorter than the fixed size, we padded it with a special PAD symbol. Further, we filtered out stop words according to NLTK's stop-word list. The optimization of the model was carried out using standard back propagation and an AdaGrad optimizer (Duchi et al., 2011). We allowed the error to propagate through all parts of the network and fine tune all trainable parameters, including the word and entity embeddings themselves. A single epoch with $2.6M$ mentions and $k = 5$ for corrupt example generation was used for training the model taking half a day using a 20-core CPU machine.

We have used the following methods as base line on the Wikilink dataset:

- **Yamada et al.** (2016a) have created a state-of-the-art NED system for jointly mapping entities and words onto the same vector space via the skip-gram model and using these for disambiguation. As Wikilinks has only one mention per fragment we use only the local features described by Yamada at el.

- **Cheng et al.** (2013) addressed the Disambiguation to Wikipedia, or the "Wikificaiton" task, with a combination of local and global approaches. Mentions are disambiguated locally by generating a ranked list of candidates for each mention using the GLOW Wikification system proposed by Ratinov et al. (2011b). We compare our results to the ranking step of the algorithm, were a linear Ranking SVM is trained over a set of local and global features to return the list of candidates sorted by their likelihood.

- We include Most Probable Sense (MPS) as a baseline. This baseline picks the entity with the highest $P(e|m)$ as the correct mention. This simple baseline is notoriously known to give competitive results in many NED datasets

5.2 CoNLL

CoNLL is an evaluation corpus created by Hoffart et al. (2011) commonly used for benchmarking NED solutions (Globerson et al., 2016; Hachey et al., 2013; Yamada et al., 2016a; Pershina, 2015). CoNLL was composed by manually annotating Reuters newswire articles from 1996. It contains 1393 documents from a period of 12 days split into train, development and test sets. Following previous works we have only evaluated our method on non-NIL mentions. For candidate generation we used the publicly available candidate dataset by Pershina at el. (2015) with over 99% gold sense recall.

CoNLL has a training set with 18505 non-NIL mentions, which preliminary experiments showed is not sufficient to train our model on. We therefore resorted to a more complex training method where we first trained our model on a large corpus of mentions derived from Wikipedia cross references and then fine tuned the resulting model on CoNLL training set. To derive the Wikipedia training corpus we have extracted all cross-reference links from Wikipedia along with their context, resulting in over 80 million training examples. Due to constrained resources we set $k = 1$ for corrupt example generation and trained 1 epoch, which took around 4 days to train. The resulting model was then fine-tuned on CoNLL training set, where corrupt examples were produced by considering all possible candidates for each mention.

We have also found that using traditional statistical and string based features along with our model further improves its performance. We therefor used a setting similar to Yamada et al. (2016b) where a Gradient Boosted Regression

Tree was fitted with our models prediction score as a feature along with 7 other statistical and string based features. The statistical features are prior probability $P(e)$ and conditional probability $P(e|m)$ as described above, along with a feature counting the number of candidates generated for the mention and a feature giving the maximum conditional probability of the entity for all mentions in the document. For string similarity features we used the edit distance between the mention and the entity title in Wikipedia, a feature indicating weather the mention is a prefix or postfix of the entity Wikipedia title and a feature indicating weather the Wikipedia entity title is a prefix or postfix of the mention. Following Yamada we used sklearn's GradientBoostingClassifier implementation (Pedregosa et al., 2011) with a deviance loss and set the learning rate, number of estimator and maximum depth of a tree to 0.02, 10000 and 4, respectively.

As a baseline we took the standard Most Probable Sense (MPS) prediction, which corresponds to the $\arg\max_{e \in E} P(e|m)$, where E is the group of all candidate entities. We also compare to the following papers - Lazic et al. (2015), Francis-Landau et al. (2016), He et al. (2013), Hoffart et al. (2011) and Chisholm et al. (2015b) ,as they are all strong local approaches and a good source for comparison.

5.3 Results

Wikilinks test set	
Model	**Micro accuracy**
ARNN	**64.8**
GBRT: Base + ARNN features	**66.8**
Yamada et al.(partial)	59.8 **
Cheng et al.	*
Baseline (MPS)	55.9

Table 1: Evaluation on Web-Fragment data (Wikilinks)

The micro and macro accuracy scores on CoNLL test-b are displayed in table 2.

add insights regarding the results and comparison

5.4 Model sensitivity

Where is ARNN w/o init but with attention?

CoNLL test-b		
Model	Micro accuracy	Macro accuracy
ARNN	87.3	88.6
GBRT: Base + ARNN features	86.9	89.1
Yamada et al. (partial)	90.9	92.4
Lazic et al.	86.4	-
Francis-Landau et al.	85.5	-
He et al.	84.82	83.37
Hoffart et al.	79.57	80.71
Chisholm et al.	86.1	-
Baseline (MPS)	77	77

Table 2: Local evaluation on CoNLL. Bold font denotes the models offered in this study

Noam: this should be extended elaborate on:
!! ESA embedding initialization
!! attention vs. no attention

Wikilinks test set	
Model	**Micro accuracy**
ARNN w/o ESA init.	61
ARNN w/ ESA init. w/o Attention	64.1
ARNN w/ ESA & Attention	64.8

Table 3: ARNN Model sensetivity

6 Conclusions

References

Razvan Bunescu and Razvan Bunescu. 2006. Using Encyclopedic Knowledge for Named Entity Disambiguation. *IN EACL*, pages 9—-16.

Xiao Cheng and Dan Roth. 2013. Relational Inference for Wikification. *Empirical Methods in Natural Language Processing*, (October):1787–1796.

Andrew Chisholm and Ben Hachey. 2015a. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics*, 3:145–156.

Andrew Chisholm and Ben Hachey. 2015b. Entity Disambiguation with Web Links. *Transactions of the Association for Computational Linguistics*, 3(0):145–156.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks. pages 1256–1261.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJcAI*, volume 7, pages 1606–1611.

Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2016. Collective Entity Resolution with Multi-Focal Attention. *Acl*, pages 621–631.

Zhaochen Guo and Denilson Barbosa. 2014. Entity linking with a unified semantic representation. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 1305–1310. ACM.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. 2013. Evaluating entity linking with wikipedia. *Artificial Intelligence*, 194:130–150.

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. Learning Entity Representation for Entity Disambiguation. pages 30–34.

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.

Nevena Lazic, Amarnag Subramanya, Michael Ringgaard, and Fernando Pereira. 2015. Plato: A Selective Context Model for Entity Resolution. *Transactions of the Association for Computational Linguistics*, 3:503–515.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *ACL (2)*, pages 302–308.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized page rank for named entity disambiguation. In *Proc. 2015 Annual Conference of the North American Chapter of the ACL, NAACL HLT*, volume 14, pages 238–243.

Maria Pershina. 2015. Personalized Page Rank for Named Entity Disambiguation. (Section 4):238–243.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011a. 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.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011b. Local and Global Algorithms for Disambiguation to Wikipedia. *Acl 2011*, 1:1375–1384.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016a. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. *arXiv preprint arXiv:1601.01343*, page 10.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016b. Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv:1601.01343*.