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

General comment about citations – take whatever bibs you can from the ACL anthology: http://aclweb.org/anthology/
The bibs from Google scholar lack a lot of information.

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

NED algorithms can broadly be divided into local and global approaches. Local algorithms disambiguate each mention independently using local context (e.g. the sentence in which the mention appeared), 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., 2011; Guo and Barbosa, 2014; Pershina et al., 2015).However, most of these 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.

Insert example of incoherent context *from the dataset*. This will convince the reader that your problem is real.

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 text fragments extracted from web pages, each containing a mention of a named entity. Our dataset contains 18K unique mentions linking to over 100K unique entities. This dataset is significantly larger than previously collected ones such as CoNLL (Hoffart et al., 2011), TAC KBP (Ji et al., 2010) and ACE 2010 (Bentivogli et al.,), allowing us to train a deep neural model for the task.

Why are you citing Hoffart 2011 if the task is from 2003?

Potentially missing from previous paragraph: how do existing models perform on this dataset? Do they fail? You need to convince that there is a real *need* for a new model, and that the neural approach indeed addresses some of the shortcomings of the existing SotA.

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.

Good! This explanation is much more compelling. Relating to the previous comment, do you have some analysis that shows where the previous models fail, and whether your model addresses (even partially) this failure?
For example, are there cases where the disambiguating information is far from the mention, and therefore ignored/forgotten by non-RNN models? How many of these cases exist in your dataset? Does your model significantly improve upon them?

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

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 RNNs with attention can better model noisy and short context. In addition, we evaluate our algorithm on the CoNLL dataset (Hoffart et al., 2011), a small and "clean" dataset,where it yields comparable performance to other state-of-the-art local methods. We conclude that RNNs with attention are well-suited for local disambiguation in clean and well-structured corpora, but that there is still much room for improvement in real world scenarios where text is short, noisy, and less coherent.

Is this dataset based on news corpora? If so, this would be

2 Background

2.1 Related Work

This section is a bit messy... Not sure I understand the flow. What are we trying to communicate here? Also, does it make the paper self-contained? The background should provide a reader who is not familiar with NED (but is from the general NLP/ML community) the necessary background information to be able to understand your contribution.

The first attempt to use Wikipedia as a knowledge base for local disambiguation was offered by Busco and Pasca back in 2006 (Bunescu and Bunescu, 2006). Hoffart et al. (2011) suggested a collective global disambiguation graph-based framework, named AIDA, which employed mention, context and coherency models. 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., 2011; Cheng and Roth, 2013). Chisholm and Hachey (2015a) incorporated web-link data from the Wikilinks dataset (Singh et al., 2012a) with Wikipedia to train a model for entity linking. Even though demonstrating superior results on well structured evaluations, such as CoNLL, the study does not discuss its performance on the web-link mentions, which presents a very interesting challenge. Also, they choose to model textual context in a Bag-Of-Word (BOW) fashion, eliminating any compositional information of the text. This approach is very different from our suggested recurrent network which allows to capture sematnic information between context words. More recently, a selective-context model (Lazic et al., 2015) has shown that only few context words are informative for the tasks of disambiguation.

I didn't know that someone had already used Wikilinks for NED (although as training). We must state this explicitly, and emphasize the differ-

In the aforementioned studies two important observations stand out and influenced 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 disambiguation challenges of traditional NED data sets (Ratinov et al., 2011; Hoffart et al., 2011). This observation, which seems intuitive to the human reader, has led us to concentrate on generating a strong local context-based disambiguation solution. Moreover, following the results of Lazic et al. we designed our model to have an attention mechanism, for reducing the effect of non-informative neighboring text.

Maybe the first attempt of using Deep Neural Networks (DNN) for NED was led by He et al. (He et al., 2013), which learned a similarity measure between mention-context structures and candidates from Wikipedia using stacked autoencoders. Recent studies, have suggested Convolutional Neural Nets (CNN) architectures for learning semantic similarity between all three context, mention and candidate inputs (Sun et al., 2015; Francis-Landau et al., 2016). The growing popularity of neural embeddings in NLP related tasks has inspired several researches to jointly map those inputs to the same space using the fantastic word2vec approach (Yamada et al., 2016; Melamud and Goldberger, 2014).

Where is Globerson? Noam: not DNN or ANN

In this paper, we embed both words and Wikipedia entities in the same space to form input vectors for a Recurrent Neural Network (RNN) model. In this manner, we not only exploit the sequential structure of local surrounding context, but incorporate semantic world knowledge from a much larger corpus in our representation of text. Our novel solution combines these properties with an attention module to produce state-of-the-art local NED results on several datasets.

2.2 Datasets for NED

One of the most commonly used evaluation for benchmarking the NED challenge (Globerson et al., 2016; Hachey et al., 2013; Yamada et al., 2016; Pershina, 2015) is the CoNLL corpus, which was crafted from the CoNLL 2003 Named Entity Recognition (NER) task. This evaluation was established by Hoffart et al. (2011), which manually annotated Reuters newswire articles from 1996 with corresponding entities in the YAGO knowledge base. Its 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.

How was it annotated? By experts? Which KB was it mapped to? Noam:

TAC KBP (Ji et al., 2010) is another popular dataset for testing the performance of disambiguation systems (Chisholm and Hachey, 2015b; Globerson et al., 2016; Sun et al., 2015). Similar to CoNLL, it is primarily based on news articles and was specifically designed for the task of entity linking in the Text Analysis Conference (TAC). The most broadly used version for NED is the TAC 2010 data set, which includes a training and test set of $1,070$ and $1,017$ non-NIL mentions, respectively. [1]

Some disambiguation systems are also evaluated on the ACE 2005 corpus (Ratinov et al., 2011; Francis-Landau et al., 2016). Being composed out of news reports from various sources, this corpus was extend by connecting and annotating it using Wikipedia (Bentivogli et al.,). It features $16,310$ annotated mentions when only $1,458$ of them have multiple links, hence making them challenging for NED.

A different type of NED evaluation, suggested by Ratinov et al. (2011), is completely composed out of paragraphs from Wikipdia pages. In this Wiki dataset cross-reference hyperlinks act as mentions surface, thus implying that data is edited in a more crowd-sourced fashion than the previous benchmarks.

The common ground of all former data sets is their coherent and well reviewed structure, as it may be expected from mighty information and media organizations. When focusing only on this type of data we overlook the fact that nowadays most information is consumed through many diverse sources, such as blogs, social network posts and micro-group chats, which feature significantly higher contextual noise as they lack redaction and editing.

, łł ... ? ł ?

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.

[1] This data set was not available to us, as it is distributed only to TAC participants.

3 Dataset: Entity Mentions in the Web

Think of a cool acronym for this dataset, e.g. WNED - web NED.

We introduce a new large-scale NED dataset based on text fragments from the web. Our dataset is derived from the Wikilinks corpus (Singh et al., 2012b), which was constructed by crawling the web and collecting hyperlinks (mentions) linking to Wikipedia concepts (entities) and their surrounding text (context). Wikilinks contains 40 million mentions covering 3 million entities, collected from over 10 million web pages. It was originally collected for the task of cross-document coreference, which can be seen as a relaxed version of entity linking; while coreference maps mentions of the same entity to a single cluster, it does not require said cluster to be associated with an entity in any particular knowledge base.

Consider dropping the last sentence.

Wikilinks can be seen as a large-scale, naturally-occurring, crowd-sourced dataset where thousands of human annotators provide ground truths for mentions of interest. This means that the dataset also contains various kinds of noise, including erroneous ground-truth labels, malformed mentions, and incoherent contexts. The contextual noise in particular presents an interesting test-case that supplements existing datasets such as CoNLL (Hoffart et al., 2011), ACE, and Wiki(Ratinov et al., 2011), since these datasets are all sourced from mostly coherent and well-formed text (news and Wikipedia). Wikilinks therefore emphasizes the need to understand the local context, and marginalizes the utility of coherency-based global approaches.

cite

What is this dataset? Why is it only mentioned now and not in the background/intro Noam: fixed

We prepared our dataset from the local-context version of Wikilinks,[2] and resolved ground-truth links from the 7/4/2016dump of Wikipedia. We used the *page* and *redirect* tables for resolution, and kept the database *pageid* column as a unique identifier for Wikipedia entities. We discarded mentions where the ground-truth could not be resolved (only 3% of mentions).

American or international dating? If

We collected all pairs of mention m and entity e appearing in the dataset, and computed the number of times m refers to e ($\#(m, e)$), as well as the conditional probability of e given m: $P(e|m) = \#(m, e)/\sum_{e'} \#(m, e')$. 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 deemed the former to be less interesting for the purpose of NED, and suspected the latter to be noise with high probability. To filter these cases, we kept only mentions for which at least two different entities have 10 mentions each ($\#(m, e) \geq 10$) and consist of at least 10% of occurrences $P(e|m) \geq 0.1$.

I agree with the reasoning, but we should have some sort of experiment/analysis to back this claim.

Not sure I understand the second case. Example?

Finally, we randomly split the data into train (90%), validation (10%), and test (10%),according to domains in order to prevent lexical memorization (see (?)).

That's 110%... did you mean 80% for train?

This procedureaggressively filtered the dataset and we were left with $2.6M$ training, $300K$ test and $300K$ evaluation samples.

which one? I'm assuming the dumping of non-ambiguous mentions.

These numbers are not consistent with the percentages.

4 Algorithm

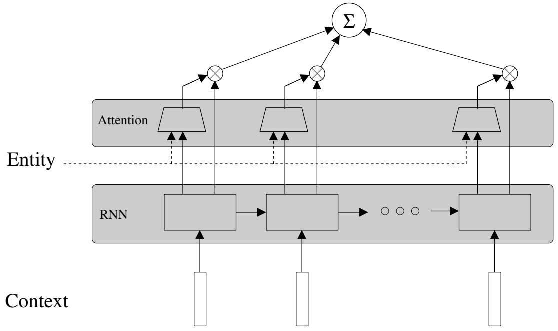
Our DNN model is a discriminative model which takes a pair of local context and candidate entity, and outputs a likelihood for the candidate entity being correct. Both words and entities are represented using embedding dictionaries and we interpret local context as a window-of-words to the left and right 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

[2] http://www.iesl.cs.umass.edu/data/wiki-links

Figure 1: Attentional RNN Architecture



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 non-linearity 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 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[3] 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

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

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-

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.

In all experiments we trained our model with 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 stop words according to NLTK's stop-word list.

Model optimization 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.

5.1 Wikilinks

When evaluating a NED system it is required to use some method for generating candidate entities. We use a simple method where given mention m we consider all candidates for whom $P(e|m) > 0$, where $P(e|m)$ is the probability of seeing entity e as a ground-truth for mention m in the training corpus. This simple method gives 97% ground-truth recall on the test set. We trained for a single epoch with $2.6M$ mentions and $k = 5$ for corrupt example generation. Training the model took half a day using a 20-core CPU machine.

YOTAM: I have better a training regime

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

- **Yamada et al.** (?) have created a state-of-the-art NED system that models entity-context similarity with word and entity embedding trained using the skip-gram model. We have obtained trained embeddings from the authors and used statistical features collected from the Wikilinks training set. We used only the local features described by Yamada at el. and trained their ranking model on Wikilinks training set.
- **Cheng et al.** (2013) have made publicly available [4] a global method which uses the GLOW system by Ratinov et al. (2011) for local disambiguation. We compare our results to the ranking step of the algorithm, without the their global component.
- 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

Due to long running time of Cheng et al. we have evaluated their method on Wikilink-Small, a smaller version of our test data with 10000 randomly sampled mentions.

5.2 CoNLL

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

[4] Available at https://cogcomp.cs.illinois.edu/page/software_view/Wikifier

each mention. For candidate generation we used the publicly available candidate dataset by Pershina at el. (2015) with over 99% gold sense recall.

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. (2016) 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 - 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

Our main evaluation results on the Wikilinks dataset are reported in Table 1. Our algorithm significantly outperforms both Yamada at el and the base line on this data by substantial margins. This result indicates that the skip-gram model used by Yamada at el. which averages the embedding vectors of all context words is non-optimal compared to our more sophisticated context modeling on this dataset. Our method outperforms the Baseline as well by a very large margin indicating our RNN model is indeed able to capture meaningful contextual features despite the noisy and short context.

Wikilinks Evaluation	
Model	**P@1**
ARNN	64.8
GBRT: Base + ARNN features	66.8
Yamada at el.	59.8
Baseline (MPS)	55.9

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

Table 2 shows evaluation results for the Wikilinks-Small test set. We used a pre-trained model supplied by Cheng at el which, similarly to the setting used for evaluating the GLOW algorithm by Ratinov at el (Ratinov et al., 2011), was not directly trained on our training set. We believe this explains their poor performance, as Wikilinks is substantially different from other NED datasets both for being noisy and for being annotated by web content authors rather then expert annotators. This last difference results in substantially different annotation patterns.

Wikilinks-Small Evaluation	
Model	**Micro accuracy**
ARNN	?
Cheng et al.	52

Table 2: Evaluation on Web-Fragment data (Wikilinks)

The micro and macro P@1 scores on CoNLL test-b are displayed in table **??**. On this dataset our model achieves reasonable results, however it cannot beat state-of-the-art results since it requires large quantities of training data to properly model the large number of parameters in the model.

add insights regarding the results and comparison

5.4 Model sensitivity

Where is ARNN w/o init but with attention?

6 Conclusions

References

Luisa Bentivogli, Pamela Forner, Claudio Giuliano, Alessandro Marchetti, Emanuele Pianta, and Kateryna Tymoshenko. Extending english ace

CoNLL test-b (Local methods)		
Model	Micro P@1	Macro P@1
PPRforNED		
RNN Attention	87.3	88.6
Yamada et al. local	90.9	92.4
Baseline (MPS)	77	77
Yago		
RNN Attention	?	?
Yamada et al. local	87.2	89.6
Francis-Landau et al.	85.5	-
Chisholm et al. local	86.1	-
Lazic et al.	86.4	-

Table 3: Evaluation on CoNLL. Bold font denotes the models offered in this study

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 4: ARNN Model sensetivity

2005 corpus annotation with ground-truth links to wikipedia.

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. *arXiv preprint arXiv:1604.00734*.

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.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. Overview of the tac 2010 knowledge base population track. In *Third Text Analysis Conference (TAC 2010)*, volume 3, pages 3–3.

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.

Oren Melamud and Jacob Goldberger. 2014. Learning Generic Context Embedding with Bidirectional LSTM.

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. 2011. Local and Global Algorithms for Disambiguation to Wikipedia. *Acl 2011*, 1:1375–1384.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012a. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. *... of Massachusetts, Amherst ...*, pages 1–14.

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

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1333–1339.

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