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? 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 (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 (Derczynski et al., 2015), or questions (Klang and Nugues, 2014), lack the sufficient coherence and context for global models to pay off. Take for example this fragment taken from the web:

> "I had no choice but to experiment with other indoor games. I was born in Atlantic City so the obvious next choice was **Monopoly**. I played until I became a successful Captain of Industry"

Potentially give citations for each domain. YOTAM: not sure how to cite these. Is the QA cite good?

This fragment is considerably less structured and with a more personal tone then news reports. It clearly references the entity *Monopoly_(Game)*, however expressions such as 'experiment', and 'Industry' can generate a lot of noise when disambiguating *Monopoly_(Game)* from the much more common entity *Monopoly* (economics term). Some sense of local semantics and syntactics must be considered in order to separate the useful signals (e.g. indoor games, played) from the noisy ones.

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.

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-YAGO (Hoffart et al., 2011), TAC KBP (Ji et al., 2010) and ACE 2010 (Bentivogli et al.,). We have found that performance of state-of-the-art methods is greatly impaired on this dataset and believe new algorithms that can better model local semantic and syntactic features are required.

Why are you citing Hoffart 2011 if the task is from 2003? YOTAM: They were the ones to annotate it for NED

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? YOTAM: We don't have any, I will see what i can do...

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-YAGO dataset (Hoffart et al., 2011), a dataset of annotated newswire articles, 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.

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. Noam: Done

In the past few years, the promising performance of global algorithms in Entity Linking (EL), Wikification and NED, established the domination of global approaches for the task of disambiguation. Nowadays, traditional and purely local solutions, such as those offered by Bunescu and Pasca (2006) and Mihalcea and Csomai (2007), tend to be embedded as local context models in sophisticated collective global disambiguation systems. In this manner, the local component of the GLOW Wikification algorithm (Ratinov et al., 2011) was exploited as part of the Relational inference system suggested by Cheng and Roth (2013). Similarly, Globerson et al. (2016) achieved state-of-the-art results by extending the local based selective-context model of Lazic et al. (2015) with an attention-like coherence mechanism.

Ablation analysis has shown more than once that coherency models boost the performance of local contextual baselines. However, it has also emphasized that the local techniques produce a

hard baseline to beat (Ratinov et al., 2011). The coherence, which is mostly regarded as a property of the dataset, can also impose false constraints on the relatedness of entities in the document. For instance, Hoffart et al. (2011) had to build a coherency test on top of his collective graph-based model to disregard globally suggested Wikipedia entities. Accordingly, around $2/3$ of CoNLL's mentions where solved without even engaging the global coherence component. These observations, which were obtained on well structured news articles and Wikipedia based documents, might be amplified in a non-coherent textual environment.The fact that our suggested Wikilinks-based evaluation (Singh et al., 2012) is crowd-sourced and much less coherent than traditional NED test-sets, led us to focus on generating a strong local context-based disambiguation solution.

Chisholm and Hachey (2015) incorporated web-link data from the same Wikilinks data source with Wikipedia to train a model for entity linking. Even though demonstrating superior results on newswire based evaluations, such as CoNLL, their study did not discuss its performance on the web-link mentions, which presents a very interesting NED challenge. Also, the textual context in that study is modeled in a Bag-Of-Word (BOW) fashion, which eliminates any compositional information of the text. Chisholm and Hachey (2015) approach is very different from our suggested algorithm which allows capturing semantic information between context words by using a dedicated Deep Neural Network architecture (DNN). .

The first published attempt of using DNN for NED was led by He et al. (2013), in which the network 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).

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. Unlike other studies, we not only exploit the sequential structure of the local surrounding context, but also incorporate semantic world knowledge from a much larger corpus in our representation of text. Moreover, following the results of Lazic et al. (2015), who claimed that only few context words have value in disambiguating the mention, we equipped our model with an attention module, which significantly reduces the effect of non-informative neighboring text. Our novel solution combines all former properties to produce state-of-the-art local NED results on the suggested Wikilinks data set.

2.2 Datasets for NED

One of the most commonly used evaluations for benchmarking the challenge of NED (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.

TAC KBP (Ji et al., 2010) is another popular dataset for testing the performance of disambiguation systems (Chisholm and Hachey, 2015; 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 EL 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 mainly composed out of news reports from various sources, this corpus was extended 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 mak-

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

Noam: lacks 's association

I didn't know that someone had already used Wikilinks for NED (although as training). We must state this explicitly, and emphasize the difference/contribution of your work. Noam: Done

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

ing them challenging for NED. A different type of 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 mention's surface, thus implying that data is edited in a more crowd-sourced fashion.

All aforementioned datasets share the property of well formed and structured content, since they are founded on the publications of recognized information and media organizations. However, when focusing only on this type of data we overlook the fact that today most information is transfered without any filtering via a variety of sources, such as blogs, social network posts and group chats. These sources incorporate significantly higher contextual noise as they lack proper redaction. Additionally, most of traditional NED evaluations were manually annotated and therefore contain a very small training set for supervised disambiguation techniques. In this study we devise a novel benchmark of crowd source text fragments that were manually linked to their corresponding Wikipedia titles by their authors. This relatively incoherent source offers a vast and natural corpus for supervised algorithms and produces an interesting case study for evaluating NED on more general and common text.

, ł ł ... ? ł ? Noam: fixed

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.

3 WikilinksNED 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., 2012), 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.

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-YAGO (Hoffart et al., 2011), TAC KBP (Ji et al., 2010) and ACE 2010 (Bentivogli et al.,), 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.

To get a sense of textual noise and entity-context coherence we have set up a small experiment where we measured the similarity between entities mentioned in WikilinksNED and their surrounding context, and compared the results to CoNLL-YAGO. We used state-of-the-art word and entity embeddings obtained from Yamada at el (Yamada et al., 2016) and computed cosine similarity between an entity embedding and the mean of context words embeddings. We compared results from all mentions in CoNLL-YAGO to a sample of 10000 web fragments taken from WikilinksNED, using a window of words of size 40 around entity mentions. On CoNLL-YAGO we found the mean similarity to be 0.188 while on WikilinksNED we got 0.163. We believe this result indicates that web fragments in WikilinksNED are indeed less coherent and noisier compared to CoNLL-YAGO documents.

We prepared our dataset from the local-context version of Wikilinks,[2] and resolved ground-truth links from the 7/4/2016dump of Wikipedia [3]. 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).

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

American or international

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

[3] https://dumps.wikimedia.org/

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 they appeared in the dataset only a handful of times and referred to different entities only a couple of times each.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 $P(e|m) \geq 0.1$. This procedure aggressively filtered our dataset and we were left with $3.2M$ mentions.

Not sure I understand the second case. Example? YOTAM: tried to improve explanation.

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

4 Algorithm

This section should be organized as follows: 4. Overview, 4.1. Model Architecture, 4.2. Training, 4.3. Embedding Initialization. Overview should refer to the diagram, which 4.1. elaborates. 4.1. should include all the formulae, as well as the rationale behind the architecture.

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.

Omer, check this again. Is it a factor or a probability?

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[4] and an output layer with two output units in a softmax. 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.

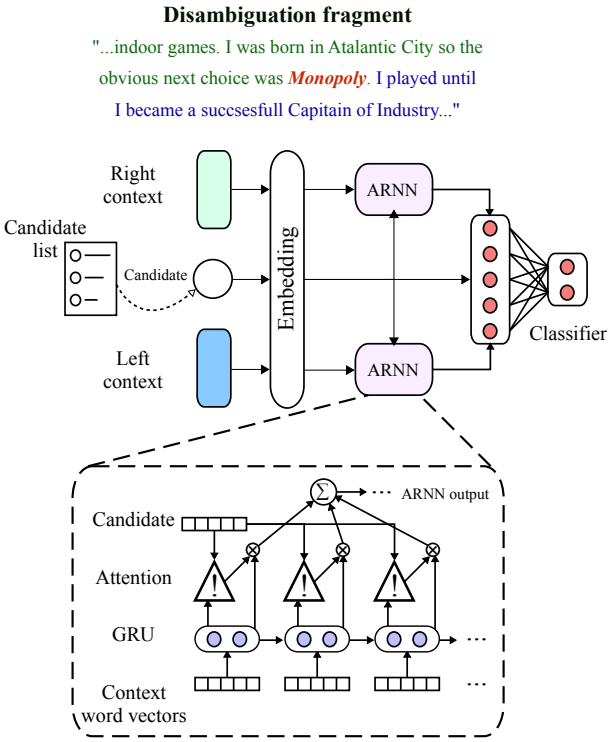

Figure 1: Our Neural Network architecture

Where is a diagram of the model? The explanation is good, but it'd be better to have something visual.

The next paragraph explains how you trained it, and pertains more to the data-sampling part of the method rather than the architecture.

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 (c,e), where e is the correct assignment for c, we produce k corrupt examples with the same context c but with a different, corrupt entity e', which is sampled uniformly 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.

Why not just from entities that can fit the same mention? It's better to train with near misses

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.

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

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.

How do I attach this figure caption without ruining the order? - "The full model appears under the disambiguation fragment. In the dashed box is a closeup on one of the Attentional GRU based component."

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

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.

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

Is this the first-ever use of attention in GRUs, or was this parametrization used before? It reads as if your contribution is inventing this mechanism, rather than using it. If this is not novel, I suggest rephrasing and citing appropriately; the explanation is very clear, but it needs to emphasize what you did *differently*.

4.2 Training initial word and entity embeddings

In the model architecture, and perhaps also in the overview, you should mention that both words and entities are embedded. In the training subsection, you should mention whether the embeddings were finetuned or not.

Training our model implicitly embeds the vocabulary of words and collection of entities in a common space. However, we find that explicitly initializing these embeddings with vectors pretrained over a large collection of unlabeled data significantly improved whatever (see Section 5). To this end, we implemented an SGNS-based approach (Mikolov et al., 2013) that simultaneously trains both word and entity vectors.

Refer to the exact subsection where you discuss this result

We used `word2vecf`[5] (Levy and Goldberg, 2014a), which allows one to train word and context embeddings using arbitrary definitions of "word" and "context" by providing a dataset of word-context pairs (w, c), rather than a textual corpus. In our usage, we define a context as an entity e. To compile a dataset of (w, e) pairs, we consider every word w that appeared in the Wikipedia article describing entity e. We limit our vocabularies to words that appeared at least 20 times in the corpus and entities that contain at least 20 words in their articles. We ran the process for 10 epochs and produced vectors of 300 dimensions; other hyperparameters were set to their defaults.

[4] 300 dimensions with ReLU, and $p = 0.5$ dropout.

[5] `http://bitbucket.org/yoavgo/word2vecf`

Levy and Goldberg (2014b) showed that SGNS implicitly factorizes the word-context PMI matrix. Our approach is doing the same for the word-entity PMI matrix, which is highly related to the word-entity TFIDF matrix used in Explicit Semantic Analysis (Gabrilovich and Markovitch, 2007).

-needs developing-

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

5 Evaluation

In this section, we describe our experimental setup and compare our model to the state of the art on two datasets: our new WikilinksNED dataset, as well as the commonly-used CoNLL-YAGO dataset (Hoffart et al., 2011). We also examine the effect of initializing our model with pre-trained word and entity embeddings.

In all experiments, our model was trained with fixed-size left and right contexts (20 words in each direction). We used a special padding symbol when the actual context was shorter than the window. Further, we filtered stopwords using NLTK's stop-word list.

Model optimization was carried out using standard backpropagation 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 WikilinksNED

Candidate Generation To isolate the effect of candidate generation algorithms, we used the following simple method for all systems: given a mention m, consider all candidate entities e that appeared as the ground-truth entity for m at least once in the training corpus. This simple method yields 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.

Baselines Since we are the first to evaluate NED algorithms on WikilinksNED, we ran a selection of existing local NED systems and compared their performance to our algorithm's. **Yamada et al.** (**?**) created a state-of-the-art NED system that models entity-context similarity with word and entity embeddings trained using the skip-gram model. We obtained the original embeddings from the authors, and trained the statistical features and ranking model on the WikilinksNED training set. Our configuration of Yamada et al.'s model used only their local features.

Cheng et al. (2013) have made their global NED system publicly available [6]. This algorithm uses GLOW (Ratinov et al., 2011) for local disambiguation. We compare our results to the ranking step of the algorithm, without the their global component. Due to the long running time of this system, we only evaluated their method on the smaller test set, which contains 10,000 randomly sampled instances from the full 320,000-example test set.

Finally, we include the **Most Probable Sense (MPS)** baseline, which selects the entity that was seen most with the given mention during training.

Results Results on WikilinksNED should be here.

5.2 CoNLL-YAGO

CoNLL-YAGO 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-YAGO 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-YAGO training set, where corrupt examples were produced by considering all possible candidates for 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

[6] https://cogcomp.cs.illinois.edu/page/software_view/Wikifier

Before or after selecting the window? If the stopwords are thrown before, the window is effectively larger. Also, why was this necessary?

Move to section 4 where you talk about optimization/training.

YOTAM: I have better a training regime

The number of mentions is redundant (mentioned in 3). The

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. (**?**) ,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 (**?**), 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-YAGO 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 train the large number of parameters in the model.

CoNLL-YAGO 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-YAGO. Bold font denotes the models offered in this study

add insights regarding the results and comparison

5.4 Error analysis

We extracted and analyzed 4227 cases of false disambiguated mentions that our ARNN model predicted. This subset was obtained from evaluating on a Wikilinks-based validation set that was not used for training.

While going over the errors, we came across several examples where the actual entity was fully contained in the prediction. For instance, gold-senses such as *'ted'* and *'macedonia'* were "falsify confused" with the entities *'ted (conference)'* and *'republic of macedonia'*. In practice, the context of those examples together with their urls links point to the same source as the one predicted by the model. This type of error was observed in almost 20% of the cases. Moreover, the conditional prior of the predicted entity in these cases was larger than the conditional of the correct sense. We believe that this indicates on good generalization ability of the model and emphasizes problems in the web sourced labeling.

A further interesting property of the errors was the distance between the false predicted entity and the correct one. This was measured by calculating the cosine similarity between our word-title embeddings. For example, the similarity between the two music related candidates, *'the fall (gorillaz album)'* and *'the fall (band)'* was 0.96. Accordingly, given that words, such as 'recorded' and 'albums', were present in the context of the mention, the decision to pick the wrong sense over the right one does not seem completely detached and irrational. In our analysis the average cosine similarity of the mistakes was estimated to be 0.925, when 0.9 was the 20st percentile of the value distribution. It is important to note that strong associations between candidates are usually expected in a disambiguation framework, since most possible senses are somewhat related. However, considering that the neural embeddings use as a main feature for our model, we believe that the particularly high error similarity shows that even the wrong predicted entities are strongly related to right senses.

fix and report with Yamadas embeddings

Additionally, we wanted to estimate how many of our Lastly, we wanted to check how much our model differs from the MPS baseline by visualizing the error distribution of the conditional prior feature. In this way we could quantify the diversity of the model's prediction. The full error distribution is displayed in. We saw that 64% of the error predictions were not most frequent entities. the average conditional prior of those non-mps mistakes was 0.197% with 0.664 coefficient of variation. Moreover, we observed that 25st percentile of the error was smaller than 0.09%. This error analysis reveals that in most of its mistakes our model does not simply captures [continue]

5.5 Model sensitivity

Where is ARNN w/o init but with 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 init. w/ Attention	64.8

Table 4: ARNN Model sensetivity

6 Conclusions

References

Luisa Bentivogli, Pamela Forner, Claudio Giuliano, Alessandro Marchetti, Emanuele Pianta, and Kateryna Tymoshenko. Extending english ace 2005 corpus annotation with ground-truth links to wikipedia.

Razvan C Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *EACL*, volume 6, 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. 2015. Entity disambiguation with web links. *Transactions of the Association for Computational Linguistics*, 3:145–156.

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*.

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32–49.

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.

Marcus Klang and Pierre Nugues. 2014. Named entity disambiguation in a question answering system. In *The Fifth Swedish Language Technology Conference (SLTC 2014)*.

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.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM.

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. 2012. 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*.