Local Named Entity Disambiguation for Noisy Web Fragments with Neural Attention-RNNs

Anonymous ACL submission

Abstract

We address the task of Named Entity Disambiguation (NED) in a web setting. We present WikilinksNED, a large-scale NED dataset of short web-page fragments containing mentions manually linked to Wikipedia by web content authors. In contrast to existing news-based datasets the web data is noisier and less coherent. We propose a model based on Attention-RNNs to capture useful features from short and noisy context. We evaluate our model on both WikilinksNED and a standard, smaller, news-based dataset. We find our model greatly outperforms existing state-of-the-art methods on WikilinksNED while achieving reasonable performance on the smaller dataset. We conclude our model can efficiently model noisy context given sufficient training. We analyze our results and show there is room for further improvement on such data.

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 of 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; Globerson et al., 2016). 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.

In this work, we investigate the task of NED in a setting where only local and noisy context is avail-

able. 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.

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.

Noam: The next sentence sounds odd...

We demonstrate our model greatly outperforms existing state-of-the-art NED algorithms on the 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 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. We conclude that RNNs with attention are well-suited for local disambiguation in real-worlds scenarios where context is noisy and less coherent, but analysis of our results reveals that there is still room for improvement.

2 Background

2.1 Related Work

Early work on Named Entity Disambiguation, such as Bunescu and Pasca (2006) and Mihalcea and Csomai (2007) have focused on local approaches where a mention is disambiguated using hand-crafted statistical and contextual features. While providing a hard-to-beat baseline (Ratinov et al., 2011), increasing attention was recently given to global approaches, which add a layer of sophistication on top of local approaches by considering the coherency of entities assignment within a document. For example the local component of the GLOW 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.

However, Hoffart et al. (2011) have found that in many cases global coherency impose false constraints on the relatedness of entities within a document. They used a coherency test to disregard globally suggested Wikipedia entities, and found around $2/3$ of CoNLL's mentions where solved without engaging the global component. We believe that these observations, which were obtained on well structured news articles and Wikipedia based documents, might be amplified in a noisy, non-coherent textual environment such as web content. We have therefore focused on obtaining a strong local, context-based disambiguation solution.

The first published attempt of using DNNs for NED was by He et al. (2013), which used stacked auto-encoders to learn a similarity measure between mention-context structures and entity candidates. Recently the increasing popularity of DNNs has inspired a number of works that used Convolutional Neural Nets (CNN) for learning semantic similarity between context, mention and candidate inputs (Sun et al., 2015; Francis-Landau et al., 2016). Neural Embedding techniques have also inspired a number of works that jointly map context and entity to the same embedded space (Yamada et al., 2016; Melamud and Goldberger, 2014).

In this paper, we train a Recurrent Neural Net-

work (RNN) model, which unlike CNNs and most methods purely based on embeddings, are naturally adapt to exploit the sequential structure of text. Moreover, Lazic et al. (2015) have used a probabilistic attention-like model and have shown only few context words have value in disambiguating a mention. We therefor experimented with a neural version of an attention mechanism.

Chisholm and Hachey (2015) experimented with using web-link data from the Wikilinks corpus (Singh et al., 2012) for training a disambiguation algorithm. They have shown that despite the noisy nature of web data, augmenting Wikipedia derived data with web-links can lead to improved performance on standard datasets. We make use of the same Wikilinks corpus, but are interested in web data as a noisy test case in-itself rather then using it for training alone. Moreover, in contrast to Chisholm, we use DNNs to automatically capture useful features from the noisy data.

Commonly used benchmarks for NED systems have mostly focused on news-based corpora. CoNLL-YAGO is a dataset based on Reuters newswire articles that was created by Hoffart et al. (2011) by hand-annotating the CoNLL 2003 Named Entity Recognition task dataset with YAGO entities. It contains 1393 documents split into train, development and test sets. TAC KBP 2010 (Ji et al., 2010) is another, smaller, dataset for NED based on news articles. ACE 2005 corpus is another news based dataset annotated by Bentivogli et al. (). Ratinov et al. (2011) have used a random sample of paragraphs from Wikipedia for evaluation, however they did not make the precise sample they used publicly available.

Our WikilinksNED dataset is substantially different from currently available datasets since these are all based on high quality content from either news-articles or Wikipedia, while WikilinksNED is a test-case for generally noisier, less coherent and lower quality data. The annotation process in significantly different as well, as our dataset reflects the annotation preferences of real-world website authors and is not annotated by experts, or a high-quality community effort in the case of Wikipedia. It is also significantly larger in size, being almost two orders of magnitude larger then all news-based datasets.

Recently, a number of Tweet based datasets have been composed as well (Locke, 2009; **?**; **?**; **?**). These represent a much more extreme case then our dataset in terms of noise, shortness and spelling variations, and are much smaller in size. Due to the unique nature of Tweet data, proposed algorithms tend to be substantially different from algorithms used for other NED tasks.

3 WikilinksNED Dataset: Entity Mentions in the Web

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 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.

Noam: we might need to add STD as the difference is very small(not sure if this is statistically significant)

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,[1] and resolved ground-truth links using a Wikipedia dump from April 2016[2]. 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).

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 website 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 Attention-RNN (ARNN) components which process each side and produce a fixed length vector representation. The resulting vectors along with the entity embedding are then fed into a classifier network with two output units that are trained to emit the likelihood of the candidate being a correct or corrupt assignment.

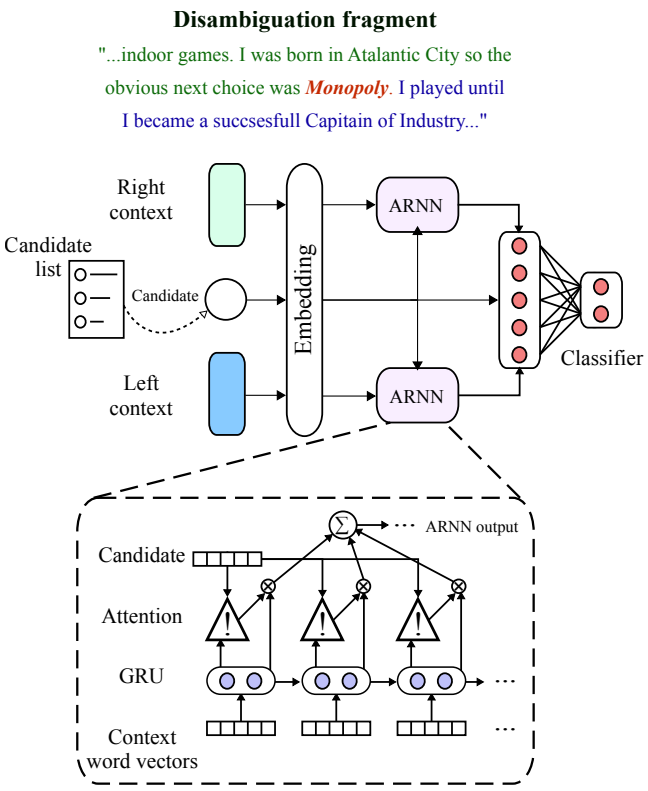

Figure 1: Architecture of our Neural Network model. A close up of the Attentional RNN component appears in the dashed box.

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

4.1 Model Architecture

Our architecture has two main component: a duo of ARNN components and a classifier. The classifier network consists of a hidden layer[3] 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.

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

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

[3] 300 dimensions with ReLU, and $p = 0.5$ dropout.

The ARNN components each process one side of the context (left and right)[4]. The ARNNs are based on a general RNN unit fitted with an attention mechanism, and their mechanics 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 our implementation we have used a standard GRU unit (Cho et al., 2014), but any RNN can be a drop-in replacement. We fit the RNN unit with an additional attention mechanism, a mechanism commonly used for state-of-the-art encoder-decoder models where the output of the decoder at time step t is used to decide which parts of the encoder output are important for decoding time step $t+1$ (Bahdanau et al., 2014; **?**). Since our model lacks a decoder, we use the entity embedding as a control signal for the attention mechanism.

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. A softmax function then normalizes the attention values and the final output o_{attn} is computed 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 follow Bahdanau et al. (2014) and parametrize the 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]Right context is fed into the ARNN in reverse order

4.2 Training

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'. We have considered two alternatives for corrupt sampling:

Near-Misses: Sampling out of the candidate set of each mention. We have found this to be more effective where the training data reliably reflects the test-set distribution.

All-Entity: Sampling from the entire dictionary of entities. Better suited to cases where the training data or candidate generation does not reflect the test-set well. Has an added benefit of allowing us to utilize unambiguous training examples where only a single candidate is found.

In our evaluation we specify exactly which approach was used for each experiment and provide an empirical comparison of the two approaches.

We sample corrupt examples uniformly in both cases. This matches the distribution of positive examples to the prior probability of the entities for All-Entity sampling and the conditional prior for Near-Misses – In effect biasing the network towards more popular entities. We note that preliminary experiments revealed that corrupt-sampling according to the distribution of entities in the dataset, rather then uniform sampling (as is done by Mikolov at el. (2013)) produces an interesting entity-context similarity measure, however it does not perform well in our settings due to the lack of biasing toward popular entities.

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. We found the performance of our model substantially improves for the first few epochs and then continues to

slowly converge with marginal gains, and therefore trained all models for 8 epochs with $k = 5$ for corrupt example generation.

4.3 Embedding Initialization

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 pre-trained over a large collection of unlabeled data significantly improved both performance and training speed (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.

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).

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 different corrupt example generation schemes, and 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 prior to selecting the window in order to focus on more informative words.

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

5.1 WikilinksNED

Training we use Near-Misses corrupt-sampling which was found to perform well due to a large training data that represents the test-data well.

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.

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 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 We used standard micro P@1 accuracy for evaluation on WikilinksNED. Experimental results are reported in Table 1. Our algorithm significantly outperforms Yamada at el. on this data by over 5 points. 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 mar-

[6] `https://cogcomp.cs.illinois.edu/page/software\_view/Wikifier`

Wikilinks Test-Set Evaluation		
Model	**Sampled Test Set (10K)**	**Full Test Set (300K)**
Baseline (MPS)	60	59.6
Cheng et al.	50.7	-
Yamada at el.	67.6	66.9
Our Attention-RNN	73.2	73
Our RNN, w/o Attention	??	??

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

gin indicating our RNN model is indeed able to capture meaningful contextual features despite the noisy and short context.

When running Cheng at el (2013) we have 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. This has resulted in poor performance, emphasizing the greater importance of training a model directly on the training set compared to existing datasets based on news corpora and annotated by experts.

We find that the attention mechanism significantly improves results by attending to the most discriminative parts of the context given a specific candidate mention.

5.2 CoNLL-YAGO

Training 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. We first trained our model on a large corpus of Wikipedia derived data, and fine-tuned on CoNLL-YAGO training set. We then used the model in a similar setting to Yamada et al. (2016) where a GBRT was trained with our model as a feature along with the statistical and string based features defined by Yamada.

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. We set $k = 5$ for corrupt example generation and trained for 1 epoch with All-Entity corrupt-sampling. 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.

Candidate Generation For comparability with existing methods we used two publicly available candidates datasets:

- PPRforNED - Pershina at el. (2015)
- YAGO - Hoffart at el. (2011)

Baselines 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), Yamada at el. (2016), and Chisholm et al. (**?**), as they are all strong local approaches and a good source for comparison.

Results The micro and macro P@1 scores on CoNLL-YAGO test-b are displayed in table 2. 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. Further, the relative cleanliness of the data allows existing approaches to perform very well and marginalizes the effect of utilizing a deep and powerful neural model.

CoNLL-YAGO test-b (Local methods)		
Model	Micro P@1	Macro P@1
PPRforNED		
Our ARNN + GBRT	87.3	88.6
Yamada et al. local	90.9	92.4
Yago		
Our ARNN + GBRT	83.3	86.3
Yamada et al. local	87.2	89.6
Francis-Landau et al.	85.5	-
Chisholm et al. local	86.1	-

Table 2: Evaluation on CoNLL-YAGO.

5.3 Effects of initialized embeddings and corrupt-sampling schemes

We performed a study of the effects of using pre-initialized embeddings for our model, and of using either All-Entity or Near-Misses corrupt-sampling. The evaluation was done on a 10% sample of the evaluation set of the WikilinksNED corpus and can be seen in Table 3. We have found that using pre-initialized embeddings gives a much better starting point for training compared to random initialization. This results in faster convergence and a significant performance gain.

Comparison of All-Entity and Near-Misses corrupt-sampling reveals that when using Near-Misses, our model achieves significantly improved performance. We attribute this difference both to the more efficient nature of training with near misses, and to Near-Misses preserving the conditional-prior of entities which is a stronger signal then the prior probability preserved by All-Entity corrupt-sampling.

YOTAM: ** = Missing one more epoch for Near-misses, random init. - tommorow

Wikilinks Evaluation-Set	
Model	**Micro accuracy**
All-Entity, with init.	70
All-Entity, random init.	67.1
Near-misses, with init.	72.5
Near-misses, random init.	67.2 ∗ ∗

Table 3: Corrupt-sampling and Initialization

5.4 Discussion//Conclusions

We believe these results demonstrate that our ARNN model is better then existing state-of-the-art techniques at modeling noisy context when sufficient amounts of training examples are available. This allows us to greatly outperform state-of-the-art algorithms which are found to be suboptimal in a noisy environment such as web content.

However the gap between results of all systems tested on both CoNLL-YAGO and WikilinksNED indicates that web mentions are indeed a more challenging test case then news-based corpora. We believe this to be an important and challenging real-world scenario where noise in the form of mis-structured text, incoherent topics and grammatical/spelling mistakes are present. This scenario represents a distinct test-case that lays in between the standard new-based datasets and the much noisier Tweeter data that has been receiving increasing attention lately. We believe recursive neural models are a promising direction for this task, while there is still much room for improvement.

5.5 Error analysis

We randomly sampled and manually analyzed 200 individual cases of false disambiguations that our neural model generated. This error subset was obtained from a Wikilinks-based validation set that was not used for training.

Working with crowd-labeled data, we expected some of the mentions to be associated with wrong Wikipedia titles. Accordingly, we found out that 18% (36/200) of the error-labeled predictions were not false, where 21% of the mistakes originated in wrong labeling of the author and 34% were predictions with an equivalent meaning as the correct entity. Interestingly, in 11.5% (23/200) of the error events, the model suggested a more convincing solution than the original author by using specific hints from the context. In this manner, the mention *'Supreme leader'* , which was contextually associated to the Iranian leader Ali Khamenei, was linked by our model with *'supreme leader of Iran'* while the "correct" tag was the general *'supreme leader'* entity. From the remaining 164 error-entries, 15.5% were cases where a Wikipedia disambiguation-page was chosen as either the correct or predicted entity (2.5% and 13%, respectively). We decided to discount this error type from further analysis, as it is logically equivalent to remain unresolved. Eventually we ended up with 131 error cases to analyze.

First, we noticed that in 32% of the errors (42/131) the model selected an entity that can be understood as a specific (7%) or general (25%) realization of the correct solution. For example, instead of predicting *'Fu Manchu mustache'* for a facial hair related text, the model addressed the mention with the fictional character *'Fu Manchu'*, with whom the beard style originated. We further investigated and saw that in almost half of those cases, either the predicted or the correct web-page contained a non-trivial link to the other page, thus implying a strong correlation between both entities. Overall, we observed this type of co-reference relation in 26% of the error cases (35/131). A

closer look discovered two prominent types of coreference errors that occurred repeatedly in the data. The first category was of a film/book/theater type of error (9%), where the actual and the predicted entities were a different display of the same narrative. Even though having a different jargon and producers, those fields share extremely similar content, which may explain why they tend to be frequently confused by the algorithm. The second issue of the model was in differentiating between adjectives that are used to describe properties of a nation. Particularity, mentions such as *'Germanic'*, *'Chinese'* and *'Dutch'* were falsely assigned to entities that describe language instead of people, and vice versa. We observed this type of mistake in 8.4% of the errors (12/131).

A further interesting trend was the prediction policy of the model in cases where the entity had low-count. We defined low-count entities as entities which appeared less than 10 times in the data, and hence were probably not seen in the model's training period. We saw that the model followed the MPS in 75% of the low-count events. This shows that entity-mention pairs, which are alien to the model, tend to bias its prediction towards the baseline. Further, the amount of generalization error in low-count conditions was also significant (35.7%), as the uncertainty of the prediction was captured by a similar but less specific entity suggestion.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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.

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.

Brian William Locke. 2009. Named entity recognition: Adapting to microblogging.

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.

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*.