Named Entity Disambiguation for Noisy Text

Anonymous ACL submission

Abstract

We address the task of Named Entity Disambiguation (NED) for noisy text. We present WikilinksNED, a large-scale NED dataset of fragments taken from web-pages, that is significantly noisier and more challenging then existing news-based datasets. We propose a model based on Attention-RNNs to model the sequential nature of text, and attend to the useful signals in it. We describe novel methods for sampling the training and for initializing word and entity embeddings, and demonstrate their importance for model performance. We evaluate both on WikilinksNED and on a standard, smaller, news-based dataset and find our model significantly outperforms existing state-of-the-art methods on WikilinksNED while achieving comparable performance on the smaller dataset.

1 Introduction

Named Entity Disambiguation (NED) is the task of linking mentions of entities in text to a given knowledge base, such as Freebase or Wikipedia.

Noam: to much 'NED' in the beginning of a sentence consider - The disambiguation task is a key component in Entity Linking (EL) systems , which typically both address ...

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 (Berant and Liang, 2014), as well as other NLP tasks.

Current research on NED is mostly driven by a number of standard datasets, such as CoNLL-YAGO (Hoffart et al., 2011), TAC KBP (Ji et al., 2010) and ACE (Bentivogli et al., 2010). A wide variety of algorithms were developed to address these datasets (CITE). However, most of the standard datasets are based on news corpora and Wikipedia, which are naturally coherent, well-structured, and rich in context. Other domains, such as web page fragments, social media, or questions, are often short, noisy and less coherent.

Noam: I don't see the negation/problem in the however. This sentences should provide the "problem" with current methods. Consider - Most standard NED benchmarks are based on news corpora and Wikipedia, which are naturally coherent, well-structured, and rich in context. However, when focusing only on this type of data we overlook the fact that today information is typically transfered unfiltered via a variety of sources such as (why is short noisy data an interesting study case?)...

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

This short fragment is considerably less structured and with a more personal tone than 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 must be considered in order to separate the useful signals (e.g. indoor games, played) from the noisy ones.

We create a dataset of 3.2M short text fragments extracted from web pages, each containing a mention of a named entity. Our dataset is two orders of magnitude larger than previously collected datasets, and contains 18K unique mentions linking to over 100K unique entities. We have empirically found it to be significantly noisier and more challenging than standard datasets.

We propose a novel neural network architecture based on recurrent neural networks (RNNs) with an attention mechanism. 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 using an attention model to reduce the impact of noise

Noam: is the attention explanation necessary to the intro? can't you simply say attention (getting rid of the '...by assigning weights...")?

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, and demonstrate its importance for model performance.

Our experiments show that our model significantly outperforms existing state-of-the-art NED algorithms on WikilinksNED, suggesting that RNNs with attention are able to model short and noisy context. In addition, we evaluate our algorithm on CoNLL-YAGO (Hoffart et al., 2011), a dataset of annotated newswire articles. We use a simple domain adaptation technique since CoNLL-YAGO lacks a large enough training set for our model, and achieve comparable results to other state-of-the-art methods.

To better understand our results we performed an error analysis of the WikilinksNED experiment. We found that many false errors originate from annotation noise, limiting to some extent achievable performance on this dataset. Three additional key sources of error were identified, and should be addressed in future work: Lack of sufficient training for long-tail entities, selecting overly specific or general entities, and failing to distinguish entities that are semantically highly related (e.g. a book and a movie of the same narrative).

2 Related Work

Local vs Global NED Early work on Named Entity Disambiguation, such as Bunescu and Paşca (2006) and Mihalcea and Csomai (2007) have focused on local approaches where a mention is disambiguated using hand-crafted statistical and contextual features. While local approaches provide a hard-to-beat baseline (Ratinov et al., 2011), increasing attention was recently given to global approaches. These add a layer of sophistication on top of local approaches by considering the coherency of entity assignments 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.

Global algorithms have significantly outperformed the local approach on standard datasets (Guo and Barbosa, 2014; Pershina et al., 2015; Globerson et al., 2016). However, global approaches are difficult to apply in domains where only short and noisy text is available. For example **?**) requires disambiguating many tweets together for applying a global model.

Neural Approaches The first published attempt of using deep neural networks (DNNs) for NED was by He et al. (2013), that used stacked auto-encoders to learn a similarity measure between mention-context structures and entity candidates. Recently, the increasing popularity of DNNs 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 measure entity-context relatedness using entity and context embeddings (Yamada et al., 2016; Hu et al., 2015).

In this paper, we train a Recurrent Neural Network (RNN) model, which unlike CNNs and embeddings, are designed to exploit the sequential nature of text. Moreover, we implement a neu-

ral attention mechanism, inspired by results from Lazic et al. (2015) that successfully used a probabilistic attention-like model for NED.

Noisy Data Chisholm and Hachey (2015) have shown that despite the noisy nature of web data, augmenting Wikipedia derived data with weblinks from the Wikilinks corpus (Singh et al., 2012) can lead to improved performance on standard datasets. Our work focuses on using a subset of Wikilinks to construct a noisy test case initself, rather then using it for training alone. Moreover, we argue that manually designing features for noisy text can be difficult and therefore use DNNs to automatically capture useful features.

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. (2010). 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 either experts, or a high-quality community effort in the case of Wikipedia. It is also significantly larger in size, being two orders of magnitude larger then existing news-based datasets.

Recently, a number of Twitter-based datasets were compiled as well (Fromreide et al., 2014). 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 The WikilinksNED Dataset: Entity Mentions in the Web

We introduce WikilinksNED, 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 especially due to incoherent contexts. The contextual noise presents an interesting test-case that supplements existing datasets that are sourced from mostly coherent and well-formed text (news and Wikipedia).

To get a sense of textual noise 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 embeddings of the correct entity assignment and the mean of context words. We compared results from all mentions in CoNLL-YAGO to a sample of 50000 web fragments taken from WikilinksNED, using a window of words of size 40 around entity mentions. We have found the similarity between context and the correct entity is indeed lower for web mentions, and found this result to be statistically significant with very high probability ($p < 10^{-5}$) . This result indicates that web fragments in WikilinksNED are indeed 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

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

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

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.

Finally, we randomly split the data into train (90%), validation (10%), and test (10%), according to website domains in order to minimize lexical memorization (Levy et al., 2015).

4 Algorithm

Our DNN model is a discriminative model which takes a pair of local context and candidate entity, and outputs a probability-like score 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 a probability-like score of the candidate being a correct or corrupt assignment.

4.1 Model Architecture

Figure 1 illustrates the main components of our architecture: an embedding layer, a duo of ARNNs, each processing one side of the context (left and right)[3], and a classifier.

Embedding The embedding layer first embeds both the entity and the context words as vectors[4].

[3]Right context is fed into the ARNN in reverse order

[4]We use vectors with 300 dimensions for both words and entities

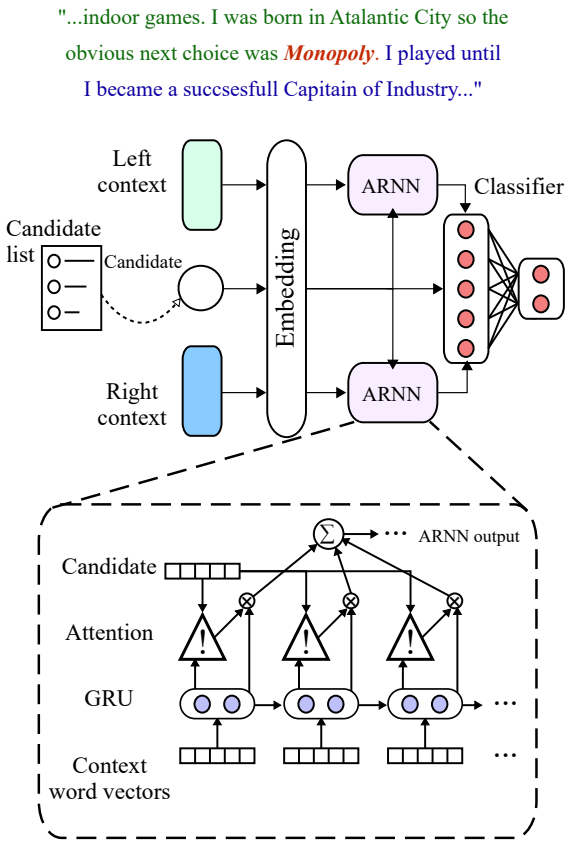

Figure 1: The architecture of our Neural Network model. A close-up of the Attention-RNN component appears in the dashed box.

ARNN The ARNN unit is composed from an RNN and an attention mechanism. 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 using some function f, and an output is computed using g. 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 our implementation we 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, commonly used with state-of-the-art encoder-decoder models (Bahdanau et al., 2014; Xu et al., 2015). 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.

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

The function r computes an attention value at each step, using the RNN output o_t and the candidate entity $v_{candidate}$. The final output vector o_{attn} is a fixed-size vector, which is the sum of all the output vectors of the RNN weighted according to the attention values. 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.

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

Classifier The classifier network consists of a hidden layer[5] and an output layer with two output units in a softmax. The output units are trained by optimizing a cross-entropy loss function.

4.2 Training

We assume our model is only given examples of correct entity assignments during training and therefore use *corrupt-sampling*, where we 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 considered two alternatives for corrupt sampling and provide an empirical comparison of the two approaches (Section 5):

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.

We sample corrupt examples uniformly in both alternatives since with uniform sampling the ratio between the number of positive and negative examples of an entity is higher for popular entities, thus biasing the network towards popular entities. In the All-Entity case, this ratio is approximately proportional to the prior probability of the entity.

We note that preliminary experiments revealed that corrupt-sampling according to the distribution of entities in the dataset (as is done by Mikolov at el. (2013)), rather then uniform sampling, 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-sampling.

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 pretrained over a large collection of unlabeled data significantly improved performance (see Section 5.3). To this end, we implemented an SGNS-based approach (Mikolov et al., 2013) that simultaneously trains both word and entity vectors.

We used `word2vecf`[6] (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

[5] 300 dimensions with ReLU, and $p = 0.5$ dropout.

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

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	72.1	72.2

Table 1: Evaluation on noisy web data (WikilinksNED)

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-sampling 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. Our model was implemented using the Keras (Chollet, 2015) and Tensorflow (Abadi et al., 2015) libraries.

5.1 WikilinksNED

Training we use Near-Misses corrupt-sampling which was found to perform well due to a large training set that represents the test set 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. (2016) 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[7]. 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. Experimental results comparing our model with the baselines are reported in Table 1. Our RNN model significantly outperforms Yamada at el. on this data by over 5 points, indicating that the more expressive RNNs are indeed beneficial for this task. We find that the attention mechanism further improves our results by a small, yet statistically significant, margin.

When running Cheng at el (2013) we used a pre-trained model supplied by the authors, which, similarly to the setting used for evaluating the

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

GLOW algorithm by Ratinov at el (Ratinov and Roth, 2011), 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.

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. To fit our model to this dataset we first used a simple domain adaptation technique and then incorporated a number of basic statistical and string based features.

Domain Adaptation We used a simple domain adaptation technique where we first trained our model on an available large corpus of label data derived from Wikipedia, and then trained the resulting model on the smaller training set of CoNLL (Mou et al., 2016). The Wikipedia corpus was built by extracting all cross-reference links along with their context, resulting in over 80 million training examples. We trained our model with All-Entity corrupt sampling for 1 epoch on this data. The resulting model was then adapted to CoNLL-YAGO by training 1 epoch on CoNLL-YAGO's training set, where corrupt examples were produced by considering all possible candidates for each mention as corrupt-samples (Near-Misses corrupt sampling).

Additional Features We proceeded to use the model in a similar setting to Yamada et al. (2016) where a Gradient Boosting Regression Tree (GBRT) (Friedman, 2001) model was trained with our model's prediction as a feature along with a number of statistical and string based features defined by Yamada. The statistical features include entity prior probability, conditional probability, number of candidates for the given mention and maximum conditional probability of the entity in the document. The string based features include edit distance between mention and entity title and two boolean features indicating whether the entity title starts or ends with the mention and vice versa. The GBRT model parameters where set to the values reported as optimal by Yamada[8].

[8]Learning rate of 0.02; maximal tree depth of 4; 10,000 trees.

Candidate Generation For comparability with existing methods we used two publicly available candidates datasets: (1) PPRforNED - Pershina at el. (2015); (2) YAGO - Hoffart at el. (2011).

Baselines As a baseline we took the standard Most Probable Sense (MPS) prediction, which selects the entity that was seen most with the given mention during training. We also compare to the following papers - Francis-Landau et al. (2016), Yamada at el. (2016), and Chisholm et al. (2015), 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 comparable results, however it does not outperform the state-of-the-art, probably because of the relatively small training set and our reliance on domain adaptation.

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 results in significant performance gains, due to the better starting point. We have also found that using Near-Misses, our model achieves significantly improved performance. We attribute this difference to the more efficient nature of training with near misses.

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

Table 3: Corrupt-sampling and Initialization

6 Error Analysis

We randomly sampled and manually analyzed 200 individual cases of prediction errors made by our model. This set was obtained from WikilinksNED's validation set that was not used for training.

Working with crowd-sourced data, we expected some errors to result from noise in the ground truths themselves. Indeed, we found that 19.5% (39/200) of the prediction errors were not false, out of which 5% (2) where wrong labels, 33% (13) were predictions with an equivalent meaning as the correct entity, and in 61.5% (24) our 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.

In addition, 15.5% (31/200) were cases where a Wikipedia disambiguation-page was chosen as either the correct or predicted entity (2.5% and 14%, respectively). We considered the rest of the 130 errors as true semantic errors, and analyzed them in-depth.

First, we noticed that in 31.5% of the true errors (41/130) our model selected an entity that can be understood as a specific (6%) or general (25%) realization of the correct solution. For example, instead of predicting *'Aroma of wine'* for a text on the scent and flavor of Turkish wine, the model assigned the mention *'Aroma'* with the general *'Odor'* entity. We observed that in 26% (34/130) of the error cases, the predicted entity had a very strong semantic relationship to the correct entity. A closer look discovered two prominent types of 'almost correct' errors occurred repeatedly in the data. The first was a film/book/theater type of error (8.4%), where the actual and the predicted entities were a different display of the same narrative. Even though having different jargon and producers, those fields share extremely similar content, which may explain why they tend to be frequently confused by the algorithm. A third (4/14) of those cases were tagged as truly ambiguous even for human reader. The second prominent type of 'almost correct' errors where 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 (11/130).

Error type	**Fraction**	
False errors		
Not errors	19.5%	(39/200)
- Annotation error	5%	(2/39)
- Better suggestion	61.5%	(24/39)
- Equivalent entities	33%	(13/39)
Disambiguation page	15.5%	(31/200)
True semantic errors		
Too specific/general	31.5%	(41/130)
'almost correct' errors	26%	(34/130)
insufficient training	21.5%	(28/130)

Table 4: Error distribution in 200 samples. Categories of true errors are not fully distinct.

Another interesting type of errors where in cases where the correct entity had insufficient training. We defined insufficient training errors as errors where the correct entity appeared less than 10 times in the training data. We saw that the model followed the MPS in 75% of these cases, showing that our model tends to follow the baseline in such cases. Further, the amount of generalization error in low-count conditions was also significant (35.7%), as our model tended to select more general entities.

7 Conclusions

Our results indicate that the expressibility of attention-RNNs indeed allows us to extract useful features from noisy context, when sufficient amounts of training examples are available. This allows our model to significantly out-perform existing state-of-the-art models. We find that both using pre-initialized embedding vocabularies, and the corrupt-sampling method employed are very

important for properly training our model.

However, the gap between results of all systems tested on both CoNLL-YAGO and WikilinksNED indicates that mentions with noisy context are indeed a challenging test. We believe this to be an important real-world scenario, that represents a distinct test-case that fills a gap between existing new-based datasets and the much noisier Twitter data (Ritter et al., 2011) that has received increasing attention. We find recurrent neural models are a promising direction for this task, while there is still room for improvement.

Finally, our error analysis shows a number of possible improvements that should be addressed. Since we use the training set for candidate generation, non-nonsensical candidates (i.e. disambiguation pages) cause our model to err and should be removed from the candidate set. In addition, we observe that lack of sufficient training for long-tail entities is still a problem, even when a large training set is available. We believe this, and some subtle semantic cases (book/movie) can be at least partially addressed by considering semantic properties of entities, such as types and categories. We intend to address these issues in future work.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Luisa Bentivogli, Pamela Forner, Claudio Giuliano, Alessandro Marchetti, Emanuele Pianta, and Kateryna Tymoshenko, 2010. *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, chapter Extending English ACE 2005 Corpus Annotation with Ground-truth Links to Wikipedia, pages 19–27. Coling 2010 Organizing Committee.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425. Association for Computational Linguistics.

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1787–1796. Association for Computational Linguistics.

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

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics.

Franois Chollet. 2015. keras. `https://github.com/fchollet/keras`.

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

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1256–1261. Association for Computational Linguistics.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Hege Fromreide, Dirk Hovy, and Anders Søgaard. 2014. Crowdsourcing and annotating ner for twitter #drift. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 1606–1611.

Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringaard, and Fernando Pereira. 2016. Collective entity resolution with multi-focal attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 621–631. Association for Computational Linguistics.

Zhaochen Guo and Denilson Barbosa. 2014. Entity linking with a unified semantic representation. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 1305–1310, New York, NY, USA. ACM.

Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. 2013. Learning entity representation for entity disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–34. Association for Computational Linguistics.

Johannes Hoffart, Amir Mohamed 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 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics.

Zhiting Hu, Poyao Huang, Yuntian Deng, Yingkai Gao, and Eric Xing. 2015. Entity hierarchy embedding. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1292–1300. 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 of Computational Linguistics*, 3:503–515.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2177–2185.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976. Association for Computational Linguistics.

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*, CIKM '07, pages 233–242, New York, NY, USA. ACM.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 479–489. Association for Computational Linguistics.

Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized page rank for named entity disambiguation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 238–243. Association for Computational Linguistics.

Lev-Arie Ratinov and Dan Roth. 2011. GLOW TAC-KBP2011 entity linking system. In *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1375–1384. Association for Computational Linguistics.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks:

A large-scale cross-document coreference corpus labeled via links to wikipedia. *University of Massachusetts, Amherst, Tech. Rep. 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 Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1333–1339.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2048–2057.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2016. Joint learning of the embedding of words and entities for named entity disambiguation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 250–259. Association for Computational Linguistics.