Effective Use of Context in Noisy Entity Linking

David Mueller
Department of Computer Science
Johns Hopkins University*
dam@jhu.edu

Greg Durrett
Department of Computer Science
The University of Texas at Austin
gdurrett@cs.utexas.edu

Abstract

To disambiguate between closely related concepts, entity linking systems need to effectively distill cues from a mention's textual context. We investigate several techniques for using these cues in the task of noisy entity linking on short texts. Our starting point is a state-of-the-art attention-based model from prior work; while this model's attention typically identifies context that is topically relevant, it fails to identify some of the most indicative context words, especially those exhibiting lexical overlap with the true title. Augmenting the model with convolutional networks over characters still leaves it largely unable to pick up on these cues compared to sparse features that target them directly, indicating that automatically learning how to identify relevant character-level context features is a hard problem. Armed with these sparse features, our final system[1] outperforms past work on the WikilinksNED test set by 2.8% absolute.

1 Introduction

Effectively using an entity mention's context to disambiguate it is the crux of the entity linking task: in isolation, the mention *Richard Wright* could refer to three possible entities in Wikipedia's knowledge base corresponding to an artist, a musician, or an author. Previous work in this area has distilled context information by exploiting tf-idf features (Cucerzan, 2007; Milne and Witten, 2008; Ratinov et al., 2011), global link coherence (Hoffart et al.; Sil and Florian, 2016), cues from coreference (Cheng and Roth, 2013; Hajishirzi et al., 2013; Durrett and Klein, 2014), convolutional neural networks (Sun et al.; Francis-Landau et al., 2016), or more sophisticated neural architectures (Gupta et al., 2017; Sil et al., 2018). These approaches typically focus on aggregating information from a mix of sources, including long-range information from the textual context or other linked entities. While this approach is suitable for entity linking settings such as newswire (Bentivogli, 2010) and Wikipedia (Ratinov et al., 2011), we cannot always rely on this information in other settings like Twitter (Guo et al., 2013; Fang and Chang, 2014; Huang et al., 2014; Dredze et al., 2016), Snapchat (Moon et al., 2018), other web platforms (Eshel et al., 2017), or dialogue systems (Bowden et al., 2018). We need models that can make effective use of limited context windows in noisy settings.

In this work, we investigate this problem of effectively using context in the setting of the WikilinksNED dataset from Eshel et al. (2017). The examples in this dataset, which consists of 3.2 million entity disambiguation examples derived from Wikilinks (Singh et al., 2012), have at most 20 words of context on either side and usually no other mentions of the entity being disambiguated. We build off a state-of-the-art attentive LSTM model from prior work (Eshel et al., 2017) and show that despite its good performance, it fails to resolve some examples that human readers would find trivial. For example, disambiguating the identity of the song *Down* in Figure 1 is easy if we can recognize the nearby string *Jay Sean* in the context, but the model sometimes fails to do this.

We explore the performance of a standard attention mechanism as well as two modifications. First, we inject character information into the model through character-level CNNs; these give the model a deeper ability to recognize character correspondences between the context and entity title. However, these convolutional filters struggle to learn useful features in this noisy context and ultimately do not help performance. By contrast, sparse features explicitly targeting these overlaps

*Work done while at UT Austin.

[1]Code available at https://github.com/davidandym/wikilinks-ned

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1024–1029
Brussels, Belgium, October 31 - November 4, 2018.

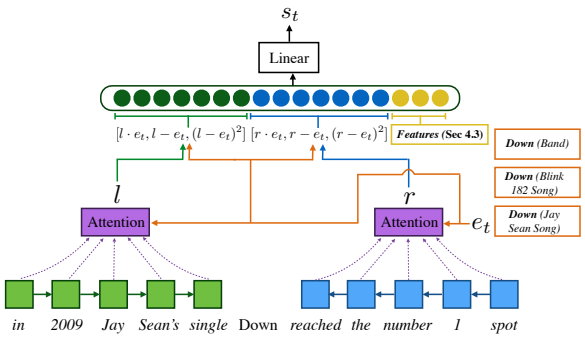

Figure 1: Neural entity linking model. *Down* has three possible link targets: in the attentive variants of our model, each target computes attention weights over GRUs that consume the left and right context. These representations are passed into a layer that compares them to the entity's embedding, yielding a final score which is normalized over all possible link targets.

appear to be more successful. We investigate the relation between our model's attention and what the sparse features learn. Our final model, using these features, achieves an accuracy of 75.8% on this dataset, substantially outperforming our baseline model as well as results from prior work.

2 Basic Model

The WikilinksNED dataset consists of entity mentions in context scraped from the web, with gold annotation derived from the fact that those mentions originally appeared with hyperlinks to Wikipedia. We denote the mention text (i.e., anchor text of the hyperlink) by m, and denote the left and right context of the mention by c_l and c_r respectively; these are at most 20 words. For this dataset, we can assume that the possible linked titles for a mention have been seen in training, and the main task is instead to disambiguate between them and identify the gold title t^*. We therefore follow prior work (Eshel et al., 2017) and take as candidates all gold entities in the training set whose mention was m rather than relying on a separate candidate generation scheme.

Our model places a distribution over titles $P(t|m, c_l, c_r)$, where t takes values in the set of candidate Wikipedia titles for that mention. This model, depicted in Figure 1, roughly follows that of Eshel et al. (2017), with some key differences, as we discuss in the rest of this section.

Embedding contexts Given an example of the form (m, c_l, c_r), our model first uses a GRU layer (Cho et al., 2014) over each context to convert c_l and c_r into continuous vector representations l and r, respectively. Our word embeddings are trained over Wikipedia as described in the following paragraph.

Embedding entities We follow the method of Eshel et al. (2017) for generating entity embeddings, using `word2vecf` (Levy and Goldberg, 2014) to jointly train word and entity embeddings simultaneously using Wikipedia article text. Each title t is associated in turn with each content word w in the article, yielding a set of (w, t) pairs that are consumed by the training procedure. This yields a set of title embeddings e_t which we can treat as distributed representations of entities.

Entity-context comparison We systematically compare the representations for l, r, and e_t as follows:

$$[l \cdot e_t, r \cdot e_t, l - e_t, r - e_t, (l - e_t)^2, (r - e_t)^2]$$

where $\cdot$ denotes the conventional dot product and the other comparisons are elementwise. These features form the input to a final feedforward layer which produces a real-valued score s_t for the given title. Repeating this computation for each title, our model's distribution is $P(t|m, c_l, c_r) = \text{softmax}_t(s_t)$.

Training Because our model involves substantial computation for each possible title, we want to limit the set of titles considered during training. For each example we consider, we construct a set T containing the gold title and 4 negative "distractor" titles from the candidate set. Unlike Eshel et al. (2017), we structure training as a multiclass decision among these titles rather than a binary prediction problem over each title as gold or not. We run our model over the candidates $t \in T$ to produce the distribution $P(t|m, c_l, c_r)$ and train to maximize the log probablity $\log P(t^*|m, c_l, c_r)$ of the gold title.

Results The model set forth in this section is the basis for the remaining models in this paper; we call it the GRU model as that is the only context encoding mechanism it uses. As shown in Table 1, this GRU model gets a score of 73.4 on the WikilinksNED development set. In the next section, we explore techniques for using the context in a more sophisticated way to improve further on this result.

Model	Accuracy on Test (%)
Eshel et al. (2017)	73.0
Eshel system release	72.2
GRU+ATTN	74.5
GRU+ATTN+FEATS	**75.8**
Model	**Accuracy on Dev (%)**
GRU	73.4
GRU+ATTN	74.4
GRU+ATTN+FEATS	**74.9**
GRU+ATTN+CNN	73.8

Table 1: Results on the WikilinksNED dev and test sets. Our model including features achieves state-of-the-art performance on the test set, compared to both the reported numbers from Eshel et al. (2017) as well as their released software. Incorporating character CNNs surprisingly leads to lower performance compared to these simple features.

3 Exploiting Context Cues

3.1 Attention

One way to improve over the basic GRU model is to use attention over the context based on the title under consideration. The attention we use is a modified version of the dot product attention (Luong et al., 2015) used by Eshel et al. (2017), allowing the model to weight the importance of the outputs of the GRU at each time step. Each context (left and right) has its own attention weights. For a given side of context and candidate t, the attention first computes a transformation of the entity embedding e_t as follows: $q_t = \tanh(We_t)$. This allows the model to learn an attention query q_t distinct from the candidate embedding e_t. The model then computes attention probabilities α_i for each GRU output o_i, normalized over the entire sequence of GRU outputs (of length n):

$$\alpha_i = \text{softmax}_i(q_t \cdot o_i)$$

The resulting probability distribution is used to take a weighted sum of GRU outputs to get a representation a:

$$a = \sum_i^n \alpha_i o_i$$

We compute a_l and a_r independently and symmetrically for the left and right context. These vectors are then fed forward through the model as the final continuous representation of the left or right context, l or r respectively.

Results In Table 1, we see that our model with attention (GRU+ATTN) outperforms our basic GRU model by around 1% absolute. It also outperforms the roughly similar model of Eshel et al. (2017) on the test set: this gain is due to a combination of factors including the improved training procedure and some small modeling changes.[2] However, our attention scheme is not without its shortcomings, as we now discuss.

3.2 Shortcomings of Attention

One common and frustrating error our model makes is failing to correctly disambiguate mentions whose contexts share similar words or character overlap with the gold entity's actual Wikipedia title. In these instances, the model fails to attend correctly to words that we, as human readers, would most likely see as disambiguating terms. For instance, in this example's left context:

> ...known also for the B.P. Koirala Institute of Health Sciences, one of the biggest government hospital. The indigenous **people** of Dharan are *Limbu* ...

the model fails to identify **people** as a critical term for disambiguation. This failure is partially due to the model's sole reliance on distributed representations: the embedding for **people** and the title embedding for *Limbu People* need to somehow contain enough common information for the model to associate these, identify **people** as an important token, and use it to disambiguate between candidate titles such as *Limbu People*, *Limbu Language*, and *Limbu Alphabet*. Moreover, with such noisy, unstructured context, it is difficult for the model to learn to rely on other grammatical or semantic cues (such as *are* indicating that the title is probably a plural noun, which *alphabet* and *language* are not).

3.3 Character CNNs

One way to address these issues in the model is to exploit more fine-grained character-level information. This circumvents the need to separately learn a distributed correspondence between terms with lexical overlap, and is especially useful when these terms may be unknown words; for example, a year mentioned within a context is often

[2] Note that in Eshel et al. (2017), the authors point out that their dataset has a high percentage of errors (35% of the errors made by their model are spurious), meaning that the skyline on this task is likely not higher than 90%.

unknown and therefore assigned an UNK embedding, even if that year matches exactly with a year in the gold candidate's title.

One solution to this is to allow our model to consult character-level information, which past models have done successfully for named entity recognition (Chiu and Nichols, 2015; Lample et al., 2016; Ma and Hovy, 2016), text classification (Zhang et al., 2015), and POS tagging (Santos and Zadrozny, 2014). We use convolutional neural networks (CNNs) to distill character representations of words into vectors that we concatenate with our word representations. We additionally use character CNNs over entity titles and concatenate these representations with the title embeddings e_t, to allow the model to learn to characterize similarities between contexts and entity titles. Our CNNs use window sizes of 6 and 100 filters each; these values were selected through hyperparameter tuning on the development set.

Table 1 shows the impact of incorporating character CNNs (GRU+ATTN+CNN). Surprisingly, these have a mild negative impact on performance. One possible explanation of this is that it causes the model to split its attention between semantically important and lexically similar context terms. Consider the following example:

> really think Final Fight could be a lot of a fun as a vigilante justice **movie** with a high quotient of hand-to-hand fight sequences. Think *The Warriors*

The gold title is *The Warrior (film)* and the base model correctly places 90% of its attention weight on the word **movie** when calculating attention for this title. However, the character-level CNN model only places 60% of its attention weight on it, distributing its attention values more evenly across the rest of the words. Such cases are frequent: the average highest weight given by attention in GRU+ATTN+CNN is about 6% lower than the average highest attention weight given by GRU+ATTN. The CNNs seem to have generally decreased the model's confidence in what context clues are key for disambiguation, leading to lower performance. We will return to more analysis of this in Section 4.

3.4 Lexical Feature Set

To determine whether character level overlap between the entity title and context is useful, we take a more direct approach to incorporating that information into our model and build a set of sparse features that directly target it.

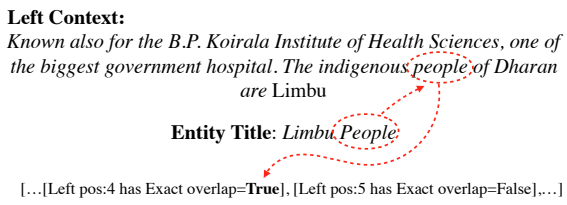

Figure 2: An example of feature generation from an example. Here, because the word *people* occurs in the title and in the left context 4 words away from the mention, the indicator feature [Pos=4,Match=ExactWord,Context=Left] fires in the feature set.

Figure 2 shows an example of how our features are computed. We fire features on each word in the context that is either an exact match or a substring of a word in the candidate title; *people* is the relevant token here. We conjoin that match information with whether the word is in the left or right context along with the bucketed offset of the word from the mention. This feature set is then appended to the vector comparison features to form the input to the model's feedforward layer (see Figure 1).

Table 1 shows the results of stacking these features on top of our model with attention (GRU+ATTN+FEATS). We see our highest development set performance and correspondingly high test performance from this model. This indicates that character-level information is useful for disambiguation, but character CNNs as we incorporated them are not able to distill it as effectively as sparse features can. Our model augmented with these sparse features achieves state-of-the-art results on the test set.

4 Attention and "Obvious" Terms

Now that we have identified features which seem useful for this entity linking problem, we can ask how the tokens attended by our attention mechanism compare to those singled out by the features.

Table 2 contains statistics regarding the attention values of our GRU+ATTN and GRU+ATTN+CNN model on a subset of examples that both models got wrong. We define accuracy as the percentage of examples in which the model gives the highest attention to a word

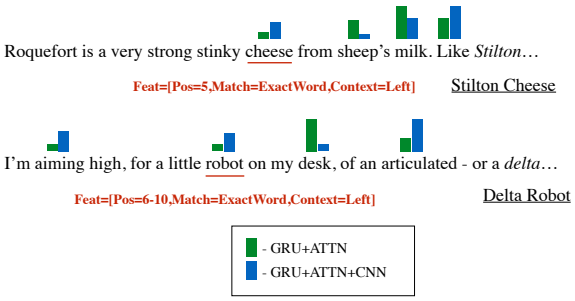

Figure 3: Examples of our models putting high attention weight into irrelevant context words, not acknowledging the relevance of disambiguating terms that share lexical overlap with the correct title. We display the weight given to the top 4 attended words above each word for two of our models.

that contains one of our lexical features, out of all examples where such a feature exists anywhere. The reported probability mass is the total attention mass that the model puts into words that associated with lexical features, averaged over all examples where such features exist. We see that the model frequently fails to exploit this information, and moreover the addition of CNNs does not strongly improve this.

Figure 3 shows examples of this behavior. In the first example, rather than identifying *cheese* as a salient term, both models instead focus more heavily on *milk* and *like*. Similarly, in the second example, the model fails to recognize the importance of *robot* in the context.

One possible reason that CNNs don't help more is that the sparse features only trigger on a subset of examples. Because the CNNs process every example, they may not see enough examples of lexical overlap to pick up on it, and instead try to augment what the word embedding model is already doing with subword information, which ends up being unstable for this task. Naturally, words with these overlap characteristics are not always the most disambiguating term. However, in light of noisy contexts, when the standard representation of context fails to be sufficient for allowing the model to disambiguate, we want the model to be able to leverage this character level information to help it make intuitive decisions, which the CNN fails to do.

5 Conclusion

In this paper, we observed that in noisy entity linking settings on short texts, neural models relying on attention do not always pick up on the correct context clues, even when those clues exhibit very obvious surface overlap with the correct entity title. These models can perform better when augmented with sparse features explicitly targeting this kind of lexical overlap: our system using these features achieves state-of-the-art disambiguation accuracy on the WikilinksNED dataset. By contrast, automatically learning learning fine-grained character-level features with CNNs in this context is hard. More exploration is needed to better understand what inductive biases are necessary for an entity linking system to make maximally effective use of the information available to it.

Context	Acc (%)	Prob Mass (%)
GRU+ATTN L	0.41	0.32
GRU+ATTN R	0.36	0.30
GRU+ATTN+CNN L	0.46	0.32
GRU+ATTN+CNN R	0.36	0.28

Table 2: Our models' attention "accuracy": how often each model's maximally-attended word also triggered a feature to fire. Prob Mass indicates the average sum of attention scores over feature-triggering words. All values are computed over a sample of 10,000 examples that each model got wrong.

Acknowledgments

This work was partially supported by NSF Grant IIS-1814522, a Bloomberg Data Science Grant, and an equipment grant from NVIDIA. The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources used to conduct this research. Thanks as well to the anonymous reviewers for their helpful comments.

References

Luisa Bentivogli. 2010. Extending English ACE 2005 Corpus Annotation with Ground-truth Links to Wikipedia. In *Proceedings of COLING*.

Kevin K. Bowden, Jiaqi Wu, Shereen Oraby, Amita Misra, and Marilyn Walker. 2018. SlugNERDS: A Named Entity Recognition Tool for Open Domain Dialogue Systems. In *Proceedings of LREC*.

Xiao Cheng and Dan Roth. 2013. Relational Inference for Wikification. In *Proceedings of EMNLP*.

Jason P. C. Chiu and Eric Nichols. 2015. Named Entity Recognition with Bidirectional LSTM-CNNs. In *Proceedings of ACL*.

Kyunghyun Cho, Bart van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of EMNLP*.

Silviu Cucerzan. 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of EMNLP-CoNLL*.

Mark Dredze, Nicholas Andrews, and Jay DeYoung. 2016. Twitter at the Grammys: A Social Media Corpus for Entity Linking and Disambiguation. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*.

Greg Durrett and Dan Klein. 2014. A Joint Model for Entity Analysis: Coreference, Typing, and Linking. In *TACL*.

Yotam Eshel, Noam Cohen, Kira Radinsky, Shaul Markovitch, Ikuya Yamada, and Omer Levy. 2017. Named Entity Disambiguation for Noisy Text. In *Proceedings of CoNLL*.

Yuan Fang and Ming-Wei Chang. 2014. Entity Linking on Microblogs with Spatial and Temporal Signals. In *TACL*.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks. In *Proceedings of NAACL*.

Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To Link or Not to Link? A Study on End-to-End Tweet Entity Linking. In *Proceedings of NAACL*.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity Linking via Joint Encoding of Types, Descriptions, and Context. In *Proceedings of EMNLP*.

Hannaneh Hajishirzi, Leila Zilles, Daniel S. Weld, and Luke Zettlemoyer. 2013. Joint Coreference Resolution and Named-Entity Linking with Multi-Pass Sieves. In *Proceedings of EMNLP*.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust Disambiguation of Named Entities in Text. In *Proceedings of EMNLP*.

Hongzhao Huang, Yunbo Cao, Xiaojiang Huang, Heng Ji, and Chin-Yew Lin. 2014. Collective Tweet Wikification based on Semi-supervised Graph Regularization. In *Proceedings of ACL*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of NAACL*.

Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *Proceedings of ACL*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of EMNLP*.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of ACL*.

David Milne and Ian H. Witten. 2008. Learning to Link with Wikipedia. In *Proceedings of CIKM*.

Seungwhan Moon, Leonardo Neves, and Vitor Carvalho. 2018. Multimodal Named Entity Disambiguation for Noisy Social Media Posts. In *Proceedings of the ACL*.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and Global Algorithms for Disambiguation to Wikipedia. In *Proceedings of NAACL*.

Cicero Dos Santos and Bianca Zadrozny. 2014. Learning Character-level Representations for Part-of-Speech Tagging. In *Proceedings of ICML*.

Avirup Sil and Radu Florian. 2016. One for all: Towards language independent named entity linking. In *Proceedings of ACL*.

Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. 2018. Neural cross-lingual entity linking. In *Proceedings of AAAI*.

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.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. Modeling Mention, Context and Entity with Neural Networks for Entity Disambiguation. In *Proceedings of IJCAI*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Proceedings of NIPS*.