Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks

Matthew Francis-Landau, Greg Durrett and **Dan Klein**
Computer Science Division
University of California, Berkeley
{mfl,gdurrett,klein}@cs.berkeley.edu

Abstract

A key challenge in entity linking is making effective use of contextual information to disambiguate mentions that might refer to different entities in different contexts. We present a model that uses convolutional neural networks to capture semantic correspondence between a mention's context and a proposed target entity. These convolutional networks operate at multiple granularities to exploit various kinds of topic information, and their rich parameterization gives them the capacity to learn which n-grams characterize different topics. We combine these networks with a sparse linear model to achieve state-of-the-art performance on multiple entity linking datasets, outperforming the prior systems of Durrett and Klein (2014) and Nguyen et al. (2014).[1]

1 Introduction

One of the major challenges of entity linking is resolving contextually polysemous mentions. For example, *Germany* may refer to a nation, to that nation's government, or even to a soccer team. Past approaches to such cases have often focused on collective entity linking: nearby mentions in a document might be expected to link to topically-similar entities, which can give us clues about the identity of the mention currently being resolved (Ratinov et al., 2011; Hoffart et al., 2011; He et al., 2013; Cheng and Roth, 2013; Durrett and Klein, 2014). But an even simpler approach is to use context information from just the words in the source document itself to make sure the entity is being resolved sensibly in context. In past work, these approaches have typically relied on heuristics such as tf-idf (Ratinov et al., 2011), but such heuristics are hard to calibrate and they capture structure in a coarser way than learning-based methods.

In this work, we model semantic similarity between a mention's source document context and its potential entity targets using convolutional neural networks (CNNs). CNNs have been shown to be effective for sentence classification tasks (Kalchbrenner et al., 2014; Kim, 2014; Iyyer et al., 2015) and for capturing similarity in models for entity linking (Sun et al., 2015) and other related tasks (Dong et al., 2015; Shen et al., 2014), so we expect them to be effective at isolating the relevant topic semantics for entity linking. We show that convolutions over multiple granularities of the input document are useful for providing different notions of semantic context. Finally, we show how to integrate these networks with a preexisting entity linking system (Durrett and Klein, 2014). Through a combination of these two distinct methods into a single system that leverages their complementary strengths, we achieve state-of-the-art performance across several datasets.

2 Model

Our model focuses on two core ideas: first, that topic semantics at different granularities in a document are helpful in determining the genres of entities for entity linking, and second, that CNNs can distill a block of text into a meaningful topic vector.

Our entity linking model is a log-linear model that places distributions over target entities t given a mention x and its containing source document. For now, we take $P(t|x) \propto \exp w^\top f_C(x, t; \theta)$, where f_C produces a vector of features based on CNNs with parameters θ as discussed in Section 2.1. Section 2.2 describes how we combine this simple model with a full-fledged entity linking system. As shown in the middle of Figure 1, each feature in f_C

[1] Source available at github.com/matthewfl/nlp-entity-convnet

Proceedings of NAACL-HLT 2016, pages 1256–1261,
San Diego, California, June 12-17, 2016.

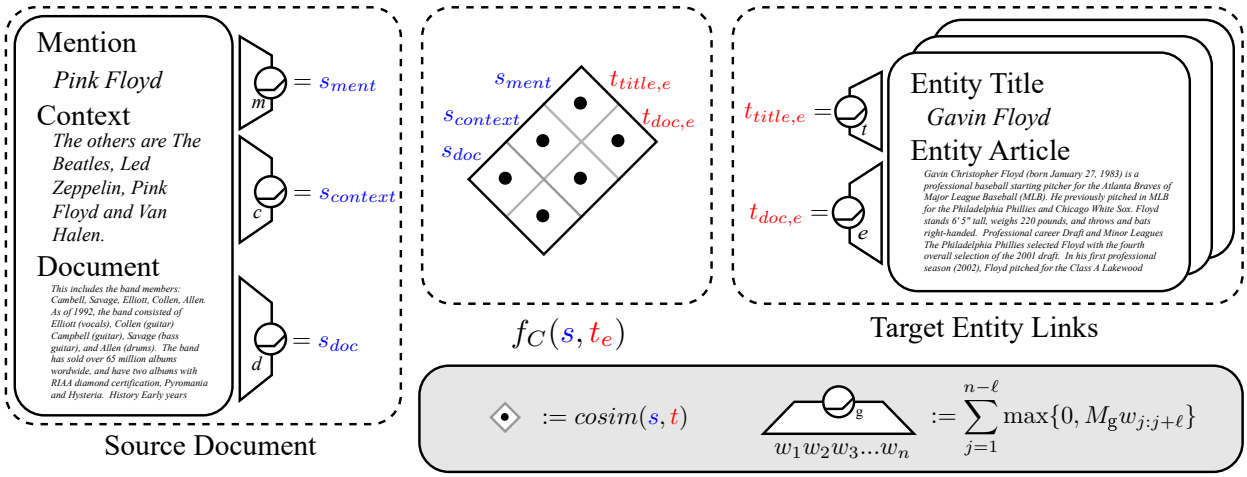

Figure 1: Extraction of convolutional vector space features $f_C(x, t_e)$. Three types of information from the input document and two types of information from the proposed title are fed through convolutional networks to produce vectors, which are systematically compared with cosine similarity to derive real-valued semantic similarity features.

is a cosine similarity between a topic vector associated with the source document and a topic vector associated with the target entity. These vectors are computed by distinct CNNs operating over different subsets of relevant text.

Figure 1 shows an example of why different kinds of context are important for entity linking. In this case, we are considering whether *Pink Floyd* might link to the article `Gavin_Floyd` on Wikipedia (imagine that *Pink Floyd* might be a person's nickname). If we look at the source document, we see that the immediate source document context around the mention *Pink Floyd* is referring to rock groups (*Led Zeppelin*, *Van Halen*) and the target entity's Wikipedia page is primarily about sports (*baseball starting pitcher*). Distilling these texts into succinct topic descriptors and then comparing those helps tell us that this is an improbable entity link pair. In this case, the broader source document context actually does not help very much, since it contains other generic last names like *Campbell* and *Savage* that might not necessarily indicate the document to be in the music genre. However, in general, the whole document might provide a more robust topic estimate than a small context window does.

2.1 Convolutional Semantic Similarity

Figure 1 shows our method for computing topic vectors and using those to extract features for a potential Wikipedia link. For each of three text granularities in the source document (the mention, that mention's immediate context, and the entire document) and two text granularities on the target entity side (title and Wikipedia article text), we produce vector representations with CNNs as follows. We first embed each word into a d-dimensional vector space using standard embedding techniques (discussed in Section 3.2), yielding a sequence of vectors $w_1, \ldots, w_n$. We then map those words into a fixed-size vector using a convolutional network parameterized with a filter bank $M \in \mathbb{R}^{k \times d\ell}$. We put the result through a rectified linear unit (ReLU) and combine the results with sum pooling, giving the following formulation:

$$\text{conv}_g(w_{1:n}) = \sum_{j=1}^{n-\ell} \max\{0, M_g w_{j:j+\ell}\} \quad (1)$$

where $w_{j:j+\ell}$ is a concatenation of the given word vectors and the max is element-wise.[2] Each convolution granularity (mention, context, etc.) has a distinct set of filter parameters M_g.

This process produces multiple representative topic vectors s_{ment}, $s_{context}$, and s_{doc} for the source document and t_{title} and t_{doc} for the target entity, as shown in Figure 1. All pairs of these vectors between the source and the target are then compared using cosine similarity, as shown in the middle of Figure 1. This yields the vector of features $f_C(s, t_e)$ which indicate the different types of similarity; this

[2]For all experiments, we set $\ell = 5$ and $k = 150$.

vector can then be combined with other sparse features and fed into a final logistic regression layer (maintaining end-to-end inference and learning of the filters). When trained with backpropagation, the convolutional networks should learn to map text into vector spaces that are informative about whether the document and entity are related or not.

2.2 Integrating with a Sparse Model

The dense model presented in Section 2.1 is effective at capturing semantic topic similarity, but it is most effective when combined with other signals for entity linking. An important cue for resolving a mention is the use of link counts from hyperlinks in Wikipedia (Cucerzan, 2007; Milne and Witten, 2008; Ji and Grishman, 2011), which tell us how often a given mention was linked to each article on Wikipedia. This information can serve as a useful prior, but only if we can leverage it effectively by targeting the most salient part of a mention. For example, we may have never observed *President Barack Obama* as a linked string on Wikipedia, even though we have seen the substring *Barack Obama* and it unambiguously indicates the correct answer.

Following Durrett and Klein (2014), we introduce a latent variable q to capture which subset of a mention (known as a *query*) we resolve. Query generation includes potentially removing stop words, plural suffixes, punctuation, and leading or tailing words. This processes generates on average 9 queries for each mention. Conveniently, this set of queries also defines the set of candidate entities that we consider linking a mention to: each query generates a set of potential entities based on link counts, whose unions are then taken to give on the possible entity targets for each mention (including the null link). In the example shown in Figure 1, the query phrases are *Pink Floyd* and *Floyd*, which generate `Pink_Floyd` and `Gavin_Floyd` as potential link targets (among other options that might be derived from the *Floyd* query).

Our final model has the form $P(t|x) = \sum_q P(t, q|x)$. We parameterize $P(t, q|x)$ in a log-linear way with three separate components:

$$P(t, q|x) \propto \exp\left(w^\top (f_Q(x, q) + f_E(x, q, t) + f_C(x, t; \theta))\right)$$

f_Q and f_E are both sparse features vectors and are taken from previous work (Durrett and Klein, 2014). f_C is as discussed in Section 2.1. Note that f_C has its own internal parameters θ because it relies on CNNs with learned filters; however, we can compute gradients for these parameters with standard backpropagation. The whole model is trained to maximize the log likelihood of a labeled training corpus using Adadelta (Zeiler, 2012).

The indicator features f_Q and f_E are described in more detail in Durrett and Klein (2014). f_Q only impacts which query is selected and not the disambiguation to a title. It is designed to roughly capture the basic shape of a query to measure its desirability, indicating whether suffixes were removed and whether the query captures the capitalized subsequence of a mention, as well as standard lexical, POS, and named entity type features. f_E mostly captures how likely the selected query is to correspond to a given entity based on factors like anchor text counts from Wikipedia, string match with proposed Wikipedia titles, and discretized cosine similarities of tf-idf vectors (Ratinov et al., 2011). Adding tf-idf indicators is the only modification we made to the features of Durrett and Klein (2014).

3 Experimental Results

We performed experiments on 4 different entity linking datasets.

- ACE (NIST, 2005; Bentivogli et al., 2010): This corpus was used in Fahrni and Strube (2014) and Durrett and Klein (2014).
- CoNLL-YAGO (Hoffart et al., 2011): This corpus is based on the CoNLL 2003 dataset; the test set consists of 231 news articles and contains a number of rarer entities.
- WP (Heath and Bizer, 2011): This dataset consists of short snippets from Wikipedia.
- Wikipedia (Ratinov et al., 2011): This dataset consists of 10,000 randomly sampled Wikipedia articles, with the task being to resolve the links in each article.[3]

[3] We do not compare to Ratinov et al. (2011) on this dataset because we do not have access to the original Wikipedia dump they used for their work and as a result could not duplicate their results or conduct comparable experiments, a problem which was also noted by Nguyen et al. (2014).

	ACE	CoNLL	WP	Wiki[4]
Previous work				
DK2014	79.6	—	—	—
AIDA-LIGHT	—	84.8	—	—
This work				
Sparse features	83.6	74.9	81.1	81.5
CNN features	84.5	81.2	87.7	75.7
Full	**89.9**	**85.5**	**90.7**	**82.2**

Table 1: Performance of the system in this work (Full) compared to two baselines from prior work and two ablations. Our results outperform those of Durrett and Klein (2014) and Nguyen et al. (2014). In general, we also see that the convolutional networks by themselves can outperform the system using only sparse features, and in all cases these stack to give substantial benefit.

We use standard train-test splits for all datasets except for WP, where no standard split is available. In this case, we randomly sample a test set. For all experiments, we use word vectors computed by running word2vec (Mikolov et al., 2013) on all Wikipedia, as described in Section 3.2.

Table 1 shows results for two baselines and three variants of our system. Our main contribution is the combination of indicator features and CNN features (Full). We see that this system outperforms the results of Durrett and Klein (2014) and the AIDA-LIGHT system of Nguyen et al. (2014). We can also compare to two ablations: using just the sparse features (a system which is a direct extension of Durrett and Klein (2014)) or using just the CNN-derived features.[5] Our CNN features generally outperform the sparse features and improve even further when stacked with them. This reflects that they capture orthogonal sources of information: for example, the sparse features can capture how frequently the target document was linked to, whereas the CNNs can capture document context in a more nuanced way. These CNN features also clearly supersede the sparse features based on tf-idf (taken from (Ratinov et al., 2011)), showing that indeed that CNNs are better at learning semantic topic similarity than heuristics like tf-idf.

In the sparse feature system, the highest weighted features are typically those indicating the frequency that a page was linked to and those indicating specific lexical items in the choice of the latent query variable q. This suggests that the system of Durrett and Klein (2014) has the power to pick the right span of a mention to resolve, but then is left to generally pick the most common link target in Wikipedia, which is not always correct. By contrast, the full system has a greater ability to pick less common link targets if the topic indicators distilled from the CNNs indicate that it should do so.

[4]The test set for this dataset is only 40 out of 10,000 documents and subject to wide variation in performance.

[5]In this model, the set of possible link targets for each mention is still populated using anchor text information from Wikipedia (Section 2.2), but note that link counts are not used as a feature here.

	ACE	CoNLL	WP
$cosim(s_{doc}, t_{doc})$	77.43	79.76	72.93
$cosim(s_{ment}, t_{title})$	80.19	80.86	70.25
All CNN pairs	84.85	86.91	82.02

Table 2: Comparison of using only topic information derived from the document and target article, only information derived from the mention itself and the target entity title, and the full set of information (six features, as shown in Figure 1). Neither the finest nor coarsest convolutional context can give the performance of the complete set. Numbers are reported on a development set.

3.1 Multiple Granularities of Convolution

One question we might ask is how much we gain by having multiple convolutions on the source and target side. Table 2 compares our full suite of CNN features, i.e. the six features specified in Figure 1, with two specific convolutional features in isolation. Using convolutions over just the source document (s_{doc}) and target article text (t_{doc}) gives a system[6] that performs, in aggregate, comparably to using convolutions over just the mention (s_{ment}) and the entity title (t_{title}). These represent two extremes of the system: consuming the maximum amount of context, which might give the most robust representation of topic semantics, and consuming the minimum amount of context, which gives the most focused representation of topics semantics (and which more generally might allow the system to directly memorize train-test pairs observed in training). However, neither performs as well as the combination of all CNN features, showing that the different granularities capture complementary aspects of the entity linking task.

[6]This model is roughly comparable to Model 2 as presented in Sun et al. (2015).

destroying missiles . spy planes	has died of his wounds	him which was more depressing
and destroying missiles . spy	vittorio sacerdoti has told his	a trip and you see
by U.N. weapons inspectors .	his bail hearing , his	" i can see why
inspectors are discovering and destroying	bail hearing , his lawyer	i think so many americans
are discovering and destroying missiles	died of his wounds after	his life from the depression
an attack using chemical weapons	from scott peterson 's attorney	trip and you see him
discovering and destroying missiles .	's murder trial . she	, but dumb liberal could
attack munitions or j-dam weapons	has told his remarkable tale	i can see why he
sanctions targeting iraq civilians ,	murder trial . she asking	one passage . you cite
its nuclear weapons and missile	trial lawyers are driving doctors	think so many americans are

Table 3: Five-grams representing the maximal activations from different filters in the convolution over the source document (M_{doc}, producing s_{doc} in Figure 1). Some filters tend towards singular topics as shown in the first and second columns, which focus on weapons and trials, respectively. Others may have a mix of seemingly unrelated topics, as shown in the third column, which does not have a coherent theme. However, such filters might represent a superposition of filters for various topics which never cooccur and thus never need to be disambiguated between.

	ACE	CoNLL	WP
Google News	87.5	89.6	83.8
Wikipedia	89.5	90.6	85.5

Table 4: Results of the full model (sparse and convolutional features) comparing word vectors derived from Google News vs. Wikipedia on development sets for each corpus.

3.2 Embedding Vectors

We also explored two different sources of embedding vectors for the convolutions. Table 4 shows that word vectors trained on Wikipedia outperformed Google News word vectors trained on a larger corpus. Further investigation revealed that the Google News vectors had much higher out-of-vocabulary rates. For learning the vectors, we use the standard word2vec toolkit (Mikolov et al., 2013) with vector length set to 300, window set to 21 (larger windows produce more semantically-focused vectors (Levy and Goldberg, 2014)), 10 negative samples and 10 iterations through Wikipedia. We do not fine-tune word vectors during training of our model, as that was not found to improve performance.

3.3 Analysis of Learned Convolutions

One downside of our system compared to its purely indicator-based variant is that its operation is less interpretable. However, one way we can inspect the learned system is by examining what causes high activations of the various convolutional filters (rows of the matrices M_g from Equation 1). Table 3 shows the n-grams in the ACE dataset leading to maximal activations of three of the filters from M_{doc}. Some filters tend to learn to pick up on n-grams characteristic of a particular topic. In other cases, a single filter might be somewhat inscrutable, as with the third column of Table 3. There are a few possible explanations for this. First, the filter may generally have low activations and therefore have little impact in the final feature computation. Second, the extreme points of the filter may not be characteristic of its overall behavior, since the bulk of n-grams will lead to more moderate activations. Finally, such a filter may represent the superposition of a few topics that we are unlikely to ever need to disambiguate between; in a particular context, this filter will then play a clear role, but one which is hard to determine from the overall shape of the parameters.

4 Conclusion

In this work, we investigated using convolutional networks to capture semantic similarity between source documents and potential entity link targets. Using multiple granularities of convolutions to evaluate the compatibility of a mention in context and several potential link targets gives strong performance on its own; moreover, such features also improve a pre-existing entity linking system based on sparse indicator features, showing that these sources of information are complementary.

Acknowledgments

This work was partially supported by NSF Grant CNS-1237265 and a Google Faculty Research Award. Thanks to the anonymous reviewers for their helpful comments.

References

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

Xiao Cheng and Dan Roth. 2013. Relational Inference for Wikification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Silviu Cucerzan. 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2015. Question Answering over Freebase with Multi-Column Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Greg Durrett and Dan Klein. 2014. A Joint Model for Entity Analysis: Coreference, Typing, and Linking. In *Transactions of the Association for Computational Linguistics (TACL)*.

Angela Fahrni and Michael Strube. 2014. A latent variable model for discourse-aware concept and entity disambiguation. In Gosse Bouma and Yannick Parmentier 0001, editors, *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 491–500. The Association for Computer Linguistics.

Zhengyan He, Shujie Liu, Yang Song, Mu Li, Ming Zhou, and Houfeng Wang. 2013. Efficient collective entity linking with stacking. In *EMNLP*, pages 426–435. ACL.

Tom Heath and Christian Bizer. 2011. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 1st edition.

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 (EMNLP)*.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep Unordered Composition Rivals Syntactic Methods for Text Classification. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Heng Ji and Ralph Grishman. 2011. Knowledge Base Population: Successful Approaches and Challenges. In *Proceedings of the Association for Computational Linguistics (ACL)*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

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 (NIPS) 26*, pages 3111–3119.

David Milne and Ian H. Witten. 2008. Learning to Link with Wikipedia. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)*.

Dat Ba Nguyen, Johannes Hoffart, Martin Theobald, and Gerhard Weikum. 2014. AIDA-light: High-Throughput Named-Entity Disambiguation. In *Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd International World Wide Web Conference (WWW)*.

NIST. 2005. The ACE 2005 Evaluation Plan. In *NIST*.

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 (ACL)*, pages 1375–1384.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning Semantic Representations Using Convolutional Neural Networks for Web Search. In *Proceedings of the 23rd International Conference on World Wide Web (WWW)*.

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.

Matthew D. Zeiler. 2012. AdaDelta: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701.