Learning Word Representations with Cross-Sentence Dependency for End-to-End Co-reference Resolution

Hongyin Luo
CSAIL, MIT
hyluo@mit.edu

James Glass
CSAIL, MIT
glass@mit.edu

Abstract

In this work, we present a word embedding model that learns cross-sentence dependency for improving end-to-end co-reference resolution (E2E-CR). While the traditional E2E-CR model generates word representations by running long short-term memory (LSTM) recurrent neural networks on each sentence of an input article or conversation separately, we propose linear sentence linking and attentional sentence linking models to learn cross-sentence dependency. Both sentence linking strategies enable the LSTMs to make use of valuable information from context sentences while calculating the representation of the current input word. With this approach, the LSTMs learn word embeddings considering knowledge not only from the current sentence but also from the entire input document. Experiments show that learning cross-sentence dependency enriches information contained by the word representations, and improves the performance of the co-reference resolution model compared with our baseline.

1 Introduction

Co-reference resolution requires models to cluster mentions that refer to the same physical entities. The models based on neural networks typically require different levels of semantic representations of input sentences. The models usually need to calculate the representations of word spans, or mentions, given pre-trained character and word-level embeddings (Turian et al., 2010; Pennington et al., 2014) before predicting antecedents. The mention-level embeddings are used to make co-reference decisions, typically by scoring mention pairs and making links (Lee et al., 2017; Clark and Manning, 2016a; Wiseman et al., 2016). Long short-term memories (LSTMs) are often used to encode the syntactic and semantic information of input sentences.

Articles and conversations include more than one sentences. Considering the accuracy and efficiency of co-reference resolution models, the encoder LSTM usually processes input sentences separately as a batch (Lee et al., 2017). The disadvantage of this method is that the models do not consider the dependency among words from different sentences, which plays a significant role in word representation learning and co-reference predicting. For example, pronouns are often linked to entities mentioned in other sentences, while their initial word vectors lack dependency information. As a result, a word representation model cannot learn an informative embedding of a pronoun without considering cross-sentence dependency in this case.

It is also problematic if we encode the input document considering cross-sentence dependency and treat the entire document as one sentence. An input article or conversation can be too long for a single LSTM cell to memorize. If the LSTM updates itself for too many steps, gradients will vanish or explode (Pascanu et al., 2013), and the co-reference resolution model will be very difficult to optimize. Regarding the entire input corpus as one sequence instead of a batch also significantly increases the time complexity of the model.

To solve the problem that traditional LSTM encoders, which treat the input sentences as a batch, lack an ability to capture cross-sentence dependency, and to avoid the time complexity and difficulties of training the model concatenating all input sentences, we propose a cross-sentence encoder for end-to-end co-reference (E2E-CR). Borrowing the idea of an external memory module from Sukhbaatar et al. (2015), an external memory block containing syntactic and semantic information from context sentences is added to the standard LSTM model. With this context memory block, the proposed model is able to encode

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4829–4833
Brussels, Belgium, October 31 - November 4, 2018.

input sentences as a batch, and also calculate the representations of input words by taking both target sentences and context sentences into consideration. Experiments showed that this approach improved the performance of co-reference resolution models.

2 Related Work

2.1 Co-reference Resolution

A popular method of co-reference resolution is mention ranking (Durrett and Klein, 2013). Reading each mention, the model calculates co-reference scores for all antecedent mentions, and picks the mention with the highest positive score to be its co-reference. Many recent works are based on this approach. Durrett and Klein (2013) designed a set of feature templates to improve the mention-ranking model. Peng et al. (2015) proposed a mention-ranking model by jointly learning mention heads and co-references. Clark and Manning (2016a) proposed a reinforcement learning framework for the mention ranking approach. Based on similar ideas but without using parsing features, the authors of Lee et al. (2017) proposed the current state-of-the-art model which uses neural networks to embed mentions and calculate mention and antecedent scores. Lee et al. (2018) applied ELMo embeddings (Peters et al., 2018) to improve within-sentence dependency modeling and word representation learning. Wiseman et al. (2016) and Clark and Manning (2016b) proposed models using global entity-level features.

2.2 Language Representation Learning

Distributed word embeddings has been used as the basic unit of language representation for over a decade (Bengio et al., 2003). Pre-trained word embeddings, for example GloVe (Pennington et al., 2014) and Skip-Gram (Mikolov et al., 2013) are widely used as the input of natural language processing models.

Long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) are widely used for sentence modeling. A single-layer LSTM network was applied in the previous state-of-the-art co-reference model (Lee et al., 2017) to generate word and mention representations. To capture dependency of longer distances, Campos et al. (2017) proposed a recurrent model that outputs hidden states by skipping input tokens.

Recently, memory networks (Sukhbaatar et al., 2015) have been applied in language modeling (Cheng et al., 2016; Tran et al., 2016). Applying an attention mechanism on memory cells, memory networks allow the model to focus on significant words or segments for classification and generation tasks. Previous works have shown that applying memory blocks in LSTMs also improves long-distance dependency extraction (Yogatama et al., 2018).

3 Learning Cross-Sentence dependency

To improve the word representation learning model for better co-reference resolution performance, we propose two word representation models that learn cross-sentence dependency.

3.1 Linear Sentence Linking

Instead of treating the entire input document as separate sentences and encode the sentences as a batch with an LSTM, the most direct way to consider cross-sentence dependency is to initialize LSTM states with the encodings of adjacent sentences. We name this method linear sentence linking (LSL).

In LSL, we encode input sentences with a 2-layer bidirectional LSTM. Give input sentences $[s_1, s_2 \dots s_n]$, the outputs of the first layer are $[[\overrightarrow{s}_1; \overleftarrow{s}_1], [\overrightarrow{s}_2; \overleftarrow{s}_2], \dots [\overrightarrow{s}_n; \overleftarrow{s}_n]]$. In the second LSTM layer, the initial state of the forward LSTM of s_i is initialized as

$$\overrightarrow{S}_i = [\overrightarrow{c}^2_0; [\overrightarrow{s}_{i-1}; \overleftarrow{s}_{i-1}]]$$

while the backward state is initialized as

$$\overleftarrow{S}_i = [\overleftarrow{c}^2_0; [\overrightarrow{s}_{i-1}; \overleftarrow{s}_{i-1}]]$$

where c^i_0 stands for the initial cell of the i-th layer, and x stands for the final output of the LSTMs in first layer. We then concatenate the outputs of the forward and backward LSTMs in the second layer as the word representations for co-reference prediction.

3.2 Attentional Sentence Linking

It is difficult for LSTMs to embed enough information about a long sentence into a low-dimensional distributed vector. To collect richer knowledge from neighbor sentences, we propose a long short-term recurrent memory module and an attention mechanism to improve sentence linking.

To describe the architecture of the proposed model, we focus on adjacent input sentences s_{i-1}

and s_i. We present the input embeddings of the j-th word in the i-th sentence with $x_{i,j}$.

3.2.1 Long Short-Term Memory RNNs

To solve the traditional recurrent neural networks, Hochreiter and Schmidhuber (1997) proposed the LSTM architecture. The detail of recurrent state updating in LSTMs $h_t = f_{lstm}(x_t, h_{t-1}, c_{t-1})$ is shown in following equations.

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
c_t &= f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
h_t &= o_t \odot tanh(c_t)
\end{aligned}
$$

where x_t is the input embedding and h_t is the output representation of the t-th word.

3.2.2 LSTMs with Cross-Sentence Attention

We design an LSTM module with cross-sentence attention for capturing cross-sentence dependency. We name this method attentional sentence linking (ASL). Considering input word $x_{i,t}$ in the i-th sentence and all words from the previous sentence $X_{i-1} = [x_{i-1,1}, x_{i-1,2}, \ldots, x_{i-1,m}]$, we regard the matrix X_{i-1} as an external memory module and calculate an attention on its cells, where each cell contains a word embedding.

$$\alpha_j = \frac{e^{c_j}}{\sum_k e^{c_k}} \tag{1}$$

$$c_k = f_c([x_{i,t}; h_{t-1}; x_{i-1,k}]^T) \tag{2}$$

With the attention distribution $\boldsymbol{\alpha}$, we can get a vector summarizing related information from s_{i-1},

$$v_{i-1} = \sum_j \alpha_j \cdot x_{i-1,j} \tag{3}$$

The model decides if it needs to pay more attention on the current input or cross-sentence information with a context gate.

$$g_t = \sigma(f_g([x_{i,t}; h_{t-1}; v_{i-1}]^T)) \tag{4}$$

$$\hat{x}_{i,t} = g_t \cdot x_{i,t} + (1 - g_t) \cdot v_{i-1} \tag{5}$$

$\sigma(\cdot)$ stands for the Sigmoid function. The word representation of the target word is calculated as

$$h_{i,t} = f_{lstm}(\hat{x}_{i,t}, h_{i,t-1}, c_{i,t-1}) \tag{6}$$

where f_{lstm} stands for standard LSTM update described in section 3.2.1.

3.3 Co-reference Prediction

In this work, we apply the mention-ranking end-to-end co-reference resolution (E2E-CR) model proposed by Lee et al. (2017) for co-reference prediction. The word representations applied in E2E-CR model is formed by concatenating pre-trained word embeddings and the outputs of LSTMs. In our work, we represent words by concatenating pre-trained word embeddings and the outputs of LSL- and ASL-LSTMs.

4 Experiments

We train and evaluate our model on the English corpus of the CoNLL-2012 shared task (Pradhan et al., 2012). We implement our model based on the published implementation of the baseline E2E-CR model (Lee et al., 2017) [1]. Our implementation is also available online for reproducing the results reported in this paper [2]. In this section, we first describe our hyperparameter setup, and then show the experimental results of previous work and our proposed models.

4.1 Model and Hyperparameter Setup

In practice, the LSTM modules applied in our model have 200 output units. In ASL, we calculate cross-sentence dependency using a multi-layer perceptron with one hidden layer consisting of 150 hidden units. The initial learning rate is set as 0.001 and decays 0.001% every 100 steps. The model is optimized with the Adam algorithm (Kingma and Ba, 2014). We randomly select up to 40 continuous sentences for training if the input is too long. In co-reference prediction, we select 250 candidate antecedents as our baseline model.

4.2 Experiment Results and Discussion

We evaluate our model on the test set of the CoNLL-2012 shared task. The performance of previous work and our model are shown in Table 1. We mainly focus on the average F1 score of MUC, B^3, and CEAF metrics. Comparing with the baseline model that achieved 67.2% F1 score, the ASL model improved the performance by 0.6% and achieved 67.8% average F1. Experiments

[1] https://github.com/kentonl/e2e-coref

[2] https://github.com/luohongyin/coatt-coref

Models	MUC			B^3			Ceafe			Avg.
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	F1
Wiseman et al. (2016)	77.5	69.8	73.4	66.8	57.0	61.5	62.1	53.9	57.7	64.2
Clark and Manning (2016b)	78.9	69.8	74.0	70.1	57.0	62.9	62.5	55.8	59.0	65.3
Clark and Manning (2016a)	79.2	70.4	74.6	69.9	58.0	63.4	63.5	55.5	59.2	65.7
Lee et al. (2017)	78.4	73.4	75.8	68.6	61.8	65.0	62.7	59.0	60.8	67.2
E2E-CR + LSL	81.0	71.5	76.0	72.6	59.4	65.3	65.0	57.5	61.0	67.4
E2E-CR + ASL	79.2	73.7	**76.4**	69.4	62.1	**65.6**	64.0	58.9	**61.4**	**67.8**

Table 1: Experimental results of previous models and cross-sentence dependency learning models on the CoNLL-2012 shared task.

- I remember receiving **an SMS like this one** last year before it snowed since snowfall would affect road conditions in Beijing to a large extent. - Uh-huh . However, **it** did not give people such a special feeling as it did this time.
- **Reporters** are tired of the usual stand ups. - **They** want to be riding on a train or walking in the rain or something to get attention .
- Planned terrorist bombing that ripped a 20 x 40 - foot hole in the **Navy destroyer USS Cole** in the Yemeni port of Aden. - **The ship** was there for refueling.
- Yemeni authorities claimed they have detained **over 70 people** for questioning. - **These** include some Afghan - Arab volunteers.

Table 2: Examples predictions of the ASL model and the baseline model.

show that the models that consider cross-sentence dependency significantly outperform the baseline model, which encodes each sentence from the input document separately.

Experiments also indicated that the ASL model has better performance than the LSL model, since it summarizes extracts context information with an attention mechanism instead of simply viewing sentence-level embeddings. This gives the model a better ability to model cross-sentence dependency.

Examples for comparing the performance of the ASL model and the baseline are shown in Table 2. Each example contains two continuous sentences with co-references distritubed in different sentences. Underlined spans in bold are target mentions and annotated co-references. Spans in green are ASL predictions, and spans in red are baseline predictions. A prediction on "-" means that no mention is predicted as a co-reference.

Table 2 shows that the baseline model, which does not consider cross-sentence dependency, has difficulty in learning the semantics of pronouns whose co-references are not in the same sentence. The pretrained embeddings of pronouns are not informative enough. In the first example, "it" is not semantically similar with "SMS" in GloVe without any context, and in this case, "it" and "SMS" are in different sentences. As a result, if reading this two sentences separately, it is hard for the encoder to represent "it" with the semantics of "SMS". This difficulty makes the co-reference resolution model either prediction a wrong antecedent mention, or cannot find any co-reference.

However, with ASL, the model learns the semantics of pronouns with an attention to words in other sentences. With the proposed context gate, ASL takes knowledge from context sentences if local inputs are not informative enough. Based on word represents enhanced with cross-sentence dependency, the co-reference scoring model can make better predictions.

5 Conclusion and Future Work

We proposed linear and attentional sentence linking models for learning word representations that captures cross-sentence dependency. Experiments showed that the embeddings learned by proposed models successfully improved the performance of the state-of-the-art co-reference resolution model, indicating that cross-sentence dependency plays an important role in semantic learning in articles and conversations consists of multiple sentences. It worth exploring if our model can improve the performance of other natural language processing

applications whose inputs contain multiple sentences, for example, reading comprehension, dialog generation, and sentiment analysis.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Víctor Campos, Brendan Jou, Xavier Giró-i Nieto, Jordi Torres, and Shih-Fu Chang. 2017. Skip rnn: Learning to skip state updates in recurrent neural networks. *arXiv preprint arXiv:1708.06834*.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

Kevin Clark and Christopher D Manning. 2016a. Deep reinforcement learning for mention-ranking coreference models. *arXiv preprint arXiv:1609.08667*.

Kevin Clark and Christopher D Manning. 2016b. Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*.

Greg Durrett and Dan Klein. 2013. Easy victories and uphill battles in coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1971–1982.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. *arXiv preprint arXiv:1804.05392*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318.

Haoruo Peng, Kai-Wei Chang, and Dan Roth. 2015. A joint framework for coreference resolution and mention head detection. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 12–21.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 1–40. Association for Computational Linguistics.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Ke Tran, Arianna Bisazza, and Christof Monz. 2016. Recurrent memory networks for language modeling. *arXiv preprint arXiv:1601.01272*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

Sam Wiseman, Alexander M Rush, and Stuart M Shieber. 2016. Learning global features for coreference resolution. *arXiv preprint arXiv:1604.03035*.

Dani Yogatama, Yishu Miao, Gabor Melis, Wang Ling, Adhiguna Kuncoro, Chris Dyer, and Phil Blunsom. 2018. Memory architectures in recurrent neural network language models. In *International Conference on Learning Representations. https://openreview. net/forum*.