Modeling Mention, Context and Entity with Neural Networks for Entity Disambiguation

Yaming Sun†, Lei Lin†* , Duyu Tang†, Nan Yang‡, Zhenzhou Ji†, Xiaolong Wang†
†Harbin Institute of Technology, Harbin, China
‡Microsoft Research, Beijing, China
sunyamingdacong@163.com, {linl, wangxl}@insun.hit.edu.cn,
dytang.hit@hotmail.com, nyang.ustc@gmail.com, jzz@pact518.hit.edu.cn

Abstract

Given a query consisting of a mention (name string) and a background document, entity disambiguation calls for linking the mention to an entity from reference knowledge base like Wikipedia. Existing studies typically use hand-crafted features to represent mention, context and entity, which is labor-intensive and weak to discover explanatory factors of data. In this paper, we address this problem by presenting a new neural network approach. The model takes consideration of the semantic representations of mention, context and entity, encodes them in continuous vector space and effectively leverages them for entity disambiguation. Specifically, we model variable-sized contexts with convolutional neural network, and embed the positions of context words to factor in the distance between context word and mention. Furthermore, we employ neural tensor network to model the semantic interactions between context and mention. We conduct experiments for entity disambiguation on two benchmark datasets from TAC-KBP 2009 and 2010. Experimental results show that our method yields state-of-the-art performances on both datasets.

1 Introduction

Entity disambiguation is a fundamental task in the field of natural language processing [Zheng *et al.*, 2010; Ratinov *et al.*, 2011; Han *et al.*, 2011; Kataria *et al.*, 2011; Sen, 2012], and a crucial step for knowledge base population [Ji and Grishman, 2011]. Given a document and a mention which is usually a text span occurred in the document, entity disambiguation targets at mapping the mention to an entity from reference knowledge base like Wikipedia[1]. For example, given a text span "*<u>President Obama</u>*" in the document "*After campaigning on the promise of health care reform, <u>President Obama</u> gave a speech in March 2010 in Pennsylvania.*" as input, the purpose of entity disambiguation is to link the mention "*<u>President Obama</u>*" in this context to an entity in the Wikipedia. The ground truth in this example is *Barack Obama*[2].

Previous studies in the literature typically regard entity disambiguation as a ranking problem, and utilize similarity measure to compare the context of a mention with the text associated with a candidate entity (e.g. the text in the corresponding page in reference KB). Since the performance of entity disambiguation is heavily dependent on the choice of feature representations of mention and entity, a variety of algorithms are developed to effectively represent them for obtaining a better disambiguation performance. Representative mention features include document surface feature like lexical and part-of-speech tags of context words, entropy based representations [Mendes *et al.*, 2011], structured text representations like dependency paths and topic feature representations [Taylor Cassidy *et al.*, 2011]. Typical entity features include name tagging, KB infoboxes, synonyms and semantic categories [Chen and Ji, 2011; Cassidy *et al.*, 2012].

Feature engineering is important but labor intensive and insufficient to disentangle the underlying explanatory factors of data. In natural language processing community, an effective feature learning approach is to compose the representation of a text (e.g. phrase, sentence or document) based on the representation of words using neural network [Mitchell and Lapata, 2010; Socher *et al.*, 2013b; Kalchbrenner *et al.*, 2014]. For the task of entity disambiguation, [He *et al.*, 2013a] uses deep neural network to learn the representations of an input document containing mention as well as a KB document referring to a candidate entity. They feed a document as input and employ Stacked Denoising Auto-encoders [Vincent *et al.*, 2008] to produce the semantic representation. However, we argue that the approach is not effective enough as it ignores the mention which is to be linked. Let us again take the example of "*<u>President Obama</u>*" as given before, the document representations towards two different mentions "*<u>President Obama</u>*" and "*<u>Pennsylvania</u>*" are identical according to He et al. [2013a]'s approach. This is problematic for entity disambiguation because the target to be linked is the mention rather than a document.

In this paper, we introduce a new neural network approach that simultaneously takes consideration of mention, context and entity for entity disambiguation. The neural architecture

*Corresponding author.

[1] https://www.wikipedia.org/

[2] http://en.wikipedia.org/wiki/Barack_Obama

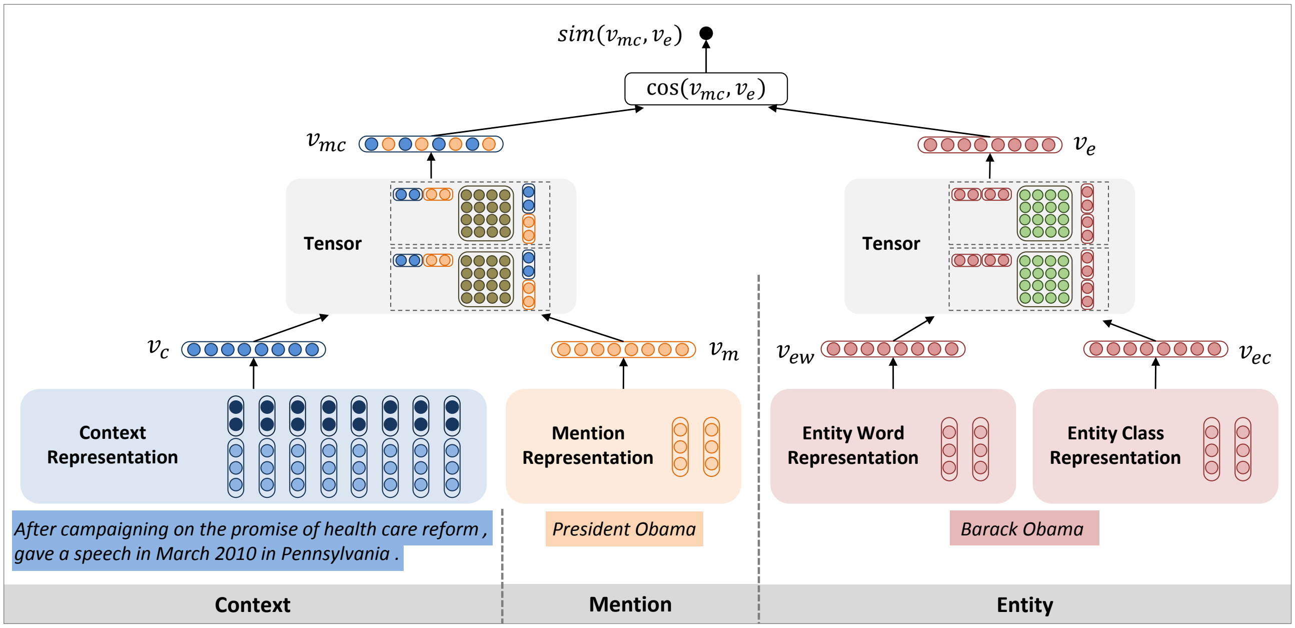

Figure 1: The proposed neural network method for entity disambiguation. In this example, the mention "*President Obama*" comes from an original document "*After campaigning on the promise of health care reform, President Obama gave a speech in March 2010 in Pennsylvania.*". The candidate entity in this example is "*Barack Obama*".

is illustrated in Figure 1. We cast entity disambiguation as a ranking task by comparing the similarities between an input (mention, context) pair and candidate entities. Specifically, we embed mention, context and entity in continuous vector space to capture their semantic representations. The variable-sized context are modeled with convolutional neural networks. Since a closer context word might be more informative than a farther one for disambiguating a mention, we also embed the distance between a context word and the mention in continuous vector space. Furthermore, we exploit low-rank neural tensor network to model the semantic composition between context and mention. We design a ranking type hinge loss function, and collect 1M anchor text from Wikipedia as training data for parameter learning without using any manual annotation.

We apply the neural network to entity disambiguation on two benchmark datasets from Text Analysis Conference-Knowledge Base Population (TAC-KBP) in 2009 and 2010. We compare to the top-performed systems in KBP evaluation along with state-of-the-art methods [Han and Sun, 2011; He *et al.*, 2013a]. Experimental results show that our method yields state-of-the-art performances on both datasets.

The main contributions of this work are as follows.

- We present a new neural network approach which effectively captures the semantics of mention, context and entity simultaneously for entity disambiguation.
- We factor in context words as well as their position information with convolutional neural network, and leverage low-rank neural tensor network to model semantic composition between mention and context.
- We report empirical results on two benchmark datasets from KBP 2009 and KBP 2010. We show that the proposed method yields state-of-the-art performances on both datasets.

2 Methodology

We describe the proposed neural network for entity disambiguation in this section. We first give an overview of the approach, followed by the methods for modeling context, mention and entity, respectively. Afterwards, we describe the use of our method for entity disambiguation and the strategy for model training.

2.1 An Overview of the Approach

A bird-view of the proposed neural network for entity disambiguation is given in Figure 1. As is shown, the input includes three parts, namely a mention, the context of mention and a candidate entity from reference knowledge base. The output of our neural network stands for the similarity between a candidate entity and a pair of mention and context. Specifically, we learn the continuous representations of context words with convolution neural networks, and produce its semantic composition with mention using a neural tensor network (detailed in Section 2.2). Meanwhile, we learn the continuous representation of a candidate entity. We then apply the learned representations of context, mention and entity for calculating the similarity between a candidate entity and a given (mention, context) pair, which will be conveniently applied to entity disambiguation (detailed in Section 2.3).

2.2 Modeling Context, Mention and Entity

We describe our method for learning continuous representations of mention, context and entity as well as their semantic composition in this section.

Context Modeling The representation of a **context** is composed of the representations of words it contains according to the principal of compositionality [Frege, 1892]. In addition, we argue that representation of a context is also influenced by the distance between a context word and the mention. This is based on the consideration that a closer context word might be more informative than a farther one for disambiguating the mention.

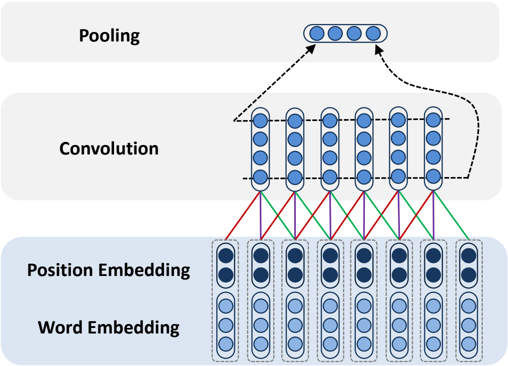

Figure 2: Context modeling with convolutional neural network. The input of context convolution includes word embedding and position embedding. The weights with same color (e.g. red, purple or green) are shared cross different filters.

To this end, the vector of each context word is made up of two parts: a word embedding $e_w = L_w i_w$ and a position embedding $e_p = L_p i_p$, where $L_w \in \mathbb{R}^{d_w \times |V_w|}$ and $L_p \in \mathbb{R}^{d_p \times |V_p|}$ are the lookup tables of words and positions, respectively; d_w and d_p are the dimensions of word vector and position vector, respectively; i_w and i_p are binary vectors which is zero in all positions except at the w-th and p-th index. The position of a context word is its distance to the mention in a given piece of text. Since the number of context words is of variable length, we use convolutional neural network, which is a state-of-the-art semantic composition approach [Kalchbrenner *et al.*, 2014; Kim, 2014], to produce a fixed-length vector for a context. The convolution layer is a list of linear layers whose parameters are shared in different filter windows, as given in Figure 2. Formally, suppose the filter window size of each convolution layer is K, the output vector of a convolution layer is calculated as follow.

$$O_{conv} = W_{conv} in_{conv} + b_{conv} \tag{1}$$

where $W_{conv} \in \mathbb{R}^{hl \times K \cdot (d_w + d_p)}$, hl is the output length of convolution layer, $in_{conv} \in \mathbb{R}^{K \cdot (d_w + d_p)}$ is the concatenation of representations of K words in a filter window, $b_{conv} \in \mathbb{R}^{hl}$. The subsequent pooling layer captures the global information of the context, and outputs a fixed-length vector for context with variable length. In this paper, we use average pooling layer but the method can naturally incorporate other pooling functions like max pooling or k-max pooling [Kalchbrenner *et al.*, 2014].

Mention Modeling Since a **mention** is typically one to three words, we simply represent them as the average of embeddings of words it contains [Socher *et al.*, 2013b]. Recall that we cast entity disambiguation as a ranking task, which requires the similarity between a candidate entity and a pair of mention and its context. Under this perspective, we need to calculate the representation of an input document consisting of mention and context based on the representation of mention, the representation of context and their semantic compositionality [Frege, 1892]. We employ neural tensor network [Socher *et al.*, 2013c] as the composition function because it is a state-of-the-art performer in the field of vector based semantic composition [Mitchell and Lapata, 2010]. Standard neural tensor network with rank 3 is a list of bilinear layer, each of which conducts bilinear operation on two input vectors and outputs a scalar. A bilinear layer is typically parameterized by a matrix $M \in \mathbb{R}^{N \times N}$, where N is the dimension of each input vector.

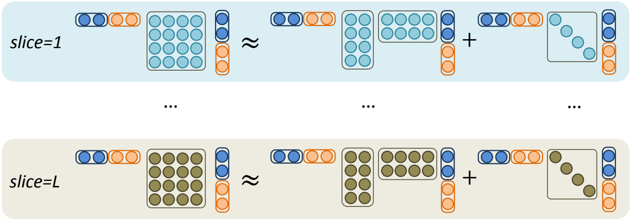

Figure 3: An illustration of low-rank neural tensor network for modeling the semantic composition between mention and context.

In this paper, we follow [Socher *et al.*, 2013c] and represent each input as the concatenation of mention vector and context vector. To decrease the number of parameters in standard neural tensor network, we make low rank approximation that represent each matrix by two low-rank matrices plus diagonal, as illustrated in Figure 3. Formally, the parameter of the i-th slice is $M_i^{appr} = M_{i1} \times M_{i2} + diag(m_i)$, where $M_{i1} \in \mathbb{R}^{N \times r}$, $M_{i2} \in \mathbb{R}^{r \times N}$, $m_i \in \mathbb{R}^N$. The output of neural tensor layer is formalized as follows.

$$v_{mc} = [v_m; v_c]^T [M_i^{appr}]^{[1:L]} [v_m; v_c] \tag{2}$$

where $[v_m; v_c] \in \mathbb{R}^N$ is the concatenation of mention vector v_m and context vector v_c; $[M_i^{appr}]^{[1:L]}$ is the low-rank tensor that defines multiple low-rank bilinear layers, L is the slice number of neural tensor network which is also equal to the output length of v_{mc}.

Entity Modeling We model the semantics of an **entity** in knowledge base from two aspects: entity surface words and entity class. For example, the surface words of entity *Barack Obama* are *barack* and *obama*. Entity class of an entity is a word or a phrase provided in infobox of reference knowledge base, which indicates the category information of the entity. For example, the class of *Barack Obama* is *president of the united states*. We use the embeddings of class words to represent the semantics of entity class. This is based on the consideration that entity classes are semantically related with each other in a continuous vector space rather than independent in a discrete vector space. Since surface words and class words are both short and variable-sized, we average them separately to produce entity word vector and entity class vector [Socher *et al.*, 2013b]. In order to encode the interactions between these two vectors, we use low-rank neural tensor network as detailed above to produce the final entity representation.

2.3 Entity Disambiguation

We apply the learned representation of candidate entity as well as composed context and mention for entity disambiguation in a ranking based framework. Given the representation of a candidate entity v_e and the representation of a mention context pair v_{mc}, we use the cosine similarity between these two vectors to represent their semantic relatedness, namely

$$sim(e, mc) = cosine(v_e, v_{mc}) \tag{3}$$

In the prediction process, we calculate the similarity between a context mention pair with each candidate entity, and select the closest one as the final result. For effectively training the model, we devise a ranking type loss function as given in Equation 4. The basic idea is that the output score of a correct entity should be larger than the score of a randomly selected candidate entity by a margin of 1.

$$loss = \sum_{(m,c) \in T} max(0, 1 - sim(e, mc) + sim(e', mc)) \tag{4}$$

where e is the gold standard entity, and e' is a *corrupted* entity which is randomly selected from the entire entity vocabulary of reference KB.

2.4 Model Training

It is commonly accepted that large training data is crucial for obtaining a better performance if one uses neural network. In order to obtain massive training data without any manual annotation, we collect queries (including mention and context) and corresponding target entities using anchor text from Wikipedia. For example, a document containing anchor text *President Obama* linked to entity *Barack Obama* will be regarded as a gold disambiguation instance, whose mention is *President Obama* and ground truth is *Barack Obama*. We train the word embeddings on our training set with SkipGram [Mikolov *et al.*, 2013] which is integrated into the widely used *word2vec* toolkit[3]. We set the dimension of word vector as 50, window size as 5. We convert all words to lowercase, normalize digit number with a special symbol. The vocabulary size of the word embedding is 1.63M. We train the neural network by taking derivative of the loss through back-propagation with respect to the whole set of parameters. The parameters of linear layer and low-rank neural tensor network are initialized from a uniform distribution $U(-rnd, rnd)$, where $rnd = 0.01$. We empirically set the learning rate as 0.01, the window size of convolution neural network as 2, the output length of neural tensor layer as 30.

3 Experiment

In this section, we describe experimental settings and empirical results on the task of entity disambiguation.

3.1 Experimental Setting

We conduct experiments on two benchmark datasets for entity disambiguation from Text Analysis Conference-Knowledge Base Population (TAC-KBP[4]) in 2009 [McNamee and Dang, 2009] and 2010 [Ji and Grishman, 2011]. TAC-KBP officially provides a collection of queries, each of which contains a mention and its background document. Participants are asked to link the mention in a query to an entity from the officially provided reference knowledge base. Each entity in reference knowledge base is accompanied with its infobox and description text.

We follow the experiment protocols as described in [He *et al.*, 2013a], and use only non-NIL queries (target entities are in KB) from KBP 2009 and KBP 2010 for testing. The numbers of non-NIL queries from KBP 2009 and KBP 2010 are 1,675 and 1,020, respectively. The reference knowledge base contains 818,741 entities, and 2,344 entity classes in total. For model training, we collect anchor texts which contain the entities covered by the reference knowledge base. We finally get 1M instances as training data to train our neural network. We use micro-averaged accuracy as the evaluation metric, which measures whether a top-ranked entity candidate is the ground truth.

We use several heuristic rules to obtain candidate entities for a given query as detailed below. We save the entities which are (a) exact matches of a given mention, (b) the anchor entities of a mention in Wikipedia, (c) the redirected entities of a mention if they are contained in redirect pages in Wikipedia, (d) the entities whose minimum edit distance with the mention is smaller than two. To reduce the number of candidates, we use the context of mention to filter out some candidates with simple string matching rules. The final recalls of our candidate entities on KBP 2009 and KBP 2010 are 90.08% and 91.17%, respectively.

3.2 Experimental Results

We report empirical results of our method as well as baseline methods for entity disambiguation on two benchmark datasets from TAC-KBP 2009 and 2010.

The methods presented in this work can be divided into four models with incremental integration of semantics. We describe the details of these four models as follows.

- **Model 1.** We only use the semantics of mention and candidate entity surface words, without using contexts of mention or class information of entity. We simply average the word vectors of a mention and an entity as their representations. This is analogous to the method used in [Blanco *et al.*, 2015].
- **Model 2.** We use the semantics of mention, context words, and candidate entity in this model. We extend Model 1 by using convolutional neural network to capture the semantics of context words. In Model 2, we simply concatenate the mention vector and context vector without capturing their interactions. For the entity component, we integrate the entity class information and concatenate its vector with entity surface word vector as the entity representation.

[3] https://code.google.com/p/word2vec/

[4] http://www.nist.gov/tac/

- **Model 3.** We extend Model 2 by taking position information of context words into consideration. We embed each position into a continuous vector space, and concatenate it with context word vector for subsequent use in convoluational neural network. We use concatenation as the semantic composition function in both mention-context part and entity part.
- **Model 4.** We extend Model 3 by incorporating the interactions between (a) context vector and mention vector as well as (b) entity surface word vector and entity class vector. We use low-rank neural tensor network to model semantic composition in both components, which is detailed in Section 2.2.

We report empirical results of our models and baseline methods on TAC-KBP 2009 and 2010 test datasets. The official 1st, 2nd and 3rd ranked systems of KBP 2009 and KBP 2010 are marked as Rank 1, Rank 2 and Rank 3. We also compare with a generative entity disambiguation method [Han and Sun, 2011] and a state-of-the-art neural network approach [He *et al.*, 2013a]. Our methods are abbreviated as Model 1-4. Experimental results are illustrated in Table 1.

Method	KBP 2009	KBP 2010
Rank 1	77.25	80.59
Rank 2	76.54	75.20
Rank 3	75.88	73.73
[Han *et al.*, 2011]	79.00	–
[He *et al.*, 2013a]	–	80.97
Model 1	73.85	75.98
Model 2	80.47	81.56
Model 3	80.75	**83.92**
Model 4	**82.26**	81.07

Table 1: Experimental results on the test set of TAC-KBP 2009 and 2010 for entity disambiguation. Evaluation metric is micro-averaged accuracy (in KB). The best result is in **bold**.

We can find that our method (Model 3, Model 4) yields the best performance on both datasets compared with many strong baselines. The performance of Model 1 is relatively low because it only utilizes the surface word-level semantics of mention and entity, but ignores the crucial contextual information of a mention. Model 2 obtains significant performance boost over Model 1 by integrating semantic representations of context words. Besides, we surprisingly find that Model 2 outperforms the best baseline methods on both datasets. This result verifies the effectiveness of context information for the task of entity disambiguation. Comparing Model 3 with Model 2, we can find an improvement (0.28% and 2.36% in accuracy) is further achieved by incorporating the position information of context words. This is intuitive since a closer context word might be more informative than a farther one for disambiguating the mention. Comparing Model 4 with Model 3, we can see that neural tensor network is more powerful than vector concatenation for semantic composition on TAC-KBP 2009 dataset. The reason lies in that neural tensor network better captures the semantic interactions between mention and context.

3.3 Model Analysis

We investigate the influential factors of our method for entity disambiguation in this part.

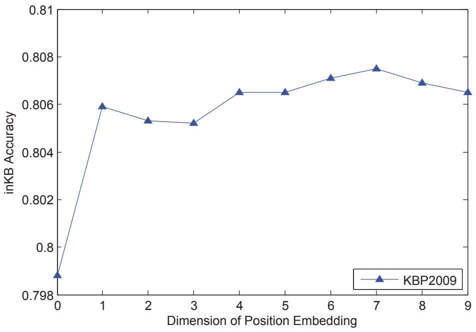

Figure 4: Experiments of our neural network (Model 3) for entity disambiguation with different dimensions of position embedding.

We first explore the effect of position embedding on KBP 2009. Specifically, we investigate how the dimension of position embedding affects the performance of our Model 3 for entity disambiguation. We vary the dimension of position embedding δ_p from 0 to 9, increased by 1. Results with different dimensions of position embedding are given in Figure 4. The model with $\delta_p = 0$ stands for Model 2 without using position information. We can find that position embedding is effective for entity disambiguation, because all models with $\delta_p > 0$ outperform the model with $\delta_p = 0$. Model 3 performs better when δ_p is 7.

We vary the rank size of low-rank neural tensor network in Model 4. The best performances on TAC-KBP 2009 and 2010 datasets are achieved at rank size equals to 1 and 2, respectively. For one iteration, the training time costs of Model 4 with different rank sizes are illustrated in Figure 5. We can find that the time cost is (almost) linearly increased along with rank size. This is because the parameter number of low-rank neural tensor network is linearly increased along with rank size. We run experiments on one computer with 64G memory 24 core Inter Xeon CPU.

4 Related Work

We briefly review existing studies on entity disambiguation and neural network approaches for natural language processing in this section.

4.1 Entity Disambiguation

Entity disambiguation is typically regarded as a ranking task, which calls for measuring the similarity between context of a mention and the text associated with a candidate entity (e.g. the text in the corresponding page in KB). Existing algorithms for entity disambiguation can be generally divided into local approaches and global (collective) approaches. The former [Zheng *et al.*, 2010; Mendes *et al.*, 2011; Ji and Grishman, 2011] uses local statistics of a mention m_i and an entity title t_i. The latter [Han *et al.*, 2011;

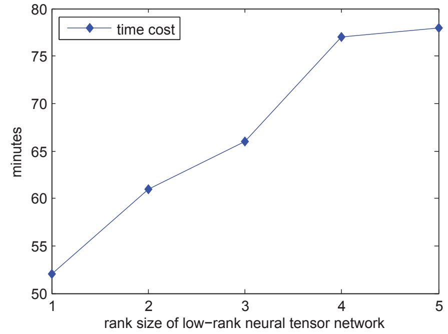

Figure 5: The training time costs of our neural network (Model 4) with different rank sizes.

Ratinov *et al.*, 2011; He *et al.*, 2013b] takes consideration of all mentions in a given document simultaneously. Both directions require the semantic relatedness between mention m_i and entity t_i. Representative mention features in the literature include document surface feature such as lexical and part-of-speech tags of context words, entropy based representations [Mendes *et al.*, 2011], structured text representations such as dependency paths and topic feature representation [Taylor Cassidy *et al.*, 2011]. Typical entity features include name tagging, KB infoboxes, synonyms and semantic categories [Chen and Ji, 2011; Cassidy *et al.*, 2012]. Since feature engineering is time-consuming and weak to discover underlying explanatory factors of data, it is desirable to learn features automatically from data. Under this perspective, [He *et al.*, 2013a] investigate Stacked Denoising Autoencoder to learn continuous representation of context text and entity document. Unlike dominating existing studies that use hand-crafted features, we learn discriminative features with neural network automatically from data. Our method differs from [He *et al.*, 2013a] in two aspects. On one hand, we use continuous representations of context positions to capture the distance between a context word and the mention. On the other hand, we explicitly model the semantic composition between context vector and mention vector with low-rank neural tensor network.

4.2 Neural Network for Natural Language Processing

We briefly introduce neural network approaches for natural language processing (NLP) in literature. Existing neural network approaches can be divided into two directions. One is learning continuous representation of words [Mikolov *et al.*, 2013]. Another direction focuses on semantic composition [Mitchell and Lapata, 2010] in order to obtain the representation of phrases, sentences and documents. The semantic representation of text can be effectively used as features for a variety of NLP tasks, including machine translation [Cho *et al.*, 2014], syntactic parsing [Socher *et al.*, 2013a], discourse parsing [Li *et al.*, 2014], relation classification [Zeng *et al.*, 2014], sentiment analysis [Socher *et al.*, 2013c; Tang *et al.*, 2014; Li, 2014], part-of-speech tagging and named entity recognition [Collobert *et al.*, 2011]. Our approach for modeling variable-sized context representation is relevant to the field of vector based semantic composition [Mitchell and Lapata, 2010]. Representative algorithms in this field are recursive neural network [Socher *et al.*, 2013c] and convolutional neural network [Kalchbrenner *et al.*, 2014; Kim, 2014]. These methods are on the basis of the principal of compositionality, which states that the representation of a text (e.g. a sentence or a document) is composed from the representations of the words it contains. In this work, we prefer convolutional neural network as it does not rely on a fixed parse tree and is a state-of-the-art performer in this field. We take consideration of the continuous representations of context positions, which has been exploited as useful signals for relation classification [Zeng *et al.*, 2014] and semantic role labeling [Collobert *et al.*, 2011]. We model the semantic composition of context and mention with neural tensor network, which has been explored as powerful composition function for sentiment analysis [Socher *et al.*, 2013c] and knowledge base completion [Socher *et al.*, 2013b]. Our strategy for approximating standard neural tensor network with a low-rank form is inspired by [Socher *et al.*, 2012], which represents each matrix with a low-rank approximation.

5 Conclusion

We present a new neural network approach in this work for entity disambiguation. The model leverages the semantics of mention, context and entity as well as their compositionality in a unified way. We represent contexts with convolutional neural network, and encode the positions of context words in continuous space for capturing the distance between context word and mention. We use low-rank neural tensor network to model semantic composition between context and mention as well as entity surface words and entity class. We apply the model to entity disambiguation on TAC-KBP 2009 and 2010 datasets. Empirical results show that the model outperforms previous studies on both datasets. We show that incorporating semantics of contexts significantly boosts the performance on entity disambiguation.

Acknowledgments

We gratefully acknowledge the helpful discussions with Shujie Liu and Yuhang Guo. We thank the anonymous reviewers for their insightful feedbacks. This work was partly supported by National Natural Science Foundation of China (No. 61300114, No. 61272383).

References

[Blanco *et al.*, 2015] Roi Blanco, Giuseppe Ottaviano, and Edgar Meij. Fast and space-efficient entity linking in queries. In *WSDM*, pages 2061–2069, 2015.

[Cassidy *et al.*, 2012] Taylor Cassidy, Heng Ji, Lev-Arie Ratinov, Arkaitz Zubiaga, and Hongzhao Huang. Analysis and enhancement of wikification for microblogs with context expansion. In *COLING*, pages 441–456, 2012.

[Chen and Ji, 2011] Zheng Chen and Heng Ji. Collaborative ranking: A case study on entity linking. In *EMNLP*, pages 771–781, 2011.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014.

[Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 2011.

[Frege, 1892] Gottlob Frege. On sense and reference. *Ludlow (1997)*, pages 563–584, 1892.

[Han and Sun, 2011] Xianpei Han and Le Sun. A generative entity-mention model for linking entities with knowledge base. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 945–954. Association for Computational Linguistics, 2011.

[Han *et al.*, 2011] Xianpei Han, Le Sun, and Jun Zhao. Collective entity linking in web text: a graph-based method. In *SIGIR*, pages 765–774. ACM, 2011.

[He *et al.*, 2013a] Zhengyan He, Shujie Liu, Mu Li, Ming Zhou, Longkai Zhang, and Houfeng Wang. Learning entity representation for entity disambiguation. In *ACL*, pages 30–34, 2013.

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

[Ji and Grishman, 2011] Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In *ACL*, pages 1148–1158, 2011.

[Kalchbrenner *et al.*, 2014] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *ACL*, pages 655–665, 2014.

[Kataria *et al.*, 2011] Saurabh S Kataria, Krishnan S Kumar, Rajeev R Rastogi, Prithviraj Sen, and Srinivasan H Sengamedu. Entity disambiguation with hierarchical topic models. In *SIGKDD*, pages 1037–1045. ACM, 2011.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751, 2014.

[Li *et al.*, 2014] Jiwei Li, Rumeng Li, and Eduard Hovy. Recursive deep models for discourse parsing. In *EMNLP*, pages 2061–2069, 2014.

[Li, 2014] Jiwei Li. Feature weight tuning for recursive neural networks. *Arxiv preprint*, 1412.3714, 2014.

[McNamee and Dang, 2009] Paul McNamee and Hoa Trang Dang. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*, pages 111–113, 2009.

[Mendes *et al.*, 2011] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[Mitchell and Lapata, 2010] Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429, 2010.

[Ratinov *et al.*, 2011] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *ACL*, pages 1375–1384, 2011.

[Sen, 2012] Prithviraj Sen. Collective context-aware topic models for entity disambiguation. In *WWW*, pages 729–738. ACM, 2012.

[Socher *et al.*, 2012] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *EMNLP*, pages 1201–1211, 2012.

[Socher *et al.*, 2013a] Richard Socher, John Bauer, Christopher D. Manning, and Ng Andrew Y. Parsing with compositional vector grammars. In *ACL*, pages 455–465, 2013.

[Socher *et al.*, 2013b] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*, pages 926–934, 2013.

[Socher *et al.*, 2013c] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642, 2013.

[Tang *et al.*, 2014] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL*, pages 1555–1565, 2014.

[Taylor Cassidy *et al.*, 2011] Zheng Chen Taylor Cassidy, Javier Artiles, Heng Ji, Hongbo Deng, Lev-Arie Ratinov, Jing Zheng, Jiawei Han, and Dan Roth. Cuny-uiuc-sri tac-kbp2011 entity linking system description. In *Proceedings Text Analysis Conference (TAC2011)*, 2011.

[Vincent *et al.*, 2008] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103. ACM, 2008.

[Zeng *et al.*, 2014] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via a convolutional deep neural network. In *COLING*, pages 2335–2344, 2014.

[Zheng *et al.*, 2010] Zhicheng Zheng, Fangtao Li, Minlie Huang, and Xiaoyan Zhu. Learning to link entities with knowledge base. In *NAACL*, pages 483–491, 2010.