Embedding Wikipedia Title Based on Its Wikipedia Text and Categories

Chi-Yen Chen, Wei-Yun Ma
Institue of Information Science
Academia Sinica
Taipei, Taiwan
{*ccy9332, ma*}*@iis.sinica.edu.tw*

***Abstract*—Distributed word representation is widely used in many NLP tasks and knowledge-based resources also provide valuable information. Comparing to conventional knowledge bases, Wikipedia provides semi-structural data other than structural data. We argue that a Wikipedia title's categories can help complement the title's meaning besides Wikipedia text, so the categories should be utilized to improve the title's embedding. We propose two directions of using categories, cooperating with conventional context-based approaches, to generate embeddings of Wikipedia titles. We conduct extensively large scale experiments on the generated title embeddings on Chinese Wikipedia. Experiments on word similarity task and analogical reasoning task show that our approaches significantly outperform conventional context-based approaches.**

***Keywords*-word embedding, Wikipedia, Wikipedia category, Knowledge base**

I. Introduction

Distributed word representations have been widely used for various NLP tasks. Many approaches have been proposed to learn word embedding from a large corpus [1], [2], [3], [4], [5], [6].

Knowledge embedding, which embeds knowledge graphs into a continuous vector space while preserving the original properties of the graph, has also attracted considerable research efforts [7], [8], [9], [10]. Instead of using mere graph node pair as a feature, GAKE [11] is designed to learn knowledge embedding by utilizing the graph's structural information.

Researchers have also explored to make full use of knowledge-based resources, such as Wikidata, Freebase [12] and WordNet [13] to improve context-based embeddings. Models that learn word embedding through both knowledge bases and text jointly were proposed as well [14], [8], [15], [16]. Among the various knowledge bases, for each name entity, Wikipedia provides not only structural data, i.e, knowledge graphs via infoboxes, but also the nonstructural data, i.e, Wikipedia text, and semi-structural data, i.e, the title's categories. Most Wikipedia categories are long noun phrases other than noun words, so they are sometimes able to provide more complete information than info-boxes. In our approaches, we define the graph nodes as Wikipedia categories with the form of long noun phrase. To the best of our knowledge, we are the first to utilize noun phrases or Wikipedia category information to improve word embedding.

In this paper, we argue that a Wikipedia title's categories can help define or complement the meaning of the title besides the Wikipedia text, so the categories should be utilized to improve the title's embedding. We propose two directions of using categories, cooperating with conventional context-based approaches, to generate embeddings of Wikipedia titles. One direction is to first obtain the embedding of each category, followed by adding them to context-based embedding which is trained from Wikipedia text. The other direction is to first decide which category can most represent the title, and then get the category's embedding in order to concatenate with the context-based embedding. We conduct extensively large scale experiments on the generated title embeddings on Chinese Wikipedia. Experiments on word similarity task and analogical reasoning task show that our approaches significantly outperform the conventional context-based approaches.

II. Data Background

Wikipedia provides data in three structural extents: nonstructural, i.e, content, semi-structural, i.e, categories, and structural data, i.e, info-boxes. Categories are usually long noun phrases and provide more information than info-boxes. The page *Albert Einstein*, for example, is in categories of *ETH Zurich alumni*, *ETH Zurich faculty* and 20 more. The categories provide information that Einstein was both alumni and faculty of ETH Zurich while in info-box, only Einstein was related to ETH Zurich is shown.

We download the Chinese version of Wikipedia dump file in September 2016, which compromised 1,243,319 articles at a time. There are 244,430,247 words in the Chinese Wikipedia corpus and each title has 2.16 categories in average.

III. Method

A. Context-based Embedding

Data preprocessing, such as word segmentation is necessary for Chinese data. We segment Wikipedia text via CKIP Chinese Word Segmentation System [17], [18]. We also collect Wikipedia titles as our lexicon to identify titles during word segmentation. On the other hand, to get general descriptions of categories, we do not apply the lexicon on segmenting Wikipedia categories. After preprocessing, we gain the Chinese Wikipedia corpus and apply the Skip-gram model [4] to obtain 300 dimensional context-based embeddings.

B. Categories Embedding

We propose approaches of acquiring embeddings through Wikipedia categories, which can partially represent the corresponding title. We name the embeddings as Wikipedia categories embedding, categories embedding for short.

Given a title t and its corresponding categories from C^1 to C^n, each category C^i has been word-segmented as K words from W_1^i to W_K^i. We use c^i and w_j^i to represent embedding of category C^i and word W_j^i.

1) Average of Category Words: We acquire categories embedding $e_{category}$ by averaging every category of a single title. Considering the completeness of category information, we obtain a category embedding by averaging all words in the category. Process of computing $e_{category}$ is shown as following:

$$e_{category} = \frac{1}{n}\sum_{i=1}^{n} c^i, \tag{1}$$

where

$$c^i = \frac{1}{K}\sum_{j=1}^{K} w_j^i. \tag{2}$$

2) Average of Category headwords: By observing Chinese linguistic structure, generally, the headword of each category C^i is the last word W_K^i. We presume that other words besides the headword may bring some noisy information; thus, in this method, we acquire $e_{category}$ by averaging only every category headword of a single title and do not consider any other words. Computing process is shown as following:

$$e_{category} = \frac{1}{n}\sum_{i=1}^{n} c^i, \tag{3}$$

where

$$c^i = w_K^i. \tag{4}$$

3) Weighted Sum of Categories (WSC): We assume that each category of a title has different degree of representative and its representative depends on its headword's occurrence in context of the title. Therefore, we assert that categories should be treated distinctively according to their representative degrees. In this section, we acquire $e_{category}$ by summing up categories of a single title with the occurrence of category headword in context. Considering category information completeness, we obtain a category embedding by averaging all words' embeddings in the category but apply a different weight d to the headword. Computing process is shown as following:

$$e_{category} = \sum_{i=1}^{n} a_i c^i, \tag{5}$$

where a_i is the category headword frequency in context with normalization and

$$c^i = \frac{1}{K+d-1}\left(\sum_{j=1}^{K-1} w_j^i + d w_K^i\right). \tag{6}$$

where d is the weight added to the headword.

4) Top N Categories: Considering category occurrence in context, we argue that categories with higher frequency should gain more attention. Therefore, we only compute categories embedding $e_{category}$ by summing up top 1, top 2, and top 3 categories relatively. Computing process is shown as following:

$$e_{category} = \sum_{i=1}^{N} c^i, \tag{7}$$

where $N = 1, 2, 3$ and

$$c^i = \frac{1}{K}\sum_{j=1}^{K} w_j^i. \tag{8}$$

C. Wikipedia Title Embedding

Wikipedia title embedding, short for title embedding, is the improved word embedding with combination of context embedding and categories embedding. We propose two approaches to acquire title embedding. One is linear combination and the other is concatenation.

1) Linear Combination: We acquire title embedding e_{title} by linear combining context embedding and categories embedding. Process of computing e_{title} is shown as following:

$$e_{title} = \alpha * e_{context} + (1-\alpha) * e_{category}, \tag{9}$$

where $0 < \alpha < 1$, $e_{context}$ is obtained from III-A and $e_{category}$ is obtained from III-B.

2) Concatenation: The method is to obtain category embedding of the title via methods in III-B. Then we concatenate it with the context-based embedding.

IV. Experiments

A. Evaluation Set Translation

There are barely evaluation sets with large enough amount of data for Chinese word embeddings. Therefore, we translate three evaluation sets from English to Chinese and check manually, two for word similarity task and one for analogical reasoning task. We get 3,000 word pairs by translating **MEN-3k** [19] dataset, 287 word pairs by translating **MTurk-287** [20] dataset, and 11,126 analogical questions by translating **Google analogy dataset** [3]. After finishing our research, we will release the Chinese datasets.

1) Difficulties: We encounter some obstacles while translating the datasets. In word similarity datasets, some English word pairs have slight differences which are even unnoticeable in Chinese. For instance, word pairs such as *(stairs, staircase)*, both words mean stairs in most dictionary. We look these words up in various dictionary resources to select different but appropriate meaning in Chinese. In analogical reasoning questions set, some English words are difficult to find an appropriate mapping in Chinese words because of the differences in word usages. For example, in syntactic relationship type questions, such as adjective-adverb relation, e.g. *apparent, apparently*. We discard questions in syntactic relationship type.

B. Word Similarity Tasks

Word similarity task datasets contain relatedness scores for word pairs; the cosine similarity of the two word embeddings should have high correlation. We have two datasets: **MEN-3k** and **MTurk-287** in Chinese version. To get development set and testing set, we split both datasets in halves, i.e, 1,500 word pairs in both development and testing set for **MEN-3k** and 143 word pairs for development set and 144 word pairs in testing set for **MTurk-287**.

Table I shows result of Linear Combination in section III-C1.We tune the weight α via development set and then apply the best α, which $\alpha = 0.9$, on testing set. Comparing to the baseline, our proposed methods get significant improvement.

Method	MEN-3k		MTurk-287	
	dev	test	dev	test
Skip-gram	67.10	64.60	50.40	59.40
Avg(words)	**67.80**	**65.20**	50.20	59.60
Avg(headwords)	67.70	**65.20**	50.10	**60.20**
WSC (d=1)	67.50	65.10	50.10	59.40
WSC (d=2)	67.50	65.10	50.10	59.50
WSC (headwords)	67.50	65.10	49.90	59.50
Top 3 categories	67.70	**65.20**	50.00	59.70
Top 2 categories	67.70	**65.20**	50.00	59.50
Top 1 category	**67.80**	**65.20**	50.20	59.60

Table I
SPEARMAN CORRELATION ON WORD SIMILARITY TASK. ALL EMBEDDING ARE 300 DIMENSIONS.

Table II shows the result of Concatenation in section III-C2. We, in fact, realize concatenation in two ways. The first procedure, Integrated Match (Int. Match), we concatenate the category embedding right after the context embedding into a 600 dimensional embedding, which is shown as below:

$$e_{title} = e_{context} || e_{category}, \tag{10}$$

where $e_{category}$ is obtained from methods in III-B. For the other procedure, Individual Match (Ind. Match), we first obtain the context embedding of a title and its corresponding category embeddings via slight combination of methods in III-B for each word pair in word similarity data set Q. Then, we have four combination of embeddings pair to select the pair with highest cosine similarity score. The procedure will be illustrated in Algorithm 1.

Algorithm 1 Individual Match

procedure INDIVIDUAL MATCH(Q)
  for each word pairs (A, B) in Q **do**
    $A_{list} = e_{context}, e_{category}$ of A
    $B_{list} = e_{context}, e_{category}$ of B
    $\hat{a}, \hat{b} = max_{a,b} CosSim(a, b)$
      where $a \in A_{list}$ and $b \in B_{list}$
    $Sim(A, B) = CosSim(\hat{a}, \hat{b})$
  end for
end procedure

The result of Integrated Match comparing to the baseline, 600 dimensional context embedding obtained from skip-gram model [4], is unsatisfied to our initial assumption. The result of Individual Match comparing to the baseline, 300 dimensional context embedding obtained from skip-gram model, validates our presumption. For the categories, we obtain its category embeddings via two methods applying to the top one category (using $N = 1$ in III-B4)): averaging category words and applying category headword relatively according to methods in III-B1 and III-B2.

Method		MEN-3k		MTurk-287	
		dev	test	dev	test
baseline	SG-300	67.10	64.60	50.40	59.40
	SG-600	67.30	65.40	50.10	56.50
Int. Match		65.40	61.60	49.30	59.65
Ind. Match	Avg(w of c)	73.20	77.90	82.50	64.30
	hword of c	**75.40**	**81.40**	**83.60**	**68.90**

Table II
SPEARMAN CORRELATION ON WORD SIMILARITY TASK. EMBEDDING IN SG-600 AND INT. MATCH ARE 600 DIMENSIONS. ALL THE OTHER EMBEDDING ARE 300 DIMENSIONS. w DENOTES WORDS IN THE TOP ONE CATEGORY c.

Comparing to the baseline, our proposed method, Ind. Match, get significant improvement, though Int. Match does not conquer the baseline. As in Ind. Match, each word in word pairs has several independent embedding candidates to choose from, i.e., each word can choose from either its context embedding or its category embedding, it can therefore choose the embedding with better representative and achieve a better performance.

C. Analogical Semantics Tasks

Analogical reasoning dataset is compromised of analogous word pairs, i.e, pairs of tuples of word relations that follow a common syntactic relation. We use translated **Google dataset** and split it in halves as development and testing sets. Each set contains 5,563 questions.

Table III shows result of Linear Combination in III-C1. We tune the weight α via development set and then apply the best α, which is $\alpha = 0.9$, on testing set. Comparing to the baseline, our proposed methods get significant improvement.

Method	Google	
	dev	test
Skip-gram	53.12	34.71
Avg(words)	**55.71**	**36.33**
Avg(headwords)	53.66	35.65
WSC (d=1)	**54.43**	**35.12**
WSC (d=2)	54.14	35.00
WSC (head word)	53.17	34.96
Top 3 categories	54.57	35.30
Top 2 categories	54.84	35.38
Top 1 category	**55.51**	**36.10**

Table III
ACCURACY ON ANALOGICAL REASONING TASK. ALL EMBEDDINGS ARE 300 DIMENSIONS.

V. Discussion

According to the experiment result, linear combination using category embedding which obtains from averaging all category words has the best performance. This circumstance contradicts one of our initial assumptions: categories have different extent of representative and their representative depend on the occurrence in context of the title. That is, we assume that methods stress on headwords should have better performances; however, it turns out that our assumption is not rigorous enough. Perhaps the categories have no distinctive representation degree. Or if distinguishable representative degree exists, the extent is related to other factors, such as the position of category appears in context. It is plausible that the category which denotes the first sentence in the context deserves most attention.

Although the effect of representative degree of categories is unclear in linear combination, the improvement is obvious in Individual Match. We apply Ind. Match only in word similarity task due to the word coverage of Wikipedia title in question and computing complexity. It is likely that the representative degree matters only in some tasks and methods applied. In either circumstances, that categories can provide valuable information for improving context-based embedding is undeniable.

VI. Conclusion

In this paper, we argue that a Wikipedia title's categories can help define or complement the meaning of the title besides the title's Wikipedia text, so the categories should be utilized to improve the title's embedding. We purpose two directions of utilizing Wikipedia categories to improve context-based embedding trained from Wikipedia text. Experiments on both word similarity task and analogical reasoning task show that our approaches significantly outperform conventional context-based approaches.

In the future, it is worth investigating whether the categories have distinct degree of representative and if so, what factors affect the degree.

References

[1] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[2] P. Dhillon, J. Rodu, D. Foster, and L. Ungar, "Two step cca: A new spectral method for estimating vector models of words," *arXiv preprint arXiv:1206.6403*, 2012.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[5] R. Lebret and R. Collobert, "Rehabilitation of count-based models for word vector representations," *arXiv preprint arXiv:1412.4930*, 2014.

[6] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[7] A. Bordes, J. Weston, R. Collobert, Y. Bengio *et al.*, "Learning structured embeddings of knowledge bases." in *AAAI*, 2011.

[8] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes." in *AAAI*, 2014, pp. 1112–1119.

[9] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion." in *AAAI*, 2015, pp. 2181–2187.

[10] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix." in *ACL (1)*, 2015, pp. 687–696.

[11] J. Feng, M. Huang, Y. Yang, and X. Zhu, "Gake: Graph aware knowledge embedding." in *COLING*, 2016, pp. 641–651.

[12] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. AcM, 2008, pp. 1247–1250.

[13] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[14] M. Yu and M. Dredze, "Improving lexical embeddings with semantic knowledge." in *ACL (2)*, 2014, pp. 545–550.

[15] D. Bollegala, M. Alsuhaibani, T. Maehara, and K.-i. Kawarabayashi, "Joint word representation learning using a corpus and a semantic lexicon." in *AAAI*, 2016, pp. 2690–2696.

[16] H.-Y. Wang and W.-Y. Ma, "Integrating semantic knowledge into lexical embeddings based on information content measurement," *EACL 2017*, p. 509, 2017.

[17] W.-Y. Ma and K.-J. Chen, "A bottom-up merging algorithm for chinese unknown word extraction," in *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*. ACL, 2003, pp. 31–38.

[18] W. Y. Ma and K. J. Chen, "Introduction to ckip chinese word segmentation system for the first international chinese word segmentation bakeoff," in *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*. Association for Computational Linguistics, 2003, pp. 168–171.

[19] E. Bruni, G. Boleda, M. Baroni, and N.-K. Tran, "Distributional semantics in technicolor," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2012, pp. 136–145.

[20] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch, "A word at a time: computing word relatedness using temporal semantic analysis," in *Proceedings of the 20th international conference on world wide web*. ACM, 2011, pp. 337–346.