Learning Multi-Prototype word representation based on subword information

Bo Liu, Caigao Jiang
20143509, 20238976
bliuab@cse.ust.hk, cjiangad@ust.hk

Abstract

In pursuit of better performance in NLP tasks, word representation is required to consider comprehensive information such as the semantics and syntax information. Modeling polysemy and morphology are proven to be beneficial for learning the word representation. The multi-prototype method incorporates the polysemy according to the context. Considering morphology using the subword method alleviate the challenges of rare words. In this project, we simultaneously consider polysemy and morphology in pursuit of even better performance. Our proposed word representation method is trained on Wikipedia subset corpora. The qualitative and quantitative empirical results demonstrate the superiority of our proposed method.

Introduction

Learning meaningful word representation is vital for the majority of natural language processing (NLP) tasks. The word representation models each word in a continuous vector space. The words with similar information should be embedded nearby to each other. For instance, Spain and France or Madrid and Paris with similar semantics meaning should lie close in the distributed representation. In pursuit of an effective word representation, comprehensive information, particularly semantics (Mikolov et al. 2013a) and syntax (Andreas and Klein 2014), are suggested to be embedded.

Embedding more structure knowledge usually leads to more effective representations. In this project, we focus on polysemy and morphology. One major limitation of classical word representation method is that only one single representation is utilized for each word. However, the specific word, e.g. *Check* in Figure 1, may have multiple meanings. Mixing multiple meanings using a single representation is clearly problematic. Recent multi-prototype method cluster each word according to its context and learn representation for each word cluster separately (Huang et al. 2012).

In recent years, large collections of work have been proposed to learn rich word representations. However, most of the popular models represent each word without considering the internal structure of words. In this way, each word is presented as a distinct vector by a word vocabulary (Bojanowski et al. 2016). Therefore, it would be problematic to

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

"Check" /tʃɛk/

1. Verify:
Customs officers have the right to check all luggage.

2. Win in Chess:
He moves his knight to check my king again

3. Cheque:
Let's get the bank check.

Figure 1: The example of polysemy for word "*Check*"

infer the representations for these words that occur rarely in the vocabulary. To model rare word better, it is reasonable to incorporate morphological information to learn a character-level representation. For instance, in Figure 2, each word is analyzed into stems, root words, prefixes, and suffixes. Intuitively, the occurrences of *un*, *fortunate*, and *ly* in different words benefit the representation of *unfortunately*. For frequent words, the subword information enriches the representation as well.

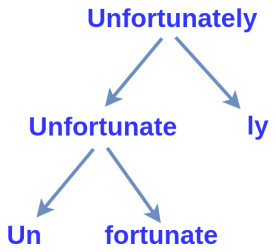

Figure 2: The example of morphology for word "*Unfortunately*"

Incorporating polysemy or morphology are proven to enrich word representation. Nevertheless, as far as we know, no existing methods consider polysemy and morphology simultaneously. In this project, we learn representation for each word using skip-gram model with subword information. Based on the learned representation, we use the multi-prototype method to distinguish multiple meanings of the specific words followed by learning representation for each word cluster. We verify the effectiveness of our methods using Wikipedia corpora and human similarity judgment.

Table 1: Comparison of multi-prototype methods for modelling polysemy

	Local Context	Global Context	Infer #Cluster	Clustering Method
Reisinger and Mooney 2010b	✓			Spherical K-Means
Huang et al. 2012	✓	✓		Spherical K-Means
Reisinger and Mooney 2010a	✓		✓	Tiered Clustering
Tian et al. 2014	✓		✓	Iterative Estimation

Related Work

Existing multi-prototype methods cluster each word according to its context as shown in Figure 1. The representation for each word cluster denotes one among multiple meanings. (Reisinger and Mooney 2010b) and (Huang et al. 2012) utilizes spherical k-means to cluster each word. Such methods suffer from two major limitations. Firstly, the number of clusters is fixed for all words. Then, the number of clusters is required to be tuned manually. (Huang et al. 2012) usually performs better in that it not only considers local context but also global context.

(Reisinger and Mooney 2010a) and (Tian et al. 2014) consider the polysemy with Bayesian method. More specifically, (Reisinger and Mooney 2010a) proposed the *Tiered clustering* method which models each word as Dirichlet Process Mixture Model (DPMM). *Tiered clustering* enjoys great advantages of inferring the number of clusters for each word automatically. (Tian et al. 2014) models the clustering and word representation learning as a uniform Bayesian model. And it estimates the cluster and word representation iteratively. We compare all multi-prototype methods in Table 1.

A large body of work have investigated the morphological representation of a word in the past. (Bojanowski et al. 2016) introduced a new approach to obtain both word-level and character-level representation by taking into sub-word information based on the skip-gram model. The word embedding is represented as the sum of each word n-gram and character n-gram vector representation. (Sperr, Niehues, and Waibel 2013) present a character encoding method for learning rich word representation by taking the letter into consideration. They capitalized restricted Boltzmann machines in order to better generalize to the rare word in the machine translation task. Furthermore, (Luong and Manning 2016) proposed a hybrid word-character model based on the recurrent neural network in machine translation task in order to address word sparsity problem for rare word. In the field of sentiment analysis, (dos Santos and Gatti 2014) capitalized deep convolution neural network to obtain morphology-level information and sentence-level information for short text in sentiment classification task. Although this kind of methods all takes local or global context into consideration, they do not incorporate polysemy information. The detailed comparison is shown in the Table 2

Methodology

In this section, we introduce our method to learn multi-prototype word representation combine with sub-word information.

Table 2: Comparison of word representation using sub-word information

	Local Context	Global Context	Polysemy	Method
Bojanowski 2016	✓	✓		Word2vec
Sperr 2013	✓	✓		RBM
Luong and Manning 2016	✓	✓		RNN
Santos and Gatti 2014	✓	✓		CNN

Skip-gram model

Let's briefly look back the skip-gram model, which is proposed in (Mikolov et al. 2013a). The idea of the skip-gram model is to use the current central word as the input vector to the hidden linear neurons, and predict this word appearing in a constant range of context. Assuming that we have a word vocabulary with the size of W, in this way, each of word can be represented as a one-hot vector. More specifically, given the training corpus represented as a sequence of words $w_1, w_2, ..., w_T$, the objective of the continuous skip-gram model is to maximize the average log likelihood:

$$\frac{1}{T}\sum_{T}^{t=1}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{1}$$

where the constant c is the size of context words surrounding w_t. The probability of assigning a context word a central word w_t is parametrized using the word vectors. The basic skip-gram formulation to define the probability $p(w_{t+j}|w_t)$ is the softmax function:

$$p(w_O|w_I) = \frac{\exp({v^{'}_{w_o}}^T v_{w_I})}{\sum_{W}^{w=1}\exp({v^{'}_{w}}^T v_{w_I})} \tag{2}$$

where v_w and $v^{'}_w$ are the input and output vector representations of the central word w. Denote a scoring function s, which is to take score that maps the pair of central word and its context. A common choice for parametrization for scoring function s is to take the scalar product:

$$s(w_t, w_c) = {v^{'}_{w_o}}^T v_{w_I} \tag{3}$$

An alternative to the softmax function is Noise Contrastive Estimation(NCE), which is to approximately maximize the log likelihood of the softmax. Negative sampling is usually defined by (Mikolov et al. 2013b):

$$\log\sigma({v^{'}_{w_o}}^T v_{w_I}) + \sum\mathbb{E}_{w_i\sim p_n(w)}[\log\sigma({v^{'}_{w_i}}^T v_{w_I})] \tag{4}$$

The following figure shows how the conventional skip-gram model works:

Sub-word model

From the above conventional skip-gram model, it is not surprising to see that this kind of model only considers the

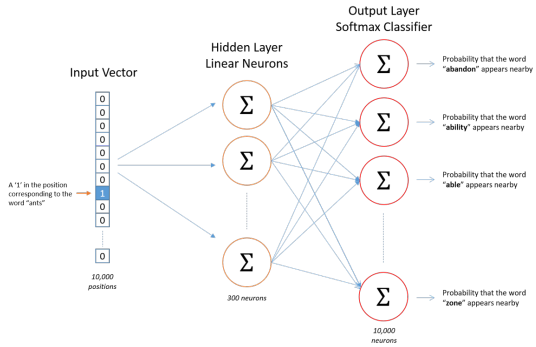

Figure 3: The skip-gram model: the word "ants" is represented using it context words.

word-level information and ignores the morphological information. In this section, we thus give the formulation of how to integrate the sub-word information into a unified skip-gram model by redefining a new scoring function s. Given a central word w, we define the character-level n-gram as $g_w \subset 1, 2, .., G$ and thus allocate the representation z_g to the n-grams g. Therefore, the new formulation of scoring function can be(Bojanowski et al. 2016):

$$s(w, c) = \sum_{g \in g_w} z_g^T v_c \tag{5}$$

From this equation and above analysis, we can infer that each word is represented as a word level representation and character level representation as well.

Learn multi-prototype representation with sub-word information

In this section, we give our solution on how to extract polysemy information and combine sub-word information into a unified model in order to get richer multi-prototype representations of words. Take the following example to illustrate how we extract multi-prototype information for each word. Assuming that there are five words "bank" in one document. The first step is to extract all the context where appears the word "bank", furthermore, each context will be represented as an embedding vector. Thus, we get five context vector presentation for one word "bank". Finally, the k-mean clustering is performed on those context vector. The following figure shows how it works.

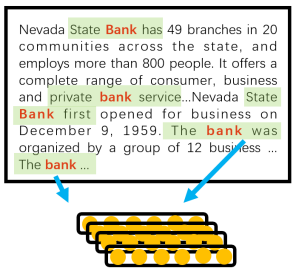

Figure 4: The example of extracting multi-prototype information

Furthermore, the following pseudocode shows how our proposed algorithm works.

Algorithm 1 Learn multi-prototype representation with sub-word information

Input: $1million$ English Wikipedia training corpus with vocabulary size W, and denote each word as w;

Output: The learned representations of words w: w_c

1: **repeat**
2: Learn representation w_s for word w using skip-gram model with sub-word information g_s
3: Extract word2vec embedding w_{em} for word w
4: Extract context representation w_{con}^i, and $i = 1, 2, ..n$; n is the number that word w appears in the corpus.
5: Capitalize the extracted context representation to do k-means clustering in order to get updated representation w_c^j, and $j = 1, 2, ...m$; m is the number of clusters.
6: Treat the word in w_c^j as m different words, and perform stage 1 again.
7: **return** w_c

Experiments

In this section, we verify the effectiveness of our proposed method based on both qualitative and quantitative results. Firstly, we list the nearest neighbor of example words based on our learned word representations. We prove that our method is capable of distinguishing multiple meanings of the specific word. Then, we show that our proposed method achieves improved performance in human similarity judgment experiments. We also discuss how the number of clusters influences the correlation with human judgments.

Implementation details

In this project, we train our word representation using the latest Wikipedia corpora (Nov. 2016). Due to efficiency issues, we randomly sample 1 million articles from Wikipedia corpora. Across our experiments, we fix the representation vector to be dimension 50 both in morphological and multi-prototype information extraction. Moreover, we ignore the words which appear less than 5 times. In terms of mult-prototype, we consider 10 word neighbors as the context for clustering. In the construction of sub-word n-gram dictionary, we keep all n-gram with the length between 3 and 6.

Nearest Neighbors

In this part, we aim to prove that our proposed method distinguishes multiple meaning of words. For the sake of simplicity, we fix two clusters for each word. Based on cosine similarity calculated on our learned embeddings, we further obtain the top4 nearest neighbor for each word cluster. The nearest neighbors from the same word source are ignored.

We demonstrate nearest neighbors in Table 3. For instance, *Master-1* and *Master-2* denotes the first and second cluster for word *Master* respectively. Obviously, for words *Master*, *Left*, and *Bank*, different word clusters successfully capture different meanings of the specific word. Unfortunately, bad cases exist in our results. For instance, two clusters of word *Band* represent similar meanings related with music.

Table 3: Top4 nearest neighbor for word clusters

Query	Top4 Nearest Neighbor
Master-1	bachelor, degree, graduate, faculty
Master-2	designer, knight, lord, baron
Left-1	leaving, faced, stayed, returned
Left-2	right, front, face, behind
Bank-1	side, corner, shore, river
Bank-2	stock, fund, corporation, capital
Band-1	duo, album, guitarist, rock
Band-2	drummer, guitarist, beatles, trio

Table 4: Examples of human similarity judgement dataset

Word 1	Word 2	Human Judgement	Cosine Similarity
tiger	cat	7.35	0.65
book	paper	7.46	0.73
stock	phone	1.62	0.54
stock	life	0.92	-0.157

Human similarity judgement

For quantitative analysis, we utilize WS353 dataset (Finkelstein et al. 2001) and rare word dataset (RW) (Luong, Socher, and Manning 2013). RW dataset is utilized to evaluate whether the subword information benefits the embeddings for rare words. The datasets provide many pairs of words accompanied with their similarity based on human judgement. The cosine similarity between embedded representations is also obtained. We calculate the Spearman's rank correlation between human judgment and cosine similarity to evaluate our word representations. We show an example of the dataset in Table 4.

In particular, based on our learned word representation, *stock* and *life* are negative correlated. In my view, the loss function of n-gram model maximizes the correlation between words within a window and minimize the correlation between random sampled words. As a result, we reckon that the negative cosine similarity is due to the pair *stock* and *life* almost never occur together. The correlation between *stock* and *life* are minimized towards negative when they are randomly drawn. In conclusion, we hypothesize that the negative correlation means that the pairs of word rarely occurs together.

Figure 5 shows the Spearman's rank correlation with respect to the number of clusters. Moreover, we also compare different baselines with and without subword information. The baselines without subword information are denoted as *Word2Vec*.

According to Figure 5, we obtain the following conclusions. Firstly, on both dataset, a multi-prototype method with 2 clusters or more achieves improved performance compared to the method without multi-prototype. This clearly proves that distinguishing multiple meanings with multiple representations is beneficial. On WS353 dataset, however, the performance for a multi-prototype method with 5 or 10 clusters becomes inferior. We reckon that more clusters make the training data for each word cluster more sparse, thereby leading to decreasing performance. We hypothesize that more clusters may be beneficial based on full scale Wikipedia dataset.

Secondly, on WS353 dataset, the methods with and without subword information perform comparable. In contrast, on RW dataset, the representation incorporating subword information outperforms the baseline consistently. Such result clear shows that our skip-gram model with subword information benefits embeddings for rare words significantly.

Finally, on both datasets, the best performance is achieved when both multi-prototype and subword information are considered.

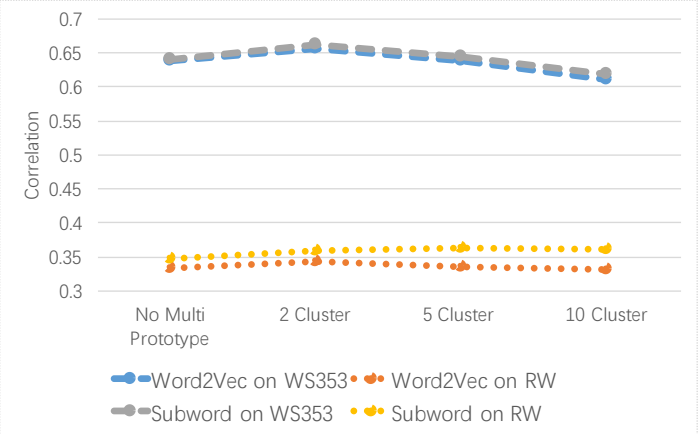

Figure 5: Performance with respect to #clusters

Conclusions

In this project, we pursue a better word representation by considering polysemy and morphology simultaneously. We utilize the skip-gram model with subword information on the clusters of words. Both qualitative and quantitative results verify our motivation that incorporating polysemy and morphology simultaneously leads to an improved word representation. In the future, we plan to infer the number of clusters automatically. Moreover, considering multiple meanings for stems, root words, prefixes, and suffixes is also attractive for us.

References

Andreas, J., and Klein, D. 2014. How much do word embeddings encode about syntax? In *ACL (2)*, 822–827.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

dos Santos, C. N., and Gatti, M. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, 69–78.

Finkelstein, L.; Gabrilovich, E.; Matias, Y.; Rivlin, E.; Solan, Z.; Wolfman, G.; and Ruppin, E. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, 406–414. ACM.

Huang, E. H.; Socher, R.; Manning, C. D.; and Ng, A. Y. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th*

Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, 873–882. Association for Computational Linguistics.

Luong, M.-T., and Manning, C. D. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv preprint arXiv:1604.00788*.

Luong, T.; Socher, R.; and Manning, C. D. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*, 104–113.

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

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

Reisinger, J., and Mooney, R. 2010a. A mixture model with sharing for lexical semantics. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 1173–1182. Association for Computational Linguistics.

Reisinger, J., and Mooney, R. J. 2010b. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 109–117. Association for Computational Linguistics.

Sperr, H.; Niehues, J.; and Waibel, A. 2013. Letter n-gram-based input encoding for continuous space language models. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, 30–39.

Tian, F.; Dai, H.; Bian, J.; Gao, B.; Zhang, R.; Chen, E.; and Liu, T.-Y. 2014. A probabilistic model for learning multi-prototype word embeddings. In *COLING*, 151–160.