Explaining Word Embeddings via Disentangled Representation

Keng-Te Liao
National Taiwan University
d05922001@ntu.edu.tw

Cheng-Syuan Lee
National Taiwan University
r07922055@ntu.edu.tw

Zhong-Yu Huang
National Taiwan University
r06944047@ntu.edu.tw

Shou-de Lin
National Taiwan University
sdlin@csie.ntu.edu.tw

Abstract

Disentangled representations have attracted increasing attention recently. However, how to transfer the desired properties of disentanglement to word representations is unclear. In this work, we propose to transform typical dense word vectors into disentangled embeddings featuring improved interpretability via encoding polysemous semantics separately. We also found the modular structure of our disentangled word embeddings helps generate more efficient and effective features for natural language processing tasks.

1 Introduction

Disentangled representations are known to represent interpretable factors in separated dimensions. This property can potentially help people understand or discover knowledge in the embeddings. In natural language processing (NLP), works of disentangled representations have shown notable impacts on sentence and document-level applications. For example, Larsson et al. (2017) and Melnyk et al. (2017) proposed to disentangle sentiment and semantic of sentences. By manipulating sentiment factors, the machine can rewrite a sentence with different sentiment. Brunner et al. (2018) also demonstrated sentence generation while more focusing on syntactic factors such as part-of-speech tags. For document-level applications, Jain et al. (2018) presented a learning algorithm which embeds biomedical abstracts disentangling populations, interventions and outcomes. Regarding word-level disentanglement, Athiwaratkun and Wilson (2017) proposed mixture of Gaussian models which can disentangle meanings of polysemous words into two or three clusters. It has a connection with unsupervised sense representations (Camacho-Collados and Pilehvar, 2018) which is an active research topic in the community.

In this work, we focus on word-level disentanglement and introduce an idea of transforming dense word embeddings such as GloVe (Pennington et al., 2014) or word2vec (Mikolov et al., 2013b) into disentangled word embeddings (DWE). The main feature of our DWE is that it can be segmented into multiple sub-embeddings or sub-areas as illustrated in Figure 1. In the figure, each sub-area encodes information relevant to one specific topical factor such as *Animal* or *Location*. As an example, we found words similar to “turkey” are “geese”, “flock” and “goose” in the *Animal* area, and the similar words turn into “Greece”, “Cyprus” and “Ankara” in the *Location* area.

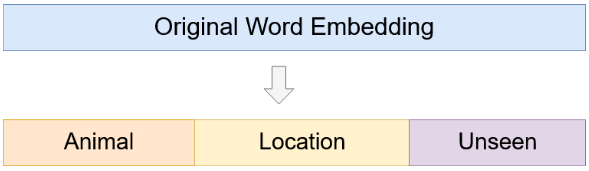

Figure 1: Disentangled embedding with factors *Animal*, *Location* and *Unseen*.

We also found our DWE generally satisfies the *Modularity* and *Compactness* properties proposed by Higgins et al. (2018) and Ridgeway and Mozer (2018) which can be a definition of general-purpose disentangled representations. Also, our DWE can have the following advantages:

- **Explaining Underlying Knowledge**
The multi-senses of words can be extracted and separately encoded despite the learning algorithm of the original word embeddings (e.g. GloVe) does not do disambiguation. As a result, the encoded semantic can be presented in an intuitive way for examination.

- **Modular and Compact Features**
Each sub-area of our DWE can itself be informative features. The advantage is that people

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 720–725
December 4 - 7, 2020. ©2020 Association for Computational Linguistics

are free to abandon features in sub-areas irrelevant to the given downstream tasks while still achieving competitive performance. In Section 4, we show that using the compact features is not only efficient but also helps improve performance on downstream tasks.

- **Quality Preservation**
  In addition to higher interpretability, our DWE preserves co-occurrence statistics information in the original word embeddings. We found it also helps preserve the performance on downstream tasks including word similarity, word analogy, POS-tagging, chunking, and named entity recognition.

2 Obtaining Disentangled Word Representations

2.1 Problem Definition

Our goal is transforming N d-dimensional dense word vectors $X \in \mathbb{R}^{N \times d}$ into disentangled embeddings $Z \in \mathbb{R}^{N \times d}$ by leveraging a set of binary attributes $\mathcal{A} = \{a_1, ..., a_M\}$ labelled on words.

Z is expected to have two properties. The first one is preserving word features encoded in X. More specifically, we require $XX^T \approx ZZ^T$ as pointed out by Levy and Goldberg (2014) that typical dense word embeddings can be regarded as factorizing co-ocurrence statistics matrices.

The second property is that Z can be decomposed into $M+1$ sub-embedding sets $Z_{a_1}, ..., Z_{a_M}$ and Z_{unseen}, where each sub-embedding set encodes information only relevant to the corresponding attribute. For example, Z_{a_1} is expected to be relevant to a_1 and irrelevant to $a_2, ..., a_M$. Information in X not relevant to any attributes in $\mathcal{A}$ is then encoded in Z_{unseen}. An example of transforming X into Z with two attributes, *Animal* and *Location*, is illustrated in Figure 1.

For modelling the relevance between sub-embeddings and attributes, we use mutual information $\mathcal{I}(Z_a, a)$ as learning objectives, where a is an arbitrary attribute in $\mathcal{A}$.

2.2 Transformation with Quality Preservation

We obtain Z by transforming X by a matrix $W \in \mathbb{R}^{d \times d}$. That is, $Z = XW$. To ensure $XX^T \approx ZZ^T$, an additional constraint $WW^T = I$ is included. $ZZ^T = (XW)(XW)^T = X(WW^T)X^T = XX^T$ if $WW^T = I$ holds.

2.3 Optimizing $\mathcal{I}(Z_a, a)$

Let $z_{a,i}$ be the i-th row in Z_a. By derivation, $\mathcal{I}(Z_a, a) =$

$$\Sigma_{i=1}^{N} p(z_i) p(a|z_{a,i}) \big[\log p(a|z_{a,i}) - \log p(a)\big]$$

$$\approx \frac{1}{N} \Sigma_{i=1}^{N} p(a|z_{a,i}) \big[\log p(a|z_{a,i}) - \log p(a)\big]$$

We let $\log p(a)$ be constant and replace $p(a|z)$ with a parametrized model $q_\theta(a|z)$. By experiments, we found logistic regression with parameter θ is sufficient to be $q_\theta(a|z)$. Intuitively, high $\mathcal{I}(Z_a, a)$ means Z_a are informative features for a classifier to distinguish whether words has attribute a.

When increasing $\mathcal{I}(Z_a, a)$ by optimizing $q_\theta(a|z)$, we found a strategy helping generate higher quality Z. The strategy is letting Z_a be features to reconstruct original vectors for words having attribute a. For words with a, the approach becomes a semi-supervised learning architecture which attempts to predict labels and reconstruct inputs simultaneously.

The loss function $\mathcal{L}(W, \theta, \phi)$ for maximizing $\mathcal{I}(Z_a, a)$ is as follow:

$$\frac{-1}{N} \Sigma_{i=1}^{N} q_\theta(a|z_{a,i}) + \lambda \mathbb{I}_{a,i} ||x_i - \phi(z_{a,i})||_2^2$$
$$\mathbb{I}_{a,i} = \begin{cases} 1 & \text{when } i\text{-th word has attribute } a \\ 0 & \text{when } i\text{-th word does not have } a \end{cases} \tag{1}$$

where ϕ is single and fully-connected layer, x_i is the original i-th word's vector in X, and λ is a hyper-parameter. We set $\lambda = \frac{1}{d}$ in all experiments.

2.4 Learning to Generate Sub-embedding Z_a

As discussed in 2.3 that high $\mathcal{I}(Z_a, a)$ indicates Z_a are informative features for classification, we propose to regard sub-embedding generation as a feature selection problem. More specifically, we apply sparsity constraint on Z. Ideally, when predicting a, a smaller number of dimensions of Z are selected as the informative features, which are regarded as Z_a.

In this work, we use *Variational Dropout* (Kingma et al., 2015; Molchanov et al., 2017) as the sparsity constraint. At each iteration of training, a set of multiplicative noise ξ is sampled from a normal distribution $\mathcal{N}(1, \alpha_a = \frac{p_a}{1-p_a})$ and injected on Z. That is, the prediction and reconstruction is done by $\theta(\xi \odot Z)$ and $\phi(\xi \odot Z)$. The parameter $\alpha_a \in \mathbb{R}^d$ is jointly learned with W, θ, and ϕ. Afterwards, d-dimensional dropout rates

$p_a = sigmoid(\log \alpha_a)$ can be obtained. For each attribute a in $\mathcal{A}$, the dimensions with dropout rates lower than 50% are normally regarded as Z_a.

We would like to emphasize that the learned dropout rates are not binary values. Therefore, deciding the length of sub-embeddings can actually depend on users preferences or tasks requirements. For example, users can obtain more compact and pure Z_a by selecting dimensions with dropout rates lower than 10%, or get more thorough yet less disentangled Z_a by setting the threshold be 70%.

To encourage disentanglement when handling multiple attributes, we include additional loss functions on dropout rates. Let a M-dimensional vector P be $1 - p_a$ for all a in $\mathcal{A}$ in a specific dimension. The idea is to minimize $\prod_{i=1}^{M} P_i$ with constraint $\Sigma_{i=1}^{M} P_i = 1$. The optimal solution is that the dimension is relevant to only one attribute a' where $1 - p_{a'} \approx 1$. In implementation, we minimize the following loss function

$$\Sigma_{i=1}^{M} \log P_i + \beta ||\Sigma_{i=1}^{M} P_i - 1||_2^2 \qquad (2)$$

We set $\beta = 1$ in the experiments, and equation 1 and 2 are optimized jointly.

To generate Z_{unseen}, we initially select a set of dimensions and constrain their dropout rates be always larger than 50%. The number of dimensions of Z_{unseen} is a hyper-parameter. After selection, we do not apply equation 2 on the selected dimensions.

3 Evaluation

3.1 Word Embeddings and Attributes

We transform 300-dimensional GloVe[1] into DWE. The 300-dimensional GloVe is denoted by GloVe-300. For word attributes $\mathcal{A}$, we use labels in **WordStat**[2]. WordStat contains 45 kinds of attributes labeled on 70,651 words. Among the attributes, we select 5 high-level and easily understandable attributes: *Artifact*, *Location*, *Animal*, *Adjective (ADJ)* and *Adverb (ADV)* for our experiments. The number of words labelled with these 5 attributes is 13,337. After training, all pre-trained GloVe vectors are transformed by the learned matrix W (i.e. XW) for downstream evaluations.

The number of learned dimensions for each attribute is illustrated in Figure 2, where the threshold of dropout rates for dimension selection is 50%.

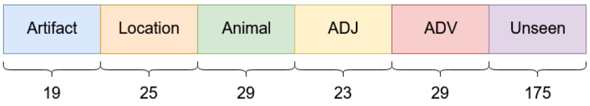

Figure 2: Disentangled embedding with five attributes: *Artifact*, *Location*, *Animal*, *Adjective* and *Adverb*. The remaining dimensions are viewed as *Unseen*.

	MEN	SimLex	BATS	GA
GloVe-300	0.749	0.369	**18.83**	**63.58**
DWE	**0.764**	**0.390**	18.75	62.30

Table 1: Word similarity and analogy performance.

	POS	Chunking	NER
GloVe-300	65.0	64.9	65.2
DWE	**67.2**	**66.3**	**66.1**

Table 2: POS-tag, chunking and NER performance.

3.2 Evaluation of Quality Preservation

We firstly examine whether DWE can preserve features encoded in GloVe-300. The examination is done by intrinsic evaluations including the following tasks and datasets.

- Word Similarity: **Marco, Elia and Nam (MEN)** (Bruni et al., 2014) and **SimLex-999** (Hill et al., 2015).
- Word Analogy: **Bigger Analogy Test Set (BATS)** (Gladkova et al., 2016), **Google Analogy (GA)** (Mikolov et al., 2013a).
- POS tagging, Chunking and Named Entity Recognition (NER): **CoNLL 2003** (Sang and Meulder, 2003; Li et al., 2017).
- QVEC-CCA[3] (Tsvetkov et al., 2015): The performance is measured by semantic and syntactic CCA.

As shown in Table 1, 2 and 3, DWE can preserve performance of GloVe-300 on various NLP tasks. Probably due to the additional information of word attributes, DWE can have slightly better performance than GloVe-300 on seven of the tasks

3.3 Attribute Classification

We design an attribute classification task for examining whether the DWE can meet requirements described in Section 2.3. We use logistic regression and take sub-embeddings Z_a as input features for

[1] https://nlp.stanford.edu/projects/glove/

[2] https://provalisresearch.com/products/content-analysis-software/

[3] https://github.com/ytsvetko/qvec

	Semantic	Syntactic
GloVe-300	0.473	0.341
DWE	**0.474**	**0.348**

Table 3: QVEC-CCA evaluation.

	Artifact	Location	Animal	ADJ	ADV
$Z_{artifact}$	**77.8**	71.0	68.0	65.5	71.2
$Z_{location}$	59.2	**83.8**	64.0	60.5	69.8
Z_{animal}	58.5	67.5	**84.2**	60.2	71.0
Z_{adj}	69.8	70.7	68.2	**82.0**	72.5
Z_{adv}	59.0	72.5	71.8	71.5	**84.2**
Z_{unseen}	54.8	70.0	66.5	60.2	68.8

Table 4: Attribute classification accuracies (%).

verifying the performance of classification by cross-validation. For each attribute, We randomly sample 400 data for testing. The numbers of positive and negative data for testing are balanced. Therefore, a random predictor would get around 50% accuracy in each classification task.

The binary classification accuracies are shown in Table 4. Take the second column of Table 4 for example. For distinguishing whether a word can be location, taking $Z_{location}$ as features for training a classifier achieves the highest accuracy 83.8%. On the other hand, the accuracy reported in the second row of Table 4 implies that $Z_{location}$ are less informative features for other attributes. Similar results can also be observed for other attributes.

3.4 Disentangled Interpretability

We provide some examples to demonstrate that words having ambiguous or different aspects of semantics can be disentangled. Table 5 shows the results of nearby words. As can be seen, querying a word in Z_a with different attributes can help discover the ambiguous semantics implicitly encoded in the original word vectors X. The results also show that Z_{unseen} does capture meaningful information having little relevance to given attributes.

4 Application: Compact Features for Downstream Tasks

Here we demonstrate an application of the *modularity* and *compactness* properties of our DWE. We firstly aim to show the sub-embeddings can directly be informative features and can outperform GloVe with the same number of dimensions. With the high interpretability, selecting relevant sub-embeddings could be intuitive. Secondly, we will demonstrate that if deciding to fine-tune word vectors for a given downstream task, by using our DWE, we can focus on updating the relevant sub-embedding instead of the whole embedding. The advantage is that it reduces the number of learning parameters. Also, it could be regarded as a dimensional and interpretable regularization technique reducing overfitting.

Query	**Vectors**	**Nearby Words**
turkey	Z_{animal}	geese, flock, goose
turkey	$Z_{location}$	greece, cyprus, ankara
mouse	Z_{animal}	mice, rat, rats
mouse	$Z_{artifact}$	keyboard, joystick, buttons
japan	$Z_{location}$	korea, vietnam, singapore
japan	Z_{unseen}	japanese, yakuza, yen
apple	$Z_{artifact}$	macintosh, software, mac
apple	Z_{unseen}	mango, cherry, tomato

Table 5: Results of nearby words.

We take a sentiment analysis task, IMDB movie review classification(Maas et al., 2011), for experiments. Intuitively, *ADJ* and *ADV* should be the most relevant attributes in $\mathcal{A}$. We then select 50 dimensions from $Z \in \mathbb{R}^{300}$ with the lowest dropout rates in *ADJ* and *ADV* sub-areas for comparing with 50-dimensional GloVe[4] (GloVe-50). The embeddings with the selected dimensions are denoted by $Z_{adj+adv}$-50. When tuning our DWE with the classifier, we update the 52 dimensions ($Z_{adj} \in \mathbb{R}^{23}$ and $Z_{adv} \in \mathbb{R}^{29}$) of DWE and compare it with GloVe-300.

The document representations for classification is averaged word embeddings. The classifier is a logistic regression. When tuning the input word embeddings, we update the embeddings with gradient propagated from the classifier.

The results are listed in Table 6. From the table, we can see $Z_{adj+adv}$-50 directly outperforms GloVe-50 without tuning. A possible explanation is that GloVe-50 is forced to encode information less relevant to the sentiments, making it less effective than $Z_{adj+adv}$-50 in this task.

In the fine-tuning experiments, DWE can show slightly higher accuracy than GloVe-300 by updating only 52 instead of 300 dimensional features.

[4] https://nlp.stanford.edu/projects/glove/

Feature	Without Tuning	After Tuning
GloVe-50	76.55	86.72
$Z_{adj+adv}$-50	79.78	87.60
GloVe-300	**83.85**	87.72
DWE	83.67	**87.84**

Table 6: Classification accuracies (%) on IMDB dataset.

5 Conclusion

In this work, we propose a new definition and learning algorithm for obtaining disentangled word representations. As a result, the disentangled word vectors can show higher interpretability and preserve performance on various NLP tasks. We can also see the ambiguous semantics hidden in typical dense word embeddings can be extracted and separately encoded. Finally, we showed the disentangled word vectors can help generate compact and effective features for NLP applications. In the future, we would like to investigate whether similar effects can be found from non-distributional or contextualized word embeddings.

References

Ben Athiwaratkun and Andrew Gordon Wilson. 2017. Multimodal word distributions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1645–1656.

E. Bruni, N.-K. Tran, and M. Baroni. 2014. Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)*, 49:1–47.

Gino Brunner, Yuyi Wang, Roger Wattenhofer, and Michael Weigelt. 2018. Natural language multitasking: Analyzing and improving syntactic saliency of hidden representations. *CoRR*, abs/1801.06024.

José Camacho-Collados and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *J. Artif. Intell. Res.*, 63:743–788.

A. Gladkova, A. Drozd, and S. Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8—-15.

Irina Higgins, David Amos, David Pfau, Sébastien Racanière, Loïc Matthey, Danilo J. Rezende, and Alexander Lerchner. 2018. Towards a definition of disentangled representations. *CoRR*, abs/1812.02230.

F. Hill, R. Reichart, and A. Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Sarthak Jain, Edward Banner, Jan-Willem van de Meent, Iain J Marshall, and Byron C. Wallace. 2018. Learning disentangled representations of texts with application to biomedical abstracts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4683–4693. Association for Computational Linguistics.

Durk P Kingma, Tim Salimans, and Max Welling. 2015. Variational dropout and the local reparameterization trick. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2575–2583. Curran Associates, Inc.

Maria Larsson, Amanda Nilsson, and Mikael Kågebäck. 2017. Disentangled representations for manipulation of sentiment in text. *CoRR*, abs/1712.10066.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

Bofang Li, Tao Liu, Zhe Zhao, Buzhou Tang, Aleksandr Drozd, Anna Rogers, and Xiaoyong Du. 2017. Investigating different syntactic context types and context representations for learning word embeddings. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2421–2431.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

Igor Melnyk, Cícero Nogueira dos Santos, Kahini Wadhawan, Inkit Padhi, and Abhishek Kumar. 2017. Improved neural text attribute transfer with non-parallel data. *CoRR*, abs/1711.09395.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Dmitry Molchanov, Arsenii Ashukha, and Dmitry P. Vetrov. 2017. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 2498–2507.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *In EMNLP*.

Karl Ridgeway and Michael C. Mozer. 2018. Learning deep disentangled embeddings with the f-statistic loss. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 185–194. Curran Associates, Inc.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *CoRR*, cs.CL/0306050.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proc. of EMNLP*.