A Comparative Study of Embedding Models in Predicting the Compositionality of Multiword Expressions

Navnita Nandakumar **Bahar Salehi** **Timothy Baldwin**
School of Computing and Information Systems
The University of Melbourne
Victoria 3010, Australia
nnandakumar@student.unimelb.edu.au {salehi.b,tbaldwin}@unimelb.edu.au

Abstract

In this paper, we perform a comparative evaluation of off-the-shelf embedding models over the task of compositionality prediction of multiword expressions ("MWEs"). Our experimental results suggest that character- and document-level models do capture some aspects of MWE compositionality and are effective at modelling varying levels of compositionality, but ultimately are not as effective as a simple word2vec baseline. However they have the advantage over word-level models that they do not require token-level identification of MWEs in the training corpus.

1 Introduction

In recent years, the study of the semantic idiomaticity of multiword expressions ("MWEs": Baldwin and Kim (2010)) has focused on *compositionality prediction*, a regression task involving the mapping of an MWE onto a continuous scale, representing its compositionality either as a whole or for each of its component words (Reddy et al., 2011; Ramisch et al., 2016; Cordeiro et al., to appear). In the case of *couch potato* "an idler who spends much time on a couch (usually watching television)", e.g., on a scale of $[0, 1]$ the overall compositionality may be judged to be 0.3, and the compositionality of *couch* and *potato* as 0.8 and 0.1, respectively. The main motivation for the study of compositionality is to better understand the semantic of the compound and the semantic relationships between the component words of the MWEs, which has applications in various information retrieval and natural language processing tasks (Venkatapathy and Joshi, 2006; Acosta et al., 2011; Salehi et al., 2015b).

Separately, there has been burgeoning interest in learning distributed representations of words and their meanings, starting out with word embeddings (Mikolov et al., 2013; Pennington et al., 2014) and now also involving the study of character- and document-level models (Baroni et al., 2014; Le and Mikolov, 2014; Bojanowski et al., 2017; Conneau et al., 2017). This work has been applied in part to predicting the compositionality of MWEs (Salehi et al., 2015a; Hakimi Parizi and Cook, 2018), work that this paper builds on directly, in performing a comparative study of the performance of a range of off-the-shelf representation learning methods over the task of MWE compositionality prediction.

Our contributions are as follows: (1) we show that, despite their effectiveness over a range of other tasks, recent off-the-shelf character- and document-level embedding learning methods are inferior to simple word2vec at modelling MWE compositionality; and (2) we demonstrate the utility of using paraphrase data in addition to simple lemmas in predicting MWE compositionality.

2 Related work

The current state-of-the-art in compositionality prediction involves the use of word embeddings (Salehi et al., 2015a). The vector representations of each component word (e.g. *couch* and *potato*) and the overall MWE (e.g. *couch potato*) are taken as a proxy for their respective meanings, and compositionality of the MWE is then assumed to be proportional to the relative similarity between each of the components and overall MWE embedding. However, word-level embeddings require token-level identification of each MWE in the training corpus, meaning that if the set of MWEs changes, the model needs to be retrained. This limitation led to research on character-level models, since character-level models can implic-

Navnita Nandakumar, Bahar Salehi and Timothy Baldwin. 2018. A Comparative Study of Embedding Models in Predicting the Compositionality of Multiword Expressions. In *Proceedings of Australasian Language Technology Association Workshop*, pages 71–76.

itly handle an unbounded vocabulary of component words and MWEs (Hakimi Parizi and Cook, 2018). There has also been work in the extension of word embeddings to document embeddings that map entire sentences or documents to vectors (Le and Mikolov, 2014; Conneau et al., 2017).

3 Embedding Methods

We use two character-level embedding models (fastText and ELMo) and two document-level models (doc2vec and infersent) to compare with word-level word2vec, as used in the state-of-the-art method of Salehi et al. (2015a). In each case, we use canonical pre-trained models, with the exception of word2vec, which must be trained over data with appropriate tokenisation to be able to generate MWE embeddings, as it treats words atomically and cannot generate OOV words.

3.1 Word-level Embeddings

Word embeddings are mappings of words to vectors of real numbers. This helps create a more compact (by means of dimensionality reduction) and expressive (by means of contextual similarity) word representation.

word2vec We trained word2vec (Mikolov et al., 2013) over the latest English Wikipedia dump.[1] We first pre-processed the corpus, removing XML formatting, stop words and punctuation, to generate clean, plain text. We then iterated through 1% of the corpus (following Hakimi Parizi and Cook (2018)) to find every occurrence of each MWE in our datasets and concatenate them, assuming every occurrence of the component words in sequence to be the compound noun (e.g. every *couch potato* in the corpus becomes *couchpotato*). We do this because instead of a single embedding for the MWE, word2vec generates separate embeddings for each of the component words, owing to the space between them. If the model still fails to generate embeddings for either the MWE or its components (due to data sparseness), we assign the MWE a default compositionality score of 0.5 (neutral). In the case of paraphrases, we compute the element-wise average of the embeddings of each of the component words to generate the embedding of the phrase.

[1] Dated 02-Oct-2018, 07:23

3.2 Character-level Embeddings

In a character embedding model, the vector for a word is constructed from the character n-grams that compose it. Since character n-grams are shared across words, assuming a closed-world alphabet,[2] these models can generate embeddings for OOV words, as well as words that occur infrequently. The two character-level embedding models we experiment with are fastText (Bojanowski et al., 2017) and ELMo (Peters et al., 2018), as detailed below.

fastText We used the 300-dimensional model pre-trained on Common Crawl and Wikipedia using CBOW. fastText assumes that all words are whitespace delimited, so in order to generate a representation for the combined MWE, we remove any spaces and treat it as a fused compound (e.g. *couch potato* becomes *couchpotato*). In the case of paraphrases, we use the same word averaging technique as we did in word2vec.

ELMo We used the ElmoEmbedder class in Python's allennlp library.[3] The model was pre-trained over SNLI and SQuAD, with a dimensionality of 1024.

Note that the primary use case of ELMo is to generate embeddings in context, but we are not providing any context in the input, for consistency with the other models. As such, we are knowingly not harnessing the full potential of the model. However, this naive use of ELMo is not inappropriate as the relative compositionality of a compound is often predictable from its component words only, even for novel compounds such as *giraffe potato* (which has a plausible compositional interpretation, as a potato shaped like a giraffe) vs. *couch intelligence* (where there is no natural interpretation, suggesting that it may be non-compositional).

3.3 Document-level Embeddings

Document-level embeddings aim to learn vector representations of documents (sentences or even paragraphs), to generate a representation

[2] Which is a safe assumption for languages with small-scale alphabetic writing systems such as English, but potentially problematic for languages with large orthographies such as Chinese (with over 10k ideograms in common use, and many more rarer characters) or Korean (assuming we treat each Hangul syllable as atomic).

[3] `options_file = https://bit.ly/2CInZPV, weight_file = https://bit.ly/2PvNqHh`

of its overall content in the form of a fixed-dimensionality vector. The two document-level embeddings used in this research are doc2vec (Le and Mikolov, 2014) and infersent (Conneau et al., 2017), as detailed below.

doc2vec We used the gensim implementation of doc2vec (Lau and Baldwin, 2016; Řehůřek and Sojka, 2010), pretrained on Wikipedia data using the word2vec skip-gram models pretrained on Wikipedia and AP News.[4]

infersent We used two versions of infersent of 300 dimensions, using the inbuilt `infersent.build_vocab_k_words` function to train the model over the 100,000 most popular English words, using: (1) GloVe (Pennington et al., 2014) word embeddings ("infersent$_{\text{GloVe}}$"); and (2) fastText word embeddings ("infersent$_{\text{fastText}}$").

4 Modelling Compositionality

In order to measure the overall compositionality of an MWE, we propose the following three broad approaches.

4.1 Direct Composition

Our first approach is to directly compare the embeddings of each of the component nouns with the embedding of the MWE via cosine similarity, in one of two ways: (1) pre-combine the embeddings for the component words via element-wise sum, and compare with the embedding for the MWE ("Direct$_{\text{pre}}$"); and (2) compare each individual component word with the embedding for the MWE, and post-hoc combine the scores via a weighted sum ("Direct$_{\text{post}}$"). Formally:

$$\begin{aligned}\text{Direct}_{\text{pre}} &= \cos(\mathbf{mwe}, \mathbf{mwe_1} + \mathbf{mwe_2}) \\ \text{Direct}_{\text{post}} &= \alpha \cos(\mathbf{mwe}, \mathbf{mwe_1}) + \\ &\quad (1-\alpha)\cos(\mathbf{mwe}, \mathbf{mwe_2})\end{aligned}$$

where: $\mathbf{mwe}$, $\mathbf{mwe_1}$, and $\mathbf{mwe_2}$ are the embeddings for the combined MWE, first component and second component, respectively;[5] $\mathbf{mwe_1}+\mathbf{mwe_2}$ is the element-wise sum of the vectors of each of the component words of the MWE; and $\alpha \in [0, 1]$ is a scalar which allows us to vary the weight of the respective components in predicting the compositionality of the compound. The intuition behind both of these methods is that if the MWE appears in similar contexts to its components, then it is compositional.

Emb. method	Direct$_{\text{pre}}$	Direct$_{\text{post}}$
word2vec	**0.684**	**0.710** ($\alpha = 0.3$)
fastText	0.223	0.285 ($\alpha = 0.3$)
ELMo	0.056	0.399 ($\alpha = 0.0$)
doc2vec	−0.049	0.025 ($\alpha = 0.0$)
infersent$_{\text{GloVe}}$	0.413	0.500 ($\alpha = 0.5$)
infersent$_{\text{infersent}}$	0.557	0.610 ($\alpha = 0.5$)

Table 1: Pearson correlation coefficient for compositionality prediction results on the REDDY dataset.

4.2 Paraphrases

Our second approach is to calculate the similarity of the MWE embedding with that of its paraphrases, assuming that we have access to paraphrase data.[6] We achieve this using the following three formulae:

$$\text{Para\_first} = \cos(\mathbf{mwe}, \mathbf{para_1})$$

$$\text{Para\_all}_{\text{pre}} = \cos(\mathbf{mwe}, \sum_i \mathbf{para_i})$$

$$\text{Para\_all}_{\text{post}} = \frac{1}{N}\sum_{i=1}^{N} \cos(\mathbf{mwe}, \mathbf{para_i})$$

where $\mathbf{para_1}$ and $\mathbf{para_i}$ denote the embedding for the first (most popular) and i-th paraphrases, respectively.

We apply this method to RAMISCH only, since REDDY does not have any paraphrase data (see Section 5.1 for details).

4.3 Combination

Our final approach ("Combined") is based on the combination of the direct composition and paraphrase methods, as follows:

$$\begin{aligned}\text{Combined} = \beta \max\big(&\text{Direct}_{\text{pre}}, \text{Direct}_{\text{post}}\big) + \\ (1-\beta) \max\big(&\text{Para\_first}, \text{Para\_all}_{\text{pre}}, \\ &\text{Para\_all}_{\text{post}}\big)\end{aligned}$$

where $\beta \in [0, 1]$ is a scalar weighting factor to balance the effects of the two methods. The choice

[4] https://github.com/jhlau/doc2vec/blob/master/README.md

[5] Noting that all MWEs are binary in our experiments, but equally that the methods generalise trivially to larger MWEs.

[6] Each paraphrase shows an interpretation of the compound semantics. e.g. *olive oil* is "oil from olive"

Emb. method	Direct$_{pre}$	Direct$_{post}$	Para_first	Para_all$_{pre}$	Para_all$_{post}$	Combined
word2vec	**0.667**	**0.731** ($\alpha = 0.7$)	**0.714**	**0.822**	**0.880**	**0.880** ($\beta = 0.0$)
fastText	0.395	0.446 ($\alpha = 0.7$)	0.569	0.662	0.704	0.704 ($\beta = 0.0$)
ELMo	0.139	0.295 ($\alpha = 0.0$)	0.367	0.642	0.664	0.669 ($\beta = 0.2$)
doc2vec	−0.146	0.048 ($\alpha = 1.0$)	0.405	0.372	0.401	0.419 ($\beta = 0.3$)
infersent$_{GloVe}$	0.321	0.427 ($\alpha = 0.7$)	0.639	0.704	0.741	0.774 ($\beta = 0.5$)
infersent$_{fastText}$	0.274	0.380 ($\alpha = 0.8$)	0.615	0.781	0.783	0.783 ($\beta = 0.0$)

Table 2: Pearson correlation coefficient for compositionality prediction results on the RAMISCH dataset.

of the max operator here to combine the sub-methods for each of the direct composition and paraphrase methods is that all methods tend to underestimate the compositionality (and empirically, it was superior to taking the mean).

5 Experiments

5.1 Datasets

We evaluate the models on the following two datasets, which are comprised of 90 English binary noun compounds each, rated for compositionality on a scale of 0 (non-compositional) to 5 (compositional). In each case, we evaluate model performance via the Pearson's correlation coefficient (r).

REDDY This dataset contains scores for the compositionality of the overall MWE, as well as that of each component word (Reddy et al., 2011); in this research, we use the overall compositionality score of the MWE only, and ignore the component scores.

RAMISCH Similarly to REDDY, this dataset contains scores for the overall compositionality of the MWE as well as the relative compositionality of each of its component words, in addition to paraphrases suggested by the annotators, in decreasing order of popularity (Ramisch et al., 2016); in this research, we use the overall compositionality score and paraphrase data only.

5.2 Results and Discussion

The results of the experiments on REDDY and RAMISCH are presented in Tables 1 and 2, respectively. In this work, we simplistically present the results for the best α and β values for each method over a given dataset, meaning we are effectively peaking at our test data. Sensitivity of the α hyper-parameter is shown in Figures 1 and 2, for the REDDY and RAMISCH datasets, respectively.

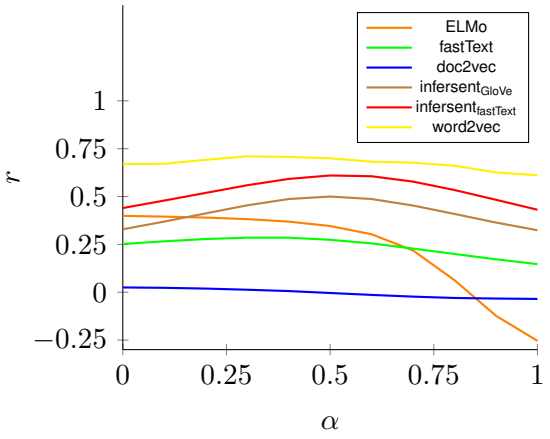

Figure 1: Sensitivity analysis of α (REDDY)

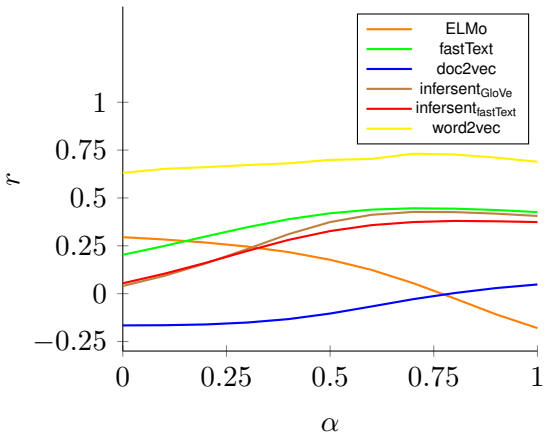

Figure 2: Sensitivity analysis of α (RAMISCH)

The first observation to be made is that none of the pretrained models match the state-of-the-art method based on word2vec, despite the simplicity of the method. ELMo and doc2vec in particular perform worse than expected, suggesting that their ability to model non-compositional language is limited. Recall, however, our comment about

using ELMo naively, in not including any context when generating the embeddings for the component words and, more importantly, the overall MWE. The results show that doc2vec performs better when representing paraphrases, and struggles with compounds without sentential context.

In Table 1, we find $\text{Direct}_{\text{post}}$ to produce a higher correlation in all cases, with α ranging from 0.0 to 0.5, suggesting that the second element (= head) contributes more to the overall compositionality of the MWE than the first element (= modifier); this is borne out in Figure 1.

In Table 2, on the other hand, we find that, with the exception of ELMo, the α values favour the modifier of the MWE over the head (i.e. $\alpha > 0.5$; also seen in Figure 2), implying that the former is more significant in predicting the compositionality of the MWE. The reason for the mismatch between the two datasets is not immediately clear, other than the obvious data sparsity.

We also see that the paraphrases achieve a higher correlation across all models, suggesting this is a promising direction for future study. The low β values for Combined also confirm that the paraphrase methods have greater predictive power than the direct composition methods. Among the paraphrase experiments, we find that $\text{Para\_all}_{\text{post}}$— the average of the similarities of the MWE with each of its paraphrases — consistently achieves the best results. We hypothesize that the paraphrases provide additional information regarding the compounds that further help determine their compositionality.

6 Conclusions and Future Work

This paper has investigated the application of a range of embedding generation methods to the task of predicting the compositionality of an MWE, either directly based on the MWE and its component words, or indirectly based on paraphrase data for the MWE. Our results show that modern character- and document-level embedding models are inferior to the simple word2vec approach at the task. We also show that paraphrase data captures valuable data regarding the compositionality of the MWE.

Since we have achieved such promising results with the paraphrase data, it might be interesting to consider other possible settings in future tests. While none of the other approaches could outperform word2vec, it is useful to note that they were pretrained and, as such, did not require any manipulation of the training corpus in order to generate vector embeddings of the MWEs. This means they can be applied to new datasets without the need for retraining and are, therefore, more robust.

In future work, we intend to train the models used in our study on a fixed corpus, to compare their performance in a more controlled setting. We will also do proper tuning of the hyperparameters over held-out data, and plan to experiment with other languages.

References

Otavio Acosta, Aline Villavicencio, and Viviane Moreira. 2011. Identification and treatment of multiword expressions applied to information retrieval. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*. Portland, USA, pages 101–109.

Timothy Baldwin and Su Nam Kim. 2010. Multiword expressions. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of Natural Language Processing*, CRC Press, Boca Raton, USA. 2nd edition.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*. Baltimore, USA, pages 238–247.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 670–680.

Silvio Cordeiro, Aline Villavicencio, Marco Idiart, and Carlos Ramisch. to appear. Unsupervised compositionality prediction of nominal compounds. *Computational Linguistics* .

Ali Hakimi Parizi and Paul Cook. 2018. Do character-level neural network language models capture knowledge of multiword expression compositionality? In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*. pages 185–192.

Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Proceedings of the 1st Workshop on Representation Learning for NLP*. pages 78–86.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*. Beijing, China, pages 1188–1196.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at the International Conference on Learning Representations*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. pages 2227–2237.

Carlos Ramisch, Silvio Cordeiro, Leonardo Zilio, Marco Idiart, and Aline Villavicencio. 2016. How naked is the naked truth? a multilingual lexicon of nominal compound compositionality. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. pages 156–161.

Siva Reddy, Diana McCarthy, and Suresh Manandhar. 2011. An empirical study on compositionality in compound nouns. In *Proceedings of 5th International Joint Conference on Natural Language Processing*. pages 210–218.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. pages 45–50.

Bahar Salehi, Paul Cook, and Timothy Baldwin. 2015a. A word embedding approach to predicting the compositionality of multiword expressions. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 977–983.

Bahar Salehi, Nitika Mathur, Paul Cook, and Timothy Baldwin. 2015b. The impact of multiword expression compositionality on machine translation evaluation. In *Proceedings of the NAACL HLT 2015 Workshop on Multiword Expressions*. Denver, USA, pages 54–59.

Sriram Venkatapathy and Aravind K Joshi. 2006. Using information about multi-word expressions for the word-alignment task. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*. pages 20–27.