Neural word embeddings with multiplicative feature interactions for tensor-based compositions

Joo-Kyung Kim[†], **Marie-Catherine de Marneffe**[‡], **Eric Fosler-Lussier**[†]
[†]Department of Computer Science and Engineering,
[‡]Department of Linguistics,
The Ohio State University, Columbus, OH 43210, USA
kimjook@cse.ohio-state.edu, mcdm@ling.ohio-state.edu,
fosler@cse.ohio-state.edu

Abstract

Categorical compositional distributional models unify compositional formal semantic models and distributional models by composing phrases with tensor-based methods from vector representations. For the tensor-based compositions, Milajevs et al. (2014) showed that word vectors obtained from the continuous bag-of-words (CBOW) model are competitive with those from co-occurrence based models. However, because word vectors from the CBOW model are trained assuming additive interactions between context words, the word composition used for the training mismatches to the tensor-based methods used for evaluating the actual compositions including pointwise multiplication and tensor product of context vectors. In this work, we show whether the word embeddings from extended CBOW models using multiplication or tensor product between context words, reflecting the actual composition methods, can show better performance than those from the baseline CBOW model in actual tasks of compositions with multiplication or tensor-based methods.

1 Introduction

In recent years, there has been a surge of interest in using word vectors for modeling semantics. Mikolov et al. (2013a,b) introduced `word2vec` that includes the continuous bag-of-words (CBOW) model and the skip-gram model.[1] These models have been most widely used for generating word vectors to be used for word related tasks because of their efficient but still effective architectures. The CBOW model takes the mean vector of projections of the context words and use it to predict the target word as the following objective function:[2]

$$\frac{1}{T}\sum_{t=1}^{T}\ln p\left(w_t\left|\frac{1}{2c}\sum_{-c\leq j\leq c, j\neq 0}p_{t+j}\right.\right), \quad (1)$$

where T is the total number of words in a corpus, w_t is the tth word, p_t is the tth word vector, and c is the half window size.

Milajevs et al. (2014) showed that the word vectors generated from the CBOW model are competitive with those from co-occurrence based models for both simple arithmetic compositions and tensor-based compositions for categorical compositional distributional models (Coecke et al., 2010).[3]

Categorical compositional distributional models represent compositional semantics with algebra of Pregroup by representing each grammatical reduction as a linear map in vector spaces (Coecke et al., 2010; Kartsaklis et al., 2012). For example, *cats like milk* consists of a subject noun, a transitive verb requiring a subject and an object, and an object noun, respectively. In Pregroup grammar, the types of the three words in this example are n, $\left(n^r s n^l\right)$, and n, respectively, where n is a noun, n^r can be combined with a n in the left, n^l can be combined with a n in the right, and s is a declarative statement. Then,

[1] `https://code.google.com/p/word2vec`

[2] Although sum is used in Mikolov et al. (2013a), the current version of `word2vec` implementation uses mean.

[3] Although Milajevs et al. (2014) described that the skip-gram model was used to generate the word vectors, the CBOW model was actually used in their work.

Proceedings of NAACL-HLT 2015, pages 143–150,
Denver, Colorado, May 31 – June 5, 2015.

they can be reduced to represent the entire phrase with single entity as follows:

$$n\left(n^r s n^l\right) n \rightarrow 1 s n^l n \rightarrow 1 s 1 \rightarrow s \qquad (2)$$

In the reduction, n^r is composed with the left n resulting in an identity element, 1. Then, n^l is composed with the right n resulting in another 1. Because 1 is an identity element, $1s1$ is reduced to s.

Since there is no specification of actual implementation of the composition in categorical compositional distributional models, different composition methods have been introduced; they are reviewed in Section 2. However, there are few studies about the vector representation of single words regarding those compositions.

One issue of using the word vectors from the CBOW model as the constituent vectors for tensor-based composition is that their assumptions of the composition are different. Word embeddings of the CBOW model are trained with an additive context composition, which is the mean of the context projection. However, most tensor-based compositions use point-wise multiplication or tensor product as composition operators. This means that there is a mismatch between the composition method used for the training of the underlying word vectors and the actual composition methods we evaluate.

To alleviate the mismatch, we introduce extensions of the CBOW model with multiplicative interactions between word projections to obtain word embeddings more suitable for the tensor-based compositions. For four datasets, evaluating different types of compositions, we show that those extensions of the CBOW model improve the performance of the actual composition tasks with multiplication or tensor product operations.

2 Tensor-based compositions

Prior to discussing the modification to the CBOW algorithm, we review different composition methods used in the literature (Table 1).

Addition and *Multiplication* are compositions by point-wise addition and multiplication, respectively (Mitchell and Lapata, 2008). They can be done simply without any other information, but they cannot reflect word orders and grammatical structures.

Mitchell and Lapata (2008, 2009) showed that composition by multiplication can be more effective than composition by addition because additive models compose by considering the content altogether whereas multiplicative models focus on the content relevant to the composition by scaling each element of one with the strength of the corresponding element of the other. Using multiplication as the composition method could be unstable in the previous work because multiplication with zero or negative values changes the value abruptly (Mitchell and Lapata, 2009). In our models, however, these instability issues could be alleviated since the training model adapt the constituent word vectors to be proper for the composition by multiplication. Mitchell and Lapata (2010) also showed that the tensor product is effective to represent composition because it allows the interactions between different features in different vectors whereas point-wise multiplication can interact with only the same feature in different vectors. Therefore, we also examine an extension of the CBOW model using tensor product for modeling local context.

There are neural network models using multiplicative interactions in the architectures. Sum-Product Networks use layer-wise multiplicative interactions (Poon and Domingos, 2011; Cheng et al., 2014) and multiplicative recurrent neural networks use multiplication of hidden state outputs from previous time step with the current word projections (Sutskever et al., 2011; Irsoy and Cardie, 2014). These approaches capture multiplicative interactions with hidden layer outputs. Our approach instead utilizes multiplicative interactions in the training of the CBOW model, making the embedded vector spaces more in tune with the compositions of end tasks.

The third to the last composition methods of Table 1 shows tensor-based composition methods for representing phrases consist of subjects, transitive verbs, and objects in categorical compositional distributional models. $\overline{verb} = \sum_i \overrightarrow{Sbj_i} \otimes \overrightarrow{Obj_i}$ represents a verb with the subjects and the objects of the verb across the corpus. The subject and the object of each transitive verb required for calculating $\overline{verb}$ are identified from the dependency tree of PukWaC 1.0 dataset, which consists of web documents in .uk domain crawled with the medium-frequency words from the British National Corpus (BNC) (Burnard,

Method	Phrase	Composition formula	Reference
Addition	$w_1 w_2 ... w_n$	$\overrightarrow{w_1} + \overrightarrow{w_2} + ... + \overrightarrow{w_n}$	Mitchell and Lapata (2008)
Multiplication		$\overrightarrow{w_1} \odot \overrightarrow{w_2} \odot ... \odot \overrightarrow{w_n}$	
Relational	Sbj Verb Obj	$\overline{verb} \odot (\overrightarrow{Sbj} \otimes \overrightarrow{Obj})$	Grefenstette and Sadrzadeh (2011a)
Kronecker		$\widetilde{verb} \odot (\overrightarrow{Sbj} \otimes \overrightarrow{Obj})$	Grefenstette and Sadrzadeh (2011b)
Copy sbj.	Sbj Verb Obj	$\overrightarrow{Sbj} \odot (\overline{Verb} \times \overrightarrow{Obj})$	Kartsaklis et al. (2012)
Copy obj.		$\overrightarrow{Obj} \odot (\overline{Verb}^\top \times \overrightarrow{Sbj})$	
Frob. add.	Sbj Verb Obj	$(\overrightarrow{Sbj} \odot (\overline{Verb} \times \overrightarrow{Obj})) + (\overrightarrow{Obj} \odot (\overline{Verb}^\top \times \overrightarrow{Sbj}))$	Kartsaklis and Sadrzadeh (2014)
Frob. mult.		$(\overrightarrow{Sbj} \odot (\overline{Verb} \times \overrightarrow{Obj})) \odot (\overrightarrow{Obj} \odot (\overline{Verb}^\top \times \overrightarrow{Sbj}))$	
Frob. outer		$(\overrightarrow{Sbj} \odot (\overline{Verb} \times \overrightarrow{Obj})) \otimes (\overrightarrow{Obj} \odot (\overline{Verb}^\top \times \overrightarrow{Sbj}))$	

Table 1: Tensor-based composition methods (Milajevs et al., 2014).

2007) as the seeds (Baroni et al., 2009; Johansson, 2007). [4] $\widetilde{verb} = \overrightarrow{verb} \otimes \overrightarrow{verb}$ represents a verb as the tensor product of the corresponding verb vector. Those methods consider the relations between transitive verbs and their subjects and objects. Therefore, we can represent their compositions more effectively. Recursive neural tensor networks also use tensor product information in the recursive composition (Socher et al., 2013), but they require training labels and only support binary compositions.

Relational and *Kronecker* represent each phrase by the multiplication of the verb matrix to the tensor product of the subject and the object (Grefenstette and Sadrzadeh, 2011a,b). Although they can represent interactions between subjects and objects as well as the verbs, it is difficult to compose them with other phrases in a uniform way since the result dimensionality is the square of the original vectors. In addition, dealing with large dimensional tensors is not very scalable.

The fifth to the last composition methods use Frobenius operators for the compositions (Kartsaklis et al., 2012), which can resolve the dimensionality issues by maintaining the original dimensionality through matrix-vector multiplication. In *Copy subject*, the verb matrix $\overline{verb}$ is multiplied with the object vector and then composed with the subject vector by point-wise multiplication. *Copy object* is opposite in terms of the positions of the subject and the object. These two methods are different ways of diagonal placement of a plane into a cube (Kartsaklis et al., 2012). The last three methods, *Frobenius addition*, *multiplication*, and *outer product*, represent different combinations of *Copy subject* and *Copy object* (Kartsaklis and Sadrzadeh, 2014).

[4] Available at `http://wacky.sslmit.unibo.it/doku.php?id=corpora`.

3 Extending the CBOW model with multiplicative interactions between word projections

As briefly discussed in the introduction, the CBOW model is an additive model in terms of the composition since the mean of the context word projections is used to predict the target word. As many composition methods in Table 1 use multiplication or tensor product as the composition operators, if these operators are used to compose the contexts in the CBOW model, then the training process can optimize the model to consider their word embeddings to be composed with those multiplicative operations. Therefore, we can train word embeddings that are more suitable for the composition methods that we are evaluating.

In the CBOW model, the point-wise mean of the word projections is used to predict the target word as shown in Equation 1. In addition to the baseline, we experimented with adding different multiplicative terms as shown in Table 2. The added terms are selected to reflect the operations of composition methods in Table 1 and their combinations. In the expressions, p_i is the projection of the ith input context word and c is the size of the context window, which is the number of neighboring words used as the input for each direction.

The second model, *mult*, uses only the multiplication of projections, which best fits to the composition by point-wise multiplication. The third and the

	Type	Expression
1	mean (baseline, Milajevs et al. (2014))	$\sum_{-c \leq i \leq c, i \neq 0} p_i / 2c$
2	pointwise multiplication	$\prod_{-c \leq i \leq c, c \neq 0} p_i$
3	mean + pointwise multiplication	mean $+ \prod_{-c \leq i \leq c, c \neq 0} p_i$
4	concat{mean, pointwise multiplication}	concat{mean, $\prod_{-c \leq i \leq c, c \neq 0} p_i$}
5	mean + projection of p_{i-1} and p_{i+1}	mean $+ W_p$concat$\{p_{i-1}, p_{i+1}\}$
6	projection of tensor product of p_{i-1} and p_{i+1}	$W_{tp}(p_{i-1} \otimes p_{i+1})$
7	mean + projection of tensor product of p_{i-1} and p_{i+1}	mean $+ W_{tp}(p_{i-1} \otimes p_{i+1})$

Table 2: Different outputs of the projection layer. p_i is the projection of the ith input context word, c is the size of the context window, and W_p and W_tp are projection matrices.

fourth models evaluate the performance when both the additive and multiplicative interactions are used together since their combination has been shown to be effective (Mitchell and Lapata, 2008). The third model adds the additive terms and multiplicative terms whereas the fourth model concatenates these terms so that they influence the output separately.

In the fifth to the last models, we try to further use the information from p_{i-1} and p_{i+1}, which are the projections of the nearest neighbor words of the ith target word in the training corpus. The fifth model concatenates p_{i-1} and p_{i+1} and project to the original dimension with a projection matrix W_p. This result is added to the baseline model so that information from the nearest words considering the order can be used to estimate the target. W_p is also updated during the training.

In the sixth model, since the tensor-based compositions are used as Table 1 and they can represent multiplicative interactions between different features, we use the tensor product of the projections of (p_{i-1} and p_{i+1}). The tensor product output is also projected to the original dimensionality by multiplying a projection matrix W_{tp}, which is also updated during the training. Although this model can use more powerful interactions of neighbor words, it can only use the information from the nearest neighbor words and it cannot use two word sentences in the training corpus for the training. To deal with these issues, in the last model, we combine the mean with the projection of the tensor product.

4 Experiment results

To evaluate the five different CBOW-based models proposed in Section 3, we use the following datasets: similarity of transitive verbs with multiple senses from Grefenstette and Sadrzadeh (2011a), three-word sentence similarity from Kartsaklis and Sadrzadeh (2014), paraphrase detection from Dolan et al. (2013), and dialog act tagging for the Switchboard corpus (Godfrey et al., 1992) from Stolcke et al. (2000). These are all the datasets evaluated in Milajevs et al. (2014)'s work as well. Each phrase in the first two datasets is fixed as a subject, a transitive verb, and an object whereas the length of each phrase in the last two datasets is arbitrary.

There are several differences between our word vectors and the ones used in Milajevs et al. (2014). First, we use BNC as the training set while Milajevs et al. (2014) use pretrained word vectors from `word2vec` that are trained using GoogleNews dataset. To reduce the size of projection matrices, all the words are lower-cased and words occurring 20 times or less are converted to the words' POS tags. Second, instead of negative sampling, our models use hierarchical softmax as the objective function, where each word is represented as a leaf node of Huffman tree since hierarchical softmax is better for training with infrequent words (Mikolov et al., 2013b). Third, we use gradient clipping for more stable training since gradient can be fluctuating when the projections are multiplied. All the other parameters for the training are the same as those used for Milajevs et al. (2014)'s experiments.

Using the mean as the network combination function can be considered a reimplementation of Milajevs et al. (2014)'s system subject to the changes mentioned above. We trained the CBOW-based models and obtained 300 dimensional word vectors, which are with the same dimensionality used in Mikolov et al. (2013a,b); Milajevs et al. (2014).

Task	Method	(Milajevs et al., 2014)	mean	mult	mean + mult	concat {mean,mult}	mean + nbr_prj	nbr_outer_prj	mean + nbr_outer_prj
Similarity of transitive verbs	Verb only	0.107	0.130	0.014	0.136	0.204	0.187	0.072	**0.250**
	Addition	0.149	0.066	0.012	0.046	-0.030	0.100	0.111	**0.145**
	Multiplication	0.095	0.160	**0.249**	0.058	0.219	0.113	0.050	0.204
	Kronecker	0.117	0.160	0.160	0.121	0.229	0.168	0.047	**0.245**
	Relational	0.362	0.330	0.276	0.319	0.280	0.344	0.316	**0.365**
	Copy sbj.	0.131	0.249	0.064	0.262	0.209	0.262	0.168	**0.290**
	Copy obj.	0.456	0.302	0.361	0.329	**0.382**	0.300	0.371	0.322
	Frob. add.	0.359	0.337	0.293	0.345	0.288	0.349	0.250	**0.355**
	Frob. mult.	0.239	0.270	0.252	0.255	0.189	0.293	0.196	**0.309**
	Frob. outer.	0.375	0.330	0.275	0.339	0.351	0.329	0.293	**0.387**
Sentence similarity	Verb only	0.561	0.528	0.360	0.520	0.531	0.527	0.260	**0.536**
	Addition	0.689	0.728	0.572	0.738	**0.770**	0.722	0.401	0.706
	Multiplication	0.341	0.062	**0.625**	0.178	0.440	0.110	0.269	0.220
	Kronecker	0.561	0.206	**0.623**	0.277	0.501	0.203	0.003	0.457
	Relational	0.618	0.505	**0.665**	0.540	0.527	0.525	0.157	0.574
	Copy sbj.	0.405	0.390	0.453	0.353	0.436	0.396	0.139	**0.454**
	Copy obj.	0.655	0.481	**0.607**	0.487	0.500	0.488	0.190	0.510
	Frob. add.	0.585	0.489	**0.610**	0.407	0.528	0.439	0.210	0.501
	Frob. mult.	0.387	0.211	**0.608**	0.323	0.419	0.335	0.065	0.349
	Frob. outer.	0.622	0.504	**0.664**	0.510	0.544	0.524	0.165	0.569

Table 3: Spearman's ρ on the similarity of transitive verbs with multiple senses (top) and three-word sentence similarity (bottom). The mean column can be considered an implementation of the Milajevs et al. (2014)'s model on the BNC corpus.

4.1 Fixed phrases (three-word)

Table 3 shows the experiment results for the three-word phrases. The first column represents the two evaluation tasks, the second column is the composition methods described in Table 1, and the third column shows the results of neural word embeddings (NWE) from previous work (Milajevs et al., 2014).[5] Bold entries in the table indicate the highest scores among our models.

In the datasets, human annotators rated each phrase pair for semantic similarity (from 1 "no similarity" to 7 "high similarity"). As each unique phrase pair is judged by multiple people, following Milajevs et al. (2014), we took the mean of the ratings to set the rating of each unique pair. Scores in the table entries are Spearman's ρs. A high value of Spearman's ρ in the table means that the similarity of the composed phrases in the vector space is highly correlated with the semantic similarity of the phrases judged by humans. Therefore, if a model shows high scores, it reflects that the model is good at representing the semantics for those short phrases.

4.1.1 Similarity of transitive verbs

The top of Table 3 shows the results for the similarity of 199 three-word phrase (subject, transitive verb, and object) pairs introduced in Grefenstette and Sadrzadeh (2011a).[6] In each phrase pair, the transitive verbs are the same but the subjects and the objects are different for each other. We try to identify the senses of a transitive verb with the different contexts. For example, "meet" is a verb with multiple senses. If the given subject is "system" and the object is "specification", "meet" would be semantically closer to "satisfy" than "visit". Then, given "system meets specification" and "system satisfies specification" as a pair, the judge would give a high rating for the similarity of the verbs.

Our results were not consistently better than Milajevs et al. (2014)'s results. However, considering that the model used for the previous work and our baseline (*mean*) are similar CBOW models, the performance difference would mainly due to the different training sets (GoogleNews and BNC). Among our models, adding tensor product result to the mean (*mean+nb_outer_prj*) showed the best performance in most types of compositions. Interestingly, the

[5]The word vectors are available at `https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit`.

[6]Available at `http://www.cs.ox.ac.uk/activities/compdistmeaning/GS2011data.txt`.

power seems to come from the combination of mean and the tensor product, as both individually perform worse than the combination. For this dataset, as the verbs are the same for both phrases in each pair, the subjects and the objects play important roles for the verb disambiguation. When a transitive verb is an ith word in a sentence denoted as w_i, in many cases, the subject and the object are w_{i-1} and w_{i+1}, respectively. Since tensor product of the $i-1$th word projection and the $i+1$th word projection can represent multiplicative interactions between different features of the two words, considering the tensor product of those projections in the model could be helpful to identify the transitive verbs.

4.1.2 Similarity of three-word phrases

The bottom of Table 3 shows the evaluation results on the similarity of 109 three-word phrase pairs with human judged ratings from (Kartsaklis and Sadrzadeh, 2014).[7] For example, the similarities of two sentences like "programme offer support" and "service provide help" are evaluated.

In this evaluation, considering the interleaved words with tensor product (*mean+nb_outer_prj*) still showed better performance than the baseline (*mean*) for the most composition methods except addition. However, the multiplication only model, *mult*, showed the best performance in most cases except when verb only or addition were used as the composition methods.

4.2 Arbitrary length phrases

The three-word phrases in the previous section are useful for the evaluation of the tensor-based compositions since we do not need to care about the structural variations of the phrases. However, we would be more interested in phrases where the lengths are not fixed. As each phrase can have different length, we cannot use the tensor-based compositions used for the fixed-length phrases. Therefore, we evaluated the composition of each phrase by only using point-wise addition and multiplication.

Table 4 shows the accuracies of classification tasks given arbitrary length phrases as the inputs. The results evaluate whether composition of arbitrary length phrases can be well represented with the word vectors from the proposed models.

4.2.1 Paraphrase detection

The top of Table 4 shows the binary classification accuracies on the Microsoft Research Paraphrase Corpus (Dolan et al., 2013), which consists of arbitrary length phrase pairs. In this dataset, each phrase pair comes with a binary label: 1 if the phrases were judged to be paraphrases, 0 otherwise. The minimum, mean, and maximum lengths of the phrases in the training set are 6, 19.8, and 35, respectively.

With this dataset, we can evaluate if our models work well for representing general phrases. Following the setting of Milajevs et al. (2014)'s work, we trained a linear binary classifier on 2000 phrase pairs and tested on 1726 phrase pairs. The classifier is trained to find the threshold of cosine similarity deciding if two phrases are paraphrases or not.

Comparing to the baseline CBOW model, there were no significant gain in the proposed models for the composition by addition. However, using multiplication of the projections (*mult*) showed significantly better performance when composed by multiplication, and started to show statistical insignificance to additive composition methods when tested by McNemar's test with p-value 0.05.

4.2.2 Dialog act tagging

The bottom of Table 4 shows the classification accuracies of dialog act tagging (Stolcke et al., 2000) on the Switchboard corpus (Godfrey et al., 1992). Switchboard is a collection of about 2400 telephone dialogs among 543 speakers in the United States. Each utterance is assigned one of 42 dialog-act tags, which summarize syntactic, semantic and pragmatic information about the turns (e.g., yes/no question, yes answer, agree).[8] The minimum, mean, and maximum lengths of the phrases in the training set are 0, 34.1, and 549, respectively. Zero length phrases exist because of the preprocessing, and they are ignored.

The task in this section is identifying the dialog act tags from given utterances. Following Milajevs and Purver (2014); Milajevs et al. (2014), we used the first 1115 utterances as the training set and the

[7] Available at `http://www.cs.ox.ac.uk/activities/compdistmeaning/emnlp2013_turk.txt`.

[8] The tags are described in `http://web.stanford.edu/~jurafsky/ws97/manual.august1.html`.

Task	Method	(Milajevs et al., 2014)	mean	mult	mean + mult	concat {mean,mult}	mean + nbr_prj	nbr_outer_prj	mean + nbr_outer_prj
Paraphrase detection	Addition	0.73	0.686	0.665	**0.690**	0.688	0.689	0.684	0.688
	Multiplication	0.42	0.393	**0.652**	0.388	0.587	0.387	0.412	0.371
Dialog act tagging	Addition	0.63	**0.638**	0.636	0.633	0.636	0.636	0.565	0.626
	Multiplication	0.58	0.522	**0.606**	0.593	0.515	0.581	0.573	0.598

Table 4: Accuracies on the paraphrase detection (top) and the dialog act tagging (bottom). The mean column can be considered an implementation of the Milajevs et al. (2014)s model on our training set.

following 19 utterances as the test set. We also concatenated utterances separated by an interruption by the other person (Webb et al., 2005), and we removed disfluency markers and punctuation signs. Once we have the vectors composed by either addition or multiplication for all of the utterances in the training set, the vector dimensionality is reduced to 50 by Singular Value Decomposition (SVD) and a k-nearest-neighbor classifier (k=5) is used to identify the dialog act tags.[9] The baseline (*mean*) model showed the best performance for the composition by addition and the *mult* model was the best for the composition by multiplication, but the differences were insignificant in this case.

The results on both evaluation for arbitrary length phrases support that matching the composition of contexts for the training of constituent word vectors with the actual composition methods shows better or competitive performance.

5 Discussion

We showed the experiment results on seven types of word vectors trained using different composition methods. Overall, we can see that multiplicative interactions in the CBOW models can help representing compositions that are multiplicative in nature.

Using only the multiplication of projections showed significant improvement for all the evaluated datasets when the phrases are composed with multiplications. Because the composition used for the training of word vectors is matching to the actual evaluated compositions, we can think that the word vectors are trained to represent their multiplications properly. One evidence is that the mean of word vectors of the $mult$ model is around 0.12 while the means of the other models are around 0. Since there are fewer negative elements in the word vectors of the $mult$ model, the composition by multiplication produces relatively more positive values. This possibly gives more stable results when used in multiplication-based compositions since fluctuations of the composition by multiplication with negative values is reduced. In the task of transitive verb disambiguation, since the interactions between non-adjacent subjects and objects are important, having their tensor product as a term in the model (*mean+nbr_outer_prj*) was noticeably helpful. In the task of three-word phrase similarity, using the tensor product as a term still showed better performance than using the models of *mean* and *mean+mult* in most cases except when the phrases are composed with addition. Interestingly, however, the model with only multiplication showed the best performance for most of the compositions by multiplication and tensor product.

In summary, for better representation of phrase compositions, we showed that it can be helpful to train the word embedding models by composing the input contexts of the model to be similar to the actual composition methods to be used because the word vectors are adjusted to more properly represent the composition by the composition method used. Specifically, using point-wise multiplication in the training model consistently showed better performance when the actual composition is also multiplication. The *mean+nbr_outer_prj* model, which is with the combination of mean and tensor product also showed better or similar performance for tensor-based composed phrases compared to the *mean* model and the *mean+mult* model.

One issue is that we used the word vectors of targets's neighbors to obtain tensor product terms. Since only the compositions of subjects, verbs, and objects are evaluated, we can expect better performance if only tensor products of subject-object pairs

[9] We used scikit-learn (Pedregosa et al., 2011) to run SVD and k-NN classifiers.

are used as the tensor product terms. As future work, An in-depth analysis of the strengths and weaknesses of each approach would be helpful to gain more insights about the patterns we see in the results.

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. The WaCky Wide Web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226, 2009.

Lou Burnard, editor. *Reference Guide for the British National Corpus*. Research Technologies Service at Oxford University Computing Services, 2007.

Wei-Chen Cheng, Stanley Kok, Hoai Vu Pham, Hai Leong Chieu, and Kian Ming A. Chai. Language modeling with sum-product networks. In *Proceedings of Interspeech*, 2014.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, 36:345–384, 2010.

Bill Dolan, Chris Brockett, and Chris Quirk. Microsoft research paraphrase corpus, 2013.

John J Godfrey, Edward C Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 517–520, 1992.

Edward Grefenstette and Mehrnoosh Sadrzadeh. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404, 2011a.

Edward Grefenstette and Mehrnoosh Sadrzadeh. Experimenting with transitive verbs in a discocat. In *Proceedings of the GEMS 2011 Workshop on Geometrical Models of Natural Language Semantics, EMNLP 2011*, pages 62–66, 2011b.

Ozan Irsoy and Claire Cardie. Modeling compositionality with multiplicative recurrent neural networks. *arXiv*, abs/1412.6577, 2014.

Richard Johansson. Dependency Syntax in the CoNLL Shared Task 2008, 2007.

Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. A study of entanglement in a categorical framework of natural language. In *Proceedings of the 11th Workshop on Quantum Physics and Logic (QPL)*, 2014.

Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Stephen Pulman. A unified sentence space for categorical distributional-compositional semantics: Theory and experiments. In *Proceedings of 24th International Conference on Computational Linguistics (COLING)*, pages 549–558, 2012.

Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *International Conference on Learning Representations (ICLR) workshop*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119, 2013b.

Dmitrijs Milajevs and Matthew Purver. Investigating the contribution of distributional semantic information for dialogue act classification. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 40–47, 2014.

Dmitrijs Milajevs, Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Matthew Purver. Evaluating neural word representations in tensor-based compositional settings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 708–719, 2014.

Jeff Mitchell and Mirella Lapata. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, 2008.

Jeff Mitchell and Mirella Lapata. Language models based on semantic compositon. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 430–439, 2009.

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

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *Proceedings of Uncertainty in Artificial Intelligence (UAI)*, 2011.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Chris Manning, Andrew Ng, and Chris Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642, 2013.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Carol Van Ess-Dykema, Rachel Martin, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373, 2000.

Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 1017–1024, 2011.

Nick Webb, Mark Hepple, and Yorick Wilks. Dialogue act classification based on intra-utterance features. In *Proceedings of the AAAI Workshop on Spoken Language Understanding*, 2005.