Composing noun phrase vector representations

Aikaterini-Lida Kalouli
University of Konstanz
aikaterini-lida.kalouli@uni-konstanz.de

Valeria de Paiva
University of Birmingham

Richard Crouch
Chegg
{valeria.depaiva,dick.crouch}@gmail.com

Abstract

Vector representations of words have seen an increasing success over the past years in a variety of NLP tasks. While there seems to be a consensus about the usefulness of word embeddings and how to learn them, it is still unclear which representations can capture the meaning of phrases or even whole sentences. Recent work has shown that simple operations outperform more complex deep architectures. In this work, we propose two novel constraints for computing noun phrase vector representations. First, we propose that the semantic and not the syntactic contribution of each component of a noun phrase should be considered, so that the resulting composed vectors express more of the phrase meaning. Second, the composition process of the two phrase vectors should apply suitable dimensions' selection in a way that specific semantic features captured by the phrase's meaning become more salient. Our proposed methods are compared to 11 other approaches, including popular baselines and a neural net architecture, and are evaluated across 6 tasks and 2 datasets. Our results show that these constraints lead to more expressive phrase representations and can be applied to other state-of-the-art methods to improve their performance.

1 Introduction

Vector representations of words date back to the 1990's (Landauer and Dumais, 1997) and have seen an increasing success over the past years (Mikolov et al., 2013; Pennington et al., 2014; Devlin et al., 2018). While there seems to be a consensus about the usefulness of word embeddings and how to learn them, it is still controversial how to learn representations that capture the meaning of phrases or even whole sentences (Zhu et al., 2018). Generally, two main approaches are used to compute phrase representations: non-compositional and compositional. The former treats phrases as single units and attempts to learn their representations directly from corpora, much as it is done for words (Socher et al., 2010; Mikolov et al., 2013; Yin and Schütze, 2014). These approaches ignore the components of the phrase and are not scalable to all possible phrases of a language. On the other hand, the compositional approach derives a phrase or sentence representation from the embeddings of its component words in various ways, from simple addition and average operations, e.g., Mitchell and Lapata (2010); Turney (2012), to more complex neural net architectures, e.g., Pagliardini et al. (2018); Conneau et al. (2017). However, such approaches often ignore word order and other linguistic intuitions and lead to representations that cannot truly express the meaning of the sentence, as recently discussed by Zhu et al. (2018).

We concentrate on efficient *phrase* representations which capture meaning and can be handled as sentence components. We believe that from such representations the meaning of a full sentence can be *compositionally* computed, much as in more traditional semantic theories, e.g. in the Fregean functional application. For example, if we can compute efficient representations for all possible phrases contained in constituency parsing, say NP, VP, PP, etc., we can then derive the meaning of the whole sentence by functionally applying the constituents' representations on each other. To this end, we believe that for compositional phrases there should be compositional phrase representations, while for non-compositional ones, e.g., idioms, learning direct representations from corpora might be more effective. In this paper, we focus on *bigram compositional* nominal phrase vectors of adjective-noun and noun-noun (compounds) combinations. By starting from this linguistic category, we can reliably evaluate the two constraints we propose on one of the most common con-

Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019), pages 84–95
Florence, Italy, August 2, 2019.

stituent types, namely the NP phrase. Specifically, in this work we propose two novel constraints for computing such phrase vectors that are *linguistically informed* and *intuitively explainable*. First, we propose to focus on the *semantic* – and not the syntactic – contribution of each phrase component and decide whether the syntactic head or the syntactic modifier (Marneffe et al., 2006; McDonald et al., 2013) is semantically more *significant* for the meaning of the phrase. The phrase component with the most clear contribution to the meaning of the phrase might actually be the syntactic modifier and not the syntactic head and then this word is to be treated as the semantic head for the composition. Second, we propose that for two given word embeddings that need to be composed, we should select for the composition only those dimensions of the semantic modifier embedding that are more relevant to the semantic head of the phrase. In order words, we need to pick from the semantic modifier these attributes that are more relevant to the semantic head phrase. For example, for the compositional phrase *black magic*, intuitively we want to select all dimensions of *black* that have to do something with *magic* and not others that have to do with, e.g. *t-shirt*. In this way, we can compose the representation *black magic* by combining the attributes of *magic* with the "magic-like" attributes of *black*.

The contributions of this paper are three-fold: Firstly, we propose two novel constraints for composing *linguistically informed* and *intuitively explainable* noun phrase representations and show how these approaches could benefit future composition methods. Secondly, we provide a thorough evaluation of our methods over 6 evaluation tasks, 2 datasets and 11 other methods. Thirdly, we create an evaluation dataset of nomimal phrase-unigram paraphrase pairs, which we make openly available.

2 Relevant work

Early work on representing word sequences focused on bigram compositionality and considered various simple functions, such as vector addition and averaging (Mitchell and Lapata, 2010; Blacoe and Lapata, 2012), while already Turney (2012) integrated features for more meaningful relations. This early work focused on the representation of specific syntactic constructions and specific number of words and continues to be an ongoing research topic: representations of verb phrases (Hashimoto and Tsuruoka, 2016), noun phrases (Baroni and Zamparelli, 2010; Boleda et al., 2013; Dima, 2016), a combination of the two (Zanzotto et al., 2010; Wieting et al., 2015), noun-noun compositionality (Reddy et al., 2011; Hermann et al., 2012; Cordeiro et al., 2018), noun phrases attribute meaning (Hartung et al., 2017; Shwartz and Waterson, 2018), etc. This strand of research covers a variety of approaches ranging from the simple vector arithmetics mentioned to vector-matrix composition operations (Zanzotto et al., 2010; Guevara, 2010; Baroni and Zamparelli, 2010; Boleda et al., 2013), to the functional application of word vectors (Coecke et al., 2010; Grefenstette et al., 2014) to RNNs (Wieting et al., 2015) and other supervised (Hartung et al., 2017; Shwartz and Waterson, 2018) or unsupervised approaches (Hermann et al., 2012). Particularly, recent research producing context-aware representations of words (Peters et al., 2018; Devlin et al., 2018) has already had a great impact on the performance of many of these composition functions. At the same time, another strand of research concentrates on representing arbitrarily long phrases and sentences and mainly employs neural nets architectures: bag-of-words models (Kalchbrenner et al., 2014), feature-weighted average (Yu and Dredze, 2015) models, deep averaging networks (Iyyer et al., 2015), recursive (Socher et al., 2013; Conneau et al., 2017) and convolutional NNs (Yin and Schütze, 2015), encoding-decoding architectures (Kiros et al., 2015), to name only a few. Despite the large number of such approaches, it is still not clear that the composed phrase or sentence embeddings express the intended meaning, as recently shown by Shwartz and Dagan (2019), Zhu et al. (2018) and Dasgupta et al. (2018). Even more interesting is the fact that averaging and weighted averaging approaches have been shown to outperform complex deep learning methods (White et al., 2015; Wieting et al., 2016; Arora et al., 2017). This shows potential in exploiting the merits of simpler approaches but boosting them up with more powerful intuitive and linguistic constraints, as the ones proposed in this work.

3 Proposed Constraints

3.1 Constraint One: Semantic Contribution

The (dependency) syntax informs us that in an English bigram compositional, nominal phrase, the

first word is the modifier and the second the head. However, we observe that this syntactic decision does not always coincide with the role that each word plays in the meaning of the phrase. It can be the case that the modifier is more "meaningful" for the phrase. For example, if someone says *space ship*, we would be inclined to first think of *space* than of a prototypical ship. In that sense, *space* has a more significant contribution to the meaning of the phrase than *ship* has. By contrast, in the phrase *black magic* the notion of *magic* is more prototypical for the meaning of the phrase.

Current work that aims at compositionally constructing phrase representations takes the asymmetric contribution of the phrase components into account, e.g. by assigning different weights to the modifier and the head. However, all of this work bases the contribution decision on the syntax, i.e. on the syntactic head and modifier. However, as it has already been observed for English noun-noun compositionality (Bannard et al., 2003; Reddy et al., 2011; Cordeiro et al., 2018), the first component of a noun-noun phrase, i.e. the (syntactic) modifier, might have a greater contribution to the meaning of the phrase. Similar is the literature for other linguistic phenomena, e.g., light verbs (e.g., take a shower, give a kiss) or auxiliaries where the syntactic head does not coincide with the semantic, (see, e.g., Butt, 2010), but also in traditional semantic composition (e.g., lambda calculus) the quantifier of a sentence serves as the head, although the verb is considered the syntactic head of the sentence. Although this asymmetry has been observed for nominal phrases as well, e.g. by Hartung et al. (2017) who find that adjective representations capture more of the compositional semantics of an adjective-noun phrase than nouns do and implicitly also by Mitchell and Lapata (2010), whose composition functions give more weight to the adjectives than to the nouns, to our knowledge this is the first work that actively proposes and integrates this constraint into the composition process.

To compose meaningful phrase representations, we propose to consider the semantic contribution of the syntactic head and modifier of a phrase. In other words, we need to consider which is the *semantic* head and which is the *semantic* modifier. To this end, we can use word embeddings to decide whether a phrase is *heady*, i.e. the syntactic head has a stronger semantic contribution than the syntactic modifier, *mody*, i.e. the syntactic modifier has a stronger semantic contribution than the syntactic head, or *equal*, i.e. the syntactic head and modifier have both the same contribution to the meaning of the phrase. For bigram phrases that can be paraphrased by a single synonym –called *target* from now on – (e.g. black magic = sorcery), we find that the embeddings of some targets are more similar to the syntactic modifier embedding and of some others more similar to the syntactic head embedding of the phrase. We implement this observation: we compute the cosine similarity of the syntactic head to the target and of the syntactic modifier to the target and calculate their Δ. If the Δ is more than one standard deviation under the mean of all Δs (z-score computation), then the label *equal* is given, to account for cases where both words have an equal contribution to the meaning. Otherwise, the phrase is labeled based on whether the similarity of the syntactic head to the target or the syntactic modifier to the target is greater.

Since this approach for deciding on the semantic contribution of the syntactic head and modifier relies on the similarity of each of those components to a target, it is not available for all possible phrases because there is not one suitable unigram paraphrase/synonym for each phrase. Therefore, we want to test if the semantic contribution constraint is indeed a quantifiable, inherent property of the phrases that can be learned and can thus still be applied to phrases without targets. We did initial experimenting to train a classifier with a balanced set of 1000 *headys* and 1000 *modys*.[1] The collection of this set will be described in Section 4.2. For the classifier, we used 80% of the instances for training and 20% for testing. The classifier had to learn the *mody-heady* label solely based on the embeddings of the phrase components and *without* seeing any target embedding. The best trained model so far has been a MultiLayer Perceptron (MLP) with 3 hidden layers, 70 neurons per layer, 200 iterations and random weight initialization, delivering an accuracy of 74.8%. This shows that the semantic contribution constraint is indeed an inherent property of the embeddings that can be learnt from phrases having synonym-targets and be used for labeling phrases without such targets. Further experimenting and more training data can potentially improve

[1] We left out the *equal* label for this experiment due to the low number of such training samples.

this performance further.

3.2 Constraint Two: Dimensions' selection

The first constraint allows us to formulate a further one that directly shapes the composition process of phrase representations. Precisely, we propose that a composed representation of a bigram phrase should contain attributes of the semantic head embedding and only those attributes of the semantic modifier that are more relevant to the semantic head. This means that we need to select only those dimensions of the embedding of the semantic modifier that are related to the semantic head. Let's look at an example: *black magic* is a *heady* phrase, i.e. the contribution of the syntactic head *magic* is more significant for the overall meaning of the phrase than the contribution of *black*. This becomes even clearer if we think of a target synonym such as *sorcery*: for the meaning of the word *sorcery*, *magic* has a stronger correlation than *black* has. Thus, in this example, the composed vector should include the dimensions of *magic* and only those dimensions of *black* that are relevant to *magic*. "More relevant" dimensions are formalized as dimensions that are closer together. Even in embeddings, where the vectors do not mirror the frequency co-occurrences of the given word to other words of the vocabulary in a one-to-one fashion and no matter the dimensionality reduction approach, the same dimension should be capturing similar properties across different words, since each dimension corresponds to the same neuron having produced it. Thus, the same dimensions of the two phrase components embeddings that are closer together should correspond to similar notions and closer points in the vector space.

Intuitively, this dimensions' selection implements the idea that the composition of two words results in specific semantic aspects becoming more salient. This intuition is close to the dilation model of Mitchell and Lapata (2010), which attempts to stretch a vector *v* to the direction of a vector *u* in order to compute their composed vector. It is also similar to the more traditional idea of functional application: one tensor or vector is applied to another, resulting into their compositional representation (Coecke et al., 2010; Grefenstette et al., 2014). This has also been proposed by Baroni and Zamparelli (2010) for adjective-noun composition: the nouns are vectors and corpus-learned adjective matrices apply to these vectors producing other vectors. However, this only works for adjective-noun phrases where the adjectives can be clearly defined as the functions. For handling noun-noun phrases (and potentially other phrases), both phrase constituents have to be seen as *terms* (similar to λ calculus-like *terms*) and thus as vectors that can be applied in any direction. This allows us to formulate the following functions, which perform a kind of functional application, by taking the semantic modifiers as the functions and the semantic heads as the terms applied on them.

Compositional Function 1 SD

```
function ComposeSelectedDimsVec
    selected_dims ← []
    for i = 0 to headEmbed.length do
        headDim ← headEmbed[i]
        modDim ← modEmbed[i]
        if headDim − modDim < τ then
            selected_dims.append(modDim)
        else
            selected_dims.append(headDim)
    return selected_dims
```

Compositional Function 2 MOD-SD

```
function ComposeModAndSelectedDimsVec
    mod_selected_dims ← α · modEmbed + β · SD
```

In the first compositional function 1 (*SD*) we compare each dimension of the embedding of the semantic head of the phrase with the corresponding dimension of the semantic modifier of the phrase. If their Δ is under a threshold τ, then the dimensions are taken to be close enough and thus relevant and the dimension of the semantic modifier embedding is inserted unchanged into the new vector *selected_dims*. If the Δ is greater than τ, then the dimensions are taken to be distant and thus irrelevant to each other other and the dimension of the head is inserted into *selected_dims*. The final vector is a mixed vector consisting of a combination of the original modifier and head dimensions. Based on a grid search in steps of 10% from 0 to 1, we find $\tau = 0.3$ as the best parameter for the required threshold. Note that this is different than the elementwise max operation, as we do not select the dimension with the highest value among the two but instead we always select the semantic modifier dimension as long as its difference to the semantic head dimension is smaller than τ, no matter if the semantic modifier's dimension is greater or smaller than the head's. In our

second proposed composition function (*MOD-SD*) we make use of the vector produced by function 1: we weight the entire vector *SD* by β and add it to the original embedding of the semantic modifier which is also weighted by α. This function is inspired by the well-performing weighted addition operation but instead of the original semantic head vector, it uses the constructed functional vector of 1, which captures only the semantic head-relevant attributes of the semantic modifier and the semantic head attributes. Suitable grid search in steps of 0.02 shows $\alpha = 0.32$ and $\beta = 0.68$ as the best parameters. All tuning was performed on a held-out set, consisting of the 50% of the created dataset, described in Section 4.2.

As it is clear, these two composition functions heavily depend on the head and modifier roles of the phrase and are therefore inseparably connected with the constraint of the semantic contribution proposed in Section 3.1.

4 Evaluation of the constraints

4.1 Compared Approaches

For evaluation we include baseline approaches of vector arithmetics, the popular matrix-vector composition approach and an own trained neural network. If the injection of our first constraint into those approaches boosts their performance, the semantic contribution constraint can be considered for future composition approaches, especially those aiming at simple but linguistically informed operations. On the other hand, if the composition process described in our second constraint outperforms the compared approaches, we can be confident that the dimensions' selection as proposed in the previous section is a useful intuition capturing compositionality and can be safely integrated in future composition tasks.

Baseline approaches We include baseline operations from the literature that were recently shown to outperform complex deep architectures (White et al., 2015; Wieting et al., 2016; Arora et al., 2017). We use weighted elementwise vector addition (1) and multiplication (2) (Mitchell and Lapata, 2010; Turney, 2012; White et al., 2015; Hartung et al., 2017; Arora et al., 2017) and weighted elementwise average (3) (Mikolov et al., 2013; Wieting et al., 2016). Since addition and multiplication have been shown to perform so strongly and since multiplicative models have the drawback that the presence of zeroes in either of the vectors leads to information essentially being lost, we follow Mitchell and Lapata (2010) and also include a fourth equation, combining the addition and multiplication operations (4). For the weighting we do our own fine-tuning which is specific to the dataset we use.[2] This fine-tuning also showed that for our set the distinction between weights for adjective-noun and noun-noun phrases is not beneficial, contrary to Mitchell and Lapata (2010), who set the weights based on the part-of-speech. After tuning, the parameters are set to $\alpha = \beta = 1.0$, which in practice means that the unweighted variants perform better than their weighted counterparts. We also include "easy" baselines involving only the syntactic head or the syntactic modifier of the phrase and check whether the proposed compositional functions are better than those variants with no composition at all.

(1) $wei\_add_j : r_j = \alpha m_j + \beta h_j$

(2) $wei\_mult_j : r_j = \alpha m_j \cdot \beta h_j$

(3) $wei\_aver_j : r_j = \frac{\alpha m_j + \beta h_j}{2}$

(4) $wei\_comb_j : r_j = \alpha m_j + \beta h_j + \alpha m_j \cdot \beta h_j$

Matrix-vector approaches As already discussed before, popular approaches for computing phrases representations are the various matrix-vector composition operations. Already explored by Guevara (2010), Baroni and Zamparelli (2010) and Zanzotto et al. (2010) these approaches have since been used by various researchers, e.g. Boleda et al. (2013); Dima (2016). In these approaches the two constituent vectors of a phrase u and $v \in \mathbb{R}^n$ are composed by multiplying them via two matrices $A,B \in \mathbb{R}^{n \times n}$. For Zanzotto et al. (2010) and Guevara (2010), A and B are the same for every u and v and are calculated with partial least squares regression, while for the adjective-noun composition of Baroni and Zamparelli (2010), A is set to 0 and the weight matrix B is specifically learned for each single adjective. The mathematical formulation of this approach is: $r = Au + Bv$. Given the effectiveness of this approach (see e.g. Boleda et al. (2013)), we compare our proposed functions to it. From the three works mentioned above implementing this approach, only Zanzotto et al.'s is suitable for our purposes because a) it can handle both adjective-noun and noun-noun combinations and b) its dataset is openly available.

[2] In fact, we did test with the original parameters and found out that they deliver worse performance.

Deep Learning approach Although White et al. (2015), Wieting et al. (2016) and Arora et al. (2017) found that simple operations outperform complex deep architectures, there is still value in comparing the performance of a trained neural net to the performance of the other methods. For this purpose we experimented with multiple architectures, including feedforward nets, RNNs and LSTMs, attempting to find the best that fits our data. The training (80% of the set) and testing data (20% of the set) we used will be analyzed in more detail in the next section. Briefly, the datasets consisted of pairs of embeddings of the phrase components and their unigram synonym/paraphrase. For example, the embeddings of *dog* and *house* were paired with the embedding of the synonymous *kennel*. The neural net had to learn the synonym embedding by considering the two word embeddings as input. The best performing model was a feedforward neural net with 2 hidden dense layers. We used Xavier weight initialization (Glorot and Bengio, 2010) and the ELU (Clevert et al., 2015) activation function for all layers. Our updater was ADADELTA (Zeiler, 2012) and our learning rate 0.1. The training run for 200 epochs with 0.5 global dropout.

The left-most column of Table 1 gives a better overview of all compared methods.

4.2 Data

Data collection To tune and evaluate our proposals we needed a set that contains bigram noun phrases matched to unigram paraphrase/synonym targets, so that we have a "stable, uncontroversial " representation to compare our composed representations to (see also Zanzotto et al. (2010) and Turney (2012)). In this way, we can compose the representation of each phrase of the set with each of the methods under comparison and ideally, the composed representations are very similar to the embedding of the target of the pair since phrase and target have a synonymy/paraphrase relation. This is a harder task than comparing the composed representation to a corpus-learned representation of the phrase because the target representation is "independent", i.e. it does not capture cooccurrence effects of the components of the phrase, as the corpus-learned representation does. To this end, we created a new dataset which we make openly available.[3] The creation process of the set is similar to that of Turney (2012): we extract the nouns of WordNet (Fellbaum, 1998) that have a bigram phrase synonym in their synset and pair them together, e.g., from the entry *kennel, doghouse, dog house (outbuilding that serves as a shelter for a dog)* we extract the pair *dog house - kennel* . The pairs were cleaned to exclude all proper names and were further expanded by Turney's (2012) set which has the same format.[4] This process resulted in 6109 pairs of this format. However, not all pairs are compositional; since we are interested in creating *compositional* phrase representations, we wanted to ensure that we are only evaluating on suitable pairs, as a *hot dog* can never be a composition of *hot* and *dog*. To this end, we attempted to automatically exclude non-compositional pairs by following Turney (2012), who proposes two WordNet-based approaches: the phrase is most likely compositional if a) one of the words of the phrase is also present in the gloss of this phrase (cf. the *dog house* entry) or b) the (syntactic) head noun of the phrase is also a hypernym of the phrase (e.g., *brain surgery* has *surgery* as its hypernym and it is thus compositional). We are aware that these methods cannot eliminate all unsuitable pairs, but the data is much less noisy now. Future work may attempt to do a better filtering of non-compositional pairs. 4475 pairs are left, from which we further exclude the ones where one of the words of the phrase is also the target (e.g. *abdominal muscle - abdominal*) and we get 1914 final pairs. 50% of that set forms the held-out set used for tuning purposes (Section 3) and the rest of the dataset is used for the evaluation of the methods.

We also evaluate our methods on a second dataset, the only other dataset we could find fulfilling the requirements of our task[5]: the noun-noun set created by Zanzotto et al. (2010) (*ZZ* from now on). This set contains the same data format (bigram phrases-unigram paraphrase) and includes 1066 positive examples, i.e. examples where the

[3] https://github.com/kkalouli/compositional_phrase_vectors

[4] The Turney (2012) set was also scraped from WordNet but we observed that this set and our scraped set were not subsets, probably due to changes on WordNet over the years or differences in the scraping process.

[5] The probably most popular dataset of Mitchell and Lapata (2010) was not suitable due to its format (no unigram as comparison element) and the nature of the data, i.e. no truly synonymous/paraphrastic phrases-targets, merely *similar* pairs; also observed by Wieting et al. (2015)

paraphrase is a valid one for this phrase, and 379 negative ones, where the unigram is not a paraphrase of the bigram.[6]

Data preprocessing Since the goal of this work is to examine the efficiency of the proposed constraints for the compositionality of the vectors, we use pretrained embeddings; however training more specific embeddings or using state-of-the-art context-aware embeddings (e.g. Devlin et al. (2018); Peters et al. (2018)) could be even more beneficial for the approaches. In fact, by using such contextualized embeddings, our constraints could better handle polysemous words as the base embeddings would be partly disambiguated from the context. For now, the two datasets are first matched to the pretrained GloVe (Pennington et al., 2014) embeddings,[7] so that each phrase component and target word are mapped to their embedding. Then, a module determines whether the phrase is *mody*, *heady* or *equal*, based on the procedure described in Section 3.1. This procedure results into 895/515 *heady*, 792/190 *mody* and 227/119 *equal* for our set and the ZZ set, respectively. So, pairs like *black magic - sorcery* and *body armor - cataphract* become "heady", *archeological site - dig* and *baseball player - ballplayer* "mody", and *dramatic art - dramaturgy* and *female parent - mother* "equal".

4.3 Evaluation Tasks

To compare the approaches, we employed 6 evaluations tasks, aiming at testing different semantic aspects of the phrases. Our goal is to see which of the 13 methods perform best in each of the tasks. We include popular tasks, like synonymy detection and concept clustering (see, e.g., Baroni et al., 2014; Schnabel et al., 2015), but we do not employ the human similarity judgments task. We are not convinced that semantic similarity can be scaled in a range of 1 to 7 as we are not sure how one should decide, e.g., between a 3 and a 4. Such criticisms were also discussed by Faruqui et al. (2016).

Plain similarity One of the most common intrinsic evaluation tasks is the semantic similarity between an item and a target. Since targets are part of our dataset, the simplest task is to calculate the cosine similarity between the composed vector of a phrase and the embedding of its target.

Precision This task is a modification of the analogy task of Mikolov et al. (2013). Given a phrase vector and its neighbors in the semantic space, we check if the target word is its closest neighbor (cf. Baroni and Zamparelli (2010); Mikolov et al. (2013)). The task is also undertaken for the next two closest neighbors of the phrase. Ultimately, we measure Precision@1, Precision@2 and Precision@3, respectively, for how many items of our set had their targets as neighbors at the corresponding positions.

Overlapping neighbors Here we measure how many neighbors of the phrase representation are also neighbors of the target embedding. Since embeddings capture the relational co-occurrences of words, it should be the case that the phrase and the target vectors share neighbors. This would mean that they are closer in the semantic space than items not sharing any neighbors, even if the target word itself is not a neighbor of the phrase embedding.

Synonymy detection This popular task, first applied on the TOEFL examples for word embeddings (Landauer and Dumais, 1997), is to select out of some candidate targets, the one with the highest similarity to the given word. Similarly, (cf. Turney, 2012) we create a set of 7 candidate unigrams for each given phrase: its syntactic modifier, its syntactic head, its target, a synonym of its syntactic modifier and of its syntactic head[8] and two random words. We compute the similarity of the phrase representation to each of those and create a ranked list of the 7 candidates. Targets that are lower in the ranked list are penalized and targets that are higher up are boosted; conversely for the random words. Ultimately, we obtain a score between -1 and 1, with -1 being the worst with a random word at rank 1 and the target at the last rank and 1 standing for the best case where the target is at rank 1 and the randoms last.

Clustering A popular task is concept categorization or clustering. Given a set of concepts, the task is to group them into categories. We adjust this task to measure how many of the phrase representations are clustered together with their target embedding. If the phrase vector truly expresses

[6] For our purposes, we excluded pairs containing proper names in capital due to the lack of pretrained embeddings for those, resulting in a set of 824 pairs.

[7] Trained on Wikipedia 2014 and Gigaword 5, 300 dim.

[8] Extracted from WordNet.

Method	Created dataset							ZZ dataset
	Sim	**Pr@1**	**Pr@2**	**Pr@3**	**OveNei**	**Syn**	**Clus**	**DistSim**
only_head	21.6	1.3	2.3	3.3	0.92	0.17	5.2	2.82E-31
only_mod	20.6	4.5	6.8	8.3	1.08	0.16	3.5	6.68E-52
sd (Const1 + Const2)	**28.5**	**5.6**	**8.4**	**10.3**	**1.80**	**0.23**	**6.7**	1.09E-41
mod-sd (Const1 + Const2)	**28.0**	**6.0**	**8.3**	**10.1**	**1.75**	0.17	**6.8**	**6.71E-47**
add	26.3	4.1	**7.6**	9.2	1.54	0.16	4.9	**9.43E-49**
mult	-0.5	0.0	0.0	0.0	0.01	-0.37	1.7	0.0801
aver	26.3	4.1	**7.6**	9.2	1.54	0.16	5.9	**9.43E-49**
comb	25.9	4.4	**7.6**	**10.0**	1.65	0.17	4.9	3.58E-44
add+Const1	**29.0**	**5.7**	**8.9**	**10.4**	**1.80**	0.18	**7.0**	**8.58E-47**
aver+Const1	**29.0**	**5.7**	**8.9**	**10.4**	**1.80**	0.18	4.1	**8.58E-47**
comb+Const1	**29.2**	**5.7**	**8.5**	**11.0**	**1.85**	0.19	**7.0**	6.27E-46
$feedforward\ NN$	24.0	0.2	0.2	0.2	0.36	**0.24**	0.5	-
matr-vec (Zanzotto et al., 2010)	-	-	-	-	-	-	-	1.00E-10

Table 1: Overview of all compared methods across the 6 evaluation tasks. The notation *+Const1* is added to the methods containing the semantic contribution constraint (Constraint One). The metric given for each task is the average metric across the entire dataset. Numbers in boldface mark the best performance per task. Multiple numbers may be in boldface in the same task, if there is no statistically significant difference between them.

meaning, the two should be clustered together. We use k-means clustering with 1914 clusters (as many as the pairs of our set) and 99 iterations.

Positive-Negative Similarity Distribution This task is the original used by Zanzotto et al. (2010), so we only apply it on the ZZ set. Here, we test if the distribution of the cosine similarities of the positive pairs is statistically different from the distribution of the similarities of the negative pairs: if it is, it means that the corresponding functions perform well because they can keep the two categories apart (see Zanzotto et al. (2010) for more details). As in the original experiment, the results show p-values, calculated with the Students t-test for two independent samples of different sizes: lower values characterize better models.

4.4 Results

In Table 1 we list all 13 methods compared in this work and their performance across all evaluation tasks. To test for statistical significance, the results were first grouped into categories and were analyzed using linear mixed effects regression models with the corresponding conditions (Method and Constraint) as fixed factors and random intercepts for the phrases of the dataset.[9] P-values were calculated using the Satterthwaite approximation of degrees-of-freedom in the R-package lmerTest (Kuznetsova et al., 2017). The above process was done separately for each of the evaluation tasks.

Within the separate categories, the models showed main effects and interactions of both Method and Constraint across all tasks. The proposed sd and $mod-sd$ functions (lines 3-4 of Table 1) perform statistically the same across tasks: sd does outperform $mod-sd$ in the *Syn* task but the latter outperforms the former in the *DistSim* task, so that they exhibit an equal behavior. Concerning the baselines, the methods of addition, average and combined addition-multiplication (lines 5-8) perform statistically the same across tasks but heavily outperform the multiplication approach (contrary to Mitchell and Lapata (2010) but similar to Boleda et al. (2013)). The same operations but with our semantic contribution constraint (*baselines+Const1*, lines 9-11) also perform statistically the same across tasks.

More interesting are the overall results across categories: here, there is a main effect of *Method*. In the *Sim* task, in all three precision tasks, in *OveNei* and in *Clus*, the proposed sd and $mod-sd$ together with the *baselines+Const1* are statistically best without any difference between them. In the *Syn* task, the sd and the NN[10] provide the statistically best results, with the *addition-*

[9]The models further included random slopes for the within-group factors when this improved the fit of the model, as determined by LogLikelihood comparisons, using the R-function anova().

[10]It is not surprising that the NN performs that well in this task. Since the NN is trained to learn/resemble the target embedding, its similarity to this specific target is higher than to other words on which it has not been trained. Thus, here it achieves better accuracy than in other tasks as it's the relative similarity to the target vs. to the other words that is measured.

multiplication+Const1 operation following. For the *DistSim* task, $mod - sd$, the *addition+Const1* and *average+Const1* as well as the simple baselines perform best and all methods outperform what is reported by Zanzotto et al. (2010).

5 Discussion

The first proposed constraint of this paper, the semantic contribution of heads and modifiers, proves powerful: the *+Const1* addition, average and combined addition-multiplication operations heavily outperform their counterparts without the constraint and come to be the statistically best in 5 of the 6 tasks, also outperforming the NN and Zanzotto et al.'s approach. This confirms that the semantic contribution constraint is indeed beneficial: it's the *semantic* contribution of the phrase components that should be considered for the weighting and not the syntactic role. The fact that this constraint boosts simple baselines like the ones presented here shows the potential in exploring how it could also boost other existing (deep) models.

On the other hand, the dimensions' selection constraint proposed in sd and $mod - sd$ performs statistically best in 5 of the 6 tasks. They outperform the non-compositional baselines (*only_head* and *only_mod*), showing that they indeed capture compositionality. They also outperform the standard baselines, the NN and Zanzotto et al.'s approach. This result shows the benefits of our proposed functions: selecting only those dimensions of the semantic modifier that are relevant to the head, i.e. implementing the intuition of functional application of one vector onto the other, but relying on semantic heads and modifiers as opposed to syntactic ones. Both functions have a heavier presence of semantic head dimensions than semantic modifier dimensions due to their composition process (see Section 3.2). From this we can conclude that compositional vectors are more efficient when more semantic head attributes than semantic modifier attributes are present. Between the two approaches there is no apparent difference: sd is better in the *Syn* task and $mod - sd$ in *DistSim*. Further evaluation tasks will have to determine any performance differences. $mod - sd$ might be able to capture more information because it combines the semantic modifier dimensions with the dimensions of the constructed functional vector which contains the semantic head attributes "dilated" in the direction of the semantic modifier.

The tasks included in the current evaluation show no real differences between the proposed methods sd and $mod - sd$ and the *baselines+Const1*, which might raise some doubt on the real value and powerfulness of the dimensions' selection constraint. However, the goal of this work was to test the intuition behind this approach and see whether it can compete with other state-of-the-art results. In that respect, the results are promising. Particularly, we expect that the proposed functions can be improved with further fine-tuning of the dimensions' selection process to outperform the standard baselines. On the contrary, the baseline operations have less room for improvement. We hope that future tasks can show more clearly the weaknesses and strengths of each approach. We are particularly interested in testing this approach on other types of phrases, e.g. verb phrases (VP), to see how our two constraints generalize. For example, concerning our first constraint, for English VPs containing a verb and an object, we expect all verbs to behave as semantic heads (and the objects as semantic modifiers) except for light verbs, where the objects should be the semantic heads. In fact, preliminary experimenting with VPs shows that both constraints can be extended to them with promising results.

6 Conclusion

In this paper, we proposed two novel constraints for composing linguistically-informed and intuitively-explainable nominal phrase vectors. After a thorough evaluation, we showed that these constraints lead to more expressive phrase vectors, outperforming popular baselines. Other evaluation tasks might prove more suitable for showing specific strengths and weaknesses of the proposed constraints. In the future, we wish to apply our approach to other kinds of phrases, e.g., verb phrases, and try to derive a representation for a whole sentence by iteratively combining the different constituent phrases of the sentence through the proposed constraints. Additionally, we would like to train a better semantic contribution classifier and make it openly available for use.

Acknowledgments

We would like to thank Bettina Braun und Katharina Zahner for useful feedback on the statistics, as well as Miriam Butt and the anonymous reviewers for constructive comments.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations (ICLR)*.

Colin Bannard, Timothy Baldwin, and Alex Lascarides. 2003. A Statistical Approach to the Semantics of Verb-particles. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18*, MWE '03, pages 65–72, Stroudsburg, PA, USA. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 238–247. Association for Computational Linguistics.

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193. Association for Computational Linguistics.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 546–556, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gemma Boleda, Marco Baroni, The Nghia Pham, and Louise McNally. 2013. Intensionality was only alleged: On adjective-noun composition in distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 35–46, Potsdam, Germany. Association for Computational Linguistics.

Miriam Butt. 2010. The light verb jungle : still hacking away. In Mengistu Amberber, Brett Baker, and Mark Harvey, editors, *Complex predicates : cross-linguistic perspectives on event structure*, pages 48–78. Cambridge University Press, Cambridge.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *CoRR*, abs/1511.07289.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical Foundations for a Compositional Distributional Model of Meaning. *Lambek Festschrift Linguistic Analysis*, 36.

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, Copenhagen, Denmark. Association for Computational Linguistics.

Silvio Cordeiro, Aline Villavicencio, Marco Idiart, and Carlos Ramisch. 2018. Unsupervised Compositionality Prediction of Nominal Compounds. *Computational Linguistics*, pages 1–80.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J. Gershman, and Noah D. Goodman. 2018. Evaluating Compositionality in Sentence Embeddings. *CoRR*, abs/1802.04302.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

Corina Dima. 2016. On the Compositionality and Semantic Interpretation of English Noun Compounds. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 27–39. Association for Computational Linguistics.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with Evaluation of Word Embeddings Using Word Similarity Tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Edward Grefenstette, Mehrnoosh Sadrzadeh, Stephen Clark, Bob Coecke, and Stephen Pulman. 2014. Concrete Sentence Spaces for Compositional Distributional Models of Meaning. *Computing Meaning*, 4:71–86.

Emiliano Guevara. 2010. A Regression Model of Adjective-Noun Compositionality in Distributional Semantics. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, pages 33–37. Association for Computational Linguistics.

Matthias Hartung, Fabian Kaupmann, Soufian Jebbara, and Philipp Cimiano. 2017. Learning Compositionality Functions on Word Embeddings for Modelling Attribute Meaning in Adjective-Noun Phrases. In

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 54–64. Association for Computational Linguistics.

Kazuma Hashimoto and Yoshimasa Tsuruoka. 2016. Adaptive Joint Learning of Compositional and Non-Compositional Phrase Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 205–215. Association for Computational Linguistics.

Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2012. An Unsupervised Ranking Model for Noun-noun Compositionality. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, SemEval '12, pages 132–141, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691. Association for Computational Linguistics.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland. Association for Computational Linguistics.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-Thought Vectors. *CoRR*, abs/1506.06726.

Alexandra Kuznetsova, Per Brockhoff, and Rune Christensen. 2017. lmerTest Package: Tests in Linear Mixed Effects Models. *Journal of Statistical Software, Articles*, 82(13):1–26.

Thomas Landauer and Susan Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

M. Marneffe, B. Maccartney, and C. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. European Language Resources Association (ELRA).

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *Proceedings of Workshop at ICLR*, 2013.

Jeff Mitchell and Mirella Lapata. 2010. Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1429.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features. 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 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on 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, New Orleans, Louisiana. Association for Computational Linguistics.

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. Asian Federation of Natural Language Processing.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307. Association for Computational Linguistics.

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition.

Vered Shwartz and Chris Waterson. 2018. Olive Oil is Made *of* Olives, Baby Oil is Made *for* Babies: Interpreting Noun Compounds Using Paraphrases in a

Neural Model. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 218–224, New Orleans, Louisiana. Association for Computational Linguistics.

Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. 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. Association for Computational Linguistics.

Peter D. Turney. 2012. Domain and Function: A Dual-Space Model of Semantic Relations and Compositions. *Journal of Artificial Intelligence Research*, 44:533–585.

Lyndon White, Roberto Togneri, Wei Liu, and Mohammed Bennamoun. 2015. How Well Sentence Embeddings Capture Meaning. In *Proceedings of the 20th Australasian Document Computing Symposium*, ADCS '15, pages 9:1–9:8, New York, NY, USA. ACM.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From Paraphrase Database to Compositional Paraphrase Model and Back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards Universal Paraphrastic Sentence Embeddings. *CoRR*, abs/1511.08198.

Wenpeng Yin and Hinrich Schütze. 2014. An Exploration of Embeddings for Generalized Phrases. In *Proceedings of the ACL 2014 Student Research Workshop*, pages 41–47. Association for Computational Linguistics.

Wenpeng Yin and Hinrich Schütze. 2015. Convolutional Neural Network for Paraphrase Identification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 901–911, Denver, Colorado. Association for Computational Linguistics.

Mo Yu and Mark Dredze. 2015. Learning Composition Models for Phrase Embeddings. *Transactions of the Association for Computational Linguistics*, 3:227–242.

Fabio Massimo Zanzotto, Ioannis Korkontzelos, Francesca Fallucchi, and Suresh Manandhar. 2010. Estimating Linear Models for Compositional Distributional Semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1263–1271, Stroudsburg, PA, USA. Association for Computational Linguistics.

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701.

Xunjie Zhu, Tingfeng Li, and Gerard de Melo. 2018. Exploring Semantic Properties of Sentence Embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 632–637, Melbourne, Australia. Association for Computational Linguistics.