

# Sense Contextualization in a Dependency-Based Compositional Distributional Model

**Pablo Gamallo**

Centro Singular de Investigación en  
Tecnoloxías da Información (CiTIUS)  
Universidade de Santiago de Compostela, Galiza  
pablo.gamallo@usc.es

## Abstract

Little attention has been paid to distributional compositional methods which employ syntactically structured vector models. As word vectors belonging to different syntactic categories have incompatible syntactic distributions, no trivial compositional operation can be applied to combine them into a new compositional vector. In this article, we generalize the method described by Erk and Padó (2009) by proposing a dependency-base framework that contextualize not only lemmas but also selectional preferences. The main contribution of the article is to expand their model to a fully compositional framework in which syntactic dependencies are put at the core of semantic composition. We claim that semantic composition is mainly driven by syntactic dependencies. Each syntactic dependency generates two new compositional vectors representing the contextualized sense of the two related lemmas. The sequential application of the compositional operations associated to the dependencies results in as many contextualized vectors as lemmas the composite expression contains. At the end of the semantic process, we do not obtain a single compositional vector representing the semantic denotation of the whole composite expression, but one contextualized vector for each lemma of the whole expression. Our method avoids the troublesome high-order tensor representations by defining lemmas and selectional restrictions as first-order tensors (i.e. standard vectors). A corpus-based experiment is performed to both evaluate the quality

of the compositional vectors built with our strategy, and to compare them to other approaches on distributional compositional semantics. The experiments show that our dependency-based compositional method performs as (or even better than) the state-of-the-art.

## 1 Introduction

Erk and Padó (2008) proposed a method in which the combination of two words,  $a$  and  $b$ , returns two vectors: a vector  $a'$  representing the sense of  $a$  given the selectional preferences imposed by  $b$ , and a vector  $b'$  standing for the sense of  $b$  given the (inverse) selectional preferences imposed by  $a$ . The main problem is that this approach does not propose any compositional model for sentences. Its objective is to simulate word sense disambiguation, but not to model semantic composition at any level of analysis. In Erk and Padó (2009), the authors briefly describe an extension of their model by proposing a recursive application of the compositional function. However, they only formalize the recursive application when the composite expression consists of two dependent words linked to the same head. So, they only explain how the head is contextualized by its dependents, but not the other way around. In addition, they do not model the influence of context on the selectional preferences. In other terms, their recursive model does not make use of contextualized selectional preferences.

In this article, we generalize the method described in Erk and Padó (2009) by proposing a dependency-base framework that contextualize both lemmas and selectional preferences. The main contribution of the article is to expand their model to a fully compositional framework in which syntactic dependencies are put at the core of semantic composition.

In our model, lemmas and selectional preferences are defined as unary-tensors (standard vectors), while syntactic dependencies are binary functions combining vectors in an iterative and incremental way.

For dealing with any sequence with  $N$  (lexical) words and  $N - 1$  dependencies linking them, the compositional process can be applied  $N - 1$  times dependency-by-dependency in two different ways: from left-to-right and from right-to-left. Figure 1 illustrates the incremental process of building the sense of words dependency-by-dependency from left-to-right. Given the composite expression “ $a b c$ ” and its dependency analysis depicted in the first row of the figure, several compositional processes are driven by the two dependencies involved in the analysis ( $m$  and  $n$ ). First,  $m$  is decomposed into two functions: the head function  $m_{\uparrow}$ , and the dependent one,  $m_{\downarrow}$ . The head function  $m_{\uparrow}$  takes as input the sense of the head word  $b$  and the selectional preferences of  $a$ , noted here as  $a^{\circ}$ , and returns a new denotation of the head word,  $b_{m_{\uparrow}}$ , which represents the contextualized sense of  $b$  given  $a$  at the  $m$  relation. Similarly, the dependent function  $m_{\downarrow}$  takes as input the sense of the dependent word  $a$  and the selectional preferences  $b^{\circ}$ , and returns a new denotation of the dependent word:  $a_{m_{\downarrow}}$ . The green box is used to highlight the result of each function. Next, the dependency  $n$  between  $b$  and  $c$  is also decomposed into the head and dependent functions:  $n_{\uparrow}$  and  $n_{\downarrow}$ . Function  $n_{\uparrow}$  combines the already contextualized head  $b_{m_{\uparrow}}$  with the selectional preferences  $c^{\circ}$ , and returns a still more specific sense of the head:  $b_{m_{\uparrow}+n_{\uparrow}}$ . Finally, function  $n_{\downarrow}$  takes as input the sense of the dependent word  $c$  and the already contextualized selectional preferences  $b_{m_{\downarrow}}^{\circ}$ , and builds a contextualized sense of the dependent word:  $c_{m_{\downarrow}+n_{\downarrow}}$ . At the end of the process, we have not obtained one single sense for the whole expression “ $a b c$ ”, but one contextualized sense per word:  $a_{m_{\downarrow}}$ ,  $b_{m_{\uparrow}+n_{\uparrow}}$ , and  $c_{m_{\downarrow}+n_{\downarrow}}$ . Notice that the two words involved in the direct object dependency,  $b$  and  $c$ , have been contextualized twice since they inherit the restrictions of the subject dependency. The root word,  $b$ , is directly involved in the two dependencies and, then, is assigned an intermediate contextualized sense,  $b_{m_{\uparrow}}$ , in the first combination with  $a$ .

In the second case, from right-to-left, the semantic process is applied in a similar way, but starting from the rightmost dependency,  $n$ , and

ending by the leftmost one,  $m$ . At the end of the process, three contextualized word senses are also obtained which might be slightly different from those obtained by the left-to-right algorithm. The main difference is that  $a$  is now contextualized by both  $b$  and  $c$ , while  $c$  is just contextualized by  $b$ .

The iterative application of the syntactic dependencies found in a sentence is actually the process of building the contextualized sense of all the content words constituting that sentence. So, the whole sentence is not assigned only one meaning - which could be the contextualized sense of the *root* word-, but one sense per word, being the sense of the root just one of them, as in the work described in Weir et al. (Weir et al., 2016). This allows us to retrieve the contextualized sense of all constituent words within a sentence. The contextualized sense of any word might be required in further semantic processes, namely for dealing with co-reference resolution involving anaphoric pronouns. Such an elementary operation is prevented if the sense of the phrase is just one complex sense, as in most compositional approaches.

The rest of the article is organized as follows. In Section 2, several distributional compositional approaches are introduced and discussed. Next, in Section 3, our dependency-based compositional model is described. In Section 4, a corpus-based experiment is performed to build and evaluate the quality of compositional vectors. Finally, relevant conclusions are addressed in Section 5.

## 2 Related Work

To take into account “the mode of combination”, some distributional approaches follow a strategy aligned with the formal semantics perspective in which functional words are represented as high-dimensional tensors (Coecke et al., 2010; Baroni and Zamparelli, 2010; Grefenstette et al., 2011; Krishnamurthy and Mitchell, 2013; Kartsaklis and Sadrzadeh, 2013; Baroni, 2013; Baroni et al., 2014). Using the abstract mathematical framework of category theory, they provide the distributional models of meaning with the elegant mechanism expressed by the principle of compositionality, where words interact with each other according to their type-logical identities (Kartsaklis, 2014). The categorial-based approaches define arguments as vectors while functions taking arguments (e.g., verbs or adjectives that combine with nouns) are n-order tensors, with the number

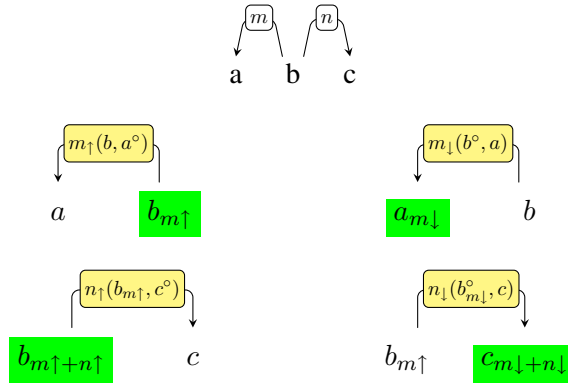


Figure 1: Syntactic analysis of the expression “ $a b c$ ” and left-to-right construction of the contextualized word senses.

of arguments determining their order. Function application is the general composition operation. This is formalized as the tensor product, which is nothing more than a generalization of matrix multiplication in higher dimensions. However, this method results in an information scalability problem, since tensor representations grow exponentially (Kartsaklis et al., 2014).

In our approach, by contrast, we operate with only two types of semantic objects: first-order tensors (or standard vectors) for lemmas and preferences, and second-order functions for syntactic dependencies. This solves the scalability problem of high-order tensors. In addition, it also prevent us giving different categorical representations to verbs in different syntactic contexts. A verb is represented as a single vector which is contextualized as it is combined with its arguments.

Some of the approaches cited above induce the compositional meaning of the functional words from examples adopting regression techniques commonly used in machine learning (Baroni and Zamparelli, 2010; Krishnamurthy and Mitchell, 2013; Baroni, 2013; Baroni et al., 2014). In our approach, by contrast, functions associated with dependencies are just basic arithmetic operations on vectors, as in the case of the first arithmetic approaches to composition (Mitchell and Lapata, 2008, 2009, 2010; Guevara, 2010; Zanzotto et al., 2010). Arithmetic approaches are easy to implement and produce high-quality compositional vectors, which makes them a good choice for practical applications (Baroni et al., 2014).

However, given that our vector space is structured and enriched with syntactic information, the vectors built by composition cannot be a sim-

ple mixture of the input vectors as in the bag-of-words approaches (Mitchell and Lapata, 2008). Our syntax-based vector representation of two related words encodes incompatible information and there is no direct way of combining the information encoded in their respective vectors. Vectors of content words (nouns, verbs, adjectives, and adverbs) live into different and incompatible spaces because they are constituted by different types of syntactic contexts. So, they cannot be merged. To combine them, on the basis of previous work (Thater et al., 2010; Erk and Padó, 2008; Melamud et al., 2015), we distinguish between direct denotation and selectional preferences within a dependency relation. Our approach is an attempt to join the main ideas of these syntax-based and structured vector space models into an entirely compositional model. More precisely, we generalize the recursive model introduced by Erk and Pado (Erk and Padó, 2009) with the addition of contextualized selection preferences.

Finally, recent works make use of deep learning strategies to build compositional vectors, such as recursive neural network models (Socher et al., 2012; Hashimoto and Tsuruoka, 2015). Still in the deep learning paradigm, special attention deserves a syntax-based compositional version of C-BOW algorithm (Pham et al., 2015). Our method, however, requires transparent and structured vector spaces to model compositionality.

### 3 The Method

In our approach, composition is modeled in terms of recursive function application on word vectors driven by binary dependencies. Each dependency stands for two functions on vectors: the *head* func-

tion and the *dependent* one. Let us consider the *nominal subject* syntactic dependency, which denotes two functions represented by the following binary  $\lambda$ -expressions:

$$\lambda \mathbf{x} \lambda \mathbf{y}^\circ \text{ nsubj}_\uparrow(\mathbf{x}, \mathbf{y}^\circ) \quad (1)$$

$$\lambda \mathbf{x}^\circ \lambda \mathbf{y} \text{ nsubj}_\downarrow(\mathbf{x}^\circ, \mathbf{y}) \quad (2)$$

where  $\text{nsubj}_\uparrow$  and  $\text{nsubj}_\downarrow$  represent the head and dependent functions, respectively;  $\mathbf{x}$ ,  $\mathbf{x}^\circ$ ,  $\mathbf{y}$ , and  $\mathbf{y}^\circ$  stand for vector variables. On the one hand,  $\mathbf{x}$  and  $\mathbf{y}$  represent the denotation of the head and dependent lemmas, respectively. They represent standard context distributions. On the other hand,  $\mathbf{x}^\circ$  represents the selectional preferences imposed by the head, while  $\mathbf{y}^\circ$  stands for the selectional preferences imposed by the dependent lemma. Selectional preferences are also vectors and the way we build them is described later.

Consider now the vectors of two specific lemmas, **cat** and **chase**, and their respective selectional preferences at the subject position. Each function application consists of multiplying the direct vector associated with a lemma and the selectional preferences imposed by the other lemma:

$$\text{nsubj}_\uparrow(\mathbf{chase}, \mathbf{cat}^\circ) = \mathbf{chase} \odot \mathbf{cat}^\circ = \mathbf{chase}_{\text{nsubj}_\uparrow} \quad (3)$$

$$\text{nsubj}_\downarrow(\mathbf{chase}^\circ, \mathbf{cat}) = \mathbf{cat} \odot \mathbf{chase}^\circ = \mathbf{cat}_{\text{nsubj}_\downarrow} \quad (4)$$

Each multiplicative operation results in a compositional vector which represents the contextualized sense of one of the two lemmas (either the head or the dependent). Component-wise multiplication has an intersective effect: the selectional preferences restricts the direct vector by assigning frequency 0 to those contexts that are not shared by both vectors. Here,  $\mathbf{cat}^\circ$  and  $\mathbf{chase}^\circ$  are selectional preferences resulting from the following vector addition:

$$\mathbf{cat}^\circ = \sum_{\mathbf{w} \in S_\downarrow(\mathbf{cat})} \mathbf{w} \quad (5)$$

$$\mathbf{chase}^\circ = \sum_{\mathbf{w} \in S_\uparrow(\mathbf{chase})} \mathbf{w} \quad (6)$$

where  $S_\downarrow(\mathbf{cat})$  returns the vector set of those verbs having *cat* as subject (except the verb *chase*).

More precisely, given the nominal subject position, the new vector  $\mathbf{cat}^\circ$  is obtained by adding the vectors  $\{\mathbf{w} | \mathbf{w} \in S_\downarrow(\mathbf{cat})\}$  of those verbs (*eat*, *jump*, etc) that are combined with the noun *cat* in that syntactic context. Component-wise addition of vectors has an union effect. In more intuitive terms,  $\mathbf{cat}^\circ$  stands for the inverse selectional preferences imposed by *cat* on any verb at the subject position. As this new vector consists of verbal contexts, it lives in the same vector space than verbs and, therefore, it can be combined with the direct vector of *chase*.

On the other hand,  $S_\uparrow(\mathbf{chase})$  in equation 6 represents the vector set of nouns occurring as subjects of *chase* (except the noun *cat*). Given the subject position, the vector  $\mathbf{chase}^\circ$  is obtained by adding the vectors  $\{\mathbf{w} | \mathbf{w} \in S_\uparrow(\mathbf{chase})\}$  of those nouns (e.g. *dog*, *man*, *tiger*, etc) that might be at the subject position of the verb *chase*.

The incremental application of head and dependent functions contextualize the representation of each word in the phrase. Incrementality also model the influence of context on the selectional preferences. The incremental left-to-right interpretation of “*the cat chased a mouse*” is illustrated in Figure 2 (without considering the meaning of determiners nor verbal tense): First, the head and dependent functions associated with the subject dependency *nsubj* build the compositional vectors  $\mathbf{chase}_{\text{nsubj}_\uparrow}$  and  $\mathbf{cat}_{\text{nsubj}_\downarrow}$ . Then, the head function associated with *doobj* produces a more elaborate chasing event,  $\mathbf{chase}_{\text{nsubj}_\uparrow + \text{doobj}_\uparrow}$ , which stands for the final contextualized sense of the root verb. In addition, the dependent function of *doobj* yields a new nominal vector,  $\mathbf{mouse}_{\text{nsubj}_\downarrow + \text{doobj}_\downarrow}$ , whose internal information only can refer to a specific animal: “*the mouse chased by the cat*”. Notice that contextualization may disambiguate ambiguous words: in the context of a chasing event, *mouse* does not refer to a computer’s device. In fact, to interpret “*the cat chased a mouse*”, it is required to interpret “*cat chased*” as a fragment that restricts the type of nouns that can appear at the direct object position: *mouse*, *rat*, *bird*, etc. In the same way “*police chases*” restricts the entities that can be chased by police officers: *thieves*, *robbers*, and so on.

In our approach, not only the lemmas are contextualized but also the selectional preferences. The contextualized selectional preferences,

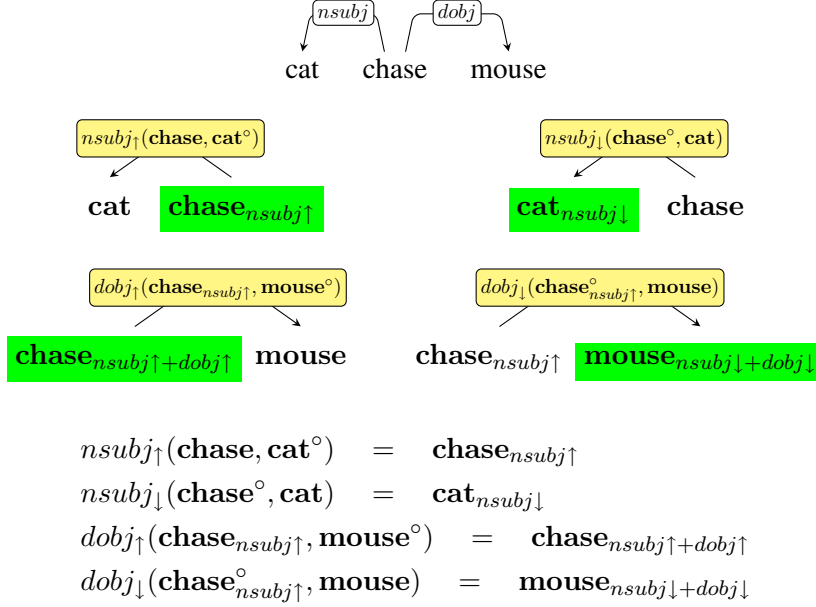


Figure 2: Syntactic analysis of the expression “the cat chased a mouse” and left-to-right construction of the contextualized word senses.

$\text{chase}^{\circ}_{nsubj_{\uparrow}}$ , are obtained as follows:

$$\text{chase}^{\circ}_{nsubj_{\uparrow}} = \text{cat}_{nsubj_{\downarrow}} \odot \sum_{w \in D_{\uparrow}(\text{chase})} w \quad (7)$$

where  $D_{\uparrow}(\text{chase})$  returns the vector set of those nouns that are in the direct object role of *chase* (except the noun *mouse*). The new vector resulting by this addition is combined by multiplication (intersection) with the contextualized dependent vector,  $\text{cat}_{nsubj_{\downarrow}}$ , to build the contextualized selectional preferences. In more intuitive terms, the selectional preferences built in equation 7 are constituted by selecting the contexts of the nouns appearing as direct object of *chase*, which are also part of *cat* after having been contextualized by the verb at the subject position. This is the major contribution with regard to the work described in Erk and Padó (2009).

The dependency-by-dependency functional application results in three contextualized word senses:  $\text{cat}_{nsubj_{\downarrow}}$ ,  $\text{chase}_{nsubj_{\uparrow}+dobj_{\uparrow}}$  and  $\text{mouse}_{nsubj_{\downarrow}+dobj_{\downarrow}}$ . They all together represent the meaning of the sentence in the left-to-right direction.

In the opposite direction, from right-to-left, the incremental process starts with the direct object dependency:

$$\begin{aligned}
dobj_{\uparrow}(\text{chase}, \text{mouse}^{\circ}) &= \text{chase}_{dobj_{\uparrow}} \\
dobj_{\downarrow}(\text{chase}^{\circ}, \text{mouse}) &= \text{mouse}_{dobj_{\downarrow}} \\
nsubj_{\uparrow}(\text{chase}_{dobj_{\uparrow}}, \text{cat}^{\circ}) &= \text{chase}_{dobj_{\uparrow}+nsubj_{\uparrow}} \\
nsubj_{\downarrow}(\text{chase}^{\circ}_{dobj_{\uparrow}}, \text{cat}) &= \text{cat}_{dobj_{\downarrow}+nsubj_{\downarrow}}
\end{aligned} \quad (8)$$

In Equation 8, the verb *chase* is first restricted by *mouse* at the direct object position, and then by its subject *cat*. In addition, this noun is restricted by the vector  $\text{chase}^{\circ}_{dobj_{\downarrow}}$ , which represents the contextualized selectional preferences built by combining  $\text{mouse}_{dobj_{\downarrow}}$  with the vectors of the nouns that are in the subject position of *chase* (except *cat*). This new compositional vector represents a very contextualized nominal concept: “the cat that chased a mouse”. The word *cat* and its specific sense can be related to anaphorical expressions by making use of co-referential relationships at the discourse level: e.g., pronoun *it*, other definite expressions (“that cat”, “the cat”), and so on.

## 4 Experiments

We carried out a corpus-based experiment based on compositional distributional similarity to check the quality of composite expressions, namely NOUN-VERB-NOUN constructions (NVN) incre-

mentally composed with *nsubj* and *dobj* dependencies.

#### 4.1 The Corpus and the Structured Vector Model

Our working corpus consists of both the English Wikipedia (dump file of November 2015<sup>1</sup>) and the British National Corpus (BNC)<sup>2</sup>. In total, the corpus contains about 2.5 billion word tokens. We used the rule-based dependency parser DepPattern (Gamallo and González, 2011; Gamallo, 2015) to perform syntactic analysis on the whole text.

Word vectors were built by computing their co-occurrences in syntactic contexts. Two different types of vectors were built from the corpus: nominal and verbal vectors. Then, for each word we filtered out non relevant contexts using simple count-based techniques inspired by those described in (Bordag, 2008; Padró et al., 2014; Gamallo, 2016), where matrices are stored in hash tables with only non-zero values. More precisely, the association between words and their contexts were weighted with the Dunning’s likelihood ratio (Dunning, 1993) and then, for each word, only the  $N$  contexts with highest likelihood scores were stored in the hash table (where  $N = 500$ ). So, the remaining contexts were removed from the hash (in standard vector/matrix representations, instead of removing contexts we should assign them zero values).

The process of matrix reduction resulted in the selection of 330,953 nouns (most of them proper names) with 236,708 different nominal contexts; and 6,618 verbs with 140,695 different verbal contexts. As the contexts of nouns and verbs are not compatible, we created two different vector spaces. Words and their contexts were stored in two hashes, one per vector space, which represent matrices containing only non-zero values. To build compositional vectors from these matrices, the strategy defined in the previous section was implemented in PERL giving rise to the software *Depfunc*<sup>3</sup>. Distributional similarity between pairs of composite expressions was performed using *Cosine*.

<sup>1</sup><https://dumps.wikimedia.org/enwiki/>

<sup>2</sup><http://www.natcorp.ox.ac.uk>

<sup>3</sup>Software and models are available at <http://gramatica.usc.es/~gamallo/prototypes.htm>

#### 4.2 NVN Composite Expressions

This experiment consists of evaluating the quality of compositional vectors built by means of the consecutive application of head and dependency functions associated with nominal subject and direct object. The experiment is performed on the dataset developed by Grefenstette and Sadrzadeh (2011a). The dataset was built using transitive verbs paired with subjects and direct objects: NVN composites.

Given our compositional strategy, we are able to compositional build several vectors that somehow represent the meaning of the whole NVN composite expression. Take the expression “*the coach runs the team*”. If we follow the left-to-right strategy (noted *nv-n*), at the end of the compositional process, we obtain two fully contextualized senses:

**nv-n\_head** The sense of the head *run*, as a result of being contextualized first by the preferences imposed by the subject and then by the preferences required by the direct object. We note *nv-n\_head* the final sense of the head in a NVN composite expression following the left-to-right strategy.

**nv-n\_dep** The sense of the object *team*, as a result of being contextualized by the preferences imposed by *run* previously combined with the subject *coach*. We note *nv-n\_dep* the final sense of the direct object in a NVN composite expression following the left-to-right strategy.

If we follow the right-to-left strategy (noted *n-nv*), at the end of the compositional process, we obtain two fully contextualized senses:

**n-nv\_head** The sense of the head *run* as a result of being contextualized first by the preferences imposed by the object and then by the subject.

**n-nv\_dep** The sense of the subject *coach*, as a result of being contextualized by the preferences imposed by *run* previously combined with the object *team*.

Table 1 shows the Spearman’s correlation values ( $\rho$ ) between individual human similarity scores and the similarity values predicted by the different versions built from our *Depfunc* system. The best score was achieved by averaging the

<i>Systems</i>	$\rho$
non-compositional (V)	0.27
Depfunc (nv_head)	0.33
Depfunc (nv_dep)	0.19
Depfunc (vn_head)	0.36
Depfunc (vn_dep)	0.38
Depfunc (nv-n_head+dep)	0.35
Depfunc (nv-n_head)	0.33
Depfunc (nv-n_dep)	0.20
Depfunc (n-vn_head+dep)	<b>0.46</b>
Depfunc (n-vn_head)	0.36
Depfunc (n-vn_dep)	0.42
Depfunc (n-vn+nv-n)	0.44
Grefenstette and Sadrzadeh (2011)	0.28
Hashimoto and Tsuruoka (2014)	0.43
Polajnar et al. (2015)	0.35

Table 1: Spearman correlation for transitive expressions using the benchmark by Grefenstette and Sadrzadeh (2011)

head and dependent similarity values derived from the *n-vn* (right-to-left) strategy. Let us note that, for NVN composite expressions, the left-to-right strategy seems to build less reliable compositional vectors than the right-to-left counterpart. Besides, the combination of the two strategies (*n-vn+nv-n*) does not improve the results of the best one (*n-vn*).<sup>4</sup>

The score value obtained by our *n-vn\_head+dep* right-to-left strategy outperforms other systems tested for this dataset: Grefenstette and Sadrzadeh (2011b); Polajnar et al. (2015), which are two works based on the categorical compositional distributional model of meaning of Coecke et al. (2010), and the neural network strategy described in Hashimoto and Tsuruoka (2015).

At the top of Table 1, we show the non-compositional baseline we have created for this dataset: similarity between single verbs. The table also shows four intermediate values resulting from comparing partial compositional constructions: the noun-verb (*nv\_head* and *nv\_dep*) and the verb-noun (*vn\_head* and *vn\_dep*) combinations. Two interesting remarks can be made from these values when they are compared with the full compositional constructions.

First, there is no clear improvement of performance if we compare the full compositional information of the two transitive constructions with the partial combinations. On the one hand, the full *nv-n* construction does not improve the scores obtained by the partial intransitive *nv*. On the other

<sup>4</sup>*n-vn+nv-n* is computed by averaging the similarities of both *n-vn\_head+dep* and *nv-n\_head+dep*

hand, *n-vn* performs slightly better than *vn* but only in the case of the dependent function which makes use of contextualized selectional preferences:  $n-vn\_dep = 0.42 / vn\_dep = 0.38$ . The low performance at the second level of composition might call into question the use of contextualized vectors to build still more contextualized senses. The scarcity problem derived from the recursive combination of contextualized vectors is an important issue which could be faced with more corpus, and which we should analyze with more complex evaluation tests.

The second remark is about the difference between the two algorithms: left-to-right and right-to-left. The scores achieved by the left-to-right algorithm (*nv*, *nv-n*) are clearly below those achieved by right-to-left (*vn*, *n-vn*). This might be due to the weak semantic motivation of the selectional preferences involved in the subject dependency of transitive constructions in comparison to the direct object one. In fact, right-to-left and left-to-right function application produces quite different vectors because each algorithm corresponds to a particular hierarchy of constituents. Change of constituency implies different semantic entailments such as we can easily observe if we consider the different levels of constituency of noun modifiers (e.g. “*fastest American runner*”  $\neq$  “*American fastest runner*”). Finally, the poor results of *nv* in this dataset might be explained because the subject role is less meaningful in transitive clauses than in intransitive ones. The subject of intransitive clauses is assigned a complex semantic role that tends to merge the notions of agent and patient. By contrast, the subject of transitive constructions tends to be just the agent of an action with an external patient.

## 5 Conclusions

In this paper, we described a distributional compositional model based on a transparent and syntactically structured vector space. The combination of two related lemmas gives rise to two vectors which represent the senses of the two contextualized lemmas. This process can be repeated until no syntactic dependency is found in the analyzed composite expression. The compositional interpretation of a composite expression builds the sense of each constituent lemma in an incremental way.

Substantial problems still remain unsolved. For instance there is no clear borderline between

compositional and non-compositional expressions (collocations, compounds, or idioms). It seems to be obvious that vectors of full compositional units should be built by means of compositional operations and predictions based on their constituent vectors. It is also evident that vectors of entirely frozen expressions should be totally derived from corpus co-occurrences of the whole expressions without considering internal constituency. However, there are many expressions, in particular collocations (such as “*save time*”, “*go mad*”, “*heavy rain*”, . . .) which can be considered as both compositional and non-compositional. In those cases, it is not clear which is the best method to build their distributional representation: predicted vectors by compositionality or corpus-observed vectors of the whole expression?

Another problem that has not been considered is how to represent the semantics of some grammatical words, namely determiners and auxiliary verbs (i.e., noun and verb specifiers). For this purpose, we think that it would be required a different functional approach, probably closer to the work described by Baroni (2014), who defines functions as linear transformations on vector spaces.

Finally, as we have outlined above, generated vectors tend to be too scarce when they are derived from the recursive combination of already contextualized vectors. Further experiments with more complex phrases and larger training corpora are required in order to deeply analyse this issue. For this purpose, we will explore the strategy defined in Kober et al. (2016) to improve sparse distributional representations.

In current work, we are defining richer semantic word models by combining WordNet features with semantic spaces based on distributional contexts (Gamallo and Pereira-Fariña, 2017). This hybrid method might also help overcome scarcity.

## Acknowledgments

This work has received financial support from a 2016 BBVA Foundation Grant for Researchers and Cultural Creators, TelePares (MINECO, ref:FFI2014-51978-C2-1-R), the Consellería de Cultura, Educación e Ordenación Universitaria (accreditation 2016-2019, ED431G/08) and the European Regional Development Fund (ERDF).

## References

- Marco Baroni. 2013. Composition in distributional semantics. *Language and Linguistics Compass* 7:511–522.
- Marco Baroni, Raffaella Bernardi, and Roberto Zamparelli. 2014. Frege in space: A program for compositional distributional semantics. *LiLT* 9:241–346.
- 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*. Stroudsburg, PA, USA, EMNLP’10, pages 1183–1193.
- Stefan Bordag. 2008. A Comparison of Co-occurrence and Similarity Measures as Simulations of Context. In *9th CICLing*, pages 52–63.
- B. Coecke, M. Sadrzadeh, and S. Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis* 36(1-4):345–384.
- Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19(1):61–74.
- Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of EMNLP*. Honolulu, HI.
- Katrin Erk and Sebastian Padó. 2009. Paraphrase assessment in structured vector space: Exploring parameters and datasets. In *Proceedings of the EACL Workshop on Geometrical Methods for Natural Language Semantics*. Athens, Greece.
- Pablo Gamallo. 2015. Dependency parsing with compression rules. In *International Workshop on Parsing Technology (IWPT 2015)*. Bilbao, Spain.
- Pablo Gamallo. 2016. Comparing explicit and predictive distributional semantic models endowed with syntactic contexts. *Language Resources and Evaluation* First online: 13 May 2016.
- Pablo Gamallo and Isaac González. 2011. A grammatical formalism based on patterns of part-of-speech tags. *International Journal of Corpus Linguistics* 16(1):45–71.
- Pablo Gamallo and Martín Pereira-Fariña. 2017. Compositional semantics using feature-based models from wordnet. In *1st Workshop on Sense, Concept and Entity Representations and their Applications, co-located at 15th Conference of the European Chapter of the Association for Computational Linguistics*. Association of Computational Linguistics (ACL), pages 1–10.
- Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011a. Experimental support for a categorical compositional distributional model of meaning. In *Conference on Empirical Methods in Natural Language Processing*.



- Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011b. Experimenting with transitive verbs in a dis-co-cat. In *Workshop on Geometrical Models of Natural Language Semantics (EMNLP-2011)*.
- Edward Grefenstette, Mehrnoosh Sadrzadeh, Stephen Clark, Bob Coecke, and Stephen Pulman. 2011. Concrete sentence spaces for compositional distributional models of meaning. In *Proceedings of the Ninth International Conference on Computational Semantics*. IWCS '11, pages 125–134.
- 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*. GEMS '10.
- Kazuma Hashimoto and Yoshimasa Tsuruoka. 2015. Learning embeddings for transitive verb disambiguation by implicit tensor factorization. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*. Association for Computational Linguistics, Beijing, China, pages 1–11. <http://www.aclweb.org/anthology/W15-4001>.
- Dimitri Kartsaklis. 2014. Compositional operators in distributional semantics. *Springer Science Reviews* 2(1-2):161–177.
- Dimitri Kartsaklis, Nal Kalchbrenner, and Mehrnoosh Sadrzadeh. 2014. Resolving lexical ambiguity in tensor regression models of meaning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Vol. 2: Short Papers)*. Association for Computational Linguistics, Baltimore, USA, pages 212–217.
- Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. 2013. Prior disambiguation of word tensors for constructing sentence vectors. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*.
- Thomas Kober, Julie Weeds, Jeremy Reffin, and David J. Weir. 2016. Improving sparse word representations with distributional inference for semantic composition. In *Proceedings of EMNLP 2016, Austin, Texas, USA*. pages 1691–1702. <http://aclweb.org/anthology/D/D16/D16-1175.pdf>.
- Jayant Krishnamurthy and Tom Mitchell. 2013. *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, Association for Computational Linguistics, chapter Vector Space Semantic Parsing: A Framework for Compositional Vector Space Models, pages 1–10.
- Oren Melamud, Ido Dagan, and Jacob Goldberger. 2015. Modeling word meaning in context with substitute vectors. In *NAACL HLT 2015, Denver, Colorado, USA*. pages 472–482. <http://aclweb.org/anthology/N/N15/N15-1050.pdf>.
- Jeff Mitchell and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*. pages 236–244.
- Jeff Mitchell and Mirella Lapata. 2009. Language models based on semantic composition. In *Proceedings of EMNLP*. pages 430–439.
- Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science* 34(8):1388–1439.
- Muntsa Padró, Marco Idiart, Aline Villavicencio, and Carlos Ramisch. 2014. Nothing like good old frequency: Studying context filters for distributional thesauri. In *Proceedings of EMNLP 2014, Doha, Qatar*. pages 419–424.
- Nghia The Pham, Germán Kruszewski, Angeliki Lazaridou, and Marco Baroni. 2015. Jointly optimizing word representations for lexical and sentential tasks with the C-PHRASE model. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. pages 971–981. <http://aclweb.org/anthology/P/P15/P15-1094.pdf>.
- Tamara Polajnar, Laura Rimell, and Stephen Clark. 2015. An exploration of discourse-based sentence spaces for compositional distributional semantics. In *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*. Association for Computational Linguistics, Lisbon, Portugal, pages 1–11. <http://aclweb.org/anthology/W15-2701>.
- Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the EMNLP-CoNLL'12*. Association for Computational Linguistics, Stroudsburg, PA, USA, pages 1201–1211. <http://dl.acm.org/citation.cfm?id=2390948.2391084>.
- Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2010. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA, pages 948–957.
- David J. Weir, Julie Weeds, Jeremy Reffin, and Thomas Kober. 2016. Aligning packed dependency trees: A theory of composition for distributional semantics. *Computational Linguistics* 42(4):727–761.
- 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.