A Generalisation of Lexical Functions for Composition in Distributional Semantics

Antoine Bride
IRIT & Université de Toulouse
antoine.bride@irit.fr

Tim Van de Cruys
IRIT & CNRS, Toulouse
tim.vandecruys@irit.fr

Nicholas Asher
IRIT & CNRS, Toulouse
nicholas.asher@irit.fr

Abstract

Over the last two decades, numerous algorithms have been developed that successfully capture something of the semantics of single words by looking at their distribution in text and comparing these distributions in a vector space model. However, it is not straightforward to construct meaning representations beyond the level of individual words – i.e. the combination of words into larger units – using distributional methods. Our contribution is twofold. First of all, we carry out a large-scale evaluation, comparing different composition methods within the distributional framework for the cases of both adjective-noun and noun-noun composition, making use of a newly developed dataset. Secondly, we propose a novel method for composition, which generalises the approach by Baroni and Zamparelli (2010). The performance of our novel method is also evaluated on our new dataset and proves competitive with the best methods.

1 Introduction

In the course of the last two decades, there has been a growing interest in distributional methods for lexical semantics (Landauer and Dumais, 1997; Lin, 1998; Turney and Pantel, 2010). These methods are based on the distributional hypothesis (Harris, 1954), according to which words that appear in the same contexts tend to be similar in meaning. Inspired by Harris' hypothesis, numerous researchers have developed algorithms that try to capture the semantics of individual words by looking at their distribution in a large corpus.

Compared to manual studies common to formal semantics, distributional semantics offers substantially larger coverage since it is able to analyze massive amounts of empirical data. However, it is not trivial to combine the algebraic objects created by distributional semantics to get a sensible distributional representation for more complex expressions, consisting of several words. On the other hand, the formalism of the λ-calculus provides us with general, advanced and efficient methods for composition that can model meaning composition not only of simple phrases, but also more complex phenomena such as coercion or composition with fine-grained types (Asher, 2011; Luo, 2010; Bassac et al., 2010). Despite continued efforts to find a general method for composition and various approaches for the composition of specific syntactic structures (e.g. adjective-noun composition, or the composition of transitive verbs and direct objects (Mitchell and Lapata, 2008; Coecke et al., 2010; Baroni and Zamparelli, 2010)), the modeling of compositionality is still an important challenge for distributional semantics. Moreover, the validation of proposed methods for composition has used relatively small datasets of human similarity judgements (Mitchell and Lapata, 2008).[1] Although such studies comparing similarity judgements have their merits, it would be interesting to have studies that evaluate methods for composition on a larger scale, using a larger test set of different specific compositions. Such an evaluation would allow us to evaluate more thoroughly the different methods of composition that have been proposed. This is one of the goals of this paper.

To achieve this goal, we make use of two different resources. We have constructed a dataset for French containing a large number of pairs of a compositional expression (adjective-noun) and a single noun that is semantically close or identical to the composed expression. These pairs have been extracted semi-automatically from

[1] A notable exception is (Marelli et al., 2014), who propose a large-scale evaluation dataset for composition at the sentence level.

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 281–291,
Beijing, China, July 26-31, 2015.

the French Wiktionary. We have also used the Semeval 2013 dataset of phrasal similarity judgements for English with similar pairs extracted semi-automatically from the English Wiktionary to construct a dataset for English for both adjective-noun and noun-noun composition. This affords us a cross-linguistic comparison of the methods.

These data sets provide a substantial evaluation of the performance of different compositional methods. We have tested three different methods of composition proposed in the literature, viz. the additive and multiplicative model (Mitchell and Lapata, 2008), as well as the lexical function approach (Baroni and Zamparelli, 2010).

The two first methods are entirely general, and take as input automatically constructed vectors for adjectives and nouns. The method by Baroni and Zamparelli, on the other hand, requires the acquisition of a particular function for each adjective, represented by a matrix. The second goal of our paper is to generalise the functional approach in order to eliminate the need for an individual function for each adjective. To this goal, we automatically learn a generalised lexical function, based on Baroni and Zamparelli's approach. This generalised function combines with an adjective vector and a noun vector in a generalised way. The performance of our novel generalised lexical function approach is evaluated on our test sets and proves competitive with the best, extant methods.

Our paper is organized as follows. First, we discuss the different compositional models that we have evaluated in our study, briefly revisiting the different existing methods for composition, followed by a description of our generalisation of the lexical function approach. Next, we report on our evaluation method and its results. The results section is followed by a section that discusses work related to ours. Lastly, we draw conclusions and lay out some avenues for future work.

2 Composition methods

2.1 Simple Models of Composition

In this section, we describe the composition models for the adjective-noun case. The extension of these models to the noun-noun case is straightforward; one just needs to replace the adjective by the subordinate noun. Admittedly, choosing which noun is subordinate in noun-noun composition may be an interesting problem but it is outside the scope of this paper. We tested three simple models of composition: a baseline method that discounts the contribution of the adjective completely, and the additive and multiplicative models of composition. The baseline method is defined as follows:

$$\text{Comp}_{\text{baseline}}(\text{adj}, \text{noun}) = \mathbf{noun}$$

The additive model adds the point-wise values of the adjective vector **adj** and noun vector **noun** using independent coefficients to provide a result for the composition:

$$\text{Comp}_{\text{additive}}(\text{adj}, \text{noun}) = \alpha\,\mathbf{noun} + \beta\,\mathbf{adj}$$

The multiplicative model consists in a point-wise multiplication of the vectors **adj** and **noun**:

$$\text{Comp}_{\text{multiplicative}}(\text{adj}, \text{noun}) = \mathbf{noun} \otimes \mathbf{adj}$$
$$\text{with} \quad (\mathbf{noun} \otimes \mathbf{adj})_i = \mathbf{noun}_i \times \mathbf{adj}_i$$

2.2 The lexical function model

Baroni and Zamparelli's (2010) lexical function model (LF) is somewhat more complex. Adjective-noun composition is modeled as the functional application of an adjective meaning (represented as a matrix) to a noun meaning (represented as a vector). Thus, the combination of an adjective and noun is the product of the matrix **ADJ** and the vector **noun** as shown in Figure 1.

Baroni and Zamparelli propose learning an adjective's matrix from examples of the vectors for **adj_noun** obtained directly from the corpus. These vectors **adj_noun** are obtained in the same way as vectors representing a single word: when the adjective-noun combination occurs, we observe its context and construct the vector from those observations. As an illustration, consider the example in 2. The word *name* appears three times modified by an adjective in the following excerpt from Oscar Wilde's *The Importance of Being Earnest*. This informs us about the co-occurrence frequencies of three vectors: one for **divine_name**, another for **nice_name**, and one for **charming_name**.

Once the **adj_noun** vectors have been created for a given adjective, we are able to calculate the **ADJ** matrix using a least squares regression that minimizes the equation $\mathbf{ADJ} \times \mathbf{adj\_noun} - \mathbf{noun}$. More formally, the problem is the following:

Find **ADJ** s.t.
$\sum_{\text{noun}} (\mathbf{ADJ} \times \mathbf{noun} - \mathbf{adj\_noun})^2$
is minimal

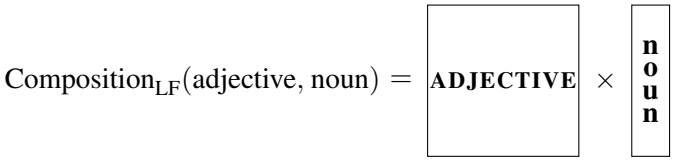

Figure 1: Lexical Function Composition

Jack: Personally, darling, to speak quite candidly, I don't much care about the name of Ernest . . . I don't think the name suits me at all.
Gwendolen: It suits you perfectly. It is a divine [name]. It has a music of its own. It produces vibrations.
Jack: Well, really, Gwendolen, I must say that I think there are lots of other much nicer [names]. I think Jack, for instance, is a charming [name].

Figure 2: Excerpt from Oscar Wilde's *The Importance of Being Earnest*

For our example, we would minimize, among others **DIVINE**×**divine_name** − **name** to get the matrix for **DIVINE**.

LF requires a large corpus, because we have to observe a sufficient number of examples of the adjective and noun combined, which are perforce less exemplified than the presence of the noun or adjective in isolation. In Figure 2, each of the occurrences of 'name' can contribute to the information in the vector **name** but none can contribute to the vector **evanescent_name**.

Baroni and Zamparelli (2010) offer an explanation of how to cope with the potential sparse data problem for learning matrices for adjectives. Moreover, recent evaluations of LF show that existent corpora have enough data for it to provide a semantics for the most frequent adjectives and obtain better results than other methods (Dinu et al., 2013b).

Nevertheless, LF has limitations in treating relatively rare adjectives. For example, the adjective 'evanescent' appears 359 times in the UKWaC corpus (Baroni et al., 2009). This is enough to generate a vector for **evanescent**, but may not be sufficient to generate a sufficient number of vectors **evanescent_noun** to build the matrix **EVANESCENT**. More importantly, for noun-noun combinations, one may need to have a LF for a combination. To get the meaning of *blood donation campaign* in the LF approach, the matrix **BLOOD_DONATION** must be combined to the vector **campaign**. Learning this matrix would require to build vectors **blood_donation_noun** for many nouns. Even if it were possible, the issue would arise again for *blood donation campaign plan*, then for *blood donation campaign plan meeting* and so forth.

In addition, LF's approach to adjectival meaning and composition has a theoretical drawback. Like Montague Grammar, it supposes that the effect of an adjective on a noun meaning is specific to the adjective (Kamp, 1975). However, recent studies suggest that the Montague approach overgeneralises from the worst case, and that the vast majority of adjectives in the world's languages are subsective, suggesting that the modification of nominal meaning that results from their composition with a noun follows general principles (Partee, 2010; Asher, 2011) that are independent of the presence or absence of examples of association.

2.3 Generalised LF

To solve these problems, we generalise LF and replace individual matrices for adjectival meanings by a single lexical function: a tensor for adjectival composition $\mathscr{A}$.[2] Our proposal is that adjective-noun composition is carried out by multiplying the tensor $\mathscr{A}$ with the vector for the adjective **adj**, followed by a multiplication with the vector **noun**, *c.f.* Figure 3.

The product of the tensor $\mathscr{A}$ and the vector **adj** yields a matrix dependent of the adjective that is multiplied with the vector **noun**. This matrix corresponds to the LF matrix **ADJ**. As indicated in Figure 4, we obtain $\mathscr{A}$ with the help of matrices obtained from the LF approach, and from vectors for single words easily obtained in distributional semantics; we perform a least square regression minimizing the norm of the matrices generated by the equations in Figure 4. Formally, the problem is

[2] A tensor generalises a matrix to several dimensions. We use a tensor in three modes. For an introduction to tensors, see (Kolda and Bader, 2009).

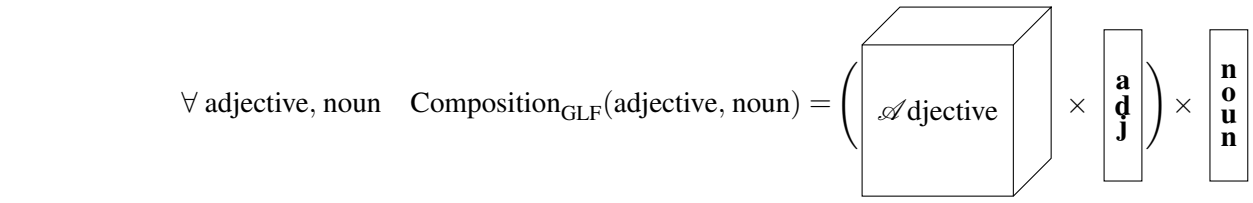

Figure 3: Composition in the generalised lexical function model

Find $\mathscr{A}$ s.t.
$\sum_{adj}(\mathscr{A} \times \mathbf{adj} - \mathbf{ADJ})^2$
is minimal

Note that our tensor is not just the compilation of the information found in the LF matrices: the adjective mode of our tensor has a limited number of dimensions, whereas the LF approach creates a separate matrix for each individual adjective. This reduction forces the model to generalise, and we hypothesise that this generalisation allows us to make proper noun modifications even in the light of sparse data.

Our approach requires learning a significant number of matrices **ADJ**. This is not a problem, since FRWaC and UKWaC provide sufficient data for the LF approach to generate matrices for a significant number of adjectives. For example, the 2000th most frequent adjective in FRWaC ('fasciste') has more than 4000 occurrences.

To return to our example of *blood donation campaign*, once the tensor $\mathscr{N}$ for noun-noun composition is learned, our approach requires only the knowledge of the vectors **blood**, **donation** and **campaign**. We would then perform the following computations:

$\mathbf{blood\_donation} = (\mathscr{N} \times \mathbf{blood}) \times \mathbf{donation}$
$\mathbf{blood\_donation\_campaign} =$
$(\mathscr{N} \times \mathbf{blood\_donation}) \times \mathbf{campaign}$

and this allows us to avoid the sparse data problem for the LF approach in generating the matrix **BLOOD_DONATION**.

Once we have obtained the tensor $\mathscr{A}$, we verify experimentally its relevance to composition, in order to check whether a tensor optimising the equations in Figure 4 would be semantically interesting.

3 Evaluation

3.1 Tasks description

In order to evaluate the different composition methods, we constructed test sets for French and English, inspired by the work of Zanzotto et al. (2010) and the SEMEVAL-2013 task *evaluating phrasal semantics* (Korkontzelos et al., 2013). The task is to make a judgement about the semantic similarity of a short word sequence (an adjective-noun combination) and a single noun. This is important, as composition models need to be able to treat word sequences of arbitrary length. Formally, the task is presented as:

With $\mathbf{comp} = \text{composition}(\text{adj}, \text{noun}_1)$
Evaluate $\text{similarity}(\mathbf{comp}, \mathbf{noun_2})$

where the 'composition' function is carried out by the different composition models. 'Similarity' needs to be a binary function, with return values 'similar' and 'non-similar'. Note, however, that the distributional approach yields a continuous similarity value (such as the cosine similarity between two vectors). In order to determine which cosine values correspond to 'similar' and which cosine values correspond to 'non-similar', we looked at a number of examples from a development set. More precisely, we carried out a logistic regression on 50 positive and 50 negative examples (separate from our test set) in order to automatically learn the threshold at which a pair is considered to be similar. Finally, we decided to use balanced test sets containing as many positive instances as negative ones.

The test set is constructed in a semi-automatic way, making use of the canonical phrasing of dictionary definitions. Take for example the definition of *bassoon* in the English Wiktionary[3], presented in Figure 5. It is quite straightforward to extract the pair (*musical_instrument*, *bassoon*) from this definition. Using a large dictionary (such as Wiktionary), it is then possible to extract a large number of positive – i.e. similar – (`adjective_noun`, `noun`) pairs.

For the construction of our test set for French, we downloaded all entries of the French Wiktionary (*Wiktionnaire*) and annotated them with

[3] `http://en.wiktionary.org/wiki/bassoon`, accessed on 26 February 2015.

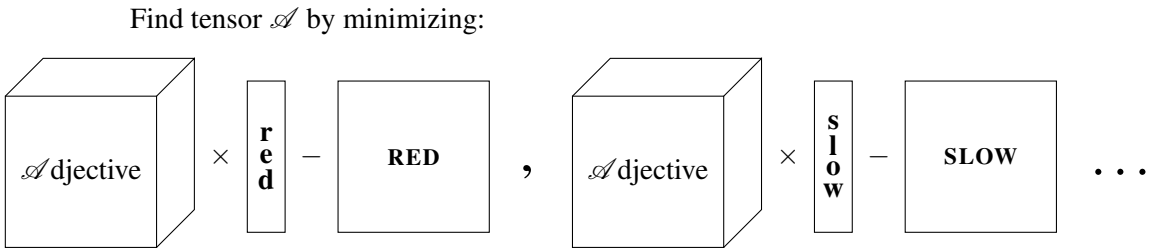

Figure 4: Learning the 𝒜djective tensor

bassoon /bəˈsuːn/ (plural bassoons)

1. A musical instrument in the woodwind family, having a double reed and, playing in the tenor and bass ranges.

Figure 5: Definition of *bassoon*, extracted from the English Wiktionary

part of speech tags, using the French part of speech tagger MElt (Denis et al., 2010). Next, we extracted all definitions that start with an adjective-noun combination. As a final step, we filtered all instances containing words that appear too infrequently in our FRWaC corpus.[4]

The automatically extracted instances were then checked manually, and all instances that were considered incorrect were rejected. This gave us a final test set of 714 positive examples.

We also created an initial set of negative examples, where we combined an existing combination of `adjective_noun1` (extracted from the French Wiktionary), with a randomly selected noun `noun2`. Again, we verified manually that the resulting (`adjective_noun1`, `noun2`) pairs constituted actual negative examples. We then created a second set of negative examples by randomly selecting two nouns (`noun1`,`noun2`) and one adjective `adjective`. The resulting pairs (`adjective_noun1`, `noun2`) were verified manually.

In addition to our new test set for French, we also experimented with the original test set of the SEMEVAL-2013 task *evaluation phrasal semantics* for English. However, the original test set lacked human oversight as 'manly behavior' was considered similar to 'testosterone' for example. We thus hand-checked the test set ourselves and extracted 652 positive pairs.

The negative pairs from the original SEMEVAL-2013 are a combination of a random noun and a random adjective-noun compositon found in the English Wiktionary. We used it as our first set of English negative examples as it is similar in construction to our first set of negative examples in French. In addition, we created a completely random negative test set for English in the same fashion we did for the second negative test set for French.

Finally, the original test set also contains noun-noun compounds so we also created a test set for that. This gave us 226 positive and negative pairs for the noun-noun composition.

3.2 Semantic space construction

In this section, we describe the construction of our semantic space. Our semantic space for French was built using the FRWaC corpus (Baroni et al., 2009) – about 1,6 billion words of web texts – which has been tagged with MElt tagger (Denis et al., 2010) and parsed with MaltParser (Nivre et al., 2006a), trained on a dependency-based version of the French treebank (Candito et al., 2010). Our semantic space for English has been built using the UKWaC corpus (Baroni et al., 2009), which consists of about 2 billion words extracted from the web. The corpus has been part of speech tagged and lemmatized with Stanford Part-Of-Speech Tagger (Toutanova and Manning, 2000; Toutanova et al., 2003), and parsed with MaltParser (Nivre et al., 2006b) trained on sections 2-21 of the Wall Street Journal section of the Penn Treebank extended with about 4000 ques-

[4] *i.e.* less than 200 times for adjectives and less than 1500 times for nouns

positive examples	random negative examples	Wiktionary-based negative examples
(*mot_court*, *abréviation*)	(*importance_fortuit*, *gamme*)	(*jugement_favorable*, *discorde*)
'short word', 'abbreviation'	'accidental importance', 'range'	'favorable judgement', 'discord'
(*ouvrage_littéraire*, *essai*)	(*penchant_autoritaire*, *ile*)	(*circonscription_administratif*, *fumier*)
'literary work', 'essay'	'authoritarian slope', isle'	'administrative district', 'manure'
(*compagnie_honorifique*, *ordre*)	(*auspice_aviaire*, *ponton*)	(*mention_honorable*, *renne*)
'honorary company', 'order'	'avian omen', 'pontoon'	'honorable mention', 'reindeer'

Table 1: A number of examples from our test set for French

tions from the QuestionBank[5].

For both corpora, we extracted the lemmas of all nouns, adjectives and (bag of words) context words. We only kept those lemmas that consist of alphabetic characters.[6] We then selected the 10K most frequent lemmas for each category (nouns, adjectives, context words), making sure to include all the words from the test set. As a final step, we created our semantic space vectors using adjectives and nouns as instances, and bag of words context words as features. The resulting vectors were weighted using *positive point-wise mutual information* (*ppmi*, (Church and Hanks, 1990)), and all vectors were normalized to unit length.

We then compared the different composition methods on different versions of the same semantic space (both for French and English): the full semantic space, a reduced version of the space to 300 dimensions using singular value decomposition (*svd*, (Golub and Van Loan, 1996)), and a reduced version of the space to 300 dimensions using non-negative matrix factorization (*nmf*, (Lee and Seung, 2000)). We did so in order to test each method in its optimal conditions. In fact:

- A non-reduced space contains more information. This might be beneficial for methods that are able to take advantage of the full semantic space (viz. the additive et multiplicative model). On the other hand, to be able to use the non-reduced space for the lexical function approach, one would have to learn matrices of size 10K $\times$10K for each adjective. This would be problematic in terms of computing time and data sparseness, as we previously noted. The same goes for our generalised approach.
- Previous research has indicated that the lexical function approach is able to achieve better results using a reduced space with *svd*. On the other hand, the negative values that result from *svd* are detrimental for the multiplicative approach.
- An *nmf*-reduced semantic space is not detrimental for the multiplicative approach.

In order to determine the best parameters for the additive model, we tested this model for different values of α and β where $\alpha+\beta=1$[7] on a development set and kept the values with the best results: $\alpha = 0.4,\ \beta = 0.6$.

3.3 Data used for regression

The LF approach and its generalisation need data in order to perform the least square regression. We thus created a semantic space for **adjective_noun** and **noun_noun** vectors using the most frequent ones in a similar way to how we created them in 3.2. Then we solved the equations in 2.2 and forth. Even though the regression data were disjoint from the test sets, for each pair, we removed some of the data that may cause overfitting.

For the lexical function tests, we remove the **adjective_noun** vector corresponding to the test pair from the regression data. For example, we do not use **short_word** to learn **SHORT** for the (short_word, abbrevation) pair.

For the generalised lexical function tests, we use the full regression data to learn the lexical functions used to train the tensor. However, we remove the **ADJECTIVE** matrix corresponding to the test pair from the (tensor) regression data. For example, we do not use **SHORT** to learn $\mathscr{A}$ for the (short_word, abbreviation) pair.

[5] `http://maltparser.org/mco/english_parser/engmalt.html`

[6] This step generally filters out dates, numbers and punctuation, which have little interest for the distributional approach.

[7] Since the vectors are normalized (*cf.* 3.2), this condition does not affect the generality of our test.

Table 2: Percentage of correctly classified pairs for (adjective_noun1, noun2) for both French and English spaces.

	baseline		multiplicative		additive		LF		generalised LF	
	fr	**en**	**fr**	**en**	**fr**	**en**	**fr**	**en**	**fr**	**en**
non-reduced	0.83	0.81	0.86	0.86	0.88	0.86	N/A		N/A	
svd	0.79	0.79	0.55	0.59	0.84	0.78	**0.93**	**0.92**	0.91	0.88
nmf	0.78	0.78	0.83	0.77	0.79	0.84	0.90	0.86	0.88	0.85

(a) Negative examples are created randomly.

	baseline		multiplicative		additive		LF		generalised LF	
	fr	**en**	**fr**	**en**	**fr**	**en**	**fr**	**en**	**fr**	**en**
non-reduced	0.80	0.79	0.83	0.81	**0.85**	0.80	N/A		N/A	
svd	0.78	0.77	0.54	0.48	0.83	0.78	**0.84**	0.79	0.81	0.77
nmf	0.78	0.78	0.79	0.78	0.83	**0.82**	0.82	**0.82**	0.81	0.80

(b) Negative examples are created from existing pairs.

Table 3: Percentage of correctly classified pairs for (noun2_noun1, noun3) with negative examples from existing pairs. Only the English space is tested.

English space	baseline	multiplicative	additive	LF	generalised LF
non-reduced	0.77	0.80	0.84	N/A	N/A
svd	0.78	0.49	**0.86**	0.83	0.82
nmf	0.79	0.82	**0.86**	**0.85**	0.83

3.4 Results

In this section, we present how the various models perform on our test sets.

3.4.1 General results

Tables 2 & 3 give an overview of the results. Note first that the baseline approach, which compares only the two nouns and ignores the subordinate adjective or noun, does relatively well on the task ($\sim$ 80% accuracy). This reflects the fact that the head noun in our pairs extracted from definitions is close to (and usually a super type of) the noun to be defined.

In addition, we observe that the multiplicative method performs badly, as expected, on the semantic space reduced with *svd*. This confirms the incompatibility of this method with the negative values generated by *svd*. Indeed, multiplying two vectors with negative values term by term may yield a third vector very far away from the other two. Such a combination does not support the subsectivity of most our test pairs. Apart from that, *svd* and *nmf* reductions do not affect the methods much.

Moreover, we observe that the multiplicative model performs better than the baseline but is bested by the additive model. We also see that additive and lexical functions often yield similar performance.

Finally, the generalised lexical function is slightly less accurate than the lexical functions. This is an expected consequence of generalisation. Nevertheless, the generalised lexical function yields sound results confirming our intuition that we can represent adjective-noun (or noun-noun) combinations by one function.

3.4.2 Adjective-noun

With random negative pairs (Table 2a), we observe that the lexical function model obtains the best results for the *svd* space. This result is significantly better than any other method on any of the spaces—e.g.,for French space, $\chi^2 = 33.49$, $p < 0.01$ when compared to the additive model for the non-reduced space which performs second.

However, with non-random negative pairs (Table 2b), LF and the additive model obtain scores that are globally equivalent for their best respec-

tive conditions — in French 0.85 for the additive non-reduced model vs. 0.84 for the LF *svd* model, a difference that is not significant ($\chi^2 = 0.20$, $p < 0.05$).

This seems to indicate that LF is especially efficient at separating out nonsense combinations. This may be caused by the fact that lexical functions learn from actual pairs. Thus, when an adjective_noun combination is bizarre, the ADJECTIVE matrix has not been optimized to interact with the **noun** vector and may lead to complete non-sense — Which is a good thing because humans would analyze the combination as such.

Finally, similar results in French and English confirm the intuition that distributional methods (and its composition models) are independent of the idiosyncrasies of a particular language; in particular they are as efficient for French as for English.

3.4.3 Noun-noun

The noun-noun tests (Table 3) yields similar results to the adjective-noun tests. This is not so surprising since noun noun compounds in English also obey a roughly subsective property: a baseball field is still a field (though a cricket pitch is perhaps not so obviously a pitch). We can see that the accuracy increase from the baseline is higher compared to adjective-noun test on the same exact spaces (Table 2b, right values). This may be due to the fact that the subordinate noun in noun-noun combinations is more important than the adjective subordinate in adjective-noun combination.

4 Related work

Many researchers have already studied and evaluated different composition models within a distributional approach. One of the first studies evaluating compositional phenomena in a systematic way is Mitchell and Lapata's (2008) approach. They explore a number of different models for vector composition, of which vector addition (the sum of each feature) and vector multiplication (the element-wise multiplication of each feature) are the most important. They evaluate their models on a noun-verb phrase similarity task. Human annotators were asked to judge the similarity of two composed pairs (by attributing a certain score). The model's task is then to reproduce the human judgements. Their results show that the multiplicative model yields the best results, along with a weighted combination of the additive and multiplicative model. The authors redid their study using a larger test set in Mitchell and Lapata (2010) (adjective-noun composition was also included), and they confirmed their initial results.

Baroni and Zamparelli (2010) evaluate their lexical function model within a somewhat different context. They evaluated their model by looking at its capacity of reconstructing the **adjective_noun** vectors that have not been seen during training. Their results show that their lexical function model obtains the best results for the reconstruction of the original co-occurrence vectors, followed by the additive model. We observe the same tendency in our evaluation results for French, although our results for English show a different picture. We would like to explore this discordance further in future work.

Grefenstette et al. (2013) equally propose a generalisation of the lexical function model that uses tensors. Their goal is to model transitive verbs, and the way we acquire our tensor is similar to theirs. In fact, they use the LF approach in order to learn VERB_OBJECT matrices that may be multiplied by a **subject** vector to obtain the **subject_verb_object** vector. In a second step, they learn a tensor for each individual verb, which is similar to how we learn our adjective tensor $\mathscr{A}$.

Coecke et al. (2010) present an abstract theoretical framework in which a sentence vector is a function of the Kronecker product of its word vectors, which allows for greater interaction between the different word features. A number of instantiations of the framework – where the key idea is that relational words (e.g. adjectives or verbs) have a rich (multi-dimensional) structure that acts as a filter on their arguments – are tested experimentally in Grefenstette and Sadrzadeh (2011a) and Grefenstette and Sadrzadeh (2011b). The authors evaluated their models using a similarity task that is similar to the one used by Mitchell & Lapata. However, they use more complex compositional expressions: rather than using compositions of two words (such as a verb and an object), they use simple transitive phrases (subject-verb-object). They show that their instantiations of the categorical model reach better results than the additive and multiplicative models on their transitive similarity task.

Socher et al. (2012) present a compositional model based on a recursive neural network. Each

node in a syntactic tree is assigned both a vector and a matrix; the vector captures the actual meaning of the constituent, while the matrix models the way it changes the meaning of neighbouring words and phrases. They use an extrinsic evaluation, using the model for a sentiment prediction task. They show that their model gets better results than the additive, multiplicative, and lexical function approach. Other researchers, however, have published different results. Blacoe and Lapata (2012) evaluated the additive and multiplicative model, as well as Socher et al.'s (2012) approach on two different tasks: Mitchell & Lapata's (2010) similarity task and a paraphrase detection task. They find that the additive and multiplicative models reach better scores than Socher et al.'s model.

Tensors have been used before to model different aspects of natural language. Giesbrecht (2010) describes a tensor factorization model for the construction of a distributional model that is sensitive to word order. And Van de Cruys (2010) uses a tensor factorization model in order to construct a three-way selectional preference model of verbs, subjects, and objects.

5 Conclusion

We have developed a new method of composition and tested it in comparison with different composition methods assuming a distributional approach. We developed a test set for French pairing nouns with adjective noun combinations very similar in meaning from the French Wiktionary. We also used an existing SEMEVAL-2013 set to create a similar test set for English both for adjective noun combination and noun noun combination. Our tests confirm that the lexical function approach by Baroni and Zamparelli performs well compared to other methods of composition, but only when the negative examples are constructed randomly. Our generalised lexical function approach fares almost equally well. It also has the advantage of being constructed from automatically acquired adjectival and noun vectors, and offers the additional advantage of countering data sparseness. However, the lexical function approach claims to perform well on more subtle cases — *e.g.* non-subsective combinations such as *stone lion*. Our test sets does not contain such cases, and so we cannot draw any conclusion on this claim.

In future work, we would like to test different sizes of dimensionality reduction, in order to optimize our generalised lexical function model. Moreover, it is possible that better results may be obtained by proposing multiple generalised lexical functions, rather than a single one. We could, e.g., try to separate the intersective adjectives from non-intersective adjectives. And finally, we would like to further explore the performance of the lexical function model and generalised lexical function model on different datasets, which involve more complex compositional phenomena.

6 Acknowledgments

We thank Dinu et al. (2013a) for their work on the DISSECT toolkit[8], which provides plenty of helpful functions for composition in distributional semantics. We also thank the OSIRIM platform[9] for allowing us to do the computations we needed. Finally, we thank the reviewers of this paper for their insightful comments.

This work is supported by a grant overseen by the French National Research Agency ANR (ANR-14-CE24-0014).

References

Nicholas Asher. 2011. *Lexical Meaning in Context: A Web of Words*. Cambridge University Press.

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, Cambridge, MA, October. Association for Computational Linguistics.

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

Christian Bassac, Bruno Mery, and Christian Retoré. 2010. Towards a Type-theoretical account of lexical semantics. *Journal of Logic, Language and Information*, 19(2):229–245.

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*, pages 546–556, Jeju Island, Korea, July. Association for Computational Linguistics.

[8] `http://clic.cimec.unitn.it/composes/toolkit/`

[9] `http://osirim.irit.fr/site/en`

Marie Candito, Benoît Crabbé, Pascal Denis, et al. 2010. Statistical french dependency parsing: treebank conversion and first results. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, pages 1840–1847.

Kenneth W. Church and Patrick Hanks. 1990. Word association norms, mutual information & lexicography. *Computational Linguistics*, 16(1):22–29.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributed model of meaning. *Lambek Festschrift, Linguistic Analysis, vol. 36*, 36.

Pascal Denis, Benoît Sagot, et al. 2010. Exploitation d'une ressource lexicale pour la construction d'un étiqueteur morphosyntaxique état-de-l'art du français. In *Traitement Automatique des Langues Naturelles: TALN 2010*.

Georgiana Dinu, Nghia The Pham, and Marco Baroni. 2013a. Dissect - distributional semantics composition toolkit. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 31–36, Sofia, Bulgaria, August. Association for Computational Linguistics.

Georgiana Dinu, Nghia The Pham, and Marco Baroni. 2013b. General estimation and evaluation of compositional distributional semantic models. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 50–58, Sofia, Bulgaria, August. Association for Computational Linguistics.

Eugenie Giesbrecht. 2010. Towards a matrix-based distributional model of meaning. In *Proceedings of the NAACL HLT 2010 Student Research Workshop*, pages 23–28. Association for Computational Linguistics.

Gene H. Golub and Charles F. Van Loan. 1996. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA.

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

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011b. Experimenting with transitive verbs in a discocat. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 62–66, Edinburgh, UK, July. Association for Computational Linguistics.

E. Grefenstette, G. Dinu, Y.-Z. Zhang, M. Sadrzadeh, and Baroni M. 2013. Multi-step regression learning for compositional distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS)*, pages 131–142, East Stroudsburg PA. Association for Computational Linguistics.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Hans Kamp. 1975. Two theories about adjectives. *Formal semantics of natural language*, pages 123–155.

Tamara G. Kolda and Brett W. Bader. 2009. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September.

Ioannis Korkontzelos, Torsten Zesch, Fabio Massimo Zanzotto, and Chris Biemann. 2013. Semeval-2013 task 5: Evaluating phrasal semantics. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 39–47, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Thomas Landauer and Susan Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychology Review*, 104:211–240.

Daniel D. Lee and H. Sebastian Seung. 2000. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pages 556–562.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL98), Volume 2*, pages 768–774, Montreal, Quebec, Canada.

Zhaohui Luo. 2010. Type-theoretical semantics with coercive subtyping. *SALT20, Vancouver*.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, S Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of SemEval 2014: International Workshop on Semantic Evaluation*.

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

J. Mitchell and M. Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

J. Nivre, J. Hall, and J. Nilsson. 2006a. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC-2006*, pages 2216–2219, Genoa, Italy.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006b. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC-2006*, pages 2216–2219.

Barbara H Partee. 2010. Privative adjectives: subsective plus coercion. *BÄUERLE, R. et ZIMMERMANN, TE, éditeurs: Presuppositions and Discourse: Essays Offered to Hans Kamp*, pages 273–285.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea, July. Association for Computational Linguistics.

Kristina Toutanova and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pages 63–70.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, pages 252–259.

Peter Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188.

Tim Van de Cruys. 2010. A non-negative tensor factorization model for selectional preference induction. *Natural Language Engineering*, 16(4):417–437.

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 2010)*, pages 1263–1271, Beijing, China, August. Coling 2010 Organizing Committee.