Fusion of Compositional Network-based and Lexical Function Distributional Semantic Models

Spiros Georgiladakis
School of ECE
National Technical University of Athens
Athens, Greece
sgeorgil@central.ntua.gr

Elias Iosif
School of ECE
National Technical University of Athens
"Athena" Research Center
Athens, Greece
iosife@central.ntua.gr

Alexandros Potamianos
School of ECE
National Technical University of Athens
"Athena" Research Center
Athens, Greece
potam@central.ntua.gr

Abstract

Distributional Semantic Models (DSMs) have been successful at modeling the meaning of individual words, with interest recently shifting to compositional structures, i.e., phrases and sentences. Network-based DSMs represent and handle semantics via operators applied on word neighborhoods, i.e., semantic graphs containing a target's most similar words. We extend network-based DSMs to address compositionality using an activation model (motivated by psycholinguistics) that operates on the fused neighborhoods of variable size activation. The proposed method is evaluated against and combined with the lexical function method proposed by (Baroni and Zamparelli, 2010). We show that, by fusing a network-based with a lexical function model, performance gains can be achieved.

1 Introduction

Vector Space Models (VSMs) have proven their efficiency at representing word semantics, which are vital components for numerous natural language applications, such as paraphrasing and textual entailment (Androutsopoulos and Malakasiotis, 2010), affective text analysis (Malandrakis et al., 2013), etc. VSMs constitute the most-widely used implementation of Distributional Semantic Models (DSMs) (Baroni and Lenci, 2010). A fundamental task addressed in the framework of DSMs is the computation of semantic similarity between words, adopting the distributional hypothesis of meaning, i.e., *"similarity of context implies similarity of meaning"* (Harris, 1954). DSMs have been successful when applied to the representation of word lexical semantics, enabling the computation of word semantic similarity (Turney and Pantel, 2010). However, the application of DSMs for representing the semantics of more complex structures, e.g., phrases or sentences, is not trivial since the meaning of such structures is the result of various compositional phenomena (Pelletier, 1994) that are inherent properties of natural language creativity. The key idea behind current approaches in semantic composition (using DSMs) is the combination of word vectors using simple functions, e.g., vector addition or multiplication (Mitchell and Lapata, 2008; Mitchell and Lapata, 2010), or other transformational functions. Regardless of the used function, the resulting representations adhere to the paradigm of VSMs, while the cosine between the (composed) vectors is used for estimating similarity. Such efforts proved to be effective when computing the similarity between two-word phrases, however, their limitations were revealed for the case of longer structures (Polajnar et al., 2014), where the composition of meaning becomes more complex. Bengio and Mikolov (2003; 2013) proposed an approach based on deep learning for building language models that address the prob-

Proceedings of CMCL 2015, pages 39–47,
Denver, Colorado, June 4, 2015.

lem of language creativity. The models appear to constantly gain support in comparison with the traditional DSMs. A preliminary comparative analysis of them is provided in (Baroni et al., 2014b) with respect to a number of tasks related to lexical semantics.

In this work, we extend a recent network-based implementation of DSMs (Iosif and Potamianos, 2015) in order to represent the semantics of compositional structures. The used framework consists of activation models motivated by semantic priming (McNamara, 2005). For each structure, an activation area (i.e, semantic neighborhood) is computed which is regarded as a sub-space within the network. The novelty of the present work is twofold. First, we propose various approaches for the creation of activation areas for compositional structures, within a framework alternative to VSMs. Second, we investigate the fusion of the proposed network-based model with VSM-based transformational approaches from the literature. In addition, we investigate the role of words as operators on the meaning of the structures they occur in by measuring their transformative degree.

The remainder of this paper is organized as follows: in Section 2 we describe work related to DSMs. In Section 3 we describe the work on which we based the proposed models. We present the proposed models in Section 4. The lexical function model is described in Section 5, and a fusion model integrating the former with network-based models is proposed. We describe the experimental procedure that we followed and evaluate the proposed models in Section 6. We elaborate on the effects of modifiers in compositional structures in Section 7, concluding in Section 8.

2 Related Work

Word-level DSMs can be categorized into unstructured, that employ a bag-of-words model, and structured, that employ syntactic relationships between words (Grefenstette, 1994; Baroni and Lenci, 2010). DSMs are typically constructed from co-occurrence statistics of word tuples. An unstructured approach for the construction of network-based DSMs was proposed in (Iosif and Potamianos, 2015), where nodes represent words, and edges are formulated according to the semantic similarity of the connected nodes. For each node, the notion of semantic neighborhood (i.e., the most semantically similar words) is utilized for estimating an improved similarity between the nodes.

Moving beyond the word-level, Turney (2012) proposed a "dual-space" model that combines relational and compositional methods for representing phrasal semantics. This approach utilized two complementary models in an attempt to address a series of phenomena that apply to compositional semantics, namely, "linguistic creativity", "order sensitivity", "adaptive capacity", and "information scalability"[1]. Three types of phrases were investigated: noun-noun (NN), adjective-noun (AN), and verb-object (VO). In (Baroni and Zamparelli, 2010), particular focus was given to the AN type, where adjectives were represented as matrices acting as functions to the vectorial representation of head nouns. Recent research efforts have been expanded to longer text segments such as sentences (Agirre et al., 2012; Agirre et al., 2013; Polajnar et al., 2014). In (Socher et al., 2012), based on the functional space proposed in (Baroni and Zamparelli, 2010), phrase constituents were treated as both a continuous vector and a parameter matrix, where the representation of sentence semantics was constructed via a recursive bottom-up procedure.

3 Baseline Network-based Model

In this section, we generalize the ideas regarding network-based DSMs presented in (Iosif and Potamianos, 2015), for the case of more complex structures. The network consists of two layers: 1) *activation*, and 2) *similarity* layer. Given a lexical unit, the first layer represents an activation area that includes a set of lexical units that are semantically related with it. The notion of "lexical unit" refers to any semantically coherent lexical structure, spanning from words (unigrams) up to word sequences (n-grams). The second layer is used for the computation of semantic similarity between two lexical units, based on their respective activation layers. The network can be defined as a graph $Q = (V, E)$ whose set of vertices V includes the lexical units un-

[1] These phenomena are defined and discussed in (Turney, 2012)

der investigation and whose set of edges E contains links between the vertices. The links between the lexical units in the network are weighted according to their pairwise semantic similarity.

3.1 Layer 1: Activation Model

The activation layer of a lexical unit, ξ, can be regarded as a sub-graph of Q, Q_ξ, also referred to as the semantic neighborhood of ξ. Its vertices (neighbors of ξ) are determined according to their semantic similarity with ξ. Given a set of lexical units, the most similar to ξ are selected as neighbors. The activation layer is motivated by the phenomenon of semantic priming (McNamara, 2005), especially for highly coherent lexical units, such as unigrams and bigrams. In the framework of DSMs, activation layers were computed for the case of unigrams in (Iosif and Potamianos, 2015), and were extended to short phrases (bigrams) in (Iosif, 2013). Consider a phrase, $i = (i_1\ i_2)$, where i_1 and i_2 denote its first and second constituent. Assuming that the N_{i_1} and N_{i_2} sets represent neighborhoods of i_1 and i_2, respectively, the neighborhood of i, N_i, was computed by taking the intersection of N_{i_1} and N_{i_2}.

3.2 Layer 2: Semantic Model

Two similarity metrics are defined for computing the similarity between two lexical units, i and j. The metrics are defined on top of their respective activation models, N_i and N_j, computed in the previous layer. This approach relies on two assumptions, namely, maximum sense and attributional similarity, for unigrams. In this work, we extend these metrics to bigrams (see Fig. 1 and Fig. 2) in order to compute the semantic similarity between two phrases, $i = (i_1\ i_2)$ and $j = (j_1\ j_2)$, exploiting their respective activation layers N_i and N_j.

Maximum Neighborhood Similarity. The key idea of this metric, M, is the computation of similarities between the constituents of phrase i (i_1 and i_2) and the members of N_j. The same is done for j_1 and j_2 and the members of N_i. The similarity between i and j (e.g., "assistant manager" and "board member" in Fig. 1) is computed by taking the maximum of the aforementioned similarities (0.50 in Fig. 1). The underlying hypothesis is that the neighborhoods encode senses that are shared between the constituents. The selection of the maximum score suggests that the similarity between i and j can be approximated by considering their closest senses (Iosif and Potamianos, 2015).

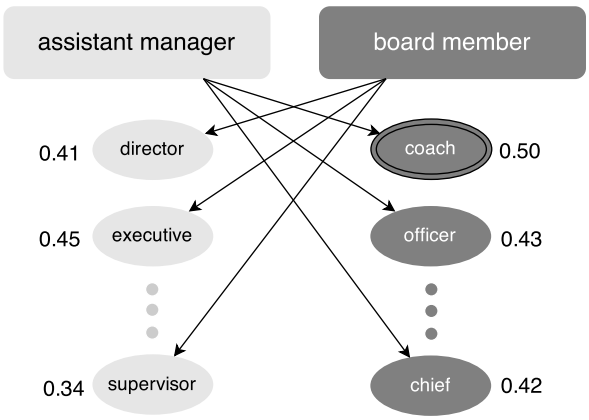

Figure 1: Maximum neighborhood similarity metric (M): bigram usecase.

Attributional Neighborhood Similarity. In this metric, R, similarities between i_1 and i_2 and the members of N_j are computed and stored into a vector. This is also done for j_1 and j_2 and the members of N_j. The correlation coefficient between the two vectors (e.g., the two right-most vectors in Fig. 2) is computed. The process is repeated, using N_i in the place of N_j, which results into another correlation coefficient. The similarity between i and j is estimated by selecting the maximum correlation coefficient. The underlying motivation is attributional similarity, i.e., the hypothesis that the neighborhoods encode semantic or affective features. Se-

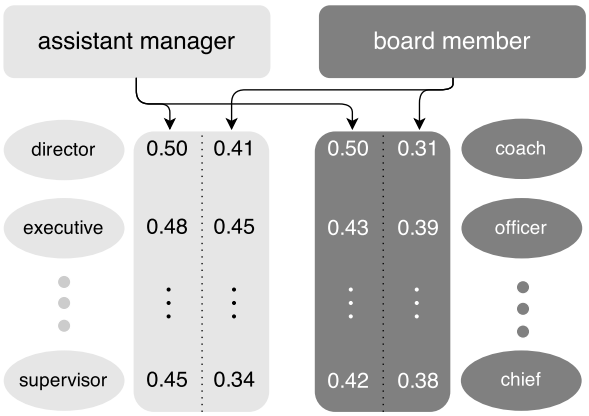

Figure 2: Attributional neighborhood similarity metric (R): bigram usecase.

mantically similar phrases are expected to exhibit correlated similarities with respect to such features (Iosif and Potamianos, 2015).

4 Extended Network-based Model

The major limitation of the model presented in Section 3 is that the neighborhoods of phrase constituents (e.g., N_{i_1} and N_{i_2}) are of fixed size. This allows the computation of an empty neighborhood for the phrase (e.g., N_i), when there is no overlap between the neighborhoods of its constituents.

In this section, we propose an extension of the aforementioned model by relaxing the hard constraint regarding the fixed size of neighborhoods. The intuition behind this idea is that the activation areas are not of the same size for all words. For example, a semantically abstract word, such as "democracy", is expected to have a larger neighborhood compared to semantically concrete words, e.g., "computer". Given a phrase, e.g., $i = (i_1\ i_2)$, in order to compute the activation N_i, we gradually extend the activation areas (i.e., sizes) of N_{i_1} and N_{i_2} until a minimum size θ for N_i is reached.

4.1 Layer 1: Activation Model

We propose three different schemes for the computation of neighborhoods. An example of those

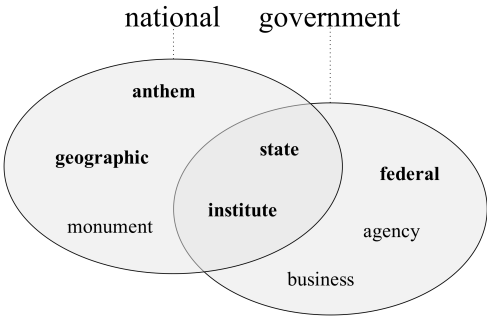

Figure 3: Activation model schemes for the phrase "national government": intersection-based, union-based, and selection of most similar neighbors (words in bold).

schemes is depicted in Fig. 3.

Scheme 1. The phrase neighborhood is computed by taking the intersection of the constituent neighborhoods, i.e., $N_i = N_{i_1} \cap N_{i_2}$. This adheres to findings from the literature of psycholinguistics suggesting that the phrase activation (and, thus, the respective meaning) should be more specific than those of its constituents (Osherson and Smith, 1981).

Scheme 2. The union of neighborhoods is used, i.e., $N_i = N_{i_1} \cup N_{i_2}$. This is motivated by the idea that, in some cases, a phrase may be associated with a larger activation area, compared to those of its constituents.

Scheme 3. The members of the phrase neighborhood are selected based on their average semantic similarity with respect to the phrase constituents. Let N_i be $\{n_1, ..., n_m, ..., n_\theta\}$, where $n_m \in \{N_{i_1} \cup N_{i_2}\}$. The N_i set can be regarded as a list, which is ranked according to $\frac{1}{2}(S(n_m, i_1) + S(n_m, i_2))$, where $S(.)$ stands for a metric of semantic similarity. This scheme is motivated by the idea that different areas of N_{i_1} and N_{i_2} may be activated given the context of words i_1 and i_2, respectively. The scheme also addresses the issue of scalability: the phrase neighborhood has the same size as the constituents' neighborhoods, enabling the recursive application of the model over longer structures.

4.2 Layer 2: Semantic Model

An extension of the M metric (described in Section 3) is proposed, along with two more metrics for computing the semantic similarity between lexical units utilizing their respective neighborhoods. The metrics are defined with respect to two lexical units, i and j, which are represented by their neighborhoods, N_i and N_j, respectively.

Average of top-k similarities (M_k). This metric extends the M metric (see Section 3) by considering the top k similarity scores instead of the maximum score. Similarity between i and j, $M_k(i, j)$, is computed by taking the arithmetic mean of the k scores.

Average of top-k pairwise similarities (P_k). Let C be a ranked list including the pairwise similarities computed between the members of N_i and N_j:

$$C = \{ \underset{\substack{x \in N_i \\ y \in N_j}}{S}(x, y)\ \}, \tag{1}$$

where $S(.)$ stands for a metric of semantic similarity. The similarity between i and j is computed as:

$$P_k(i, j) = \frac{1}{k} \sum_{l=1}^{k} c_l, \tag{2}$$

where c_l is the l–th member of C.

Hausdorff-based similarity (H). This metric is

motivated by the Hausdorff distance (Hung and Yang, 2004). Let $h(N_i, N_j)$ be defined as

$$h(N_i, N_j) = \min_{x \in N_i} \{ \max_{y \in N_j} \{S(x, y)\}\}, \quad (3)$$

where $S(.)$ is a semantic similarity metric. The similarity between i and j is computed as:

$$H(i, j) = \max\{h(N_i, N_j), h(N_j, N_i)\} \quad (4)$$

5 Fusion of Lexical Function with Network-based Models

The representation of phrase semantics requires the consideration of the consituents' functional influence on the composed meaning. For example, when considering an adjective-noun phrase, such as "bad cat", the former word ("bad") acts as an operator, i.e., *modifier*, to the latter word ("cat"), modifying its meaning. In (Baroni and Zamparelli, 2010; Baroni et al., 2014a), it was proposed that such modifications can be implemented via the use of functions that act as linear transformations in VSMs. Application of these functions is realized via matrix-by-vector multiplication as (Baroni et al., 2014a):

$$f(\alpha) =_{def} F \times a = b, \quad (5)$$

where F is the matrix-encoded function f, a is the vectorial representation of the argument α, and b is the compositional vector output. The F function is learnt according to examples of observed input and output (distributional) representations. The input is the representation of the head word, and the output is the representation of the phrase. Regression is employed for calculating the set of weights in the matrix that best approximate the observed vectors. For example, the function for the modifier "bad" is learnt by regressing over phrase examples and their head nouns, such as <*pet, bad pet*>, <*dog, bad dog*>, <*bird, bad bird*>. Using the trained set of weights and the vectorial representation of the head noun, e.g., "cat", the composite representation for the phrase "bad cat" is induced.

5.1 Fusion

The proposed network-based model, presented in Section 4, exploits the merging of word senses for computing activation areas for phrases. The model defined by (5) utilizes the transformational function of an operator for changing the meaning of a phrase. Both models (intuitively) seem to be aligned with the human process of phrase comprehension, however, there are cases that one of the models applies better than the other. Consider two example phrases, "football manager" and "successful engineer". The transformational model is expected to perform better for the latter phrase, while for the first phrase an intersection of word senses (i.e., a network-based model) seems to be more appropriate.

Based on the above considerations, we propose a fusion of the lexical function (*lf*), defined by (5), with the proposed network-based models. The fusion is aimed to model more accurately the semantic representations of complex structures. To do so, we measure the Mean Squared Error (MSE) when training the lexical function model, in order to quantify the transformative degree of the modifier under investigation. The transformative degree is used for deciding whether a network-based or a transformational model is more appropriate. Given two phrases, $i = (i_1\ i_2)$ and $j = (j_1\ j_2)$, the transformative degree $T(i, j)$ is defined as:

$$T(i, j) = \frac{1}{2}(MSE(i_1) + MSE(j_1)), \quad (6)$$

where $MSE(i_1)$ and $MSE(j_1)$ is the MSE that corresponds to modifiers i_1 and j_1, respectively. The proposed fusion metric, $\Phi^{lf}_{net}(i, j)$, used for estimating the similarity between the i and j phrases, is defined as:

$$\Phi^{lf}_{net}(i, j) = \lambda(i, j)\ S_N + (1 - \lambda(i, j))\ S_{LF}, \quad (7)$$

where S_N and S_{LF} are similarity scores computed by the network-based and lexical function models, respectively. λ is a function of i and j, computed using a sigmoid function as:

$$\lambda(i, j) = 0.5/\left(1 + \mathrm{e}^{-T(i,j)}\right). \quad (8)$$

The sigmoid function is applied in order to smooth and normalize (within [0,1]) the values of $T(i, j)$.

Finally, in addition to the aforementioned fusion, we also implement a fusion combining the *lf* and the widely-used additive (*add*) (Mitchell and Lapata, 2008; Mitchell and Lapata, 2010) model. This fusion metric, Φ^{lf}_{add}, is defined similarly to (7).

6 Experiments and Evaluation

The procedure for creating the network and conducting the experiments is described in Section 6.1. In Section 6.2, we evaluate the proposed models and compare them with results from the literature.

6.1 Experimental Procedure

We defined our vocabulary (network nodes) by intersecting the English vocabulary found in the ASPELL[2] dictionary and the Wikipedia dump[3] to derive an English vocabulary of approximately 135K words. Using it, a corpus comprising of web-harvested document snippets was constructed by downloading 1000 snippets for each word in the vocabulary. Word-level similarities were computed among all vocabulary entries' pairs. To this end, the Normalized Google Distance (G) was utilized, proposed in (Vitanyi, 2005; Cilibrasi and Vitanyi, 2007) and motivated by Kolmogorov complexity. Let G be defined as

$$G(w_1, w_2) = \frac{\max\{A\} - \log |D| w_1, w_2|}{\log |D| - \min\{A\}}, \quad (9)$$

where w_1 and w_2 are two vocabulary words under investigation, $| D |$ is the total number of documents in the corpus, $| D | w_1, w_2 |$ is the total number of documents containing both w_1 and w_2, and $A = \{\log | D | w_1 |, \log | D | w_2 |\}$. We used a variation of (9), proposed in (Gracia et al., 2006), referred to as "Google-based Semantic Relatedness" ($G^{'}$). This variation defines a similarity measure, bounded within the $[0, 1]$ range and defined as

$$G^{'}(w_1, w_2) = e^{-2G(w_1, w_2)}, \quad (10)$$

where $G(w_1, w_2)$ is computed according to (9). In this work, D denotes the sentence rather than the document, as the co-occurrence of words was defined at sentence-level. This metric was adopted based on its good performance in word-level semantic similarity tasks (Iosif and Potamianos, 2015).

Network-based model. We used sizes of $\theta = \{10, 25, 50, 100, 150, 500\}$ for the case of fixed-size neighborhoods, and $\theta = \{1, 5, ..., 40\}$ for the extended activation models described in Section 4.1. We used both the baseline and the extended activation layers for the M model, the latter being defined as M'. For M_k and P_k, we set $k = \{1, ..., 5\}$.

Transformational model. For the *lf* model described in (5), we computed co-occurence counts for bigrams occurring at least 50 times in the corpus. Positive Pointwise Mutual Information (PPMI) was applied to reweigh them. We used a) Singular Value Decomposition (SVD), and b) Non-Negative Matrix Factorization (NMF) (Lee and Seung, 2001) to reduce the dimensionality of the space down to a) 300, and b) 500 dimensions. To train *lf*, we selected corpus bigrams comprising of a *modifier* and a *noun*. We used a) Least Squares (LSR), and b) Ridge (RR) (Hastie et al., 2009) regression. The DIStributional SEmantics Composition Toolkit (DISSECT [4], (Dinu et al., 2013)) was used to implement *lf*, as well as the widely-used additive (*add*) and multiplicative (*mult*) models proposed in (Mitchell and Lapata, 2008; Mitchell and Lapata, 2010).

Fusion model. We combined the best performing model configurations on NNs (see Section 6.2) in order to implement the proposed fusion models.

6.2 Evaluation Results

For evaluation purposes, we used the widely-used Mitchell & Lapata (2010) datasets comprising of 108 noun-noun (NN), adjective-noun (AN), and verb-object (VO) phrase pairs, evaluated by human judgements and averaged per phrase pair. The models were evaluated using Spearman's correlation coefficient. Evaluation results are presented in Table 1. Due to space limitations, only the best performing network-based model configurations are reported here. Also, since the *mult* model performs poorly when the composed vectors contain negative values, as is the case with SVD, we only report results for the NMF variations for it. Finally, since training the *lf* model with RR had significantly superior performance over LSR in all configurations, we only report evaluations of the former.

The *lf* model, when using RR in combination with NMF, performs best (.76) for the case of NNs. Best performances for ANs and VOs are obtained by the *add* model (.63 and .59, respectively).

[2] http://www.aspell.net/

[3] As of the 4th quarter of 2012.

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

Model	NN	AN	VO
add (NMF300)	**.67**	.61	.53
add (NMF500)	.66	**.63**	.56
add (SVD300)	.63	.59	**.59**
add (SVD500)	.66	**.63**	**.59**
mult (NMF300)	.59	.38	.36
mult (NMF500)	.59	.36	.42
lf (NMF300, RR)	**.76**	**.46**	**.35**
lf (NMF500, RR)	.67	.41	.28
lf (SVD300, RR)	.63	.35	.26
lf (SVD500, RR)	.56	.33	.23
M (Intersection)	.56	.46	.37
M' (Intersection)	.61	**.57**	**.47**
$M_{k=3}$ (Intersection)	**.64**	.51	.41
$P_{k=3}$ (Most-similar)	.63	.46	.23
H (Intersection)	.58	.39	.26
fusion Φ^{lf}_{net}	**.80**	.54	.35
fusion Φ^{lf}_{add}	.76	**.57**	**.44**

Table 1: Performance of models on NN, AN, and VO phrase pairs. Evaluations are reported using Spearman's correlation coefficient with human ratings.

Regarding network-based models, performance is improved when using the extended activation model over the baseline. This is confirmed by the absolute 5%, 11% and 10% increase for the case of NN, AN, and VO pairs, respectively, for the M metric. All the extended network-based models perform consistently better than the baseline of M, in the case of NNs, although their performance drops for the case of ANs and VOs. In the case of P_k, the scheme that constructs neighborhoods via the selection of the most similar neighbors performs better than the intersection- or the union-based scheme.

Φ^{lf}_{add} yields no relative improvements over the best performances of the separate models. Φ^{lf}_{net} provides an improvement for the case of NNs, reaching .80, which is also the best observed performance overall. However, Φ^{lf}_{net} does not improve performance in the case of ANs and VOs.

Performance improvements when using the extended activation layer for compositional structures is consistent with experimental observations from psycholinguistics (Osherson and Smith, 1981), and shows that the activation area for phrases might be adaptive to the degree of relatedness between words.

7 Discussion

The results displayed in Table 1 for the fusion models provide an indication of the different ways in which the operator changes the meaning of a phrase. In this section, we investigate the transformational properties of phrases as defined by their modifiers. By observing the properties of modifiers, we discuss whether their use in a phrase has mainly a *transformational* or a merely *compositional* effect, based on the goodness of fit of each model, estimated during model training.

7.1 The Transformative Effect of Modifiers

Early research on compositionality involved applying the word-level semantic similarity estimation techniques to phrases using context-based, bag-of-words models, i.e., defining the structures' meaning as a function of the words in their context. Though simple and cost-effective, the aforementioned techniques fail to detect the effect that a word has to its linguistic context and the semantic changes on its meaning, e.g., a "nice" table is still a table but a "fake" or "broken" table is not.

Depending on context, a modifier can affect the meaning of the encompassing phrase in different ways. For example, the modifier "normal" changes the meaning of "normal cat" much less than the modifier "dead" in "dead cat". Moreover, the modifier effect may vary for each syntactic category. For example, verbs can be transitive or intransitive, nouns can be abstract or concrete, and adjectives can be intensional or not (Boleda et al., 2013). Words that act as *functions* on their linguistic context have attracted much interest, and have recently been successfully handled by computational models.

7.2 Estimating the Transformative Degree

We categorise modifiers based on their regression performance, when training them for the *lf* model. Specifically, we acquire the MSE of their training as a measure for deciding the degree of their transformative effect on a given head noun. Taking the MSE is a sensible approach, since regression tries to derive a close approximation to observed vectorial representations of phrases and head nouns by means of transforming the head noun vector; high error in training indicates that the *lf* model is a poor match

for this modifier. We trained the *lf* model using Ridge Regression and estimated the MSE for each modifier. In Table 2, we present example modifiers of low, neutral, and high transformative degree, as defined by their MSE score. We observe that highly-transformative modifiers have a more functional influence, when used in bigram structures. For example, in "efficient machine", "efficient" has a greater effect on the meaning of "efficient machine" rather than, e.g., "new" in "new machine". A "new machine" retains the same properties of a generic machine. However, an "efficient machine" should contain mechanisms that account for optimization of speed, cost, etc. Our observations suggest that modifiers affect the structure in which they occur in different ways. Some modifiers have a stronger effect on the meaning of the head noun, while others act merely as constituents of simple compositions. The proposed fusion of the transformational, *lf* model, with network-based or simple compositional models indicates that combining different models can yield improved performance when the transformative degree of modifiers is used as a fusion criterion.

Degree	Nouns	Adjectives	Verbs
Low	news	new	like
	service	great	buy
	business	black	help
	world	general	use
	state	good	provide
Neutral	company	various	face
	care	right	need
	community	better	cut
High	railway	old	encourage
	labour	rural	attend
	defence	elderly	remember
	personnel	efficient	satisfy
	committee	practical	suffer

Table 2: Examples of low, neutral, and high transformative modifiers.

8 Conclusions

We presented a network-based model that operates on neighborhoods of variable size to calculate similarity of compositional structures. We investigated various methods for composing neighborhoods of adjacent words and presented three metrics, motivated by psycholinguistics and metric space algebra, for estimating similarity between activation areas. Employing variable size activation improves semantic similarity performance, revealing a different activational behavior among bigrams. We also presented a fusion of the proposed models with the lexical function model based on the transformative degree of modifiers, achieving an improvement of performance for noun-noun compositions, reaching state-of-the-art performance of 80% Spearman correlation with human judgements. We further investigated the transformative degree of modifiers, and elaborated on their role as mostly *compositional* or *transformational*.

In future work, we will further investigate the role of modifiers and their application in the proposed activation composition approaches, while also explore the criteria for deriving activations and deciding on fusion strategies. We also plan to apply network-based models on longer semantic structures.

Acknowledgments

This work has been partially funded by the SpeDial project supported by the EU FP7 with grant number 611396, and the BabyAffect project supported by the Greek General Secretariat for Research and Technology (GSRT) with grant number 3610.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), in conjuction with the First Joint Conference on Lexical and Computational Semantic (*SEM 2012)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. Sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In *In* SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics*.

Ion Androutsopoulos and Prodromos Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187.

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based

semantics. *Computational Linguistics*, 36(4):673–721.

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of EMNLP 2010*, pages 1183–1193.

Marco Baroni, Raffaela Bernardi, and Roberto Zamparelli. 2014a. Frege in space: A program of compositional distributional semantics. *Linguistic Issues in Language Technology (LiLT)*, 9.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014b. Dont 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, pages 238–247.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Gemma Boleda, Marco Baroni, Nghia T. Pham, and Louise McNally. 2013. Intensionality was only alleged: On adjective-noun composition in distributional semantics. In *Proceedings of IWCS 2013*, pages 35–46.

Rudi L. Cilibrasi and Paul MB Vitanyi. 2007. The Google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):370–383.

Georgiana Dinu, Nghia The Pham, and Marco Baroni. 2013. DISSECT-DIStributional SEmantics Composition Toolkit. In *Proceedings of the 51st Annual Meeting of ACL: System Demonstrations*, pages 31–36.

Jorge Gracia, Raquel Trillo, Mauricio Espinoza, and Eduardo Mena. 2006. Querying the web: A multiontology disambiguation method. In *Proceedings of the 6th International Conference on Web Engineering (ICWE 2006)*, pages 241–248.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers.

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

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning*, volume 2. Springer.

Wen-Liang Hung and Miin-Shen Yang. 2004. Similarity measures of intuitionistic fuzzy sets based on hausdorff distance. *Pattern Recognition Letters*, 25(14):1603–1611.

Elias Iosif and Alexandros Potamianos. 2015. Similarity computation using semantic networks created from web-harvested data. *Natural Language Engineering*, 21(01):49–79.

Elias Iosif. 2013. *Network-based Distributional Semantic Models*. Ph.D. thesis, Technical University of Crete, Chania, Greece.

Daniel D. Lee and Sebastian H. Seung. 2001. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562.

Nikolaos Malandrakis, Alexandros Potamianos, Elias Iosif, and Shrikanth Narayanan. 2013. Distributional semantic models for affective text analysis. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(11):2379–2392.

T. P. McNamara. 2005. *Semantic Priming: Perspectives from Memory and Word Recognition*. Psychology Press.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *ICLR Workshop*.

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

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Daniel N. Osherson and Edward E. Smith. 1981. On the adequacy of prototype theory as a theory of concepts. *Cognition*, 9(1):35–58.

Francis J. Pelletier. 1994. The principle of semantic compositionality. *Topoi*, 13(1):11–24.

Tamara Polajnar, Laura Rimell, and Stephen Clark. 2014. Evaluation of simple distributional compositional operations on longer texts. In *9th Language Resources and Evaluation Conference (LREC 2014)*.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP-CoNLL*, pages 1201–1211.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.

Peter D. Turney. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44:533–585.

Paul Vitanyi. 2005. Universal similarity. In *Proceedings of Information Theory Workshop on Coding and Complexity*, pages 238–243.