“Not not bad” is not “bad”: A distributional account of negation

Karl Moritz Hermann **Edward Grefenstette** **Phil Blunsom**
University of Oxford Department of Computer Science
Wolfson Building, Parks Road
Oxford OX1 3QD, United Kingdom
firstname.lastname@cs.ox.ac.uk

Abstract

With the increasing empirical success of distributional models of compositional semantics, it is timely to consider the types of textual logic that such models are capable of capturing. In this paper, we address shortcomings in the ability of current models to capture logical operations such as negation. As a solution we propose a tripartite formulation for a continuous vector space representation of semantics and subsequently use this representation to develop a formal compositional notion of negation within such models.

1 Introduction

Distributional models of semantics characterize the meanings of words as a function of the words they co-occur with (Firth, 1957). These models, mathematically instantiated as sets of vectors in high dimensional vector spaces, have been applied to tasks such as thesaurus extraction (Grefenstette, 1994; Curran, 2004), word-sense discrimination (Schütze, 1998), automated essay marking (Landauer and Dumais, 1997), and so on.

During the past few years, research has shifted from using distributional methods for modelling the semantics of words to using them for modelling the semantics of larger linguistic units such as phrases or entire sentences. This move from word to sentence has yielded models applied to tasks such as paraphrase detection (Mitchell and Lapata, 2008; Mitchell and Lapata, 2010; Grefenstette and Sadrzadeh, 2011; Blacoe and Lapata, 2012), sentiment analysis (Socher et al., 2012; Hermann and Blunsom, 2013), and semantic relation classification (*ibid.*). Most efforts approach the problem of modelling phrase meaning through vector composition using linear algebraic vector operations (Mitchell and Lapata, 2008; Mitchell and Lapata, 2010; Zanzotto et al., 2010), matrix or tensor-based approaches (Baroni and Zamparelli, 2010; Coecke et al., 2010; Grefenstette et al., 2013; Kartsaklis et al., 2012), or through the use of recursive auto-encoding (Socher et al., 2011; Hermann and Blunsom, 2013) or neural-networks (Socher et al., 2012). On the non-compositional front, Erk and Padó (2008) keep word vectors separate, using syntactic information from sentences to disambiguate words in context; likewise Turney (2012) treats the compositional aspect of phrases and sentences as a matter of similarity measure composition rather than vector composition.

These compositional distributional approaches often portray themselves as attempts to reconcile the empirical aspects of distributional semantics with the structured aspects of formal semantics. However, they in fact only principally co-opt the syntax-sensitivity of formal semantics, while mostly eschewing the logical aspects.

Expressing the effect of logical operations in high dimensional distributional semantic models is a very different task than in boolean logic. For example, whereas predicates such as ‘red’ are seen in predicate calculi as functions mapping elements of some set M_{red} to $\top$ (and all other domain elements to $\bot$), in compositional distributional models we give the meaning of ‘red’ a vector-like representation, and devise some combination operation with noun representations to obtain the representation for an adjective-noun pair. Under the logical view, negation of a predicate therefore yields a new truth-function mapping elements of the complement of M_{red} to $\top$ (and all other domain elements to $\bot$), but the effect of negation and other logical operations in distributional models is not so sharp: we expect the representation for “not red” to remain close to other objects of the same domain of discourse (i.e. other colours) while being sufficiently different from the representation of ‘red’ in some manner. Exactly how textual logic would best be represented in a continuous vector space model remains an open problem.

In this paper we propose one possible formulation for a continuous vector space based representation of semantics. We use this formulation as the basis for providing an account of logical operations for distributional models. In particular, we focus on the case of negation and how it might work in higher dimensional distributional models. Our formulation separates domain, value and functional representation in such a way as to allow negation to be handled naturally. We explain the linguistic and model-related impacts of this mode of representation and discuss how this approach could be generalised to other semantic functions.

In Section 2, we provide an overview of work relating to that presented in this paper, covering the integration of logical elements in distributional models, and the integration of distributional elements in logical models. In Section 3, we introduce and argue for a tripartite representation in distributional semantics, and discuss the issues relating to providing a linguistically sensible notion of negation for such representations. In Section 4, we present matrix-vector models similar to that of Socher et al. (2012) as a good candidate for expressing this tripartite representation. We argue for the elimination of non-linearities from such models, and thus show that negation cannot adequately be captured. In Section 5, we present a short analysis of the limitation of these matrix-vector models with regard to the task of modelling non-boolean logical operations, and present an improved model bypassing these limitations in Section 6. Finally, in Section 7, we conclude by suggesting future work which will extend and build upon the theoretical foundations presented in this paper.

2 Motivation and Related Work

The various approaches to combining logic with distributional semantics can broadly be put into three categories: those approaches which use distributional models to enhance existing logical tools; those which seek to replicate logic with the mathematical constructs of distributional models; and those which provide new mathematical definitions of logical operations within distributional semantics. The work presented in this paper is in the third category, but in this section we will also provide a brief overview of related work in the other two in order to better situate the work this paper will describe in the literature.

Vector-assisted logic The first class of approaches seeks to use distributional models of word semantics to enhance logic-based models of textual inference. The work which best exemplifies this strand of research is found in the efforts of Garrette et al. (2011) and, more recently, Beltagy et al. (2013). This line of research converts logical representations obtained from syntactic parses using Bos' Boxer (Bos, 2008) into Markov Logic Networks (Richardson and Domingos, 2006), and uses distributional semantics-based models such as that of Erk and Padó (2008) to deal with issues polysemy and ambiguity.

As this class of approaches deals with improving logic-based models rather than giving a distributional account of logical function words, we view such models as orthogonal to the effort presented in this paper.

Logic with vectors The second class of approaches seeks to integrate boolean-like logical operations into distributional semantic models using existing mechanisms for representing and composing semantic vectors. Coecke et al. (2010) postulate a mathematical framework generalising the syntax-semantic passage of Montague Grammar (Montague, 1974) to other forms of syntactic and semantic representation. They show that the parses yielded by syntactic calculi satisfying certain structural constraints can be canonically mapped to vector combination operations in distributional semantic models. They illustrate their framework by demonstrating how the truth-value of sentences can be obtained from the combination of vector representations of words and multilinear maps standing for logical predicates and relations. They furthermore give a matrix interpretation of negation as a 'swap' matrix which inverts the truth-value of vectorial sentence representations, and show how it can be embedded in sentence structure.

Recently, Grefenstette (2013) showed that the examples from this framework could be extended to model a full quantifier-free predicate logic using tensors of rank 3 or lower. In parallel, Socher et al. (2012) showed that propositional logic can be learned using tensors of rank 2 or lower (i.e. only matrices and vectors) through the use of non-linear activation functions in recursive neural networks.

The work of Coecke et al. (2010) and Grefenstette (2013) limits itself to defining, rather than learning, distributional representations of logical operators for distributional models that simulate logic, and makes no pretense to the provision of operations which generalise to higher-dimensional distributional semantic representations. As for the non-linear approach of Socher et al. (2012), we will discuss, in Section 4 below, the limitations with this model with regard to the task of modelling logic for higher dimensional representations.

Logic for vectors The third and final class of approaches is the one the work presented here belongs to. This class includes attempts to define representations for logical operators in high dimensional semantic vector spaces. Such approaches do not seek to retrieve boolean logic and truth values, but to define what logical operators *mean* when applied to distributional representations. The seminal work in this area is found in the work of Widdows and Peters (2003), who define negation and other logical operators algebraically for high dimensional semantic vectors. Negation, under this approach, is effectively a binary relation rather than a unary relation: it expresses the semantics of statements such as 'A NOT B' rather than merely 'NOT B', and does so by projecting the vector for A into the orthogonal subspace of the vector for B. This approach to negation is useful for vector-based information retrieval models, but does not necessarily capture all the aspects of negation we wish to take into consideration, as will be discussed in Section 3.

3 Logic in text

In order to model logical operations over semantic vectors, we propose a tripartite meaning representation, which combines the separate and distinct treatment of domain-related and value-related aspects of semantic vectors with a domain-driven syntactic functional representation. This is a unification of various recent approaches to the problem of semantic representation in continuous distributional semantic modelling (Socher et al., 2012; Turney, 2012; Hermann and Blunsom, 2013).

We borrow from Socher et al. (2012) and others (Baroni and Zamparelli, 2010; Coecke et al., 2010) the idea that the information words refer to is of two sorts: first the semantic content of the word, which can be seen as the sense or reference to the concept the word stands for, and is typically modelled as a semantic vector; and second, the *function* the word has, which models the effect the word has on other words it combines with in phrases and sentences, and is typically modelled as a matrix or higher-order tensor. We borrow from Turney (2012) the idea that the semantic aspect of a word should not be modelled as a single vector where everything is equally important, but ideally as two or more vectors (or, as we do here, two or more *regions* of a vector) which stand for the aspects of a word relating to its *domain*, and those relating to its *value*.

We therefore effectively suggest a *tripartite representation* of the semantics of words: a word's meaning is modelled by elements representing its value, domain, and function, respectively.

The tripartite representation We argue that the tripartite representation suggested above allows us to explicitly capture several aspects of semantics. Further, while there may be additional distinct aspects of semantics, we argue that this is a minimal viable representation.

First of all, the differentiation between domain and value is useful for establishing similarity within subspaces of meaning. For instance, the words *blue* and *red* share a common domain (colours) while having very different values. We hypothesise that making this distinction explicit will allow for the definition of more sophisticated and fine-grained semantics of logical operations, as discussed below. Although we will represent domain and value as two regions of a vector, there is no reason for these not to be treated as separate vectors at the time of comparison, as done by Turney (2012).

Through the third part, the functional representation, we capture the compositional aspect of semantics: the functional representation governs how a term interacts with its environment. Inspired by the distributional interpretation (Baroni and Zamparelli, 2010; Coecke et al., 2010) of syntactically-paramatrized semantic composition functions from Montogovian semantics (Montague, 1974), we will also assume the function part of our representation to be parametrized principally by syntax and domain rather than value. The intuition behind taking domain into account in addition to syntactic class being that all members of a domain largely interact with their environment in the same fashion.

Modeling negation The tripartite representation proposed above allows us to define logical operations in more detail than competing approaches. To exemplify this, we focus on the case of negation.

We define negation for semantic vectors to be the absolute complement of a term in its domain. This implies that negation will not affect the domain of a term but only its value. Thus, *blue* and *not blue* are assumed to share a common domain. We call this naive form of negation the inversion of a term A, which we idealise as the partial inversion A^{inv} of the region associated with the value of the word in its vector representation A.

$$\begin{array}{ccc}
\begin{bmatrix} d \\ v \\ v \end{bmatrix} & \begin{bmatrix} d \\ v \\ -v \end{bmatrix} & \begin{bmatrix} d \\ v \\ -\mu v \end{bmatrix} \\
\begin{bmatrix} f \end{bmatrix} & \begin{bmatrix} f \end{bmatrix} & \begin{bmatrix} f \end{bmatrix} \\
\mathrm{W} & \mathrm{W}^{inv} & \neg\mathrm{W}
\end{array}$$

Figure 1: The semantic representations of a word W, its inverse W^{inv} and its negation $\neg W$. The domain part of the representation remains unchanged, while the value part will partially be inverted (inverse), or inverted and scaled (negation) with $0 < \mu < 1$. The (separate) functional representation also remains unchanged.

Additionally, we expect negation to have a diminutive effect. This diminutive effect is best exemplified in the case of sentiment: *good* is more positive than *not bad*, even though *good* and *bad* are antonyms of each other. By extension *not not good* and *not not not bad* end up somewhere in the middle—qualitative statements still, but void of any significant polarity. To reflect this diminutive effect of negation and double negation commonly found in language, we define the idealised diminutive negation $\neg A$ of a semantic vector A as a scalar inversion over a segment of the value region of its representation with the scalar $\mu : 0 < \mu < 1$, as shown in Figure 1.

As we defined the functional part of our representation to be predominately parametrized by syntax and domain, it will remain constant under negation and inversion.

4 A general matrix-vector model

Having discussed, above, how the vector component of a word can be partitioned into domain and value, we now turn to the partition between semantic content and function. A good candidate for modelling this partition would be a dual-space representation similar to that of Socher et al. (2012). In this section, we show that this sort of representation is not well adapted to the modelling of negation.

Models using dual-space representations have been proposed in several recent publications, notably in Turney (2012) and Socher et al. (2012). We use the class of recursive matrix-vector models as the basis for our investigation; for a detailed introduction see the MV-RNN model described in Socher et al. (2012).

We begin by describing composition for a general dual-space model, and apply this model to the notion of compositional logic in a tripartite representation discussed earlier. We identify the shortcomings of the general model and subsequently discuss alternative composition models and modifications that allow us to better capture logic in vector space models of meaning.

Assume a basic model of compositionality for such a tripartite representation as follows. Each term is encoded by a semantic vector v capturing its domain and value, as well as a matrix M capturing its function. Thus, composition consists of two separate operations to learn semantics and function of the composed term:

$$\begin{aligned} \mathbf{v}_p &= f_v(\mathbf{v}_a, \mathbf{v}_b, M_a, M_b) \\ M_p &= f_M(M_a, M_b) \end{aligned} \tag{1}$$

As we defined the functional representation to be parametrized by syntax and domain, its composition function does not require $\mathbf{v}_a$ and $\mathbf{v}_b$ as inputs, with all relevant information already being contained in M_a, M_b. In the case of Socher et al. (2012) these functions are as follows:

$$M_p = W_M \begin{bmatrix} M_a \\ M_b \end{bmatrix} \tag{2}$$

$$\mathbf{v}_p = g\left(W_v \begin{bmatrix} M_a\mathbf{v}_b \\ M_b\mathbf{v}_a \end{bmatrix}\right) \tag{3}$$

where g is a non-linearity.

4.1 The question of non-linearities

While the non-linearity g could be equipped with greater expressive power, such as in the boolean logic experiment in Socher et al. (2012)), the aim of this paper is to place the burden of compositionality on the atomic representations instead. For this reason we treat g as an identity function, and W_M, W_v as simple additive matrices in this investigation, by setting

$$g = I \quad W_v = W_M = [I \; I]$$

where I is an identity matrix. This simplification is justified for several reasons.

A simple non-linearity such as the commonly used hyperbolic tangent or sigmoid function will not add sufficient power to overcome the issues outlined in this paper. Only a highly complex non-linear function would be able to satisfy the requirements for vector space based logic as discussed above. Such a function would defeat the point however, by pushing the "heavy-lifting" from the model structure into a separate function.

Furthermore, a non-linearity effectively encodes a scattergun approach: While it may have the power to learn a desired behaviour, it similarly has a lot of power to learn undesired behaviours and side effects. From a formal perspective it would therefore seem more prudent to explicitly encode desired behaviour in a model's structure rather than relying on a non-linearity.

4.2 Negation

We have outlined our formal requirements for negation in the previous section. From these requirements we can deduce four equalities, concerning the effect of negation and double negation on the semantic representation and function of a term. The matrices J_μ and J_ν (illustrated in Figure 2) describe a partially scaled and inverted identity matrix, where $0 < \mu, \nu < 1$.

$$f_v(\text{not}, a) = J_\mu \mathbf{v}_a \tag{4}$$

$$f_M(\text{not}, a) \approx M_a \tag{5}$$

$$f_v(\text{not}, f_v(\text{not}, a)) = J_\nu J_\mu \mathbf{v}_a \tag{6}$$

$$f_M(\text{not}, f_M(\text{not}, a)) \approx M_a \tag{7}$$

$$\begin{pmatrix} 1 & & & & & \\ & \ddots & & & \mathbf{0} & \\ & & 1 & & & \\ & & & -\mu & & \\ & \mathbf{0} & & & \ddots & \\ & & & & & -\mu \end{pmatrix}$$

Figure 2: A partially scaled and inverted identity matrix J_μ. Such a matrix can be used to transform a vector storing a domain and value representation into one containing the same domain but a partially inverted value, such as W and $\neg W$ described in Figure 1.

Based on our assumption about the constant domain and interaction across negation, we can replace the approximate equality with a strict equality in Equations 5 and 7. Further, we assume that both $M_a \neq I$ and $M_a \neq 0$, i.e. that A has a specific and non-zero functional representation. We make a similar assumption for the semantic representation $\mathbf{v}_a \neq 0$.

Thus, to satisfy the equalities in Equations 4 through 7, we can deduce the values of $\mathbf{v}_{not}$ and M_{not} as discussed below.

Value and Domain in Negation Under the simplifications of the model explained earlier, we know that the following is true:

$$\begin{aligned} f_v(a, b) &= g\left(W_v \begin{bmatrix} M_a \mathbf{v}_b \\ M_b \mathbf{v}_a \end{bmatrix}\right) \\ &= I\left(\begin{bmatrix} I \; I \end{bmatrix} \begin{bmatrix} M_a \mathbf{v}_b \\ M_b \mathbf{v}_a \end{bmatrix}\right) \\ &= M_a \mathbf{v}_b + M_b \mathbf{v}_a \end{aligned}$$

I.e. the domain and value representation of a parent is the sum of the two Mv multiplications of its children. The matrix W_v could re-weight this addition, but that would not affect the rest of this analysis.

Given the idea that the domain stays constant under negation and that a part of the value is inverted and scaled, we further know that these two equations hold:

$$\begin{aligned} \forall a \in A &: f_v(\text{not}, a) = J_\mu \mathbf{v}_a \\ \forall a \in A &: f_v(\text{not}, f_v(\text{not}, a)) = J_\nu J_\mu \mathbf{v}_a \end{aligned}$$

Assuming that both semantic and functional representation across all A varies and is non-zero, these equalities imply the following conditions for the representation of *not*:

$$\begin{aligned} M_{not} &= J_\mu = J_\nu \\ \mathbf{v}_{not} &= 0 \end{aligned}$$

These two equations suggest that the term not has no inherent value ($\mathbf{v}_{not} = 0$), but merely acts as a function, inverting part of another terms semantic representation ($M_{not} = J_\mu$).

Functional Representation in Negation We can apply the same method to the functional representation. Here, we know that:

$$\begin{aligned} f_M(a,b) &= W_M \begin{bmatrix} M_a \\ M_b \end{bmatrix} \\ &= \begin{bmatrix} I \ I \end{bmatrix} \begin{bmatrix} M_a \\ M_b \end{bmatrix} \\ &= M_a + M_b \end{aligned}$$

Further, as defined in our discussion of negation, we require the functional representation to remain unchanged under negation:

$$\begin{aligned} \forall a \in A : f_M(\text{not}, a) = M_a \\ \forall a \in A : f_M(\text{not}, f_M(\text{not}, a)) = M_a \end{aligned}$$

These requirements combined leave us to conclude that $M_{not} = 0$. Combined with the result from the first part of the analysis, this causes a contradiction:

$$\begin{aligned} M_{not} &= 0 \\ M_{not} &= J_\mu \\ \implies J_\mu &= 0 \lightning \end{aligned}$$

This demonstrates that the MV-RNN as described in this paper is not capable of modelling semantic logic according to the principles we outlined. The fact that we would require $M_{not} = 0$ further supports the points made earlier about the non-linearities and setting W_M to $\begin{bmatrix} I \ I \end{bmatrix}$. Even a specific W_M and non-linearity would not be able to ensure that the functional representation stays constant under negation given a non-zero M_{not}.

Clearly, any other complex semantic representation would suffer from the same issue here—the failure of double-negation to revert a representation to its (diminutive) original.

5 Analysis

The issue identified with the MV-RNN style models described above extends to a number of other models of vector spaced compositionality. It can be viewed as a problem of uninformed composition caused by a composition function that fails to account for syntax and thus for scope.

Of course, identifying the scope of negation is a hard problem in its own right—see e.g. the *SEM 2012 shared task (Morante and Blanco, 2012). However, at least for simple cases, we can deduce scope by considering the parse tree of a sentence:

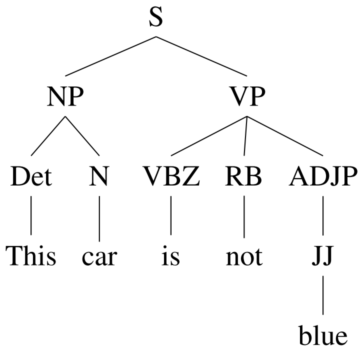

Figure 3: The parse tree for *This car is not blue*, highlighting the limited scope of the negation.

If we consider the parse tree for *this car is not blue*, it is clear that the scope of the negation expressed includes the colour but not the car (Figure 3).

While the MV-RNN model in Socher et al. (2012) incorporates parse trees to guide the order of its composition steps, it uses a single composition function across all steps. Thus, the functional representation of *not* will to some extent propagate outside of its scope, leading to a vector capturing something that is not blue, but also not quite a car.

There are several possibilities for addressing this issue. One possibility is to give greater weight to syntax, for instance by parametrizing the composition functions f_v and f_M on the parse structure. This could be achieved by using specific weight matrices W_v and W_M for each possible tag. While the power of this approach is limited by the complexity of the parse structure, it would be better able to capture effects such as the scoping and propagation of functional representations.

Another approach, which we describe in greater detail in the next section, pushes the issue of propagation onto the word level. While both approaches could easily be combined, this second option is more consistent with our aim of avoiding the implicit encoding of logic into fixed model parameters in favour of the explicit encoding in model structure.

6 An improved model

As we outlined in this paper, a key requirement for a compositional model motivated by formal semantics is the ability to propagate functional representations, but also to not propagate these representations when doing so is not semantically appropriate. Here, we propose a modification of the MV-RNN class of models that can capture this distinction without the need to move the composition logic into the non-linearity.

We add a parameter α to the representation of each word, controlling the degree to which its functional representation propagates after having been applied in its own composition step.

Thus, the composition step of the new model requires three equations:

$$M_p = W_M \begin{bmatrix} \frac{\alpha_a}{\alpha_a+\alpha_b} M_a \\ \frac{\alpha_b}{\alpha_a+\alpha_b} M_b \end{bmatrix} \tag{8}$$

$$\mathbf{v}_p = g\left(W_v \begin{bmatrix} M_a \mathbf{v}_b \\ M_b \mathbf{v}_a \end{bmatrix}\right) \tag{9}$$

$$\alpha_p = max(\alpha_a, \alpha_b) \tag{10}$$

Going back to the discussion of negation, this model has the clear advantage of being able to capture negation in the way we defined it. As $f_v(a, b)$ is unchanged, these two equations still hold:

$$M_{not} = J_\mu = J_\nu$$
$$\mathbf{v}_{not} = 0$$

However, as $f_M(a, b)$ is changed, the second set of equations changes. We use Z as the α-denominator ($Z = \alpha_a + \alpha_B$) for simplification:

$$\begin{aligned} f_M(a,b) &= W_M \begin{bmatrix} \frac{\alpha_a}{Z} M_a \\ \frac{\alpha_b}{Z} M_b \end{bmatrix} \\ &= \begin{bmatrix} I \\ I \end{bmatrix} \begin{bmatrix} \frac{\alpha_a}{Z} M_a \\ \frac{\alpha_b}{Z} M_b \end{bmatrix} \\ &= \frac{\alpha_a}{Z} M_a + \frac{\alpha_b}{Z} M_b \end{aligned}$$

Further, we still require the functional representation to remain constant under negation:

$$\forall a \in A : f_M(\text{not}, a) = M_a$$
$$\forall a \in A : f_M(\text{not}, f_M(\text{not}, a)) = M_a$$

Thus, we can infer the following two conditions on the new model:

$$\frac{\alpha_{not}}{Z} M_{not} \approx 0$$
$$\frac{\alpha_a}{Z} M_a \approx M_a$$

From our previous investigation we already know that $M_{not} = J_\mu \neq 0$, i.e. that *not* has a non-zero functional representation. While this caused a contradiction for the original MV-RNN model, the design of the improved model can resolve this issue through the α-parameter:

$$\alpha_{not} = 0$$

Thus, we can use this modified MV-RNN model to represent negation according to the principles outlined in this paper. The result $\alpha_{not} = 0$ is in accordance with our intuition about the propagation of functional aspects of a term: We commonly expect negation to directly affect the things under its scope (*not blue*) by choosing their semantic complement. However, this behaviour should not propagate outside of the scope of the negation. A *not blue car* is still very much a car, and when a film is not good, it is still very much a film.

7 Discussion and Further Work

In this paper, we investigated the capability of continuous vector space models to capture the semantics of logical operations in non-boolean cases. Recursive and recurrent vector models of meaning have enjoyed a considerable amount of success in recent years, and have been shown to work well on a number of tasks. However, the complexity and subsequent power of these models comes at the price that it can be difficult to establish which aspect of a model is responsible for what behaviour. This issue was recently highlighted by an investigation into recursive autoencoders for sentiment analysis (Scheible and Schütze, 2013). Thus, one of the key challenges in this area of research is the question of how to control the power of these models. This challenge motivated the work in this paper. By removing non-linearities and other parameters that could disguise model weaknesses, we focused our work on the basic model design. While such features enhance model power, they should not be used to compensate for inherently flawed model designs.

As a prerequisite for our investigation we established a suitable encoding of textual logic. Distributional representations have been well explained on the word level, but less clarity exists as to the semantic content of compositional vectors. With the tripartite meaning representation we proposed one possible approach in that direction, which we subsequently expanded by discussing how negation should be captured in this representation.

Having established a suitable and rigorous system for encoding meaning in compositional vectors, we were thus able to investigate the representative power of the MV-RNN model. We focused this paper on the case of negation, which has the advantage that it does not require many additional assumptions about the underlying semantics. Our investigation showed that the basic MV-RNN model is incompatible with our notion of negation and thus with any textual logic building on this proposal.

Subsequently, we analysed the reasons for this failure. We explained how the issue of negation affects the general class of MV-RNN models. Through the issue of double-negation we further showed how this issue is largely independent on the particular semantic encoding used. Based on this analysis we proposed an improved model that is able to capture such textual logic.

In summary, this paper has two key contributions. First, we developed a tripartite representation for vector space based models of semantics, incorporating multiple previous approaches to this topic. Based on this representation, the second contribution of this paper was a modified MV-RNN model that can capture effects such as negation in its inherent structure.

In future work, we would like to build on the proposals in this paper, both by extending our work on textual logic to include formulations for e.g. function words, quantifiers, or locative words. Similarly, we plan to experimentally validate these ideas. Possible tasks for this include sentiment analysis and relation extraction tasks such as in Socher et al. (2012) but also more specific tasks such as the *SEM shared task on negation scope and reversal (Morante and Blanco, 2012).

References

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

I. Beltagy, C. Chau, G. Boleda, D. Garrette, E. Erk, and R. Mooney. 2013. Montague meets markov: Deep semantics with probabilistic logical form. June.

W. Blacoe and M. Lapata. 2012. A comparison of vector-based representations for semantic composition. *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing*.

J. Bos. 2008. Wide-coverage semantic analysis with boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 277–286. Association for Computational Linguistics.

B. Coecke, M. Sadrzadeh, and S. Clark. 2010. Mathematical Foundations for a Compositional Distributional Model of Meaning. March.

J. R. Curran. 2004. *From distributional to semantic similarity*. Ph.D. thesis.

K. Erk and S. Padó. 2008. A structured vector space model for word meaning in context. *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08*, (October):897.

J. R. Firth. 1957. A synopsis of linguistic theory 1930-1955. *Studies in linguistic analysis*.

D. Garrette, K. Erk, and R. Mooney. 2011. Integrating logical representations with probabilistic information using markov logic. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 105–114. Association for Computational Linguistics.

E. Grefenstette and M. Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of EMNLP*, pages 1394–1404.

E. Grefenstette, G. Dinu, Y. Zhang, M. Sadrzadeh, and M. Baroni. 2013. Multi-step regression learning for compositional distributional semantics. In *Proceedings of the Tenth International Conference on Computational Semantics*. Association for Computational Linguistics.

E. Grefenstette. 2013. Towards a formal distributional semantics: Simulating logical calculi with tensors. *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*.

G. Grefenstette. 1994. *Explorations in automatic thesaurus discovery*.

K. M. Hermann and P. Blunsom. 2013. The role of syntax in vector space models of compositional semantics. In *Proceedings of ACL*, Sofia, Bulgaria, August. Association for Computational Linguistics.

D. Kartsaklis, M. Sadrzadeh, and S. Pulman. 2012. A unified sentence space for categorical distributional-compositional semantics: Theory and experiments. In *Proceedings of 24th International Conference on Computational Linguistics (COLING 2012): Posters*, pages 549–558, Mumbai, India, December.

T. K. Landauer and S. T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*.

J. Mitchell and M. Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL*, volume 8.

J. Mitchell and M. Lapata. 2010. Composition in Distributional Models of Semantics. *Cognitive Science*.

R. Montague. 1974. English as a Formal Language. *Formal Semantics: The Essential Readings*.

R. Morante and E. Blanco. 2012. *SEM 2012 shared task: resolving the scope and focus of negation. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 265–274, Stroudsburg, PA, USA. Association for Computational Linguistics.

M. Richardson and P. Domingos. 2006. Markov logic networks. *Machine learning*, 62(1-2):107–136.

C. Scheible and H. Schütze. 2013. Cutting recursive autoencoder trees. In *Proceedings of the International Conference on Learning Representations*.

H. Schütze. 1998. Automatic word sense discrimination. *Computational linguistics*, 24(1):97–123.

R. Socher, E. H. Huang, J. Pennington, A. Y. Ng, and C. D. Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *Advances in Neural Information Processing Systems*, 24:801–809.

R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing*, pages 1201–1211.

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

D. Widdows and S. Peters. 2003. Word vectors and quantum logic: Experiments with negation and disjunction. *Mathematics of language*, 8(141-154).

F. M. Zanzotto, I. Korkontzelos, F. Fallucchi, and S. Manandhar. 2010. Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1263–1271. Association for Computational Linguistics.