A Structured Distributional Semantic Model : Integrating Structure with Semantics

Kartik Goyal* **Sujay Kumar Jauhar*** **Huiying Li***
Mrinmaya Sachan* **Shashank Srivastava*** **Eduard Hovy**
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
{kartikgo,sjauhar,huiyingl,mrinmays,shashans,hovy}@cs.cmu.edu

Abstract

In this paper we present a novel approach (SDSM) that incorporates structure in distributional semantics. SDSM represents meaning as relation specific distributions over syntactic neighborhoods. We empirically show that the model can effectively represent the semantics of single words and provides significant advantages when dealing with phrasal units that involve word composition. In particular, we demonstrate that our model outperforms both state-of-the-art window-based word embeddings as well as simple approaches for composing distributional semantic representations on an artificial task of verb sense disambiguation and a real-world application of judging event coreference.

1 Introduction

With the advent of statistical methods for NLP, Distributional Semantic Models (DSMs) have emerged as powerful method for representing word semantics. In particular, the distributional vector formalism, which represents meaning by a distribution over neighboring words, has gained the most popularity.

DSMs are widely used in information retrieval (Manning et al., 2008), question answering (Tellex et al., 2003), semantic similarity computation (Wong and Raghavan, 1984; McCarthy and Carroll, 2003), automated dictionary building (Curran, 2003), automated essay grading (Landauer and Dutnais, 1997), word-sense discrimination and disambiguation (McCarthy et al., 2004; Schütze, 1998), selectional preference modeling (Erk, 2007) and identification of translation equivalents (Hjelm, 2007).

Systems that use DSMs implicitly make a bag of words assumption: that the meaning of a phrase can be reasonably estimated from the meaning of its constituents. However, semantics in natural language is a compositional phenomenon, encompassing interactions between syntactic structures, and the meaning of lexical constituents. It follows that the DSM formalism lends itself poorly to composition since it implicitly disregards syntactic structure. For instance, the distributions for "Lincoln", "Booth", and "killed" when merged produce the same result regardless of whether the input is "Booth killed Lincoln" or "Lincoln killed Booth". As suggested by Pantel and Lin (2000) and others, modeling the distribution over preferential attachments for each syntactic relation separately can yield greater expressive power.

Attempts have been made to model linguistic composition of individual word vectors (Mitchell and Lapata, 2009), as well as remedy the inherent failings of the standard distributional approach (Erk and Padó, 2008). The results show varying degrees of efficacy, but have largely failed to model deeper lexical semantics or compositional expectations of words and word combinations.

In this paper we propose an extension to the traditional DSM model that explicitly preserves structural information and permits the approximation of distributional expectation over dependency relations. We extend the generic DSM model by representing a word as distributions over relation-specific syntactic neighborhoods. One can think of the Structured DSM (SDSM) representation of a word/phrase as several vectors defined over the same vocabulary, each vector representing the

*Equally contributing authors

Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality, pages 20–29,
Sofia, Bulgaria, August 9 2013.

word's selectional preferences for a different syntactic argument. We argue that this representation captures individual word semantics effectively, and is better able to express the semantics of composed units.

The overarching theme of our framework of evaluation is to explore the semantic space of the SDSM. We do this by measuring its ability to discriminate between varying surface forms of the same underlying concept. We perform the following set of experiments to evaluate its expressive power, and conclude the following:

1. Experiments with single words on similarity scoring and substitute selection: SDSM performs at par with window-based distributional vectors.

2. Experiments with phrasal units on two-word composition: state-of-the-art results are produced on the dataset from Mitchell and Lapata (2008) in terms of correlation with human judgment.

3. Experiments with larger structures on the task of judging event coreferentiality: SDSM shows superior performance over state-of-the-art window-based word embeddings, and simple models for composing distributional semantic representations.

2 Related Work

Distributional Semantic Models are based on the intuition that "a word is characterized by the company it keeps" (Firth, 1957). While DSMs have been very successful on a variety of NLP tasks, they are generally considered inappropriate for deeper semantics because they lack the ability to model composition, modifiers or negation.

Recently, there has been a surge in studies to model a stronger form of semantics by phrasing the problem of DSM compositionality as one of vector composition. These techniques derive the meaning of the combination of two words a and b by a single vector $c = f(a, b)$. Mitchell and Lapata (2008) propose a framework to define the composition $c = f(a, b, r, K)$ where r is the relation between a and b, and K is the additional knowledge used to define composition.

While the framework is quite general, most models in the literature tend to disregard K and r and are generally restricted to component-wise addition and multiplication on the vectors to be composed, with slight variations. Dinu and Lapata (2010) and Séaghdha and Korhonen (2011) introduced a probabilistic model to represent word meanings by a latent variable model. Subsequently, other high-dimensional extensions by Rudolph and Giesbrecht (2010), Baroni and Zamparelli (2010) and Grefenstette et al. (2011), regression models by Guevara (2010), and recursive neural network based solutions by Socher et al. (2012) and Collobert et al. (2011) have been proposed.

Pantel and Lin (2000) and Erk and Padó (2008) attempted to include syntactic context in distributional models. However, their approaches do not explicitly construct phrase-level meaning from words which limits their applicability to real world problems. A quasi-compositional approach was also attempted in Thater et al. (2010) by a systematic combination of first and second order context vectors. To the best of our knowledge the formulation of composition we propose is the first to account for K and r within the general framework of composition $c = f(a, b, r, K)$.

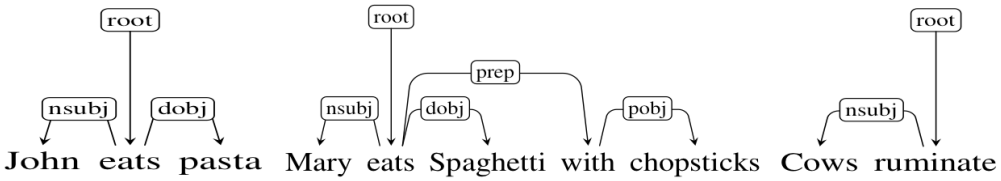
3 Structured Distributional Semantics

In this section, we describe our Structured Distributional Semantic framework in detail. We first build a large knowledge base from sample english texts and use it to represent basic lexical units. Next, we describe a technique to obtain the representation for larger units by composing their constituents.

3.1 The PropStore

To build a lexicon of SDSM representations for a given vocabulary we construct a proposition knowledge base (the PropStore) by processing the text of Simple English Wikipedia through a dependency parser. Dependency arcs are stored as 3-tuples of the form $\langle w_1, r, w_2 \rangle$, denoting occurrences of words w_1 and word w_2 related by the syntactic dependency r. We also store sentence identifiers for each triple for reasons described later. In addition to the words' surface-forms, the PropStore also stores their POS tags, lemmas, and Wordnet supersenses.

The PropStore can be used to query for preferred expectations of words, supersenses, relations, etc., around a given word. In the example in Figure 1, the query (SST(W_1)



1) { (John/NNP/john/Noun.person , nsubj, eats/VBG/eat/verb.consumption), (eats/VBG/eat/verb.consumption, dobj, pasta/NN/pasta/noun.food) }
2) { (Mary/NNP/mary/Noun.person), nsubj, (eats/VBG/eat/verb.consumption) ... }
3) { (Cows/NNP/cow/Noun.animal),nsubj,(ruminate/VBG/ruminate/verb.consumption) }

Figure 1: Sample sentences & triples

= verb.consumption, ?, dobj) i.e., "what is consumed", might return expectations [pasta:1, spaghetti:1, mice:1 . . .]. In our implementation, the relations and POS tags are obtained using the Fanseparser (Tratz and Hovy, 2011), supersense tags using sst-light (Ciaramita and Altun, 2006), and lemmas are obtained from Wordnet (Miller, 1995).

3.2 Building the Representation

Next, we describe a method to represent lexical entries as structured distributional matrices using the PropStore.

The canonical form of a concept C (word, phrase etc.) in the SDSM framework is a matrix M^C, whose entry M^C_{ij} is a list of sentence identifiers obtained by querying the PropStore for contexts in which C appears in the syntactic neighborhood of the word j linked by the dependency relation i. As with other distributional models in the literature, the content of a cell is the frequency of co-occurrence of its concept and word under the given relational constraint.

This canonical matrix form can be interpreted in several different ways. Each interpretation is based on a different normalization scheme.

1. **Row Norm**: Each row of the matrix is interpreted as a distribution over words that attach to the target concept with the given dependency relation.

$$M^C_{ij} = \frac{M_{ij}}{\Sigma_j M_{ij}} \qquad \forall i$$

2. **Full Norm**: The entire matrix is interpreted as a distribution over the word-relation pairs which can attach to the target concept.

$$M^C_{ij} = \frac{M_{ij}}{\Sigma_{i,j} M_{ij}} \qquad \forall i, j$$

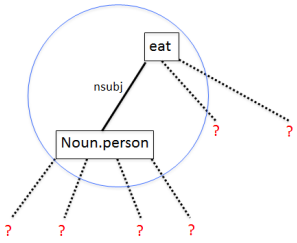


Figure 2: Mimicking composition of two words

3. **Collapsed Vector Norm**: The columns of the matrix are collapsed to form a standard normalized distributional vector trained on dependency relations rather than sliding windows.

$$M^C_j = \frac{\Sigma_i M_{ij}}{\Sigma_{i,j} M_{ij}} \qquad \forall j$$

3.3 Mimicking Compositionality

For representing intermediate multi-word phrases, we extend the above word-relation matrix symbolism in a bottom-up fashion. The combination hinges on the intuition that when lexical units combine to form a larger syntactically connected phrase, the representation of the phrase is given by its own distributional neighborhood within the embedded parse tree. The distributional neighborhood of the net phrase can be computed using the PropStore given syntactic relations anchored on its parts. For the example in Figure 1, we can compose SST(w_1) = Noun.person and Lemma(W_1) = eat with relation 'nsubj' to obtain expectations around "people eat" yielding [pasta:1, spaghetti:1 . . .] for the *object* relation ([dining room:2, restaurant:1 . . .] for the *location* relation, etc.) (See Figure 2). Larger phrasal queries can be built to answer questions like "What do people in China eat with?", "What do cows do?", etc. All of this helps

us to account for both relation r and knowledge K obtained from the PropStore within the compositional framework $c = f(a, b, r, K)$.

The general outline to obtain a composition of two words is given in Algorithm 1. Here, we first determine the sentence indices where the two words w_1 and w_2 occur with relation r. Then, we return the expectations around the two words within these sentences. Note that the entire algorithm can conveniently be written in the form of database queries to our PropStore.

Algorithm 1 ComposePair(w_1, r, w_2)

$M_1 \leftarrow$ queryMatrix(w_1)
$M_2 \leftarrow$ queryMatrix(w_2)
SentIDs $\leftarrow M_1(r) \cap M_2(r)$
return (($M_1 \cap$ SentIDs) $\cup$ ($M_2 \cap$ SentIDs))

Similar to the two-word composition process, given a parse subtree T of a phrase, we obtain its matrix representation of empirical counts over word-relation contexts. This procedure is described in Algorithm 2. Let the $E = \{e_1 \dots e_n\}$ be the set of edges in T, $e_i = (w_{i1}, r_i, w_{i2}) \forall i = 1 \dots n$.

Algorithm 2 ComposePhrase(T)

SentIDs $\leftarrow$ All Sentences in corpus
for $i = 1 \rightarrow n$ **do**
    $M_{i1} \leftarrow$ queryMatrix(w_{i1})
    $M_{i2} \leftarrow$ queryMatrix(w_{i2})
    SentIDs $\leftarrow$ SentIDs $\cap (M_1(r_i) \cap M_2(r_i))$
end for
return (($M_{11} \cap$ SentIDs) $\cup$ ($M_{12} \cap$ SentIDs) $\cdots \cup$ ($M_{n1} \cap$ SentIDs) $\cup$ ($M_{n2} \cap$ SentIDs))

3.4 Tackling Sparsity

The SDSM model reflects syntactic properties of language through preferential filler constraints. But by distributing counts over a set of relations the resultant SDSM representation is comparatively much sparser than the DSM representation for the same word. In this section we present some ways to address this problem.

3.4.1 Sparse Back-off

The first technique to tackle sparsity is to back off to progressively more general levels of linguistic granularity when sparse matrix representations for words or compositional units are encountered or when the word or unit is not in the lexicon. For example, the composition "Balthazar eats" cannot be directly computed if the named entity "Balthazar" does not occur in the PropStore's knowledge base. In this case, a query for a supersense substitute – "Noun.person eat" – can be issued instead. When supersenses themselves fail to provide numerically significant distributions for words or word combinations, a second back-off step involves querying for POS tags. With coarser levels of linguistic representation, the expressive power of the distributions becomes diluted. But this is often necessary to handle rare words. Note that this is an issue with DSMs too.

3.4.2 Densification

In addition to the back-off method, we also propose a secondary method for "densifying" distributions. A concept's distribution is modified by using words encountered in its syntactic neighborhood to infer counts for other semantically similar words. In other terms, given the matrix representation of a concept, densification seeks to populate its null columns (which each represent a word-dimension in the structured distributional context) with values weighted by their scaled similarities to words (or effectively word-dimensions) that actually occur in the syntactic neighborhood.

For example, suppose the word "play" had an "nsubj" preferential vector that contained the following counts: [cat:4 ; Jane:2]. One might then populate the column for "dog" in this vector with a count proportional to its similarity to the word cat (say 0.8), thus resulting in the vector [cat:4 ; Jane:2 ; dog:3.2]. These counts could just as well be probability values or PMI associations (suitably normalized). In this manner, the k most similar word-dimensions can be densified for each word that actually occurs in a syntactic context. As with sparse back-off, there is an inherent trade-off between the degree of densification k and the expressive power of the resulting representation.

3.4.3 Dimensionality Reduction

The final method tackles the problem of sparsity by reducing the representation to a dense low-dimensional word embedding using singular value decomposition (SVD). In a typical term-document matrix, SVD finds a low-dimensional approximation of the original matrix where columns become latent concepts while similarity structure between rows are preserved. The PropStore, as described in Section 3.1, is an order-3 tensor with w_1, w_2 and

rel as its three axes. We explore the following two possibilities to perform dimensionality reduction using SVD.

Word-word matrix SVD. In this experiment, we preserve the axes w_1 and w_2 and ignore the relational information. Following the SVD regime ($W = U\Sigma V^T$) where Σ is a square diagonal matrix of k largest singular values, and U and V are $m \times k$ and $n \times k$ matrices respectively. We adopt matrix U as the compacted concept representation.

Tensor SVD. To remedy the relation-agnostic nature of the word-word SVD matrix representation, we use tensor SVD (Vasilescu and Terzopoulos, 2002) to preserve the structural information. The mode-n vectors of an order-N tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ are the I_n-dimensional vectors obtained from $\mathcal{A}$ by varying index i_n while keeping other indices fixed. The matrix formed by all the mode-n vectors is a mode-n flattening of the tensor. To obtain the compact representations of concepts we thus first apply mode w_1 flattening and then perform SVD on the resulting tensor.

4 Single Word Evaluation

In this section we describe experiments and results for judging the expressive power of the structured distributional representation for individual words. We use a similarity scoring task and a lexical substitute selection task for the purpose of this evaluation. We compare the SDSM representation to standard window-based distributional vectors trained on the same corpus (Simple English Wikipedia). We also experiment with different normalization techniques outlined in Section 3.2, which effectively lead to structured distributional representations with distinct interpretations.

We experimented with various similarity metrics and found that the normalized cityblock distance metric provides the most stable results.

$$CityBlock(X,Y) = \frac{ArcTan(d(X,Y))}{d(X,Y)}$$

$$d(X,Y) = \frac{1}{|R|}\sum_{r \in R} d(X_r, Y_r)$$

Results in the rest of this section are thus reported using the normalized cityblock metric. We also report experimental results for the two methods of alleviating sparsity discussed in Section 3.4, namely, densification and SVD.

4.1 Similarity Scoring

On this task, the different semantic representations were used to compute similarity scores between two (out of context) words. We used a dataset from Finkelstein et al. (2002) for our experiments. It consists of 353 pairs of words along with an averaged similarity score on a scale of 1.0 to 10.0 obtained from 13–16 human judges.

4.2 Lexical Substitute Selection

In the second task, the same set of semantic representations was used to produce a similarity ranking on the Turney (2002) ESL dataset. This dataset comprises 50 words that appear in a context (we discarded the context in this experiment), along with 4 candidate lexical substitutions. We evaluate the semantic representations on the basis of their ability to discriminate the top-ranked candidate.[1]

4.3 Results and Discussion

Table 1 summarizes the results for the window-based baseline and each of the structured distributional representations on both tasks. It shows that our representations for single words are competitive with window based distributional vectors. Densification in certain conditions improves our results, but no consistent pattern is discernible. This can be attributed to the trade-off between the gain from generalization and the noise introduced by semantic drift.

Hence we resort to dimensionality reduction as an additional method of reducing sparsity. Table 2 gives correlation scores on the Finkelstein et al. (2002) dataset when SVD is performed on the representations, as described in Section 3.4.3. We give results when 100 and 500 principal components are preserved for both SVD techniques.

These experiments suggest that though afflicted by sparsity, the proposed structured distributional paradigm is competitive with window-based distributional vectors. In the following sections we show that that the framework provides considerably greater power for modeling composition when dealing with units consisting of more than one word.

[1] While we are aware of the standard lexical substitution corpus from McCarthy and Navigli (2007) we chose the one mentioned above for its basic vocabulary, lower dependence on context, and simpler evaluation framework.

Model	Finklestein (Corr.)	ESL (% Acc.)
DSM	**0.283**	0.247
Collapsed	0.260	0.178
FullNorm	0.282	0.192
RowNorm	0.236	0.264
Densified RowNorm	0.259	**0.267**

Table 1: Single Word Evaluation

Model	Correlation
matSVD100	0.207
matSVD500	0.221
tenSVD100	0.267
tenSVD500	**0.315**

Table 2: Finklestein: Correlation using SVD

5 Verb Sense Disambiguation using Composition

In this section, we examine how well our model performs composition on a pair of words. We derive the compositional semantic representations for word pairs from the M&L dataset (Mitchell and Lapata, 2008) and compare our performance with M&L's additive and multiplicative models of composition.

5.1 Dataset

The M&L dataset consists of polysemous intransitive verb and subject pairs that co-occur at least 50 times in the BNC corpus. Additionally two landmark words are given for every polysemous verb, each corresponding to one of its senses. The subject nouns provide contextual disambiguation for the senses of the verb. For each [subject, verb, landmark] tuple, a human assigned score on a 7-point scale is provided, indicating the compatibility of the landmark with the reference verb-subj pair. For example, for the pair "gun bomb", landmark "thunder" is more similar to the verb than landmark "prosper". The corpus contains 120 tuples and altogether 3600 human judgments. Reliability of the human ratings is examined by calculating inter-annotator Spearman's ρ correlation coefficient.

5.2 Experiment procedure

For each tuple in the dataset, we derive the composed word-pair matrix for the reference *verb-subj* pair based on the algorithm described in Section 3.3 and query the single-word matrix for the landmark word. A few modifications are made to adjust the algorithm for the current task:

1. In our formulation, the dependency relation needs to be specified in order to compose a pair of words. Hence, we determine the five most frequent relations between w_1 and w_2 by querying the PropStore. We then use the algorithm in Section 3.3 to compose the verb-subj word pair using these relations, resulting in five composed representations.

2. The word pairs in M&L corpus are extracted from a parsed version of the BNC corpus, while our PropStore is built on Simple Wikipedia texts, whose vocabulary is significantly different from that of the BNC corpus. This causes *null* returns in our PropStore queries, in which case we back-off to retrieving results for super-sense tags of both the words. Finally, the composed matrix and the landmark matrix are compared against each other by different matrix distance measures, which results in a similarity score. For a [subject, verb, landmark] tuple, we average the similarity scores yielded by the relations obtained in 1.

The Spearman Correlation ρ between our similarity ratings and the ones assigned by human judges is computed over all the tuples. Following M&L's experiments, the inter-annotator agreement correlation coefficient serves an upper bound on the task.

5.3 Results and Discussion

As in Section 4, we choose the cityblock measure as the similarity metric of choice. Table 3 shows the evaluation results for two word composition. Except for row normalization, both forms of normalization in the structured distributional paradigm show significant improvement over the results reported by M&L. The results are statistically significant at p-value = 0.004 and 0.001 for Full Norm and Collapsed Vector Norm, respectively.

Model	ρ
M&L combined	0.19
Row Norm	0.134
Full Norm	0.289
Collapsed Vector Norm	0.259
UpperBound	0.40

Table 3: Two Word Composition Evaluation

These results validate our hypothesis that the integration of structure into distributional semantics

as well as our framing of word composition together outperform window-based representations under simplistic models of composition such as addition and multiplication. This finding is further re-enforced in the following experiments on event coreferentiality judgment.

6 Event Coreference Judgment

Given the SDSM formulation and assuming no sparsity constraints, it is possible to calculate SDSM matrices for composed concepts. However, are these correct? Intuitively, if they truly capture semantics, the two SDSM matrix representations for "Booth assassinated Lincoln" and "Booth shot Lincoln with a gun" should be (almost) the same. To test this hypothesis we turn to the task of predicting whether two event mentions are coreferent or not, even if their surface forms differ.

While automated resolution of entity coreference has been an actively researched area (Haghighi and Klein, 2009; Stoyanov et al., 2009; Raghunathan et al., 2010), there has been relatively little work on event coreference resolution. Lee et al. (2012) perform joint cross-document entity and event coreference resolution using the two-way feedback between events and their arguments.

In this paper, however, we only consider coreferentiality between pairs of events. Formally, two event mentions generally refer to the same event when their respective actions, agents, patients, locations, and times are (almost) the same. Given the non-compositional nature of determining equality of locations and times, we represent each event mention by a triple $\mathbf{E} = (e, a, p)$ for the event, agent, and patient.

While linguistic theory of argument realization is a debated research area (Levin and Rappaport Hovav, 2005; Goldberg, 2005), it is commonly believed that event structure (Moens and Steedman, 1988) centralizes on the predicate, which governs and selects its role arguments (Jackendoff, 1987). In the corpora we use for our experiments, most event mentions are verbs. However, when nominalized events are encountered, we replace them by their verbal forms. We use SRL Collobert et al. (2011) to determine the agent and patient arguments of an event mention. When SRL fails to determine either role, its empirical substitutes are obtained by querying the PropStore for the most likely word expectations for the role. The triple (e, a, p) is thus the composition of the triples (a, rel_{agent}, e) and $(p, rel_{patient}, e)$, and hence a complex object. To determine equality of this complex composed representation we generate three levels of progressively simplified event constituents for comparison:

Level 1: Full Composition:
$M_{full} = \textit{ComposePhrase}(e, a, p)$.
Level 2: Partial Composition:
$M_{part:EA} = \textit{ComposePair}(e, r, a)$
$M_{part:EP} = \textit{ComposePair}(e, r, p)$.
Level 3: No Composition:
$M_E = queryMatrix(e)$
$M_A = queryMatrix(a)$
$M_P = queryMatrix(p)$.

To judge coreference between events **E1** and **E2**, we compute pairwise similarities $\text{Sim}(M1_{full}, M2_{full})$, $\text{Sim}(M1_{part:EA}, M2_{part:EA})$, etc., for each level of the composed triple representation. Furthermore, we vary the computation of similarity by considering different levels of granularity (lemma, SST), various choices of distance metric (Euclidean, Cityblock, Cosine), and score normalization techniques (Row-wise, Full, Column collapsed). This results in 159 similarity-based features for every pair of events, which are used to train a classifier to make a binary decision for coreferentiality.

6.1 Datasets

We evaluate our method on two datasets and compare it against four baselines, two of which use window based distributional vectors and two that employ weaker forms of composition.

IC Event Coreference Corpus: The dataset (citation suppressed), drawn from 100 news articles about violent events, contains manually created annotations for 2214 pairs of co-referent and non-coreferent events each. Where available, events' semantic role-fillers for *agent* and *patient* are annotated as well. When missing, empirical substitutes were obtained by querying the PropStore for the preferred word attachments.

EventCorefBank (ECB) corpus: This corpus (Bejan and Harabagiu, 2010) of 482 documents from Google News is clustered into 45 topics, with event coreference chains annotated over each topic. The event mentions are enriched with semantic roles to obtain the canonical event structure described above. Positive instances are ob-

	IC Corpus				ECB Corpus			
	Prec	Rec	F-1	Acc	Prec	Rec	F-1	Acc
SDSM	**0.916**	0.929	**0.922**	**0.906**	0.901	0.401	**0.564**	**0.843**
Senna	0.850	0.881	0.865	0.835	0.616	**0.408**	0.505	0.791
DSM	0.743	0.843	0.790	0.740	0.854	0.378	0.524	0.830
MVC	0.756	**0.961**	0.846	0.787	**0.914**	0.353	0.510	0.831
AVC	0.753	0.941	0.837	0.777	0.901	0.373	0.528	0.834

Table 4: Cross-validation Performance on IC and ECB dataset

tained by taking pairwise event mentions within each chain, and negative instances are generated from pairwise event mentions across chains, but within the same topic. This results in 11039 positive instances and 33459 negative instances.

6.2 Baselines:

To establish the efficacy of our model, we compare SDSM against a purely window-based baseline (DSM) trained on the same corpus. In our experiments we set a window size of three words to either side of the target. We also compare SDSM against the window-based embeddings trained using a recursive neural network (SENNA) (Collobert et al., 2011) on both datsets. SENNA embeddings are state-of-the-art for many NLP tasks. The second baseline uses SENNA to generate level 3 similarity features for events' individual words (agent, patient and action). As our final set of baselines, we extend two simple techniques proposed by Mitchell and Lapata (2008) that use element-wise addition and multiplication operators to perform composition. The two baselines thus obtained are AVC (element-wise addition) and MVC (element-wise multiplication).

6.3 Results and Discussion:

We experimented with a number of common classifiers, and selected decision-trees (J48) as they give the best classification accuracy. Table 4 summarizes our results on both datasets.

The results reveal that the SDSM model consistently outperforms DSM, SENNA embeddings, and the MVC and AVC models, both in terms of F-1 score and accuracy. The IC corpus comprises of domain specific texts, resulting in high lexical overlap between event mentions. Hence, the scores on the IC corpus are consistently higher than those on the ECB corpus.

The improvements over DSM and SENNA embeddings, support our hypothesis that syntax lends greater expressive power to distributional semantics in compositional configurations. Furthermore, the increase in predictive accuracy over MVC and AVC shows that our formulation of composition of two words based on the relation binding them yields a stronger form of composition than simple additive and multiplicative models.

Next, we perform an ablation study to determine the most predictive features for the task of determining event coreferentiality. The forward selection procedure reveals that the most informative attributes are the level 2 compositional features involving the agent and the action, as well as their individual level 3 features. This corresponds to the intuition that the agent and the action are the principal determiners for identifying events. Features involving the patient and level 1 features are least useful. The latter involves full composition, resulting in sparse representations and hence have low predictive power.

7 Conclusion and Future Work

In this paper we outlined an approach that introduces structure into distributional semantics. We presented a method to compose distributional representations of individual units into larger composed structures. We tested the efficacy of our model on several evaluation tasks. Our model's performance is competitive for tasks dealing with semantic similarity of individual words, even though it suffers from the problem of sparsity. Additionally, it outperforms window-based approaches on tasks involving semantic composition. In future work we hope to extend this formalism to other semantic tasks like paraphrase detection and recognizing textual entailment.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This work was supported in part by the following grants: NSF grant IIS-1143703, NSF award IIS-1147810, DARPA grant FA87501220342.

References

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*, EMNLP '10, pages 1183–1193, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cosmin Adrian Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1412–1422, Stroudsburg, PA, USA. Association for Computational Linguistics.

Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 594–602, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 999888:2493–2537, November.

James Richard Curran. 2003. From distributional to semantic similarity. Technical report.

Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1162–1172, Stroudsburg, PA, USA. Association for Computational Linguistics.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 897–906, Stroudsburg, PA, USA. Association for Computational Linguistics.

Katrin Erk. 2007. A simple, similarity-based model for selectional preferences.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. In *ACM Transactions on Information Systems*, volume 20, pages 116–131, January.

John R. Firth. 1957. A Synopsis of Linguistic Theory, 1930-1955. *Studies in Linguistic Analysis*, pages 1–32.

Adele E. Goldberg. 2005. *Argument Realization: Cognitive Grouping and Theoretical Extensions*.

Edward Grefenstette, Mehrnoosh Sadrzadeh, Stephen Clark, Bob Coecke, and Stephen Pulman. 2011. Concrete sentence spaces for compositional distributional models of meaning. In *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, pages 125–134, Stroudsburg, PA, USA. Association for Computational Linguistics.

Emiliano Guevara. 2010. A regression model of adjective-noun compositionality in distributional semantics. In *Proceedings of the 2010 Workshop on GEometrical Models of Natural Language Semantics*, GEMS '10, pages 33–37, Stroudsburg, PA, USA. Association for Computational Linguistics.

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1152–1161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hans Hjelm. 2007. Identifying cross language term equivalents using statistical machine translation and distributional association measures. In *Proceedings of NODALIDA*, pages 97–104. Citeseer.

Ray Jackendoff. 1987. The status of thematic roles in linguistic theory. *Linguistic Inquiry*, 18(3):369–411.

Thomas K Landauer and Susan T. Dutnais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, pages 211–240.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 489–500, Stroudsburg, PA, USA. Association for Computational Linguistics.

Beth Levin and Malka Rappaport Hovav. 2005. *Argument Realization*. Cambridge University Press.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Diana McCarthy and John Carroll. 2003. Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Comput. Linguist.*, 29(4):639–654, December.

Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), Prague, Czech Republic*, pages 48–53.

Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. 2004. Finding predominant word senses in untagged text. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, November.

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

Jeff Mitchell and Mirella Lapata. 2009. Language models based on semantic composition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 430–439, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marc Moens and Mark Steedman. 1988. Temporal ontology and temporal reference. *Computational linguistics*, 14(2):15–28.

Patrick Pantel and Dekang Lin. 2000. Word-for-word glossing with contextually similar words. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, NAACL 2000, pages 78–85, Stroudsburg, PA, USA. Association for Computational Linguistics.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 492–501, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sebastian Rudolph and Eugenie Giesbrecht. 2010. Compositional matrix-space models of language. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 907–916, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Comput. Linguist.*, 24(1):97–123.

Diarmuid Ó Séaghdha and Anna Korhonen. 2011. Probabilistic models of similarity in syntactic context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1047–1057, Stroudsburg, PA, USA. Association for Computational Linguistics.

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*, EMNLP-CoNLL '12, pages 1201–1211, Stroudsburg, PA, USA. Association for Computational Linguistics.

Veselin Stoyanov, Nathan Gilbert, Claire Cardie, and Ellen Riloff. 2009. Conundrums in noun phrase coreference resolution: making sense of the state-of-the-art. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 656–664, Stroudsburg, PA, USA. Association for Computational Linguistics.

Stefanie Tellex, Boris Katz, Jimmy J. Lin, Aaron Fernandes, and Gregory Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *SIGIR*, pages 41–47.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2010. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 948–957, Stroudsburg, PA, USA. Association for Computational Linguistics.

Stephen Tratz and Eduard Hovy. 2011. A fast, accurate, non-projective, semantically-enriched parser. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1257–1268, Stroudsburg, PA, USA. Association for Computational Linguistics.

Peter D. Turney. 2002. Mining the web for synonyms: Pmi-ir versus lsa on toefl. *CoRR*.

M. Alex O. Vasilescu and Demetri Terzopoulos. 2002. Multilinear analysis of image ensembles: Tensorfaces. In *In Proceedings of the European Conference on Computer Vision*, pages 447–460.

S. K. M. Wong and Vijay V. Raghavan. 1984. Vector space model of information retrieval: a reevaluation. In *Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '84, pages 167–185, Swinton, UK. British Computer Society.