Take and *Took*, *Gaggle* and *Goose*, *Book* and *Read*: Evaluating the Utility of Vector Differences for Lexical Relation Learning

Ekaterina Vylomova,[1] **Laura Rimell,**[2] **Trevor Cohn,**[1] and **Timothy Baldwin**[1]
[1]Department of Computing and Information Systems, University of Melbourne
[2]Computer Laboratory, University of Cambridge
evylomova@gmail.com laura.rimell@cl.cam.ac.uk {tcohn,tbaldwin}@unimelb.edu.au

Abstract

Recent work has shown that simple vector subtraction over word embeddings is surprisingly effective at capturing different lexical relations, despite lacking explicit supervision. Prior work has evaluated this intriguing result using a word analogy prediction formulation and hand-selected relations, but the generality of the finding over a broader range of lexical relation types and different learning settings has not been evaluated. In this paper, we carry out such an evaluation in two learning settings: (1) spectral clustering to induce word relations, and (2) supervised learning to classify vector differences into relation types. We find that word embeddings capture a surprising amount of information, and that, under suitable supervised training, vector subtraction generalises well to a broad range of relations, including over unseen lexical items.

1 Introduction

Learning to identify lexical relations is a fundamental task in natural language processing ("NLP"), and can contribute to many NLP applications including paraphrasing and generation, machine translation, and ontology building (Banko et al., 2007; Hendrickx et al., 2010).

Recently, attention has been focused on identifying lexical relations using word embeddings, which are dense, low-dimensional vectors obtained either from a "predict-based" neural network trained to predict word contexts, or a "count-based" traditional distributional similarity method combined with dimensionality reduction. The skip-gram model of Mikolov et al. (2013a) and other similar language models have been shown to perform well on an analogy completion task (Mikolov et al., 2013b; Mikolov et al., 2013c; Levy and Goldberg, 2014a), in the space of *relational similarity* prediction (Turney, 2006), where the task is to predict the missing word in analogies such as A:B :: C:*–?–*. A well-known example involves predicting the vector $\mathbf{queen}$ from the vector combination $\mathbf{king} - \mathbf{man} + \mathbf{woman}$, where linear operations on word vectors appear to capture the lexical relation governing the analogy, in this case OPPOSITE-GENDER. The results extend to several semantic relations such as CAPITAL-OF ($\mathbf{paris} - \mathbf{france} + \mathbf{poland} \approx \mathbf{warsaw}$) and morphosyntactic relations such as PLURALISATION ($\mathbf{cars} - \mathbf{car} + \mathbf{apple} \approx \mathbf{apples}$). Remarkably, since the model is not trained for this task, the relational structure of the vector space appears to be an emergent property.

The key operation in these models is *vector difference*, or *vector offset*. For example, the $\mathbf{paris} - \mathbf{france}$ vector appears to encode CAPITAL-OF, presumably by cancelling out the features of $\mathbf{paris}$ that are France-specific, and retaining the features that distinguish a capital city (Levy and Goldberg, 2014a). The success of the simple offset method on analogy completion suggests that the difference vectors ("DIFFVEC" hereafter) must themselves be meaningful: their direction and/or magnitude encodes a lexical relation.

Previous analogy completion tasks used with word embeddings have limited coverage of lexical relation types. Moreover, the task does not explore the full implications of DIFFVECs as meaningful vector space objects in their own right, because it only looks for a one-best answer to the particular lexical analogies in the test set. In this paper, we introduce a new, larger dataset covering many well-known lexical relation types from the linguistics and cognitive science literature. We then apply DIFFVECs to two new tasks: unsupervised and supervised relation extraction. First, we cluster the DIFFVECs to test whether the clusters map onto true lexical relations. We find that the clustering

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1671–1682,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

works remarkably well, although syntactic relations are captured better than semantic ones.

Second, we perform classification over the DIFFVECs and obtain remarkably high accuracy in a closed-world setting (over a predefined set of word pairs, each of which corresponds to a lexical relation in the training data). When we move to an open-world setting including random word pairs — many of which do not correspond to any lexical relation in the training data — the results are poor. We then investigate methods for better attuning the learned class representation to the lexical relations, focusing on methods for automatically synthesising negative instances. We find that this improves the model performance substantially.

We also find that hyper-parameter optimised count-based methods are competitive with predict-based methods under both clustering and supervised relation classification, in line with the findings of Levy et al. (2015a).

2 Background and Related Work

A lexical relation is a binary relation r holding between a word pair (w_i, w_j); for example, the pair $(\textit{cart}, \textit{wheel})$ stands in the WHOLE-PART relation. Relation learning in NLP includes relation extraction, relation classification, and relational similarity prediction. In relation extraction, related word pairs in a corpus and the relevant relation are identified. Given a word pair, the relation classification task involves assigning a word pair to the correct relation from a pre-defined set. In the Open Information Extraction paradigm (Banko et al., 2007; Weikum and Theobald, 2010), also known as unsupervised relation extraction, the relations themselves are also learned from the text (e.g. in the form of text labels). On the other hand, relational similarity prediction involves assessing the degree to which a word pair (A, B) stands in the same relation as another pair (C, D), or to complete an analogy *A*:*B* :: *C*: –*?*–. Relation learning is an important and long-standing task in NLP and has been the focus of a number of shared tasks (Girju et al., 2007; Hendrickx et al., 2010; Jurgens et al., 2012).

Recently, attention has turned to using vector space models of words for relation classification and relational similarity prediction. Distributional word vectors have been used for detection of relations such as hypernymy (Geffet and Dagan, 2005; Kotlerman et al., 2010; Lenci and Benotto, 2012; Weeds et al., 2014; Rimell, 2014; Santus et al., 2014) and qualia structure (Yamada et al., 2009). An exciting development, and the inspiration for this paper, has been the demonstration that vector difference over word embeddings (Mikolov et al., 2013c) can be used to model word analogy tasks. This has given rise to a series of papers exploring the DIFFVEC idea in different contexts. The original analogy dataset has been used to evaluate predict-based language models by Mnih and Kavukcuoglu (2013) and also Zhila et al. (2013), who combine a neural language model with a pattern-based classifier. Kim and de Marneffe (2013) use word embeddings to derive representations of adjective scales, e.g. *hot—warm—cool—cold*. Fu et al. (2014) similarly use embeddings to predict hypernym relations, in this case clustering words by topic to show that hypernym DIFFVECs can be broken down into more fine-grained relations. Neural networks have also been developed for joint learning of lexical and relational similarity, making use of the WordNet relation hierarchy (Bordes et al., 2013; Socher et al., 2013; Xu et al., 2014; Yu and Dredze, 2014; Faruqui et al., 2015; Fried and Duh, 2015).

Another strand of work responding to the vector difference approach has analysed the structure of predict-based embedding models in order to help explain their success on the analogy and other tasks (Levy and Goldberg, 2014a; Levy and Goldberg, 2014b; Arora et al., 2015). However, there has been no systematic investigation of the range of relations for which the vector difference method is most effective, although there have been some smaller-scale investigations in this direction. Makrai et al. (2013) divide antonym pairs into semantic classes such as quality, time, gender, and distance, finding that for about two-thirds of antonym classes, DIFFVECs are significantly more correlated than random. Necşulescu et al. (2015) train a classifier on word pairs, using word embeddings to predict coordinates, hypernyms, and meronyms. Roller and Erk (2016) analyse the performance of vector concatenation and difference on the task of predicting lexical entailment and show that vector concatenation overwhelmingly learns to detect Hearst patterns (e.g., *including*, *such as*). Köper et al. (2015) undertake a systematic study of morphosyntactic and semantic relations on word embeddings produced with `word2vec` (“`w2v`” hereafter; see §3.1) for English and German. They test a variety of relations including word similarity, antonyms, synonyms, hypernyms, and meronyms, in a novel analogy task. Although the set of relations tested by

Köper et al. (2015) is somewhat more constrained than the set we use, there is a good deal of overlap. However, their evaluation is performed in the context of relational similarity, and they do not perform clustering or classification on the DiffVecs.

3 General Approach and Resources

We define the task of lexical relation learning to take a set of (ordered) word pairs $\{(w_i, w_j)\}$ and a set of binary lexical relations $R = \{r_k\}$, and map each word pair (w_i, w_j) as follows: (a) $(w_i, w_j) \mapsto r_k \in R$, i.e. the "closed-world" setting, where we assume that all word pairs can be uniquely classified according to a relation in R; or (b) $(w_i, w_j) \mapsto r_k \in R \cup \{\phi\}$ where ϕ signifies the fact that none of the relations in R apply to the word pair in question, i.e. the "open-world" setting.

Our starting point for lexical relation learning is the assumption that important information about various types of relations is implicitly embedded in the offset vectors. While a range of methods have been proposed for composing word vectors (Baroni et al., 2012; Weeds et al., 2014; Roller et al., 2014), in this research we focus exclusively on DiffVec (i.e. $\mathbf{w_2} - \mathbf{w_1}$). A second assumption is that there exist dimensions, or directions, in the embedding vector spaces responsible for a particular lexical relation. Such dimensions could be identified and exploited as part of a clustering or classification method, in the context of identifying relations between word pairs or classes of DiffVecs.

In order to test the generalisability of the DiffVec method, we require: (1) word embeddings, and (2) a set of lexical relations to evaluate against. As the focus of this paper is not the word embedding pre-training approaches so much as the utility of the DiffVecs for lexical relation learning, we take a selection of four pre-trained word embeddings with strong currency in the literature, as detailed in §3.1. We also include the state-of-the-art count-based approach of Levy et al. (2015a), to test the generalisability of DiffVecs to count-based word embeddings.

For the lexical relations, we want a range of relations that is representative of the types of relational learning tasks targeted in the literature, and where there is availability of annotated data. To this end, we construct a dataset from a variety of sources, focusing on lexical semantic relations (which are less well represented in the analogy dataset of Mikolov et al. (2013c)), but also including morphosyntactic and morphosemantic relations (see §3.2).

Name	Dimensions	Training data
w2v	300	100×10^9
GloVe	200	6×10^9
SENNA	100	37×10^6
HLBL	200	37×10^6
w2v$_{\text{wiki}}$	300	50×10^6
GloVe$_{\text{wiki}}$	300	50×10^6
SVD$_{\text{wiki}}$	300	50×10^6

Table 1: The pre-trained word embeddings used in our experiments, with the number of dimensions and size of the training data (in word tokens). The models trained on English Wikipedia ("wiki") are in the lower half of the table.

3.1 Word Embeddings

We consider four highly successful word embedding models in our experiments: w2v (Mikolov et al., 2013a; Mikolov et al., 2013b), GloVe (Pennington et al., 2014), SENNA (Collobert and Weston, 2008), and HLBL (Mnih and Hinton, 2009), as detailed below. We also include SVD (Levy et al., 2015a), a count-based model which factorises a positive PMI (PPMI) matrix. For consistency of comparison, we train SVD as well as a version of w2v and GloVe (which we call w2v$_{\text{wiki}}$ and GloVe$_{\text{wiki}}$, respectively) on the English Wikipedia corpus (comparable in size to the training data of SENNA and HLBL), and apply the preprocessing of Levy et al. (2015a). We additionally normalise the w2v$_{\text{wiki}}$ and SVD$_{\text{wiki}}$ vectors to unit length; GloVe$_{\text{wiki}}$ is natively normalised by column.[1]

w2v CBOW (Continuous Bag-Of-Words; Mikolov et al. (2013a)) predicts a word from its context using a model with the objective:

$$J = \frac{1}{T}\sum_{i=1}^{T} \log \frac{\exp\left(\mathbf{w}_i^\top \sum_{j\in[-c,+c],j\neq 0} \tilde{\mathbf{w}}_{i+j}\right)}{\sum_{k=1}^{V}\exp\left(\mathbf{w}_k^\top \sum_{j\in[-c,+c],j\neq 0} \tilde{\mathbf{w}}_{i+j}\right)}$$

where $\mathbf{w}_i$ and $\tilde{\mathbf{w}}_i$ are the vector representations for the ith word (as a focus or context word, respectively), V is the vocabulary size, T is the number of tokens in the corpus, and c is the context window size.[2] Google News data was used

[1] We ran a series of experiments on normalised and unnormalised w2v models, and found that normalisation tends to boost results over most of our relations (with the exception of LexSem$_{\text{Event}}$ and Noun$_{\text{Coll}}$). We leave a more detailed investigation of normalisation to future work.

[2] In a slight abuse of notation, the subscripts of $\mathbf{w}$ do double

Relation	Description	Pairs	Source	Example
LexSem$_{\text{Hyper}}$	hypernym	1173	SemEval'12 + BLESS	(*animal*, *dog*)
LexSem$_{\text{Mero}}$	meronym	2825	SemEval'12 + BLESS	(*airplane*, *cockpit*)
LexSem$_{\text{Attr}}$	characteristic quality, action	71	SemEval'12	(*cloud*, *rain*)
LexSem$_{\text{Cause}}$	cause, purpose, or goal	249	SemEval'12	(*cook*, *eat*)
LexSem$_{\text{Space}}$	location or time association	235	SemEval'12	(*aquarium*,*fish*)
LexSem$_{\text{Ref}}$	expression or representation	187	SemEval'12	(*song*, *emotion*)
LexSem$_{\text{Event}}$	object's action	3583	BLESS	(*zip*, *coat*)
Noun$_{\text{SP}}$	plural form of a noun	100	MSR	(*year*, *years*)
Verb$_{\text{3}}$	first to third person verb present-tense form	99	MSR	(*accept*, *accepts*)
Verb$_{\text{Past}}$	present-tense to past-tense verb form	100	MSR	(*know*, *knew*)
Verb$_{\text{3Past}}$	third person present-tense to past-tense verb form	100	MSR	(*creates*, *created*)
LVC	light verb construction	58	Tan et al. (2006b)	(*give*, *approval*)
VerbNoun	nominalisation of a verb	3303	WordNet	(*approve*, *approval*)
Prefix	prefixing with *re* morpheme	118	Wiktionary	(*vote*, *revote*)
Noun$_{\text{Coll}}$	collective noun	257	Web source	(*army*, *ants*)

Table 2: Description of the 15 lexical relations.

to train the model. We use the focus word vectors, $W = \{\mathbf{w}_k\}_{k=1}^{V}$, normalised such that each $\|\mathbf{w}_k\| = 1$.

The `GloVe` model (Pennington et al., 2014) is based on a similar bilinear formulation, framed as a low-rank decomposition of the matrix of corpus co-occurrence frequencies:

$$J = \frac{1}{2}\sum_{i,j=1}^{V} f(P_{ij})(\mathbf{w}_i^{\top}\tilde{\mathbf{w}}_j - \log P_{ij})^2 \,,$$

where w_i is a vector for the left context, w_j is a vector for the right context, P_{ij} is the relative frequency of word j in the context of word i, and f is a heuristic weighting function to balance the influence of high versus low term frequencies. The model was trained on English Wikipedia and the English Gigaword corpus version 5.

The `SVD` model (Levy et al., 2015a) uses positive pointwise mutual information (PMI) matrix defined as:

$$\text{PPMI}(w, c) = \max(\log \frac{\hat{P}(w, c)}{\hat{P}(w)\hat{P}(c)}, 0) \,,$$

where $\hat{P}(w, c)$ is the joint probability of word w and context c, and $\hat{P}(w)$ and $\hat{P}(c)$ are their marginal probabilities. The matrix is factorised by singular value decomposition.

`HLBL` (Mnih and Hinton, 2009) is a log-bilinear formulation of an n-gram language model, which predicts the ith word based on context words $(i - n, \ldots, i-2, i-1)$. This leads to the following training objective:

$$J = \frac{1}{T}\sum_{i=1}^{T} \frac{\exp(\tilde{\mathbf{w}}_i^{\top}\mathbf{w}_i + b_i)}{\sum_{k=1}^{V}\exp(\tilde{\mathbf{w}}_i^{\top}\mathbf{w}_k + b_k)} \,,$$

where $\tilde{\mathbf{w}}_i = \sum_{j=1}^{n-1} C_j \mathbf{w}_{i-j}$ is the context embedding, C_j is a scaling matrix, and b_* is a bias term.

The final model, `SENNA` (Collobert and Weston, 2008), was initially proposed for multi-task training of several language processing tasks, from language modelling through to semantic role labelling. Here we focus on the statistical language modelling component, which has a pairwise ranking objective to maximise the relative score of each word in its local context:

$$J = \frac{1}{T}\sum_{i=1}^{T}\sum_{k=1}^{V} \max\big[0, 1 - f(\mathbf{w}_{i-c}, \ldots, \mathbf{w}_{i-1}, \mathbf{w}_i) \\ + f(\mathbf{w}_{i-c}, \ldots, \mathbf{w}_{i-1}, \mathbf{w}_k)\big] \,,$$

where the last $c - 1$ words are used as context, and $f(x)$ is a non-linear function of the input, defined as a multi-layer perceptron.

For `HLBL` and `SENNA`, we use the pre-trained embeddings from Turian et al. (2010), trained on the Reuters English newswire corpus. In both cases, the embeddings were scaled by the global standard deviation over the word-embedding matrix, $W_{\text{scaled}} = 0.1 \times \frac{W}{\sigma(W)}$.

For `w2v`$_{\text{wiki}}$, `GloVe`$_{\text{wiki}}$ and `SVD`$_{\text{wiki}}$ we used English Wikipedia. We followed the same preprocessing procedure described in Levy et al. (2015a),[3] i.e., lower-cased all words and removed non-textual elements. During the training phase, for each model we set a word frequency threshold of 5. For the `SVD` model, we followed the recommendations of Levy et al. (2015a) in setting the context window size to 2, negative sampling parameter to 1, eigenvalue weighting to 0.5, and context distribution smoothing to 0.75; other parameters were assigned

duty, denoting either the embedding for the ith token, $\mathbf{w}_i$, or kth word type, $\mathbf{w}_k$.

[3] Although the `w2v` model trained without preprocessing performed marginally better, we used preprocessing throughout for consistency.

their default values. For the other models we used the following parameter values: for w2v, context window = 8, negative samples = 25, hs = 0, sample = 1e-4, and iterations = 15; and for GloVe, context window = 15, x_max = 10, and iterations = 15.

3.2 Lexical Relations

In order to evaluate the applicability of the DiffVec approach to relations of different types, we assembled a set of lexical relations in three broad categories: lexical semantic relations, morphosyntactic paradigm relations, and morphosemantic relations. We constrained the relations to be binary and to have fixed directionality.[4] Consequently we excluded symmetric lexical relations such as synonymy. We additionally constrained the dataset to the words occurring in all embedding sets. There is some overlap between our relations and those included in the analogy task of Mikolov et al. (2013c), but we include a much wider range of lexical semantic relations, especially those standardly evaluated in the relation classification literature. We manually filtered the data to remove duplicates (e.g., as part of merging the two sources of LexSem$_{\text{Hyper}}$ intances), and normalise directionality.

The final dataset consists of 12,458 triples $\langle \text{relation}, \text{word}_1, \text{word}_2 \rangle$, comprising 15 relation types, extracted from SemEval'12 (Jurgens et al., 2012), BLESS (Baroni and Lenci, 2011), the MSR analogy dataset (Mikolov et al., 2013c), the light verb dataset of Tan et al. (2006a), Princeton WordNet (Fellbaum, 1998), Wiktionary,[5] and a web lexicon of collective nouns,[6] as listed in Table 2.[7]

4 Clustering

Assuming DiffVecs are capable of capturing all lexical relations equally, we would expect clustering to be able to identify sets of word pairs with high relational similarity, or equivalently clusters of similar offset vectors. Under the additional assumption that a given word pair corresponds to a unique lexical relation (in line with our definition of the lexical relation learning task in §3), a hard clustering approach is appropriate. In order to test these assumptions, we cluster our 15-relation closed-world dataset in the first instance, and evaluate against the lexical resources in §3.2.

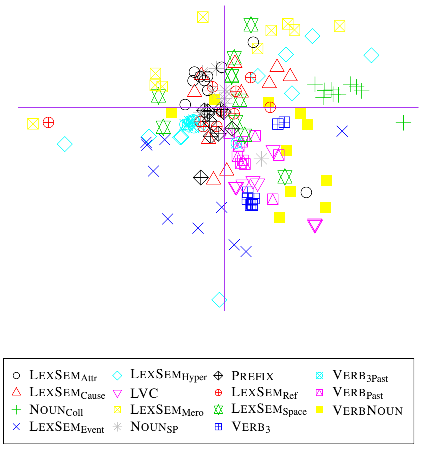

Figure 1: t-SNE projection (Van der Maaten and Hinton, 2008) of DiffVecs for 10 sample word pairs of each relation type, based on w2v. The intersection of the two axes identify the projection of the zero vector. Best viewed in colour.

As further motivation, we projected the DiffVec space for a small number of samples of each class using t-SNE (Van der Maaten and Hinton, 2008), and found that many of the morphosyntactic relations (Verb$_3$, Verb$_{\text{Past}}$, Verb$_{\text{3Past}}$, Noun$_{\text{SP}}$) form tight clusters (Figure 1).

We cluster the DiffVecs between all word pairs in our dataset using spectral clustering (Von Luxburg, 2007). Spectral clustering has two hyperparameters: the number of clusters, and the pairwise similarity measure for comparing DiffVecs. We tune the hyperparameters over development data, in the form of 15% of the data obtained by random sampling, selecting the configuration that maximises the V-Measure (Rosenberg and Hirschberg, 2007). Figure 2 presents V-Measure values over the test data for each of the four word embedding models. We show results for different numbers of clusters, from $N = 10$ in steps of 10, up to $N = 80$ (beyond which the clustering quality diminishes).[8] Observe that w2v achieves the best results, with a V-Measure value of around 0.36,[9] which is relatively constant over varying numbers of clusters. GloVe and SVD mirror this result, but are consistently below w2v at a V-Measure of around 0.31. HLBL and SENNA performed very

[4] Word similarity is not included; it is not easily captured by DiffVec since there is no homogeneous "content" to the lexical relation which could be captured by the direction and magnitude of a difference vector (other than that it should be small).

[5] http://en.wiktionary.org

[6] http://www.rinkworks.com/words/collective.shtml

[7] The dataset is available at http://github.com/ivri/DiffVec

[8] Although 80 clusters $\gg$ our 15 relation types, the SemEval'12 classes each contain numerous subclasses, so the larger number may be more realistic.

[9] V-Measure returns a value in the range $[0, 1]$, with 1 indicating perfect homogeneity and completeness.

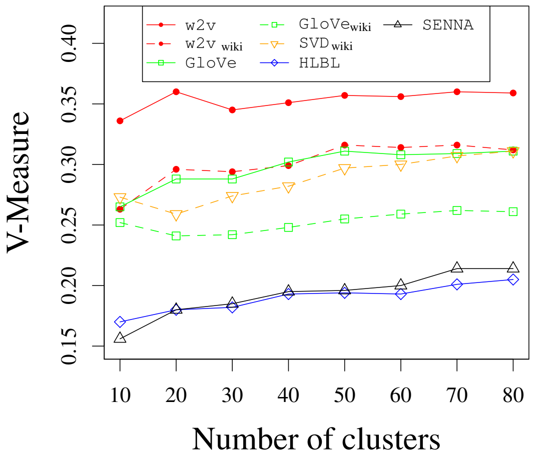

Figure 2: Spectral clustering results, comparing cluster quality (V-Measure) and the number of clusters. DIFFVECs are clustered and compared to the known relation types. Each line shows a different source of word embeddings.

	w2v	GloVe	HLBL	SENNA
LEXSEM$_{\text{Attr}}$	**0.49**	0.54	0.62	0.63
LEXSEM$_{\text{Cause}}$	**0.47**	0.53	0.56	0.57
LEXSEM$_{\text{Space}}$	**0.49**	0.55	0.54	0.58
LEXSEM$_{\text{Ref}}$	**0.44**	0.50	0.54	0.56
LEXSEM$_{\text{Hyper}}$	0.44	0.50	**0.43**	0.45
LEXSEM$_{\text{Event}}$	**0.46**	0.47	0.47	0.48
LEXSEM$_{\text{Mero}}$	**0.40**	0.42	0.42	0.43
NOUN$_{\text{SP}}$	**0.07**	0.14	0.22	0.29
VERB$_{3}$	**0.05**	0.06	0.49	0.44
VERB$_{\text{Past}}$	**0.09**	0.14	0.38	0.35
VERB$_{3\text{Past}}$	0.07	**0.05**	0.49	0.52
LVC	**0.28**	0.55	0.32	0.30
VERBNOUN	**0.31**	0.33	0.35	0.36
PREFIX	0.32	**0.30**	0.55	0.58
NOUN$_{\text{Coll}}$	**0.21**	0.27	0.46	0.44

Table 3: The entropy for each lexical relation over the clustering output for each set of pre-trained word embeddings.

similarly, at a substantially lower V-Measure than w2v or GloVe, closer to 0.21. As a crude calibration for these results, over the related clustering task of word sense induction, the best-performing systems in SemEval-2010 Task 4 (Manandhar et al., 2010) achieved a V-Measure of under 0.2.

The lower V-measure for w2v$_{\text{wiki}}$ and GloVe$_{\text{wiki}}$ (as compared to w2v and GloVe, respectively) indicates that the volume of training data plays a role in the clustering results. However, both methods still perform well above SENNA and HLBL, and w2v has a clear empirical advantage over GloVe. We note that SVD$_{\text{wiki}}$ performs almost as well as w2v$_{\text{wiki}}$, consistent with the results of Levy et al. (2015a).

We additionally calculated the entropy for each lexical relation, based on the distribution of instances belonging to a given relation across the different clusters (and simple MLE). For each embedding method, we present the entropy for the cluster size where V-measure was maximised over the development data. Since the samples are distributed nonuniformly, we normalise entropy results for each method by $\log(n)$ where n is the number of samples in a particular relation. The results are in Table 3, with the lowest entropy (purest clustering) for each relation indicated in bold.

Looking across the different lexical relation types, the morphosyntactic paradigm relations (NOUN$_{\text{SP}}$ and the three VERB relations) are by far the easiest to capture. The lexical semantic relations, on the other hand, are the hardest to capture for all embeddings.

Considering w2v embeddings, for VERB$_3$ there was a single cluster consisting of around 90% of VERB$_3$ word pairs. Most errors resulted from POS ambiguity, leading to confusion with VERB-NOUN in particular. Example VERB$_3$ pairs incorrectly clustered are: $(study, studies)$, $(run, runs)$, and $(like, likes)$. This polysemy results in the distance represented in the DIFFVEC for such pairs being above average for VERB$_3$, and consequently clustered with other cross-POS relations.

For VERB$_{\text{Past}}$, a single relatively pure cluster was generated, with minor contamination due to pairs such as $(hurt, saw)$, $(utensil, saw)$, and $(wipe, saw)$. Here, the noun *saw* is ambiguous with a high-frequency past-tense verb; *hurt* and *wipe* also have ambigous POS.

A related phenomenon was observed for NOUN$_{\text{Coll}}$, where the instances were assigned to a large mixed cluster containing word pairs where the second word referred to an animal, reflecting the fact that most of the collective nouns in our dataset relate to animals, e.g. $(stand, horse)$, $(ambush, tigers)$, $(antibiotics, bacteria)$. This is interesting from a DIFFVEC point of view, since it shows that the lexical semantics of one word in the pair can overwhelm the semantic content of the DIFFVEC (something that we return to investigate in §5.4). LEXSEM$_{\text{Mero}}$ was also split into multiple clusters along topical lines, with separate clusters for weapons, dwellings, vehicles, etc.

Given the encouraging results from our clustering experiment, we next evaluate DIFFVECs in a supervised relation classification setting.

5 Classification

A natural question is whether we can accurately characterise lexical relations through supervised learning over the DIFFVECs. For these experiments we use the w2v, w2v$_{\text{wiki}}$, and SVD$_{\text{wiki}}$ embeddings exclusively (based on their superior performance in the clustering experiment), and a subset of the relations which is both representative of the breadth of the full relation set, and for which we have sufficient data for supervised training and evaluation, namely: NOUN$_{\text{Coll}}$, LEXSEM$_{\text{Event}}$, LEXSEM$_{\text{Hyper}}$, LEXSEM$_{\text{Mero}}$, NOUN$_{\text{SP}}$, PREFIX, VERB$_3$, VERB$_{3\text{Past}}$, and VERB$_{\text{Past}}$ (see Table 2).

We consider two applications: (1) a CLOSED-WORLD setting similar to the unsupervised evaluation, in which the classifier only encounters word pairs which correspond to one of the nine relations; and (2) a more challenging OPEN-WORLD setting where random word pairs — which may or may not correspond to one of our relations — are included in the evaluation. For both settings, we further investigate whether there is a lexical memorisation effect for a broad range of relation types of the sort identified by Weeds et al. (2014) and Levy et al. (2015b) for hypernyms, by experimenting with disjoint training and test vocabulary.

5.1 CLOSED-WORLD Classification

For the CLOSED-WORLD setting, we train and test a multiclass classifier on datasets comprising $\langle \text{DIFFVEC}, r\rangle$ pairs, where r is one of our nine relation types, and DIFFVEC is based on one of w2v, w2v$_{\text{wiki}}$ and SVD. As a baseline, we cluster the data as described in §4, running the clusterer several times over the 9-relation data to select the optimal V-Measure value based on the development data, resulting in 50 clusters. We label each cluster with the majority class based on the training instances, and evaluate the resultant labelling for the test instances.

We use an SVM with a linear kernel, and report results from 10-fold cross-validation in Table 4.

The SVM achieves a higher F-score than the baseline on almost every relation, particularly on LEXSEM$_{\text{Hyper}}$, and the lower-frequency NOUN$_{\text{SP}}$, NOUN$_{\text{Coll}}$, and PREFIX. Most of the relations — even the most difficult ones from our clustering experiment — are classified with very high F-score. That is, with a simple linear transformation of the embedding dimensions, we are able to achieve near-perfect results. The PREFIX relation achieved markedly lower recall, resulting in a lower F-score, due to large differences in the predominant usages associated with the respective words (e.g., $(\textit{union}, \textit{reunion})$, where the vector for *union* is heavily biased by contexts associated with trade unions, but *reunion* is heavily biased by contexts relating to social get-togethers; and $(\textit{entry}, \textit{reentry})$, where *entry* is associated with competitions and entrance to schools, while *reentry* is associated with space travel). Somewhat surprisingly, given the small dimensionality of the input (vectors of size 300 for all three methods), we found that the linear SVM slightly outperformed a non-linear SVM using an RBF kernel. We observe no real difference between w2v$_{\text{wiki}}$ and SVD$_{\text{wiki}}$, supporting the hypothesis of Levy et al. (2015a) that under appropriate parameter settings, count-based methods achieve high results. The impact of the training data volume for pre-training of the embeddings is also less pronounced than in the case of our clustering experiment.

Relation	Baseline	w2v	w2v$_{\text{wiki}}$	SVD$_{\text{wiki}}$
LEXSEM$_{\text{Hyper}}$	0.60	0.93	0.91	0.91
LEXSEM$_{\text{Mero}}$	0.90	0.97	0.96	0.96
LEXSEM$_{\text{Event}}$	0.87	0.98	0.97	0.97
NOUN$_{\text{SP}}$	0.00	0.83	0.78	0.74
VERB$_3$	0.99	0.98	0.96	0.97
VERB$_{\text{Past}}$	0.78	0.98	0.98	0.95
VERB$_{3\text{Past}}$	0.99	0.98	0.98	0.96
PREFIX	0.00	0.82	0.34	0.60
NOUN$_{\text{Coll}}$	0.19	0.95	0.91	0.92
Micro-average	0.84	0.97	0.95	0.95

Table 4: F-scores ($\mathcal{F}$) for CLOSED-WORLD classification, for a baseline method based on clustering + majority-class labelling, a multiclass linear SVM trained on w2v, w2v$_{\text{wiki}}$ and SVD$_{\text{wiki}}$ DIFFVEC inputs.

5.2 OPEN-WORLD Classification

We now turn to a more challenging evaluation setting: a test set including word pairs drawn at random. This setting aims to illustrate whether a DIFFVEC-based classifier is capable of differentiating related word pairs from noise, and can be applied to open data to learn new related word pairs.[10]

For these experiments, we train a binary classifier for each relation type, using $\frac{2}{3}$ of our relation data for training and $\frac{1}{3}$ for testing. The test data is augmented with an equal quantity of random pairs, generated as follows:

(1) sample a seed lexicon by drawing words proportional to their frequency in Wikipedia;[11]

[10] Hereafter we provide results for w2v only, as we found that SVD achieved similar results.

[11] Filtered to consist of words for which we have embed-

Relation	Orig			+neg		
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
LexSem$_{\text{Hyper}}$	0.95	**0.92**	**0.93**	**0.99**	0.84	0.91
LexSem$_{\text{Mero}}$	0.13	**0.96**	0.24	**0.95**	0.84	**0.89**
LexSem$_{\text{Event}}$	0.44	**0.98**	0.61	**0.93**	0.90	**0.91**
Noun$_{\text{SP}}$	0.95	**0.68**	0.8	**1.00**	**0.68**	**0.81**
Verb$_{3}$	0.75	**1.00**	0.86	**0.93**	0.93	**0.93**
Verb$_{\text{Past}}$	0.94	**0.86**	**0.90**	**0.97**	0.84	**0.90**
Verb$_{\text{3Past}}$	0.76	**0.95**	0.84	**0.87**	0.93	**0.90**
Prefix	**1.00**	**0.29**	**0.44**	**1.00**	0.13	0.23
Noun$_{\text{Coll}}$	0.43	**0.74**	0.55	**0.97**	0.41	**0.57**

Table 5: Precision ($\mathcal{P}$) and recall ($\mathcal{R}$) for Open-World classification, using the binary classifier without ("Orig") and with ("+neg") negative samples .

(2) take the Cartesian product over pairs of words from the seed lexicon;

(3) sample word pairs uniformly from this set.

This procedure generates word pairs that are representative of the frequency profile of our corpus.

We train 9 binary RBF-kernel SVM classifiers on the training partition, and evaluate on our randomly augmented test set. Fully annotating our random word pairs is prohibitively expensive, so instead, we manually annotated only the word pairs which were positively classified by one of our models. The results of our experiments are presented in the left half of Table 5, in which we report on results over the combination of the original test data from §5.1 and the random word pairs, noting that recall ($\mathcal{R}$) for Open-World takes the form of relative recall (Pantel et al., 2004) over the positively-classified word pairs. The results are much lower than for the closed-word setting (Table 4), most notably in terms of precision ($\mathcal{P}$). For instance, the random pairs (*have*, *works*), (*turn*, *took*), and (*works*, *started*) were incorrectly classified as Verb$_{3}$, Verb$_{\text{Past}}$ and Verb$_{\text{3Past}}$, respectively. That is, the model captures syntax, but lacks the ability to capture lexical paradigms, and tends to overgenerate.

5.3 Open-World Training with Negative Sampling

To address the problem of incorrectly classifying random word pairs as valid relations, we retrain the classifier on a dataset comprising both valid and automatically-generated negative distractor samples. The basic intuition behind this approach is to construct samples which will force the model to learn decision boundaries that more tightly capture the true scope of a given relation. To this end, we automatically generated two types of negative distractors:

opposite pairs: generated by switching the order of word pairs, $\mathit{Oppos}_{w1,w2} = \mathbf{word_1} - \mathbf{word_2}$. This ensures the classifier adequately captures the asymmetry in the relations.

shuffled pairs: generated by replacing w_2 with a random word w_2' from the same relation, $\mathit{Shuff}_{w1,w2} = \mathbf{word_2'} - \mathbf{word_1}$. This is targeted at relations that take specific word classes in particular positions, e.g., (`VB`, `VBD`) word pairs, so that the model learns to encode the relation rather than simply learning the properties of the word classes.

Both types of distractors are added to the training set, such that there are equal numbers of valid relations, opposite pairs and shuffled pairs.

After training our classifier, we evaluate its predictions in the same way as in §5.2, using the same test set combining related and random word pairs.[12] The results are shown in the right half of Table 5 (as "+neg"). Observe that the precision is much higher and recall somewhat lower compared to the classifier trained with only positive samples. This follows from the adversarial training scenario: using negative distractors results in a more conservative classifier, that correctly classifies the vast majority of the random word pairs as not corresponding to a given relation, resulting in higher precision at the expense of a small drop in recall. Overall this leads to higher F-scores, as shown in Figure 3, other than for hypernyms (LexSem$_{\text{Hyper}}$) and prefixes (Prefix). For example, the standard classifier for Noun$_{\text{Coll}}$ learned to match word pairs including an animal name (e.g., (*plague*, *rats*)), while training with negative samples resulted in much more conservative predictions and consequently much lower recall. The classifier was able to capture (*herd*, *horses*) but not (*run*, *salmon*), (*party*, *jays*) or (*singular*, *boar*) as instances of Noun$_{\text{Coll}}$, possibly because of polysemy. The most striking difference in performance was for LexSem$_{\text{Mero}}$, where the standard classifier generated many false positive noun pairs (e.g. (*series*, *radio*)), but the false positive rate was considerably reduced with negative sampling.

5.4 Lexical Memorisation

Weeds et al. (2014) and Levy et al. (2015b) recently showed that supervised methods using DiffVecs achieve artificially high results as a result of "lexical memorisation" over frequent words asso-

dings.

[12] But noting that relative recall for the random word pairs is based on the pool of positive predictions from both models.

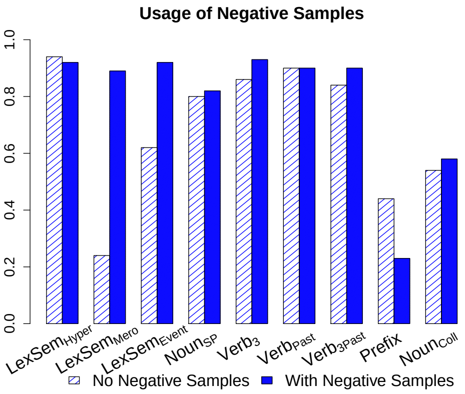

Figure 3: F-score for OPEN-WORLD classification, comparing models trained with and without negative samples.

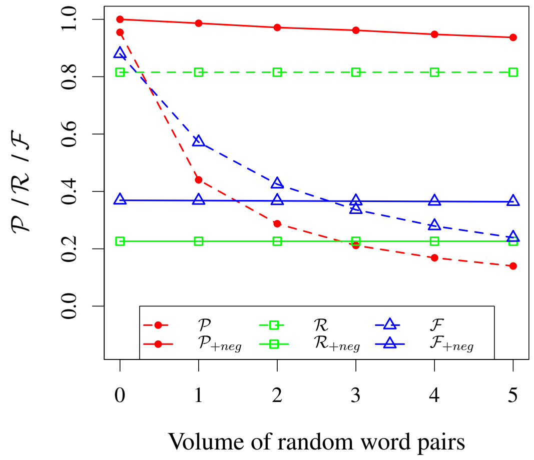

Figure 4: Evaluation of the OPEN-WORLD model when trained on split vocabulary, for varying numbers of random word pairs in the test dataset (expressed as a multiplier relative to the number of CLOSED-WORLD test instances).

ciated with the hypernym relation. For example, $(animal, cat)$, $(animal, dog)$, and $(animal, pig)$ all share the superclass *animal*, and the model thus learns to classify as positive any word pair with *animal* as the first word.

To address this effect, we follow Levy et al. (2015b) in splitting our vocabulary into training and test partitions, to ensure there is no overlap between training and test vocabulary. We then train classifiers with and without negative sampling (§5.3), incrementally adding the random word pairs from §5.2 to the test data (from no random word pairs to five times the original size of the test data) to investigate the interaction of negative sampling with greater diversity in the test set when there is a split vocabulary. The results are shown in Figure 4.

Observe that the precision for the standard classifier decreases rapidly as more random word pairs are added to the test data. In comparison, the precision when negative sampling is used shows only a small drop-off, indicating that negative sampling is effective at maintaining precision in an OPEN-WORLD setting even when the training and test vocabulary are disjoint. This benefit comes at the expense of recall, which is much lower when negative sampling is used (note that recall stays relatively constant as random word pairs are added, as the vast majority of them do not correspond to any relation). At the maximum level of random word pairs in the test data, the F-score for the negative sampling classifier is higher than for the standard classifier.

6 Conclusions

This paper is the first to test the generalisability of the vector difference approach across a broad range of lexical relations (in raw number and also variety). Using clustering we showed that many types of morphosyntactic and morphosemantic differences are captured by DIFFVECs, but that lexical semantic relations are captured less well, a finding which is consistent with previous work (Köper et al., 2015). In contrast, classification over the DIFFVECs works extremely well in a closed-world setting, showing that dimensions of DIFFVECs encode lexical relations. Classification performs less well over open data, although with the introduction of automatically-generated negative samples, the results improve substantially. Negative sampling also improves classification when the training and test vocabulary are split to minimise lexical memorisation. Overall, we conclude that the DIFFVEC approach has impressive utility over a broad range of lexical relations, especially under supervised classification.

Acknowledgments

LR was supported by EPSRC grant EP/I037512/1 and ERC Starting Grant DisCoTex (306920). TC and TB were supported by the Australian Research Council.

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2015. Random walks on context spaces: Towards an explanation of the mysteries of semantic word embeddings. arXiv:1502.03520 [cs.LG].

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction for the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-2007)*, pages 2670–2676, Hyderabad, India.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, GEMS '11, pages 1–10, Edinburgh, Scotland.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the EACL (EACL 2012)*, pages 23–32, Avignon, France.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 25 (NIPS-13)*, pages 2787–2795.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*, pages 160–167, Helsinki, Finland.

Manaal Faruqui, Jesse Dodge, Sujay Jauhar, Chris Dyer, Ed Hovy, and Noah Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics — Human Language Technologies (NAACL HLT 2015)*, pages 1351–1356, Denver, USA.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, USA.

Daniel Fried and Kevin Duh. 2015. Incorporating both distributional and relational semantics in word representations. In *Proceedings of the Third International Conference on Learning Representations (ICLR 2015)*, San Diego, USA.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 1199–1209, Baltimore, USA.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 107–114, Ann Arbor, USA.

Roxana Girju, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2007. SemEval-2007 Task 4: Classification of semantic relations between nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluation (SemEval 2007)*, pages 13–18, Prague, Czech Republic.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 Task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval 2010)*, pages 33–38, Uppsala, Sweden.

David Jurgens, Saif Mohammad, Peter Turney, and Keith Holyoak. 2012. SemEval-2012 Task 2: Measuring degrees of relational similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012)*, pages 356–364, Montréal, Canada.

Joo-Kyung Kim and Marie-Catherine de Marneffe. 2013. Deriving adjectival scales from continuous space word representations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1625–1630, Seattle, USA.

Maximilian Köper, Christian Scheible, and Sabine Schulte im Walde. 2015. Multilingual reliability and "semantic" structure of continuous word spaces. In *Proceedings of the Eleventh International Workshop on Computational Semantics (IWCS-11)*, pages 40–45, London, UK.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16:359–389.

Alessandro Lenci and Giulia Benotto. 2012. Identifying hypernyms in distributional semantic spaces. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM 2012)*, pages 75–79, Montréal, Canada.

Omer Levy and Yoav Goldberg. 2014a. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the 18th Conference on Natural Language Learning (CoNLL-2014)*, pages 171–180, Baltimore, USA.

Omer Levy and Yoav Goldberg. 2014b. Neural word embeddings as implicit matrix factorization. In *Advances in Neural Information Processing Systems 26 (NIPS-14)*.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015a. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Omer Levy, Steffen Remus, Chris Biemann, Ido Dagan, and Israel Ramat-Gan. 2015b. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics — Human Language Technologies (NAACL HLT 2015)*, pages 970–976, Denver, USA.

Márton Makrai, Dávid Nemeskey, and András Kornai. 2013. Applicative structure in vector space models. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 59–63, Sofia, Bulgaria.

Suresh Manandhar, Ioannis Klapaftis, Dmitriy Dligach, and Sameer Pradhan. 2010. SemEval-2010 Task 14: Word sense induction & disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 63–68, Uppsala, Sweden.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of the Workshop of the First International Conference on Learning Representations (ICLR 2013)*, Scottsdale, USA.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 25 (NIPS-13)*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2013)*, pages 746–751, Atlanta, USA.

Andriy Mnih and Geoffrey E Hinton. 2009. A scalable hierarchical distributed language model. In *Advances in Neural Information Processing Systems 21 (NIPS-09)*, pages 1081–1088.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems 25 (NIPS-13)*.

Silvia Necşulescu, Sara Mendes, David Jurgens, Núria Bel, and Roberto Navigli. 2015. Reading between the lines: Overcoming data sparsity for accurate classification of lexical relationships. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics (*SEM 2015)*, pages 182–192, Denver, USA.

Patrick Pantel, Deepak Ravichandran, and Eduard Hovy. 2004. Towards terascale semantic acquisition. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 771–777, Geneva, Switzerland.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543, Doha, Qatar.

Laura Rimell. 2014. Distributional lexical entailment by topic coherence. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pages 511–519, Gothenburg, Sweden.

Stephen Roller and Katrin Erk. 2016. Relations such as hypernymy: Identifying and exploiting Hearst patterns in distributional vectors for lexical entailment. *arXiv preprint arXiv:1605.05433*.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 1025–1036, Dublin, Ireland.

Andrew Rosenberg and Julia Hirschberg. 2007. V-Measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning 2007 (EMNLP-CoNLL 2007)*, pages 410–420, Prague, Czech Republic.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, pages 38–42, Gothenburg, Sweden.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems 25 (NIPS-13)*.

Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. 2006a. *Introduction to Data Mining*. Addison Wesley.

Yee Fan Tan, Min-Yen Kan, and Hang Cui. 2006b. Extending corpus-based identification of light verb constructions using a supervised learning framework. In *Proceedings of the EACL 2006 Workshop on Multi-word-expressions in a Multilingual Context*, pages 49–56, Trento, Italy.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the ACL (ACL 2010)*, pages 384–394, Uppsala, Sweden.

Peter D. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85.

Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, pages 2249–2259, Dublin, Ireland.

Gerhard Weikum and Martin Theobald. 2010. From information to knowledge: harvesting entities and relationships from web sources. In *Proceedings of the Twenty Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 65–76, Indianapolis, USA.

Chang Xu, Yanlong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. 2014. RC-NET: A general framework for incorporating knowledge into word representations. In *Proceedings of the 23rd ACM Conference on Information and Knowledge Management (CIKM 2014)*, pages 1219–1228, Shanghai, China.

Ichiro Yamada, Kentaro Torisawa, Jun'ichi Kazama, Kow Kuroda, Masaki Murata, Stijn De Saeger, Francis Bond, and Asuka Sumida. 2009. Hypernym discovery based on distributional similarity and hierarchical structures. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, pages 929–937, Singapore.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 545–550, Baltimore, USA.

A. Zhila, W.T. Yih, C. Meek, G. Zweig, and T. Mikolov. 2013. Combining heterogeneous models for measuring relational similarity. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2013)*.