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 **Tim 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 on word embeddings has shown that simple vector subtraction over pre-trained 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"). Accurate relation classification, relational similarity prediction, and wide-coverage and adaptable relation discovery can contribute to numerous 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 contextual vector space representations, particularly neural language embeddings, which are dense, low-dimensional vectors obtained from a neural network trained to predict word contexts. The skip-gram model of Mikolov et al. (2013a) and other neural language models have been shown to perform well on an analogy completion task (Mikolov et al., 2013c; Mikolov et al., 2013b), in the space of *relational similarity* prediction (Turney, 2006). Linear operations on word vectors appear to capture the lexical relation governing the analogy. A well-known example involves predicting the vector $\mathbf{queen}$ from the vector combination $\mathbf{king} - \mathbf{man} + \mathbf{woman}$, which appears to capture a gender relation. The results also extend to semantic relations such as CAPITAL-OF-COUNTRY ($\mathbf{paris} - \mathbf{france} + \mathbf{poland} \approx \mathbf{warsaw}$) and morphosyntactic relations such as PLURALISATION ($\mathbf{cars} - \mathbf{car} + \mathbf{apple} \approx \mathbf{apples}$). This is particularly remarkable because the model is not trained for this task, so the relational structure of the vector space appears to be an emergent property of the model.

The key operation in these models is *vector difference*, or *vector offset*. For example, it is the $\mathbf{paris} - \mathbf{france}$ vector that 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 semantic relation. We would then expect the vector $\mathbf{helsinki} - \mathbf{finland}$ to be quite similar, in a quantifiable way, to $\mathbf{paris} - \mathbf{france}$.

However, the now-standard analogy task is only a first step in probing the semantics and morphosyntactics of DIFFVECs. First, the analogy task does not provide coverage of many well-known lexical relation types from the linguistics and cognitive science literature. Second, because the task requires a one-best answer, it may fail to identify meaningful patterns present in the data. Third, it is focused on recall rather than precision, leaving open the question of whether all DIFFVECs encode meaningful relations. There may also be more fine-grained structure in the DIFFVECs: Fu et al. (2014) found that vector offsets representing the hypernym relation could be grouped into semantic sub-clusters, as the difference between **carpenter** and **laborer**, e.g., was quite distinct from the one between **goldfish** and **fish**.

In this paper we investigate how well DIFFVECs calculated over different word embeddings capture lexical relations from a variety of linguistic resources. We systematically study the expressivity of vector difference in distributed spaces in two ways. First, we cluster the DIFFVECs to test whether the clusters map onto true lexical relations. We explore a parameter space consisting of the number of clusters and two distance measures, and find that syntactic relations are captured better than semantic relations.

Second, we perform classification over the DIFFVECs and obtain surprisingly 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 and attempt to classify 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 engineering negative instances. We find that this improves the model performance substantially.

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. NLP tasks related to lexical relation learning include relation extraction and discovery, relation classification, and relational similarity prediction. In relation extraction, word pairs standing in a given relation are mined from a corpus. The relations may be pre-defined or, in the Open Information Extraction paradigm (Banko et al., 2007; Weikum and Theobald, 2010), the relations themselves are also learned from the text (e.g. in the form of text labels). In relation classification, the task is to assign a word pair to the correct relation, from a pre-defined set of relations. 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).

Relation extraction and discovery has involved generic semantic relations such as IS-A and WHOLE-PART, but also corpus-specific relations such as CEO-OF-COMPANY (Pantel and Pennacchiotti, 2006). Some datasets are task-specific, for example paraphrasing the relation holding between nouns in noun-noun compounds (Girju et al., 2007), or analogy questions from the American SAT exam for relational similarity (Turney et al., 2003).

Historically, approaches to relation learning have generally been supervised or semi-supervised. Relation extraction has used pattern-based approaches such as *A such as B*, either explicitly (Hearst, 1992; Kozareva et al., 2008; McIntosh et al., 2011) or implicitly (Snow et al., 2005), although not all relations are equally amenable to this style of approach (Yamada and Baldwin, 2004). Relation classification involves supervised classifiers (Chklovski and Pantel, 2004; Snow et al., 2005; Davidov and Rappoport, 2008). Relational similarity prediction has also mostly used classification based on lexico-syntactic patterns linking word pairs in text (Ó Séaghdha and Copestake, 2009; Jurgens et al., 2012; Turney, 2013), or generalised from manually crafted resources such as Princeton WordNet (Fellbaum, 1998) using techniques such as Latent Semantic Analysis (Turney, 2006; Chang et al., 2013).

Recently, attention has turned to using vector space models of words for relation classification and relational similarity. Distributional word vectors, while mostly applied to measuring semantic similarity and relatedness (Bullinaria and Levy, 2007), have also 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 neural 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 neural 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, but instead of using a single DIFFVEC, they cluster words by topic and show that the hypernym DIFFVEC 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 neural 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) divided antonym pairs into semantic classes such as quality, time, gender, and distance, and tested whether the DIFFVECs internal to each antonym class were significantly more correlated than random. They found that for about two-thirds of the antonym classes, the DIFFVECs were significantly correlated. Necşulescu et al. (2015) trained a classifier on word pairs using word embeddings in order to predict coordinates, hypernyms, and meronyms. Köper et al. (2015) undertook a systematic study of morphosyntactic and semantic relations on word embeddings produced with `word2vec` ("`w2v`" hereafter; see §3.1) for English and German. They tested 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 was performed in the context of relational similarity, and they did not perform clustering or classification on the DIFFVECs.

3 General Approach and Resources

For our purposes, 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 the vectors of the component words (Baroni et al., 2012; Weeds et al., 2014; Roller et al., 2014), in this research we consider solely DIFFVEC (i.e. $\mathbf{w_2} - \mathbf{w_1}$) and hypothesise that these DIFFVECs should capture a wide spectrum of possible lexical contrasts. 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.

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

Name	Dimensions	Training data
`w2v`	300	100×10^9
`GloVe`	200	6×10^9
`SENNA`	100	37×10^6
`HLBL`	200	37×10^6

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).

well represented in the analogy dataset of Mikolov et al. (2013c)), but also including morphosyntactic and morphosemantic relations (see §3.2).

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 et al., 2011), and `HLBL` (Mnih and Hinton, 2009). Embeddings from these sources exhibit a variety of influences, through their use of different modelling tasks, linearity, manner of relating words to their contexts, dimensionality, and scale and domain of training datasets (as listed in Table 1).

`w2v` was developed to predict a word from its context using the CBOW 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 i^{th} 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.[1] Google News data was used 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 is based on a similar bilinear formulation, framed as a low-rank decomposition of the matrix of corpus coocurrence 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 Wikipedia 2014 and the English Gigaword corpus version 5.

`HLBL` is a log-bilinear formulation of an n-gram language model, which predicts the i^{th} 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, $\{C_j\}$ are scaling matrices, and b_* bias terms.

The final model, `SENNA`, 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.

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

Our expectation is that the differences in initial training conditions will affect performance, e.g. we expect the bidirectional models to work better than the left-to-right ones, and log-linear models to outperform their non-linear counterparts, due to our use of linear vector difference.

[1] In a slight abuse of notation, the subscripts of $\mathbf{w}$ play double duty, denoting either the embedding for the i^{th} token, $\mathbf{w}_i$, or k^{th} word type, $\mathbf{w}_k$.

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 lexical relations to be binary and to have fixed directionality. Consequently we excluded symmetric lexical relations such as synonymy. We additionally constrained the dataset to the words occurring in all four pre-trained embeddings. 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 preprocessed the data to exclude all undirected relations, remove duplicate triples 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 dataset of Tan et al. (2006a), Princeton WordNet (Fellbaum, 1998), Wiktionary, and a web source, as listed in Table 2 and detailed below (wherein we define each relation relative to the directed word pair (x, y)). We will release this dataset on publication of this paper.

Lexical Semantic Relations

We constructed our dataset from the combination of the six top-level asymmetric lexical semantic relations from SemEval-2012 Task 2 (Jurgens et al., 2012) and three lexical semantic relations from BLESS (Baroni and Lenci, 2011). There is partial overlap between the two datasets, meaning that we consolidated the relations as follows:

LEXSEM$_{\mathbf{Hyper}}$: x names a class that includes entity y; e.g. (*animal*, *dog*)

LEXSEM$_{\mathbf{Mero}}$: y names a part of entity x or is an instance of class x; e.g. (*airplane*, *cockpit*)

LEXSEM$_{\mathbf{Attr}}$: y names a characteristic quality, property, or action of x; e.g. (*cloud*, *rain*)

LEXSEM$_{\mathbf{Cause}}$: y represents the cause, purpose, or goal of x or using x; e.g. (*cook*, *eat*)

LEXSEM$_{\mathbf{Space}}$: y is a thing or action that is associated with x (a location or time); e.g. (*aquarium*, *fish*)

LEXSEM$_{\mathbf{Ref}}$: x is an expression or representation of, or a plan or design for, or provides information about, y; e.g. (*song*, *emotion*)

LEXSEM$_{\mathbf{Event}}$: x refers to an action that entity y is usually involved in; e.g. (*zip*, *coat*)

Here, we have merged the class relation from SemEval'12 with the hypernymy relation from BLESS, and the part–whole relation from SemEval'12 with the meronym relation from BLESS.

Morphosyntactic Paradigm Relations

As morphosyntactic paradigm lexical relations, we include four relations from the original Mikolov et al. (2013c) DIFFVEC paper:

NOUN$_{\mathbf{SP}}$: y is the plural form (`NNS`, in Penn tagset terms) of singular noun x (an `NN`); e.g. (*year*, *years*)

VERB$_{\mathbf{3}}$: y is the 3rd person singular present-tense verb form (`VBZ`) of base-form verb x (a `VB`); e.g. (*accept*, *accepts*)

VERB$_{\mathbf{Past}}$: y is the past-tense verb form (`VBD`) of base verb x (a `VB`); e.g. (*know*, *knew*)

VERB$_{\mathbf{3Past}}$: y is the past-tense verb form (`VBD`) of 3rd person singular present-tense verb form x (a `VBZ`); e.g. (*creates*, *created*)

Morphosemantic Relations

The dataset also includes the following morphosemantic relations:

LVC: x is the light verb associated with noun y, from the "leniently"-annotated dataset of Tan et al. (2006b); e.g. (*give*, *approval*)

VERBNOUN: y is the nominalisation of verb x, as extracted (exhaustively) from Princeton WordNet v3.0; e.g. (*americanize*, *americanization*)

PREFIX: y is x prefixed with the *re* bound morpheme, as extracted (exhaustively) from Wiktionary; e.g. (*vote*, *revote*)

NOUN$_{\mathbf{Coll}}$: x is the collective noun for noun y, based on an online list;[2] e.g. (*army*, *ants*)

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

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

Relation	Pairs	Source
LexSem$_{\text{Hyper}}$	1173	SemEval'12 + BLESS
LexSem$_{\text{Mero}}$	2825	SemEval'12 + BLESS
LexSem$_{\text{Attr}}$	71	SemEval'12
LexSem$_{\text{Cause}}$	249	SemEval'12
LexSem$_{\text{Space}}$	235	SemEval'12
LexSem$_{\text{Ref}}$	187	SemEval'12
LexSem$_{\text{Event}}$	3583	BLESS
Noun$_{\text{SP}}$	100	MSR
Verb$_3$	99	MSR
Verb$_{\text{Past}}$	100	MSR
Verb$_{\text{3Past}}$	100	MSR
LVC	58	Tan et al. (2006b)
VerbNoun	3303	WordNet
Prefix	118	Wiktionary
Noun$_{\text{Coll}}$	257	Web source

Table 2: The 15 lexical relations in our dataset.

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 the resulting clusters against the lexical resources in §3.2.

As further motivation, consider Figure 1, which presents the DiffVec space for 10 samples of each class (based on a projection learned over the full dataset). The samples corresponding to the verb–verb morphosyntactic relations (Verb$_3$, Verb$_{\text{Past}}$, Verb$_{\text{3Past}}$) each form a tight cluster near the origin, spread amongst which are the verbal morphosemantic relations VerbNoun and LVC. Similarly, Noun$_{\text{SP}}$ forms another tight cluster.

We cluster the DiffVecs between all word pairs in our dataset using spectral clustering (Von Luxburg, 2007), a choice that was motivated by the fact that it is a hard clustering algorithm that can capture clusters of arbitrary geometric shape, and has achieved superior results to other (hard) clustering methods over a variety of tasks (Ng et al., 2002).

Spectral clustering has two hyperparameters: (1) the number of clusters; and (2) the pairwise similarity measure for comparing DiffVecs. We tune the hyperparameters over development data, in the form of 15% of randomly-sampled instances, selecting the configuration that maximises the V-Measure (Rosenberg and Hirschberg, 2007). V-Measure is an information-theoretic measure that combines homogeneity and completeness, and is defined in terms of normalised conditional entropy of the true classes

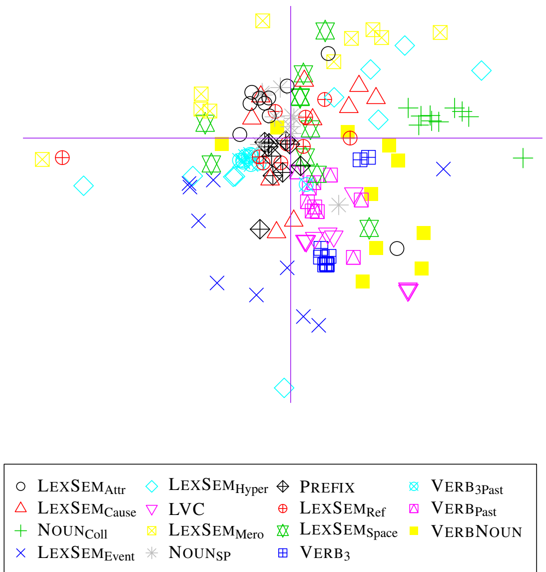

Figure 1: t-SNE projection (Van der Maaten and Hinton, 2008) of DiffVecs for 10 sample word pairs of each relation type, based on `w2v`. The intersection of the two axes identify the projection of the zero vector. Best viewed in colour.

given a clustering, and vice-versa:

$$V = \frac{2 \times \text{homogeneity} \times \text{completeness}}{\text{homogeneity} + \text{completeness}}$$

Our use of V-Measure is based on the findings of Christodoulopoulos et al. (2010), who showed for part-of-speech induction that out of seven clustering evaluation measures, V-Measure is the most effective and least sensitive to the number of clusters.

To populate the affinity matrix for spectral clustering, we scale using a Gaussian kernel:[3]

$$\exp\left(-\gamma \times \frac{\text{dist}\left(\Delta_{i,j}, \Delta_{k,l}\right)}{\sigma}\right),$$

where $\Delta_{i,j} = \mathbf{w}_j - \mathbf{w}_i$ is the vector difference between the embeddings of the i^{th} and j^{th} word types, σ is the standard deviation of the corpus $\text{dist}\left(\Delta_{i,j}, \Delta_{k,l}\right)$ values, and γ is a hyper-parameter which determines the decay rate as the distance increases. The distance metric, $\text{dist}\left(\Delta_{i,j}, \Delta_{k,l}\right)$ is Euclidean distance, while γ affects how quickly the score drops with distance: high γ values have a faster

[3]The Gaussian kernel introduces an extra non-linearity into the formulation. In preliminary experiments, we found this to outperform the basic cosine and Euclidean distances.

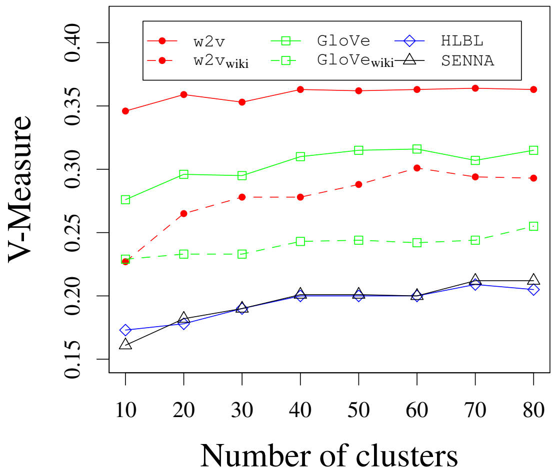

Figure 2: Spectral clustering results, comparing cluster quality (V-Measure) and the number of clusters. DIFFVECs are clustered and compared to the known relation types. Each line shows a different source of word embeddings.

	`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$_{\text{3Past}}$	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 of the four word embeddings.

decay and effectively impose a threshold distance, beyond which points are assigned a near-zero similarity value. $\gamma = 0.1$ provided the best performance over the development data, and is used in all experiments.

Note that the results of spectral clustering depend on random initialisation, so we ran several experiments using the same parameters, and average across them in the final results.

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 increasing steps of 10, up to $N = 80$ (beyond which the clustering quality diminishes).[4] Observe that `w2v` achieves the best results, with a V-Measure value of around 0.36,[5] which is relatively constant over varying numbers of clusters. `GloVe` mirrors this result, but is consistently below `w2v` at a V-Measure of around 0.31. `HLBL` and `SENNA` performed very similarly, at a substantially lower V-Measure than `w2v` or `GloVe`, closer to 0.21.

One possible explanation for the relative ordering for the results of the four methods in Figure 2 is that, for the pre-trained vectors we use: (a) `w2v` is trained over a larger corpus than `GloVe`, which is in turn trained over a much larger corpus than `SENNA` and `HLBL`; and (b) `w2v` has higher dimensionality than the other methods. To determine whether this is, indeed, the cause of the difference, we additionally report on results for `w2v` and `GloVe` over English Wikipedia (comparable to `SENNA` and `HLBL`). For the two methods, we set the dimensionality to 300, and other parameters to default values. The results are presented in the plot as `w2v`$_{\text{wiki}}$ and `GloVe`$_{\text{wiki}}$. While there is a drop in results for both methods, both perform well above `SENNA` and `HLBL`, and `w2v` still has a clear empirical advantage over `GloVe`. As such, the superiority of `w2v` would appear to be a true effect, based on which we focus exclusively on `w2v` for the remainder of our experiments .

To better understand these results, and the clustering performance over the different lexical relations, 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 non-uniformly, we normalise entropy results for each method by $\log(n)$ where n is the number of samples in a particular relation.

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

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

Table 3 presents the entropy values for each relation and embedding, with the lowest entropy (purest clustering) for each relation indicated in bold. Combining the V-Measure and entropy results we can see that the clustering does remarkably well, without any supervision in terms of either the training of the word embeddings[6] or the clustering of the DIFFVECs, nor indeed any explicit representation of the component words (as all instances are DIFFVECs). While it is hard to calibrate the raw numbers, for the somewhat related lexical semantic 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.

Looking across the different lexical relation types, the morphosyntactic paradigm relations (NOUN$_{\text{SP}}$ and the three VERB relations) are by far the easiest, with `w2v` notably achieving a perfect clustering of the word pairs for VERB$_3$. The lexical semantic relations, on the other hand, are the hardest to capture for all embeddings.

Looking in depth at the composition of the clusters, taking `w2v` as our exemplar word embedding (based on it achieving the highest V-Measure), for VERB$_3$ there was a single cluster consisting of around 90% VERB$_3$ word pairs. The remaining 10% of instances tended to include a word that was ambiguous in POS, leading to confusion with VERBNOUN in particular. Example VERB$_3$ pairs incorrectly clustered with other relations are: (*study*, *studies*), (*run*, *runs*), (*remain*, *remains*), (*save*, *saves*), (*like*, *likes*) and (*increase*, *increases*). This polysemy results in the distance represented in the vector difference for such pairs being above the average for VERB$_3$, and the word pairs consequently being clustered with word pairs associated with other cross-POS relations.

For VERB$_{\text{Past}}$, a single relatively pure cluster was generated, with minor contamination due to semantic and syntactic ambiguity with word pairs from lexical semantic relations such as (*hurt*, *saw*), (*utensil*, *saw*), and (*wipe*, *saw*). Here, the noun *saw* is ambiguous with a high-frequency past-tense verb, and for the first and last example, the first word is also ambiguous with a base verb, but from a different paradigm. A similar effect was observed for NOUN$_{\text{SP}}$. This suggests a second issue: the words in a word pair individually having the correct lexical property (in terms of verb tense/form) for the lexical relation, but not satisfying the additional paradigmatic constraint associated with the relation.

A related phenomenon was observed for NOUN$_{\text{Coll}}$, where the instances were assigned to a large mixed cluster containing word pairs where word y 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.

LEXSEM$_{\text{Mero}}$ was split into multiple clusters along domain lines, with separate clusters for weapons, dwellings, vehicles, etc. Other semantic relations were clustered in similar ways, with one cluster largely made up of (ANIMAL_NOUN, MOVEMENT_VERB) word pairs, and another comprised largely of (FOOD_NOUN, FOOD_VERB) word pairs. Interestingly, there was also a large cluster of (PROFESSION_NOUN, ACTION_VERB) pairs.

Our clustering methodology could, of course, be applied to an open-world dataset including randomly-sampled word pairs, and the resultant clusters examined to determine their relational composition, perhaps showing that relation discovery is possible using word embeddings and DIFFVECs. Instead, however, we opt to investigate open-world relation learning based on a supervised approach, as detailed in the next section.

5 Classification

A natural question is whether we can accurately characterise lexical relations based on DIFFVECs through supervised learning over the DIFFVECs. For these experiments we use the `w2v` embeddings exclusively, 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$_{\text{3Past}}$, and VERB$_{\text{Past}}$.

We consider two applications: (1) a CLOSED-

[6] With the minor exception of `SENNA`, in that the word embeddings were indirectly learned using multi-task learning.

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 memorization effect for a broad range of relation types of the sort recently identified by Weeds et al. (2014) and Levy et al. (2015) 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 $\langle \Delta_{i,j}, r \rangle$ pairs, where r is one of our nine relation types.

We use an SVM with a linear kernel and report results from 10-fold cross-validation in Table 4. Most of the relations, even the most difficult ones from our clustering experiment, are classified with very high precision and recall. 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, due to large differences in the predominant usages associated with the respective words (e.g., (*union*, *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 (*entry*, *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 (`w2v` vectors of size 300), we found that the linear SVM slightly outperformed a non-linear SVM using an RBF kernel.

As a baseline, we first cluster the data as described in §4. We run the clusterer several times over the 9-relation data to select the optimal V-Measure value based on the development data, corresponding in this case to 50 clusters. We assign to each cluster the majority class based the training instances, and evaluate the resultant labelling for the test instances. The linear SVM achieves a higher F-score than the baseline on almost every relation, particularly on $\text{LEXSEM}_{\text{Hyper}}$, and the lower-frequency NOUN_{SP}, $\text{NOUN}_{\text{Coll}}$, and PREFIX.

Relation	Baseline			SVM		
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
$\text{LEXSEM}_{\text{Hyper}}$	0.60	0.61	0.60	0.96	0.91	0.93
$\text{LEXSEM}_{\text{Mero}}$	0.93	0.88	0.90	0.97	0.98	0.97
$\text{LEXSEM}_{\text{Event}}$	0.82	0.93	0.87	0.97	0.99	0.98
NOUN_{SP}	0.00	0.00	0.00	0.83	0.83	0.83
VERB_{3}	1.00	0.98	0.99	0.99	0.97	0.98
$\text{VERB}_{\text{Past}}$	0.80	0.77	0.78	0.97	1.00	0.98
$\text{VERB}_{\text{3Past}}$	1.00	0.98	0.99	1.00	0.97	0.98
PREFIX	0.00	0.00	0.00	0.99	0.70	0.82
$\text{NOUN}_{\text{Coll}}$	0.15	0.27	0.19	0.98	0.91	0.95
MicroAvg.	0.82	0.86	0.84	0.97	0.97	0.97

Table 4: Precision ($\mathcal{P}$), recall ($\mathcal{R}$) and F-score ($\mathcal{F}$) for CLOSED-WORLD classification, for a baseline method based on clustering + majority-class labelling, and a multiclass linear SVM trained on 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 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.

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 noise samples, generated as follows:

(1) we first sample a seed lexicon by drawing words proportional to their frequency in Wikipedia;[7]
(2) next, we take the Cartesian product over pairs of words from the seed lexicon;
(3) finally, we 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 SVM classifiers with RBF kernels 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

[7]Filtered to consist of words for which we have embeddings.

Relation	Orig		+neg	
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}$
LEXSEM$_{\text{Hyper}}$	0.95	**0.92**	**0.99**	0.84
LEXSEM$_{\text{Mero}}$	0.13	**0.96**	**0.95**	0.84
LEXSEM$_{\text{Event}}$	0.44	**0.98**	**0.93**	0.90
NOUN$_{\text{SP}}$	0.95	**0.68**	**1.00**	**0.68**
VERB$_{3}$	0.75	**1.00**	**0.93**	0.93
VERB$_{\text{Past}}$	0.94	**0.90**	**0.97**	0.84
VERB$_{\text{3Past}}$	0.76	**0.95**	**0.87**	0.93
PREFIX	**1.00**	**0.29**	**1.00**	0.13
NOUN$_{\text{Coll}}$	0.43	**0.74**	**0.97**	0.41

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

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*), (*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 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.[8] 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 Memorization

Weeds et al. (2014) and Levy et al. (2015) recently showed that supervised methods using DIFFVECs achieve artificially high results as a result of "lexical memorization" over frequent words associated 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 issue, we randomly split our CLOSED-WORLD vocabulary into two lexically-disjoint partitions, which we call training and test.

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

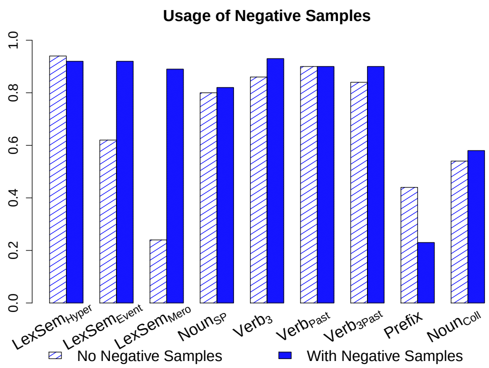

Figure 3: F-score for OPEN-WORLD classification, comparing models trained with and without negative samples.

Relation	Split			Overlap		
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
LEXSEM$_{\text{Hyper}}$	0.87	0.71	0.79	0.93	0.90	0.91
LEXSEM$_{\text{Mero}}$	0.89	0.95	0.92	0.94	0.97	0.95
LEXSEM$_{\text{Event}}$	0.89	0.96	0.92	0.95	0.98	0.97
NOUN$_{\text{SP}}$	0.7	0.33	0.45	0.9	0.43	0.58
VERB$_3$	1.00	1.00	1.00	1.00	1.00	1.00
VERB$_{\text{Past}}$	0.96	1.00	0.98	1.00	1.00	1.00
VERB$_{3\text{Past}}$	1.00	1.00	1.00	1.00	1.00	1.00
PREFIX	1.00	0.71	0.83	1.00	0.54	0.70
NOUN$_{\text{Coll}}$	0.94	0.63	0.75	0.96	0.86	0.91
MicroAvg.	0.89	0.89	0.89	0.94	0.94	0.94

Table 6: Precision ($\mathcal{P}$), recall ($\mathcal{R}$) and F-score ($\mathcal{F}$) for CLOSED-WORLD classification, where a multiclass linear SVM was trained on DIFFVEC inputs with/without overlap.

We compare the results of classification using two training datasets. The first ("Split") contains labelled pairs from the original CLOSED-WORLD where both words occur in the training partition. The second ("Overlap") relaxes the lexical partitioning by adding labelled pairs from the original CLOSED-WORLD where one word is in the training partition and the other in the test partition. The test dataset is the same in both cases, namely all labelled pairs from the original CLOSED-WORLD where both words are in the test partition. For the Overlap setting, we also downsample the training set to the same size as the training data for the Split setting. We train a multiclass classification model over the data, as described in §5.1. Results are shown in Table 6.

The results show that most of the relations maintain good classification accuracy with minimal degradation from the Overlap to the Split setting, with the exception of LEXSEM$_{\text{Hyper}}$, NOUN$_{\text{SP}}$, and NOUN$_{\text{Coll}}$. Interestingly, the biggest losses for LEXSEM$_{\text{Hyper}}$ and NOUN$_{\text{Coll}}$ are in recall, suggesting lexical memorization may play a role in retrieving triples with words seen in training. Other relations, notably LEXSEM$_{\text{Mero}}$, LEXSEM$_{\text{Event}}$, and the morphosyntactic verb paradigm relations show similar classification accuracy under the Overlap and Split conditions.

To measure the extent of lexical memorization for each relation, we calculated: (1) the difference in F-score between the "Overlap" and "Split" experiments; and (2) the average number of training instances containing each of the two words in test instances associated with that relation. Our hypothesis here is that the greater the average representation of test instances in the training data, the greater the difference in F-score, and that there will be a direct correlation between the degree of lexical overlap and the inflation in F-scores. The Pearson's correlation across the 9 relations was $r = 0.66$, lending strong support to this hypothesis.

We also report on an OPEN-WORLD experiment (see §5.2-5.3) in a split vocabulary setting. This experiment is analogous to those test sets in Levy et al. (2015) that include random pairs as confounders for the target hypernym relation. Once again, we first split our vocabulary into training and test portions, 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

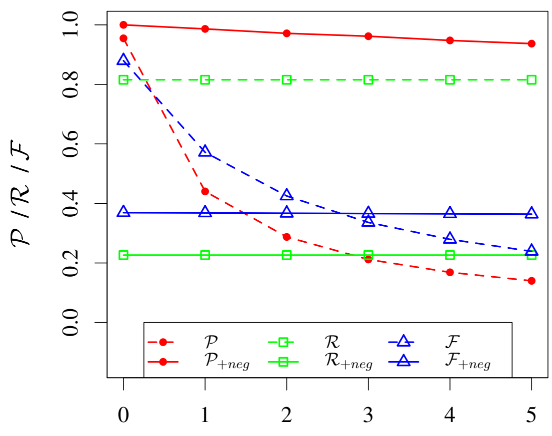

Figure 4: Evaluation of the OPEN-WORLD model when trained on split vocabulary, for varying numbers of random word pairs in the test dataset (expressed as a multiplier relative to the number of CLOSED-WORLD test instances).

random word pairs are added, as the vast majority of them don't 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.

5.5 Comparison with a Count-based Method

We also consider a "count"-based vector space model, to determine the generalisability of DIFFVEC-based relation classification. To train the model, we evaluate a word co-occurrence matrix over the same English Wikipedia corpus as used in §4 to calibrate w2v and GloVe over the same training data and dimensionality. Specifically, we use a bag-of-words context window of 3 to either side of the target word, and restrict our vocabulary to terms which occur at least 5 times in the corpus. We truncate the context matrix to the 10,000 most frequent words (similarly to Pennington et al. (2014)), scale the frequencies with the function $\log(freq_{ij} + 1)$, and finally run SVD over the context matrix. The representation of each target word is based on the first 300 columns in the output, to produce a representation of the same size as w2v.

We built a CLOSED-WORLD multi-class classifier in the same manner as described in §5.1, over the full dataset (with lexical overlap). The results are presented in Table 7, and should be contrasted with those from Table 4.

Relation	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
$\text{LexSem}_{\text{Hyper}}$	0.96	0.22	0.37
$\text{LexSem}_{\text{Mero}}$	0.78	0.97	0.87
$\text{LexSem}_{\text{Event}}$	0.76	0.98	0.85
Noun_{SP}	0.00	0.00	0.00
Verb_{3}	0.00	0.00	0.00
$\text{Verb}_{\text{Past}}$	0.00	0.00	0.00
$\text{Verb}_{\text{3Past}}$	1.00	0.01	0.02
$\text{Noun}_{\text{Coll}}$	0.00	0.00	0.00
MicroAvg.	0.74	0.78	0.71

Table 7: Precision ($\mathcal{P}$), recall ($\mathcal{R}$) and F-score ($\mathcal{F}$) for CLOSED-WORLD classification for count-based SVD model.

The first thing to notice is that the overall results are substantially lower than those for w2v. Looking at the breakdown across the different relations, we can see that the classifier heavily favours the lexical semantic relations (in particular $\text{LexSem}_{\text{Mero}}$ and $\text{LexSem}_{\text{Event}}$), so much so that only one test instance is assigned to any of the other relations (namely $\text{Verb}_{\text{3Past}}$). That is, the $\mathit{Diff}_{w1,w2}$ method works considerably less impressively over vectors learned through a count-based method. We observed similar results using non-negative sparse embeddings (Murphy et al., 2012).

6 Conclusions

This paper is the first to test the generalizability 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 memorization. Overall, we conclude that the DIFFVEC approach has impressive utility over a broad range of lexical relations, especially under supervised classification.

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.

John A Bullinaria and Joseph P Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526.

Kai-Wei Chang, Wen tau Yih, and Christopher Meek. 2013. Multi-relational latent semantic analysis. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, pages 1602–1612, Seattle, USA.

Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 33–40, Barcelona, Spain.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2010. Two decades of unsupervised pos induction: How far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 575–584, Boston, USA.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Dmitry Davidov and Ari Rappoport. 2008. Classification of semantic relationships between nominals using pattern clusters. In *Proceedings of the 46th Annual Meeting of the ACL: HLT (ACL 2008)*, pages 227–235, Columbus, USA.

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 04: Classification of semantic relations between nominals. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 13–18, Prague, Czech Republic.

Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING '92)*, pages 539–545, Nantes, France.

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 Evaluations (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.

Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. 2008. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of the 46th Annual Meeting of the ACL: HLT (ACL 2008)*, pages 1048–1056, Columbus, USA.

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, Steffen Remus, Chris Biemann, Ido Dagan, and Israel Ramat-Gan. 2015. 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.

Tara McIntosh, Lars Yencken, James R. Curran, and Timothy Baldwin. 2011. Relation guided bootstrapping of semantic lexicons. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL HLT 2011)*, pages 266–270, Portland, USA.

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)*.

Brian Murphy, Partha Pratim Talukdar, and Tom M Mitchell. 2012. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 1933–1950, Mumbai, India.

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.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, pages 849–856.

Diarmuid Ó Séaghdha and Ann Copestake. 2009. Using lexical and relational similarity to classify semantic relations. In *Proceedings of the 12th Conference of the EACL (EACL 2009)*, pages 621–629, Athens, Greece.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of COLING/ACL 2006*, pages 113–120, Sydney, Australia.

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 EACL (EACL 2014)*, pages 511–519, Gothenburg, Sweden.

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 EACL (EACL 2014)*, pages 38–42, Gothenburg, Sweden.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems 17 (NIPS-05)*, pages 1297–1304, Vancouver, Canada.

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.

Peter D. Turney, Michael L. Littman, Jeffrey Bigham, and Victor Shnayder. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03)*, pages 482–489, Borovets, Bulgaria.

Peter D. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3):379416.

Peter D. Turney. 2013. Distributional semantics beyond words: Supervised learning of analogy and paraphrase. *Transactions of the Association for Computational Linguistics*, 1:353–366.

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 and Timothy Baldwin. 2004. Automatic discovery of telic and agentive roles from corpus data. In *Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation (PACLIC 18)*, pages 115–126, Tokyo, Japan.

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 NAACL-HLT*.