Reading Between the Lines: Overcoming Data Sparsity for Accurate Classification of Lexical Relationships

Silvia Necşulescu
Universitat Pompeu Fabra
Barcelona, Spain
silvia.necsulescu@upf.edu

Sara Mendes
Universidade de Lisboa
Lisboa, Portugal
sara.mendes@clul.ul.pt

David Jurgens
McGill University
Montreal, Canada
jurgens@cs.mcgill.ca

Núria Bel
Universitat Pompeu Fabra
Barcelona, Spain
nuria.bel@upf.edu

Roberto Navigli
Università "La sapienza"
Rome, Italy
navigli@di.uniroma1.it

Abstract

The lexical semantic relationships between word pairs are key features for many NLP tasks. Most approaches for automatically classifying related word pairs are hindered by data sparsity because of their need to observe two words co-occurring in order to detect the lexical relation holding between them. Even when mining very large corpora, not every related word pair co-occurs. Using novel representations based on graphs and word embeddings, we present two systems that are able to predict relations between words, even when these are never found in the same sentence in a given corpus. In two experiments, we demonstrate superior performance of both approaches over state of the art, achieving significant gains in recall.

1 Introduction

Resources containing lexical-semantic relations such as *hypernymy* or *meronymy* have proven useful in many NLP tasks. While resources such as WordNet (Miller, 1995) contain many general relations and subsequently have seen widespread adoption, developing this type of rich resource for new languages or for new domains is prohibitively costly and time-consuming. Therefore, automated approaches are needed and in order to create such a lexical-semantic database, a first step is to develop accurate techniques for classifying the type of lexical-semantic relationship between two words.

Approaches to classifying the relationship between a word pair have typically relied on the assumption that contexts where word pairs co-occur will yield information on the semantic relation (if any) between them. Given a set of example word pairs having some relation, relation-specific patterns may be automatically acquired from the contexts in which these example pairs co-occur (Turney, 2008b; Mintz et al., 2009). Comparing these relation-specific patterns with those seen with other word pairs measures *relational similarity*, i.e., how similar is the relation holding between two word pairs. However, any classification system based on patterns of co-occurrence is limited to only those words co-occurring in the data considered; due to the Zipfian distribution of words, even in a very large corpus there are always semantically related word pairs that do not co-occur. As a result, these pattern-based approaches have a strict upper-bound limit on the number of instances that they can classify. As an alternative to requiring co-occurrence, other works have classified the relation of a word pair using *lexical similarity*, i.e., the similarity of the concepts themselves. Given two word pairs, (w_1, w_2) and (w_3, w_4), if w_1 is lexically similar to w_3 and w_2 to w_4 (i.e., are pair-wise similar) then the pairs are said to have the same semantic relation. These two sources of information are used as two independent units: *relational similarity* is calculated using co-occurrence information; *lexical similarity* is calculated using distributional information (Snow et al., 2004; Séaghdha and Copestake, 2009; Herdadelen and Baroni, 2009), and ultimately these scores are combined. Experimental evidence has shown that relational similarity cannot necessarily be revealed through lexical similarity (Turney, 2006b; Turney, 2008a), and therefore, the issue of how to collect information for word pairs that do not co-occur is still an open problem.

We propose two new approaches to representing word pairs in order to accurately classify them as instances of lexical-semantic relations – even when the pair members do not co-occur. The first approach creates a word pair representation based on a graph representation of the corpus created with dependency relations. The graph encodes the distributional behavior of each word in the pair and consequently, patterns of co-occurrence expressing each target relation are extracted from it as relational information. The second approach uses word embeddings which have been shown to preserve linear regularities among words and pairs of words, therefore, encoding lexical and relational similarities (Baroni et al., 2014), a necessary property for our task. In two experiments comparing with state-of-the-art pattern-based and embedding-based classifiers (Turney, 2008b; Zhila et al., 2013), we demonstrate that our approaches achieve higher accuracy with significantly increased recall.

2 Related work

Initial approaches to the extraction of lexical-semantic relations have relied on hand-crafted lexico-syntactic patterns to identify instances of semantic relations (Hearst, 1992; Widdows and Dorow, 2002; Berland and Charniak, 1999). These manually designed patterns are explicit constructions expressing a target semantic relation such as the pattern *X is a Y* for the relation of *hypernymy*. However, these approaches are limited because a relation may be expressed in many ways, depending on the domain, author, and writing style, which may not match the originally identified patterns. Moreover, the identification of high-quality patterns is costly and time-consuming, and must be repeated for each new relation type, domain and language. To overcome these limitations, techniques have been developed for the automatic acquisition of meaningful patterns of co-occurrence cueing a single target relation (Snow et al., 2004; Girju et al., 2006; Davidov and Rappoport, 2006).

More recent work focuses on methods for the classification of word pairs as instances of several relations at once, based on their relational similarity. This similarity is calculated using a vectorial representation for each pair, created by relying on co-occurrence contexts (Turney, 2008b; Séaghdha and Copestake, 2009; Mintz et al., 2009). These representations are very sparse due to the scarce contexts where the members of many word pairs co-occur. Moreover, many semantically related word pairs do not co-occur in corpus.

For overcoming these issues, relational similarity was combined with lexical similarity calculated based on the distributional information of words (Cederberg and Widdows, 2003; Snow et al., 2004; Turney, 2006a; Séaghdha and Copestake, 2009; Herdadelen and Baroni, 2009). However, (Turney, 2006b; Turney, 2008a) showed that relational similarity cannot be improved using the distributional similarity of words. In contrast with the previous approaches that took into account lexical and relational information as a linear combination of lexical and relational similarity scores, the present work focuses on introducing word pair representations that merge and jointly represent types of information: lexical and relational. In this way, we aim to reduce vector sparseness and to increase the classification recall.

As a first approach, we use a graph to model the distributional behavior of words. Other researchers used graph-based approaches to model corpus information for the extraction of co-hyponyms (Widdows and Dorow, 2002), hypernyms (Navigli and Velardi, 2010) or synonyms (Minkov and Cohen, 2012). Navigli and Velardi (2010) have the most similar representation to ours, creating a graph that models only definitional sentences. In contrast, our objective is to create a general representation of the whole corpus that can be used for classifying instances of several lexical semantic relations. The second approach presented in this paper, relies on word embeddings to create word pair representations. Extensive experiments have leveraged word embeddings to find general semantic relations (Mikolov et al., 2013a; Mikolov et al., 2013b; Mikolov et al., 2013c; Levy and Goldberg, 2014b). Nevertheless, only one work has applied word embeddings for classifying instances of a lexical semantic relation, specifically the relation *hyponymy-hypernymy* (Fu et al., 2014). This relation is more complex than other semantic relations tested and therefore, it is reflected in more than one offset, depending on the domain of each instance. The present work uses a machine learning approach to discover meaningful information for the semantic relations encoded in the dimensions of the embeddings.

3 Task description

The goal of this work is to classify word pairs as instances of lexical-semantic relations. Given a set of target semantic relations $R = \{r_1, \ldots, r_n\}$, and a set of word pairs $W = \{(x, y)_1, \ldots, (x, y)_n\}$, the task is to label each word pair $(x, y)_i$ with the relation $r_j \in R$ holding between its members and outputting a set of tuples $((x, y)_i, r_j)$. For this task, we propose two novel representations of word pairs (described next), which are each used to train a classifier. Following, in Section 3.1 and Section 3.2 we describe each representation and then, in Section 3.3, we describe the common classification setup used with both representations.

3.1 Graph-based Representation Model

The present section introduces a novel word pair representation model based on patterns of co-occurrence contexts, and on a graph-based corpus representation created with dependency relations. A word pair is represented as a vector of features set up with the most meaningful patterns of context and filled in with information extracted from the graph representation of the corpus. We refer to systems trained with these graph-based representations as **Gra**ph-based **C**lassification syst**E**m (GraCE).

The novelty of this system stands in the graph-based representation. Its main advantage is that all the dependency relations of a target word, extracted from different sentences, are incident edges to its corresponding node in the graph. Thus, words that never co-occur in the same context in corpus, are linked in the graph through bridging words: words that appear in a dependency relation with each member of the pair but in different sentences. With this representation we address the data sparsity issue, aiming to overcome the reported major bottleneck of previous approaches: low recall because information can only be gathered from co-occurrences in the same sentence of two related words.

Word pair representations are created in three steps:

(1). Corpus representation: the input corpus is represented as a graph;

(2). Feature selection: the input corpus is used to extract meaningful patterns of co-occurrence for each semantic relation r_i starting from an initial set of examples E;

(3). Word pair representation: the information acquired in **(1)** and **(2)** is used to create vectorial representations of target word pairs.

Next, we present an example of how the graph representation of the corpus addresses the sparsity problem in distributional data and formally introduce each step of the GraCE algorithm.

Example To illustrate the benefit of acquiring information about a word pair from the graph instead of using co-occurrence information, let us consider that given the sentences (S1) and (S2) below, we want to classify the pair *(chisel, tool)* as an instance of the relation of hypernymy.

(S1) The students learned how to handle screwdrivers, hammers and other tools.

(S2) The carpenter handles the new chisel.

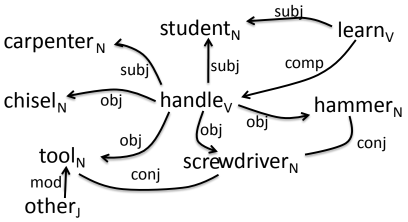

Figure 1: Dependency multigraph built from a two sentence corpus using GraCE. See text for details.

The word pair *(chisel, tool)* has a relation of hypernymy but its members do not co-occur in the same sentence. However, both words occur as objects of the verb *to handle* in different sentences, just like other hypernym word pairs such as *(hammer, tool)* and *(screwdriver, tool)* which do co-occur in the same sentence. This shows that *handle* is one of the contexts shared between these semantically related words that provide information regarding a possible semantic relatedness between them. Leveraging only the information provided by each sentence, as existing pattern based approaches do, no evidence is acquired regarding the semantic relation holding between *chisel* and *tool*. GraCE combines the dependency relations seen in each sentence in the graph shown in Figure 1. In this graph, *chisel* and *tool* are connected by a path passing through the bridging word *handle* which shows that both *chisel* and *tool* could co-occur in a sentence as objects of the verb *to handle*, although they do not in the example two-sentence corpus.

Corpus representation The goal of the first step is to generate a graph connecting semantically associated words using observed dependency relations.

Formally, the corpus is represented as a graph $\mathbb{G} = (V, E)$, where V is a set of POS-tagged lemmas in a corpus and E is the set of dependency relations connecting two lemmas from V in the corpus. From each parsed sentence of the corpus, a set of dependency relations linking the words in it is produced: $D = \{d_1 \ldots, d_{|D|}\}$, where $d = (w_i, dep, w_j)$ and w_i, w_j and dep denote POS-tagged lemmas and a dependency relation, respectively. The graph $\mathbb{G}$ is created using all the dependency relations from D.

The output of this step is a multigraph, where two words are connected by the set of edges containing all the dependency relations holding between them in the corpus.

Feature Selection The goal of the second step is to collect features associated with each relation r from the parsed input corpus. Similar to the work of Snow et al. (2004), our features are *patterns of co-occurrence contexts* created with dependency paths. For acquiring patterns of co-occurrence contexts for each relation r, we use the set of labeled examples E, assuming that all the contexts in which a word pair $(x, y)_i \in E$ co-occurs provide information about the relation r holding between its members. All the dependency paths between x and y up to three edges are extracted from the dependency graph of each sentence where $(x, y)_i$ co-occur.[1] For example, $((hammer_N, tool_N), hyper)$ is an instance of the relation of hypernymy. In the dependency graph of sentence (S1), the words $hammer_N$ (hyponym) and $tool_N$ (hypernym) are connected by the dependency path $hammer_N \xleftarrow{\text{obj}} handle_V \xrightarrow{\text{obj}} tool_N$. This path is converted into a *pattern of co-occurrence contexts* by replacing the seeds in the path with their parts of speech as follows: $N \xleftarrow{\text{obj}} handle_V \xrightarrow{\text{obj}} N$. Table 1 illustrates several examples of pattern of co-occurrence contexts.

attr.	$X_N \xrightarrow{\text{prep\_such\_as}^{-1}} tool_N \xrightarrow{\text{mod}} Y_J$
co-hypo.	$X_N \xrightarrow{\text{obj}^{-1}} use_V \xrightarrow{\text{obj}} Y_N$
action	$X_N \xrightarrow{\text{obj}^{-1}} use_V \xrightarrow{\text{conj}} Y_V$
hyper.	$X_N \xrightarrow{\text{prep\_such\_as}^{-1}} tool_N \xrightarrow{\text{conj}} Y_N$
mero.	$X_N \xrightarrow{\text{nn}^{-1}} blade_N \xrightarrow{\text{conj}} Y_N$

Table 1: Examples of relation features

For the word pairs vectorial representation, the top 5000 most meaningful patterns are considered in the final set of patterns $\mathbb{P}$ to form a feature space.[2] In order to rank the patterns, the *tf-idf score* is calculated for each pattern with respect to each lexical semantic relation. Here, $tf - idf$ is defined as $max_j(\frac{log(uniq(p_i,r_j)+1)*|R|}{|R_p|})$, where p_i is a pattern of co-occurrence, $uniq(p_i, r_j)$ is the number of unique instances of the relation r_j occurring in the pattern p_i and $|R_p|$ is the number of relations r_j whose example instances are seen occurring in the pattern p_i. Each pattern is then associated with the highest *tf-idf* score obtained across all relations.

Word pair representation Using the graph model $\mathbb{G}$ and the set of contextual patterns automatically acquired $\mathbb{P}$, each word pair (x, y) is represented as a binary distribution over each pattern from $\mathbb{P}$. Rather than using the input corpus to identify contexts of occurrence for the word pair (x, y) and match those with the acquired patterns, GraCE uses paths connecting x and y in $\mathbb{G}$. All the paths between x and y up to three edges are extracted from $\mathbb{G}$. These paths are then matched against the feature patterns from $\mathbb{P}$ and the word pair (x, y) is represented as a binary vector encoding non-zero values for all the features matching the pair's paths extracted from $\mathbb{G}$, and zero otherwise.[3] Because the graph contains combinations of multiple dependency relations, extracted from various sentences, paths not observed in the corpus can be found in the graph.

3.2 Word Embeddings Representations

The present section introduces two word pair representations based on word embeddings. We refer to

[1] Paths with more than three edges commonly connect semantically-unrelated portions of a sentence and therefore are not beneficial for the purposes of relation classification.

[2] Initial experiments tested different amounts of patterns using held out data and the best results were obtained with the top 5000 patterns.

[3] Binary weights are used because the feature values are derived observing paths in the graph, which is a generalization of the corpus; because not all paths in the graph are observed in the corpus, weighting based on path frequency would encounter the same data sparsity issue that the graph is intended to overcome.

systems based on embeddings as **W**ord **E**mbeddings **C**lassification syst**E**m (WECE). An embedding is a low-dimensional vectorial representation of a word, where the dimensions are latent continuous features and vector components are set to maximize the probability of the contexts in which the target word tends to appear. Since similar words occur in similar contexts the word embeddings learn similar vectors for similar words. Moreover, the vector offset of two word embeddings reflect the relation holding between them. For instance, Mikolov et al. (2013c) give the example that $v(king) - v(man) \approx v(queen) - v(women)$, where $v(x)$ is the embedding of the word x, indicating the vectors are encoding information on the words' semantic roles.

For learning word embeddings, we used the Skip-gram model, improved with techniques of negative sampling and subsampling of frequent words, which achieved the best results for detecting semantically similar words (Mikolov et al., 2013a; Mikolov et al., 2013b). Moreover, for a fair comparison with the GraCE system, developed with dependency relations, we also tested the results obtained with a dependency-based Skip-gram model (Levy and Goldberg, 2014a). Words occurring only once in corpus are filtered out and 200-dimensional vectors are learned.

Two embedding-based representations are considered for a relation: *WECE$_{offset}$* leverages the offset of the word embeddings, while *WECE$_{concat}$* concatenates the embeddings, both described next.

WECE$_{\mathbf{offset}}$ Representation Mikolov et al. (2013c) shows that the vectorial representation of words provided by word embeddings captures syntactic and semantic regularities and that each relationship is characterized by a relation specific vector offset. Word pairs with similar offsets can be interpreted as word pairs with the same semantic relation. Therefore, given a target word pair (x, y), the vectorial representation is calculated from the difference between its vectors, i.e., $v((x, y)) = v(x) - v(y)$. Note that this operation is dependent on the order of the arguments and is therefore potentially able to capture asymmetric relationships.

WECE$_{\mathbf{concat}}$ Representation A novel word pair representation is proposed to test if the information encoded directly in the embeddings reflects the semantic relation of the word pair.

A word pair is represented by concatenating the vectorial representation of its members. Formally, given a word pair (x, y), whose members vectorial representations are $v(x) = (x_1, x_2, \ldots, x_n)$, and $v(y) = (y_1, y_2, \ldots, y_n)$ respectively, the vectorial representation of (x, y) is defined as the concatenation of $v(x)$ and $v(y)$: $v((x, y)) = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)$ Consequently the length of $v((x, y))$ is $2n$, where n is the dimensions of the embedding space.

3.3 Relation Classification

For both representations, a supervised classifier is trained. Given a set of tuples $E = ((x, y)_i, r_i)$ of example instances for each relation $r_i \in R$, a support vector machine (SVM) multi-class classifier with a radial basis function kernel (Platt, 1999) is trained using WEKA (Hall et al., 2009) to classify each word pair based on its representation provided by a graph-based representation model (Section 3.1) or a word embeddings representation model (Section 3.2) for N different lexical relations. The SVM classifier generates a distribution over relation labels and the highest weighted label is selected as the relation holding between the members of the word pair.

4 Experiments

While several datasets have been created for detecting semantic relations between two words in context (Hendrickx et al., 2010; Segura-Bedmar et al., 2013), in our work we focus on the classification of word pairs as instances of lexical-semantic relations out of context. The performance of the GraCE and WECE systems is tested across two datasets, focusing on their ability to classify instances of specific lexical-semantic relations as well as to provide insights into the systems' generalization capabilities.

4.1 Experimental Setup

Corpora Many pattern-based systems increase the size of the input corpus in an attempt to overcome data sparsity and to achieve a better recall. Therefore, in our experiments, we train our systems using two corpora of different sizes: the British National Corpus (BNC), a 100 million-word corpus, and a Wikipedia dump created from 5 million pages and containing 1.5 billion words. The size difference allows us to measure the potential impact of increased word co-occurrence on recall. Both corpora were initially parsed with the Stanford dependency parser in the *collapsed dependency* format (Manning et al., 2014).

Embbedings WECE$_{\text{offset}}$ and WECE$_{\text{concat}}$ are implemented based on a bag-of-words (BoW) (Mikolov et al., 2013a) and based on dependency relations (Dep) (Levy and Goldberg, 2014a).

Evaluation We compare each system by reporting precision (P), recall (R) and F1 measure (F).

4.2 Comparison Systems

The two proposed models are compared with two state-of-the-art systems and one baseline system.

PAIRCLASS The PairClass algorithm (Turney, 2008b) provides a state-of-the-art pattern-based approach for extracting and classifying the relationship between word pairs and has performed well for many relation types. Using a set of seed pairs (x, y) for each relation, PairClass acquires a set of lexical patterns using the template *[0 to 1 words] x [0 to 3 words] y [0 to 1 words]*. Using the initial set of lexical patterns extracted from a corpus, additional patterns are generated by optionally generalizing each word to its part of speech. For N seed pairs, the most frequent kN patterns are retained. We follow Turney (2008b) and set $k = 20$. The patterns retained are then used as features to train an SVM classifier over the set of possible relation types.

DS$_{\text{Zhila}}$ & DS$_{\text{Levy}}$ Word embeddings have previously been shown to accurately measure relational similarity; Zhila et al. (2013) demonstrate state-of-the-art performance on SemEval-2012 Task 2 (Jurgens et al., 2012) which measures word pair similarity within a particular semantic relation (i.e., which pairs are most prototypical of a semantic relation). This approach can easily be extended to the classification setting: Given a target word pair (x, y), the similarity is computed between (x, y) and each word pair $(x, y)_i$ of a target relation r. The average of these similarity measurements was taken as the final score for each relation r.[4] Finally, the word pair is classified as an instance of the relation with the highest associated score. Two types of embeddings are used, (a) the word embeddings produced using the method of Mikolov et al. (2011), which was originally used in Zhila et al. (2013) and (b) the embeddings using the method of Levy and Goldberg (2014a), which include dependency parsing information. We refer to these as DS_{Zhila} and DS_{Levy}, respectively. The inclusion of this system enables comparing the performance impact of using an SVM classifier with our embedding-based pair representations versus classifying instances by comparing the embeddings themselves. We note a DS system represents a minimally-supervised system whose features are produced in an unsupervised way (i.e., through the embedding process) and are therefore not necessarily tuned for the task of relation classification; however, such embeddings have previously been shown to yield state-of-the-art performance in other semantic relation tasks (Baroni et al., 2014) and therefore the DS systems are intended to identify potential benefits when adding feature selection by means of the SVM in WECE systems.

BASELINE The purported benefit of the GraCE model is that the graph enables identifying syntactic features between pair members that are never observed in the corpus, which increases the number of instances that can be classified without sacrificing accuracy. Therefore, to quantify the effect of the graph, we include a baseline system, denoted BL, that uses an identical setup to GraCE but where the feature vector for a word pair is created only from the dependency path features that were observed in the corpus (as opposed to the graph). Unlike the GraCE model which has binary weighting (due to the graph properties), the baseline model's feature values correspond to the frequencies with which patterns occur; following common practice, the values are log-normalized.

4.3 Experiment 1

Both of the proposed approaches rest on the hypothesis that the graph or embeddings can enable accurate pair classification, even when pairs never cooccur in text. Therefore, in the first experiment, we test whether the recall of classification systems is improved when the word pair representation encodes information about lexical and relational sim-

[4] Additional experiments showed that using alternate ways of measuring similarity, such as using the maximum similarity for any instance of r, attained similar results.

Domain	#Co-hypo	#Hyper	#Mero
Animals	8038 (92.4%)	3039 (97.2%)	386 (89.1%)
Plants	18972 (95.5%)	1185 (97.4%)	330 (82.4%)
Vehicles	530 (82.6%)	189 (97.9%)	455 (100%)

Table 2: Distribution of K&H dataset, with the % of instances which occur in the corpora.

	BNC			Wikipedia		
	P	R	F	P	R	F
$PairClass$	76.9	4.6	8.7	77.0	11.7	20.4
BL	82.6	7.7	14.2	89.4	16.2	27.5
$GraCE$	90.7	43.8	59.0	94.0	75.5	83.7
DS_{Zhila}	31.6	15.7	21.0	32.8	22.6	26.8
DS_{Levy}	18.7	11.4	14.2	27.7	15.6	20.0
$WECE^{BoW}_{offset}$	96.0	59.1	73.1	96.8	87.7	92.0
$WECE^{BoW}_{concat}$	**97.4**	60.0	74.2	**97.6**	**89.3**	**93.2**
$WECE^{Dep}_{offset}$	87.9	63.1	73.5	95.4	86.1	90.5
$WECE^{Dep}_{concat}$	93.1	**64.7**	**76.4**	96.7	88.4	92.4

Table 3: Aggregated results obtained for the in-domain setup with the K&H dataset. Detailed results are presented in the Appendix A.

ilarity. As an evaluation dataset, we expand on the dataset of Kozareva and Hovy (2010) (K&H), which was collected from hyponym-hypernym instances from WordNet (Miller, 1995) spanning three topical domains: *animals*, *plants* and *vehicles*. Because our systems are capable of classifying instances with more than one relation at once, we enhance this dataset with instances of two more relation types: co-hyponymy and meronymy. Co-hyponyms are extracted directly from the K&H dataset: two words are co-hyponyms if they have the same direct ancestor.[5] To avoid including generic nouns, such as "migrator" in the "animal" domain, only leaf nodes are considered. The meronym instances are extracted directly from WordNet. The final dataset excludes multi-word expressions, which were not easily handled by any of the tested systems. The total number of instances considered in our experiments is presented in Table 2.

Results Table 3 presents the average of the results obtained by the systems when tested *in-domain* in a 10-fold cross-validation setup. For the *in-domain* setup, only instances from one domain are used for training and testing.

As expected, all the systems gain recall with a larger corpus, like Wikipedia, showing that the recall depends on the size of that corpus when a system acquires its distributional information directly from the input corpus and thus is dependent on the word pairs co-occurring. Indeed, in the BNC, only 19.4% of the K&H instances never co-occur, while in Wikipedia –a corpus 15 times larger than BNC– the number of co-occurrences rises only to 30.7%, demonstrating the challenge of classifying such pairs. Therefore, the real upper-bound limit for these types of systems is the amount of word pairs co-occurring in the same sentence in the corpus. The recall achieved by GraCE overcomes this limitation of pattern-based systems: 40% and 78.7% of the instances that never co-occur in BNC and in Wikipedia, respectively, are correctly classified by GraCE. This ability causes GraCE to improve the BL performance by 8.1 points in precision and 36.1 points in recall on BNC and 4.6 points in precision and 59.3 in recall on Wikipedia. Given that the BL system is constructed identically to GraCE but without using a graph, these results demonstrate the performance benefit of joining the distributional information of a corpus into a graph-based corpus representation.

Analyzing the false negatives of the GraCE classifier, we observe that even relying on a graph-based corpus representation to extract the distributional information of a word pair, many errors are still caused by the sparsity of their vectorial representation. For the word pairs that do not co-occur in the same sentence, the GraCE vector representations of correctly-classified pairs have a median of eight non-zero features, indicating that the graph was beneficial for still providing evidence of a relationship; in contast, incorrectly-classified pairs had a median of only three non-zero features, suggesting that data sparisity is still major contributor to classification error.

By combining all the distributional information into a more dense vector, WECE systems are able to improve upon GraCE's results by an average of 2.9 points in precision and 17.9 points in recall. WECE results see an increase by 62 points in precision and 46 in recall over DS_{Zhila} which used the same embeddings, highlighting the importance of the SVM classifier for learning which features of the embeddings reflect the lexical relation. Although embeddings have been argued to reflect the

[5] y is a direct ancestor of x if there is no other word z which is hypernym of x and hyponym of y.

	In-domain			Out-of-domain		
	P	R	F	P	R	F
$PairClass$	66.8	35.6	46.4	78.9	43.2	55.8
BL	79.5	51.6	62.6	71.7	40.0	51.4
$GraCE$	87.7	85.0	86.3	66.2	36.3	46.9
DS_{Zhila}	62.1	47.4	53.7	50.7	46.9	48.7
DS_{Levy}	53.0	49.2	51.0	51.1	47.5	49.2
$WECE^{BoW}_{offset}$	**90.0**	90.9	**90.4**	68.0	**66.9**	67.5
$WECE^{BoW}_{concat}$	89.9	**91.0**	**90.4**	**83.8**	57.0	67.8
$WECE^{Dep}_{offset}$	85.3	86.5	85.9	68.7	62.3	65.4
$WECE^{Dep}_{concat}$	85.9	87.0	86.5	78.2	63.8	**70.3**

Table 4: Aggregated results obtained when systems are tested with the BLESS dataset over BNC.

semantic or syntactic relations between two words (Mikolov et al., 2013c), our results suggest that additional machine learning (as done with $\text{WECE}_{\text{offset}}$) is needed to identify which dimensions of the embeddings accurately correspond to specific relationships. Between the WECE systems, $\text{WECE}_{\text{concat}}$ achieves slightly better results on the K&H dataset.

4.4 Experiment 2

In the first experiment, the proposed systems were compared to test the importance of having a representation that includes information about lexical and relational similarities for the classifier to generalize and to gain recall. Therefore, as further validation, a second experiment is carried out, where the systems have to classify word pairs from a different domain than the domains in the training set. The objective is to assess the importance of the domain-aware training instances for the classification.

The K&H dataset contains only instances from three domains and is imbalanced between the number of instances across domains and relation types. Therefore, our second experiment tests each method on the BLESS dataset (Baroni and Lenci, 2011), which spans 17 topical domains and includes five relation types, the three in K&H and (a) attributes of concepts, a relation holding between nouns and adjectives, and (b) actions performed by/to concepts a relation holding between nouns and verbs. In total, the BLESS dataset contains 14400 positive instances and an equal number of negative instances. This experiment measures the generalizability of each system and tests the capabilities of the systems for lexical-semantic relation types other than taxonomic relations.

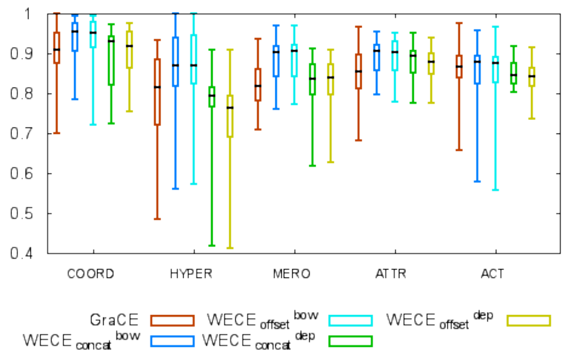

Figure 2: F1 scores distribution across domains for each proposed system and relation type over BNC corpus.

Domain-aware training instances To show the importance of the domain-aware training instances, the average results of the systems obtained for the *in-domain* setup across the BLESS dataset are compared with the average results obtained when the systems are trained *out-of-domain*. For the *out-of-domain* setup, one domain is left out from the training set and used for testing. The experiment was repeated for each domain and the average results are presented in Table 4. In this experiment, the systems are tested over the BNC corpus to show the capabilities of the systems to classify *out-of-domain* in a more reduced corpus.

Results When no examples from a domain are provided, a general significant decrease in performance is observed. The GraCE performance decreases 39.4 points in F1, while the WECE systems decrease 20.55 points in average.

The results obtained show that when the instances to be classified are less homogeneous, i.e. when the instances belong to different domains, none of the systems can achieve the level of performance reported for the in-domain setup. These were the expected results for the GraCE system due to the lexical features that it uses and which are domain dependent. However, the WECE systems are also affected by this lack of domain-aware training instances. $\text{WECE}_{\text{concat}}$ results decrease because similar embeddings are associated to similar words. When two words belong to two different topical domains, their embeddings are less similar and therefore, the SVM system cannot learn distinctive features for each lexical-semantic relation.

In-domain results per relation type In this work we are interested in creating a general approach for the classification of any lexical semantic relations instances. Figure 2 shows the box and whisker plot of the results obtained per relation type across domains in the in-domain setup over the BNC corpus.

Discussion The results confirm that the proposed systems achieve satisfactory results across all the relations, the median of the results being around 90 points in F1. The most accurate system is WECE^{bow}, which supports the assertion by Levy and Goldberg (2014a) that bag-of-word embeddings should offer superior performance to dependency-based embeddings on task involving semantic relations. Carrying out an error analysis, the lowest results of the WECE systems are obtained in the domains with the fewest training instances, making apparent that word embedding systems are dependent on the number of training instances. For these domains, GraCE achieves better results.

5 Conclusions

In this paper we have presented two systems for classifying the lexical-semantic relation of a word pair. Both are designed to address the challenge of data sparsity, i.e., classifying word pairs whose members never co-occur, in order to improve classification recall. The two main contributions are the word pair vectorial representations, one based on a graph-based corpus representation and the other one based on word embeddings. We have demonstrated that by including information about lexical and relational similarity in the word pair vectorial representation, the recall of our systems increases, overcoming the upper-bound limit of state-of-the-art systems. Furthermore, we show that our systems are able to classify target word pairs into multiple lexical semantic relation types, beyond the traditional taxonomic types. In future work, we plan to analyze the properties of the instances that can be classified with the GraCE system but not with the WECE systems.

Acknowledgments

The authors gratefully acknowledge the support of the CLARA project (EU-7FP-ITN-238405), and of the SKATER project (Ministerio de Economia y Competitividad, TIN2012-38584-C06-05).

Appendix

A Full Classifier Results

		BNC			Wikipedia		
		P	R	F	P	R	F
$PairClass$	C	84.1	3.6	6.9	92.4	9.3	16.8
	H	79.7	10.1	17.9	75.6	26.1	38.8
	M	38.6	8.5	14.0	23.9	15.5	18.8
	*	76.9	4.6	8.7	77.0	11.7	20.4
BL	C	84.4	5.6	10.4	88.8	13.8	23.9
	H	82.4	20.1	32.3	92.7	31.9	47.5
	M	69.4	12.8	21.6	77.3	14.5	24.4
	*	82.6	7.7	14.2	89.4	16.2	27.5
$GraCE$	C	90.9	43.7	59.0	94.2	78.7	85.7
	H	90.5	48.9	63.5	93.2	67.8	78.5
	M	87.5	26.3	40.4	91.8	28.7	43.7
	*	90.7	43.8	59.0	94.0	75.5	83.7
DS	C	97.2	8.0	14.8	95.5	11.5	20.5
	H	28.2	58.6	38.1	29.1	85.4	43.4
	M	8.4	36.4	13.7	8.5	48.0	14.5
	*	31.6	15.7	21.0	32.8	22.6	26.8
DS^{Dep}	C	82.0	2.6	5.0	84.0	5.2	9.8
	H	20.7	62.7	31.1	21.8	80.7	34.4
	M	5.1	26.1	8.6	11.3	43.6	17.9
	*	18.7	11.4	14.2	27.7	15.6	20.0
$WECE^{Bow}_{offset}$	C	95.9	60.4	74.1	96.6	89.7	93.0
	H	98.1	56.5	71.7	98.9	85.3	91.6
	M	88.6	38.3	53.5	90.8	51.2	65.4
	*	96.0	59.1	73.1	96.8	87.7	92.0
$WECE^{Bow}_{concat}$	C	98.2	60.6	74.9	98.5	89.8	93.9
	H	96.0	60.1	73.9	97.1	91.3	94.1
	M	81.1	45.9	58.7	77.9	68.6	72.9
	*	97.4	60.0	74.2	97.6	89.3	93.2
$WECE^{Dep}_{offset}$	C	87.0	66.5	75.4	95.1	88.1	91.5
	H	96.6	51.9	67.5	98.1	84.3	90.7
	M	83.1	26.4	40.1	88.2	44.7	59.3
	*	87.9	63.1	73.5	95.4	86.1	90.5
$WECE^{Dep}_{concat}$	C	94.0	66.7	78.0	98.0	89.2	93.4
	H	93.1	60.2	73.1	95.5	90.3	92.8
	M	67.0	35.8	46.7	69.5	62.0	65.6
	*	93.1	64.7	76.4	96.7	88.4	92.4

Table 5: Detailed results for each relation tested, coordination (C), hypernymy (H) and meronymy (M), and the aggregated results (*) obtained with K&H dataset over BNC and Wikipedia.

References

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

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of ACL*.

Matthew Berland and Eugene Charniak. 1999. Finding parts in very large corpora. In *Proceedings of ACL*, pages 57–64.

Scott Cederberg and Dominic Widdows. 2003. Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proceedings of CoNLL*, pages 111–118.

Dmitry Davidov and Ari Rappoport. 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *Proceedings of COLING-ACL*, pages 297–304.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of ACL*, volume 1.

Roxana Girju, Adriana Badulescu, and Dan I. Moldovan. 2006. Automatic discovery of part-whole relations. *Computational Linguistics*, 32(1):83–135.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. *SIGKDD Explorations Newsletter*, 11(1):10–18.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING*, pages 539–545.

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*, pages 33–38.

Amaç Herdadelen and Marco Baroni. 2009. Bagpack: A general framework to represent semantic relations. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 33–40.

David Jurgens, Peter D Turney, Saif M Mohammad, and Keith J Holyoak. 2012. Semeval-2012 Task 2: Measuring degrees of relational similarity. In *Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 356–364.

Zornitsa Kozareva and Eduard H. Hovy. 2010. A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of EMNLP*, pages 1110–1118.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Proceedings of ACL*.

Omer Levy and Yoav Goldberg. 2014b. Linguistic regularities in sparse and explicit word representations. In *Proceedings of CoNLL*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Tomas Mikolov, Anoop Deoras, Daniel Povey, Lukas Burget, and Jan Cernocky. 2011. Strategies for training large scale neural network language models. In *Proceedings of the 2011 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 196–201.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of HLT-NAACL*, pages 746–751.

George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41.

Einat Minkov and William W. Cohen. 2012. Graph based similarity measures for synonym extraction from parsed text. In *Proceedings of the Workshop on Graph-based Methods for Natural Language Processing*, pages 20–24.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-CONLL*, pages 1003–1011.

Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In *Proceedings of ACL*, pages 1318–1327.

John C. Platt. 1999. Fast training of support vector machines using sequential minimal optimization. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in kernel methods*, pages 185–208. MIT Press, Cambridge, MA, USA.

Diarmuid O Séaghdha and Ann Copestake. 2009. Using lexical and relational similarity to classify semantic relations. In *Proceedings of EACL*, pages 621–629.

Isabel Segura-Bedmar, Paloma Martınez, and Marıa Herrero-Zazo. 2013. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts. In *Proceedings of the 7th International Workshop on Semantic Evaluation*.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. In *Proceedings of NIPS*.

Peter D. Turney. 2006a. Expressing implicit semantic relations without supervision. In *Proceedings of COLING-ACL*, pages 313–320.

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

Peter D Turney. 2008a. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research (JAIR)*, 33:615–655.

Peter D. Turney. 2008b. A uniform approach to analogies, synonyms, antonyms, and associations. In *Proceedings of COLING*, pages 905–912.

Dominic Widdows and Beate Dorow. 2002. A graph model for unsupervised lexical acquisition. In *Proceedings of COLING*.

Alisa Zhila, Wen-tau Yih, Christopher Meek, Geoffrey Zweig, and Tomas Mikolov. 2013. Combining heterogeneous models for measuring relational similarity. In *Proceedings of HLT-NAACL*, pages 1000–1009.