Relation Induction in Word Embeddings Revisited

Zied Bouraoui
CRIL CNRS UMR 8188
Artois University
F-62300 Lens, France
bouraoui@cril.fr

Shoaib Jameel
School of Computing
Medway Campus
University of Kent, UK
M.S.Jameel@kent.ac.uk

Steven Schockaert
School of Computer Science
and Informatics
Cardiff University, UK
schockaerts1@cardiff.ac.uk

Abstract

Given a set of instances of some relation, the relation induction task is to predict which other word pairs are likely to be related in the same way. While it is natural to use word embeddings for this task, standard approaches based on vector translations turn out to perform poorly. To address this issue, we propose two probabilistic relation induction models. The first model is based on translations, but uses Gaussians to explicitly model the variability of these translations and to encode soft constraints on the source and target words that may be chosen. In the second model, we use Bayesian linear regression to encode the assumption that there is a linear relationship between the vector representations of related words, which is considerably weaker than the assumption underlying translation based models.

1 Introduction

It has been observed that many lexical relationships can be modelled as vector translations in a word embedding space (Mikolov et al., 2013b; Pennington et al., 2014). Even though this remarkable property is now well-established, and is commonly used as a basis for evaluating word embedding models, its potential for knowledge base completion has only been explored in a preliminary way. In this paper, we are particularly interested in the following relation induction task, which was proposed in (Vylomova et al., 2016): given a set $\{(s_1, t_1), ..., (s_n, t_n)\}$ of word pairs that are related in a given way, identify new word pairs (s, t) that are likely to be related in the same way. We will refer to s and t as the source and target word respectively; we will write $\mathbf{w}$ for the vector representation of the word w.

One natural approach is to model the considered relation using the average translation vector $\mathbf{r} = \frac{1}{n}\sum_i(\mathbf{t_i} - \mathbf{s_i})$, and to accept (s, t) as a likely instance if $\cos(\mathbf{s} + \mathbf{r}, \mathbf{t})$ is sufficiently high. While this approach was found in (Drozd et al., 2016) to work well for analogy completion, for relation induction it typically leads to too many false positives. This is illustrated in Table 1 for the case where $\mathbf{r}$ is constructed from the instances of the *capital of* relation of the BATS dataset[1]. As can be seen in the table, most of the top-ranked pairs are actually incorrect. In practice, this problem is further exacerbated by the dramatic class imbalance: for typical vocabulary sizes there are tens (or hundreds) of billions of incorrect pairs, compared to only a few hundred correct instances. While correct instances may, on average, get a higher score than incorrect instances, as a result of this imbalance there will still be many incorrect instances that receive a very high score.

Another problem relates to the use of the cosine similarity, which treats each dimension in the same way when comparing the vectors $\mathbf{r}$ and $\mathbf{t} - \mathbf{s}$. In practice, however, some dimensions of the word embedding may correspond to features of meaning that are irrelevant for the considered relationship. When we are only given one example (s, t) of a correct instance, as in the most common version of the analogy completion task, the cosine similarity is a suitable choice as we cannot determine which dimensions are most relevant. For relation induction, however, we can use the empirical variance of the translation vectors $\mathbf{t_i} - \mathbf{s_i}$ to make a more informed choice.

[1] See section 4 for more details about the experimental set-up, including how negative test examples were chosen.

word pair	cos	word pair	cos
(horse, horses)	0.84	*(baghdad,iraq)*	0.64
(boy, girl)	0.79	(aware, unaware)	0.64
(madrid, spain)	0.73	*(moscow, russia)*	0.63
(london, england)	0.69	*(berlin, germany)*	0.63
(spain, madrid)	0.68	(look, looking)	0.61
(walk, walks)	0.65	(moscow,germany)	0.59

Table 1: Cosine scores for the average translation model applied to the *capital of* relation; correct instances are shown in italics.

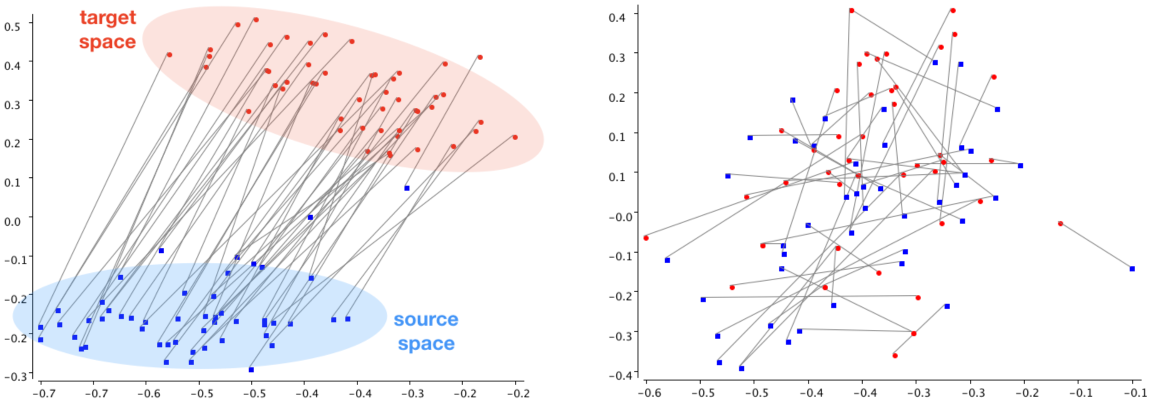
In this paper, we propose a probabilistic relation induction model that addresses both problems. First, to reduce the number of spurious instances that are detected, our model learns a soft constraint on which words are likely to occur as source and target words. Second, we use a (Bayesian estimation of a) Gaussian distribution over translation vectors to encode which features of word meaning are most important for the considered relation. We also consider a variant that is not based on translations and merely assumes that there is a linear mapping between source and target words, which we formalize using Bayesian linear regression. Being more general than the translation based approach, this model can potentially be more faithful, but it needs a larger number of training instances to be effective.

2 Related Work

Predicting Relations. At least three different types of approaches have been studied for predicting relations that are missing from a given knowledge base. First, there is a large body of work on relation extraction from text, for instance by using the known instances as a form of distant supervision (Mintz et al., 2009; Riedel et al., 2010; Surdeanu et al., 2012). The second type of approach relies on modeling statistical dependencies among the known instances of the considered relations, for instance by learning latent representations (Kok and Domingos, 2007; Speer et al., 2008; Nickel et al., 2012; Riedel et al., 2013; Bordes et al., 2013; Wang et al., 2014; Yang et al., 2015), or probabilistic rules (Schoenmackers et al., 2010; Lao et al., 2011; Wang et al., 2015). The third type of approach, which is the focus of this paper and is reviewed in more detail below, relies on vector space representations. A standard approach is to model relations as translations in the vector space (Mikolov et al., 2013b), although various other approaches have also been investigated (Weeds et al., 2014).

These three types of methods are highly complementary. While relation extraction methods can predict very fine-grained relations, they require that at least one sentence in the corpus states the relation explicitly. Statistical methods can predict relations even without access to a text corpus, but they are limited to predicting what can plausibly derived from what is already known. From a knowledge base completion point of view, the main appeal of word embeddings is that they may be able to reveal commonsense relationships which are rarely stated explicitly in text.

Modeling Relations in a Vector Space. It has been shown that word embeddings can be used to complete analogy questions of the form *a*:*b*::*c*:?, asking for a word that relates to c in the same way that b relates to a (e.g. *france*:*wine*::*germany*:?), by predicting the word w that maximizes $\cos(\mathbf{b} - \mathbf{a} + \mathbf{c}, \mathbf{w})$ (Mikolov et al., 2013b; Pennington et al., 2014). Similarly, several types of interpretable features can be modeled as directions in word embeddings. For example, in (Rothe and Schütze, 2016), it was shown that word embeddings can be decomposed in orthogonal subspaces that capture particular semantic properties, including a one-dimensional subspace (i.e. a direction) that encodes polarity. Along similar lines, in (Kim and de Marneffe, 2013) it was found that the direction defined by a word and its antonym (e.g. "good" and "bad") can be used to derive adjectival scales (e.g. bad $<$ okay $<$ good $<$ excellent). In (Gupta et al., 2015), it was shown that many types of numerical attributes can be predicted from word embeddings (e.g. GDP, fertility rate and CO_2 emissions of a country) using linear regression, again supporting the view that directions can model meaningful relations. Finally, in (Derrac and Schockaert, 2015) an unsupervised method was proposed to decompose domain-specific vector spaces into interpretable directions. For instance, in a space of movies, directions modeling terms such as "scary", "romantic" or "hilarious" were found.



(a) Superlative degree relation.

(b) Meronymy relation.

Figure 1: Superlative degree and Meronymy relations.

Several authors have focused on extracting hypernym relations from word embeddings. To decide whether a word h is a hypernym of w, in (Baroni et al., 2012) it is proposed to use an SVM with a polynomial kernel, using the concatenation of $\mathbf{h}$ and $\mathbf{w}$ as feature vector. In (Roller et al., 2014) it was shown that vector differences can lead to good results with a linear SVM, provided that the vectors are normalized, and that the squared differences of each coordinate are added as additional features. Intuitively, this allows the SVM classifier to express that h and w need to be different in particular aspects (using the vector differences) but similar in other aspects (using the squared differences). Some authors have also proposed to identify hypernyms by using word embedding models that represent words as regions or densities (Erk, 2009; Vilnis and McCallum, 2015; Jameel and Schockaert, 2017).

Beyond hypernyms, most work has focused on completing analogies. The problem of relation induction was studied in (Vylomova et al., 2016), where a linear SVM trained on vector differences was used. While strong results were obtained for several relations in a controlled setting (e.g. predicting which among a given set of relations a word pair belongs to), many false positives were obtained when random word pairs were added as negative examples. A variant of the relation induction problem was also studied in (Drozd et al., 2016), where the focus was on predicting the target word t given a valid source word s (as in analogy completion), given a set of training instances (as in relation induction). Two strong baselines were introduced in that paper, which will be explained in Section 4.1.

3 Modeling Relations

In this section, we propose two models for relation induction. We assume that we are given a set of pairs $\{(s_1, t_1), ..., (s_n, t_n)\}$ as training data, and we need to determine whether a given pair of words (s, t) are related in the same way.

3.1 Translation Model

Intuition. The source words $s_1, ..., s_n$ typically belong to some semantic or syntactic category, and their representations can often be expected to approximately belong to some relatively low-dimensional subspace of the word embedding. This is illustrated in Figure 1a for the ‘superlative degree’ relation from the BATS dataset[2] (see Section 4). Irrespective of the translation $\mathbf{t} - \mathbf{s}$, if (s, t) is a valid relation instance, we would expect $\mathbf{s}$ to be approximately in the same subspace as $\mathbf{s_1}, ..., \mathbf{s_n}$, and similar for the target words. Imposing this condition will intuitively allow us to ensure that only pairs where s and t are of the correct type are considered. As we will see, this will substantially reduce the number of false positives that are predicted by the model.

In Figure 1a we can see that there is no single translation vector that perfectly models the relation, although the translation vectors $\mathbf{t_i} - \mathbf{s_i}$ are all rather similar. This can be naturally modeled by considering

[2] In particular, the figure shows the two first principle components of the set $\{\mathbf{s_1}, ..., \mathbf{s_n}, \mathbf{t_1}, ..., \mathbf{t_n}\}$. As word vectors, we used the Skip-gram embedding that was learned from the Google news corpus.

a probability distribution over vector translations. In the example, this distribution would have a small variance along directions which are orthogonal to the average translation vector (as most vectors are almost parallel), but a larger variance along the direction of the average translation vector itself (as the translation vectors have varying lengths).

Putting everything together, we want to accept (s, t) as a valid instance if (i) $\mathbf{s}$ and $\mathbf{t}$ are sufficiently similar to the vector representations of the given source and target words, and (ii) the translation $\mathbf{t} - \mathbf{s}$ has a sufficiently high probability.

Model description. Let us write δ_s and δ_t be the event that the source word s and target word t are of the correct type, and let θ_{st} be the event that s and t are in the considered relation. Note that θ_{st} entails δ_s and δ_t, hence we have:

$$P(\theta_{st}|\mathbf{s}, \mathbf{t}) = P(\delta_s|\mathbf{s}) \cdot P(\delta_t|\mathbf{t}) \cdot P(\theta_{st}|\mathbf{s}, \mathbf{t}, \delta_s, \delta_t)$$

By making the core assumption that the relation can be modelled in terms of vector translations, together we Bayes' rule, we obtain:

$$P(\theta_{st}|\mathbf{s}, \mathbf{t}) \propto \frac{f(\mathbf{s}|\delta_s)}{f(\mathbf{s})} \cdot \frac{f(\mathbf{t}|\delta_t)}{f(\mathbf{t})} \cdot \frac{f(\mathbf{t} - \mathbf{s}|\theta_{st})}{f(\mathbf{t} - \mathbf{s}|\delta_s, \delta_t)}$$

We now discuss how the densities occurring in this latter expression can be estimated. First, $f(\mathbf{s}|\delta_s)$ models the density of words that can appear as source words in instances of the considered relation (i.e. the words which are intuitively of the correct type). We make the simplifying assumption that the vector representations of these words follow a Gaussian distribution. Note, however, that the number of available training instances n is typically small, and in particular smaller than the number of dimensions. In such cases, the covariance matrix cannot be reliably estimated, and we have to impose strong regularity assumptions. A common approach, which we will adopt, is to only allow diagonal covariance matrices. In this case, we have

$$f(\mathbf{s}|\delta_s) = \prod_{i=1}^{m} f(x_i^s|\delta_s)$$

where m is the number of dimensions in the word embedding, and we write x_i^s for the i^{th} coordinate of $\mathbf{s}$. The density $f(x_i^s|\delta_s)$ is then a univariate Gaussian distribution with an unknown mean and variance. Given the typically small number of training examples, a maximum likelihood estimation of the parameters would lead to a form of over-fitting. To address this, we use a Bayesian approach to estimate $f(x_i^s|\delta_s)$ as follows:

$$\int G(x_i^s; \mu, \sigma^2) \mathit{NI}\chi^2(\mu, \sigma^2|\mu_0, \kappa_0, \nu_0, \sigma_0^2) d\mu d\sigma$$

where G represents the Gaussian distribution and $\mathit{NI}\chi^2$ is the normal inverse χ^2 distribution. Note that in the standard relation induction setting, we have no prior information about μ and σ^2. The parameters $\mu_0, \kappa_0, \nu_0, \sigma_0^2$ can then be chosen such that they correspond to a flat prior; we refer to (Murphy, 2007) for details. For this choice, it can be shown that the integral evaluates to:

$$f(x_i^s|\delta_s) = t_{n-1}\Big(\overline{x_i}, \frac{(n+1)\sum_{j=1}^{n}(x_i^{s_j} - \overline{x_i})^2}{n(n-1)}\Big)$$

where $\overline{x_i} = \frac{1}{n}\sum_{j=1}^{n} x_i^{s_j}$ and t_{n-1} is the Student t-distribution with $n-1$ degrees of freedom. The density $f(\mathbf{t}|\delta_t)$ is evaluated in the same way. To evaluate $f(\mathbf{s})$ and $f(\mathbf{t})$, we make the simplifying assumption that the overall distribution of word vectors also follows a Gaussian distribution. Given the typical vocabulary sizes, we can reliably use the sample mean and covariance matrix as estimations of the parameters of this Gaussian.

The density $f(\mathbf{t}-\mathbf{s}|\theta_{st})$ is estimated similarly to $f(x_i^s|\delta_s)$, which corresponds to an assumption that the translations $\mathbf{t}-\mathbf{s}$ also follow a Gaussian distribution. In particular, we estimate $f(\mathbf{t}-\mathbf{s}|\theta_{st})$ as $\prod_{i=1}^{m} f(x_i^s - x_i^t|\theta_{st})$, where x_i^s is again the i^{th} coordinate of $\mathbf{s}$ and similar for x_i^t. Each univariate Gaussian $f(x_i^s -$

$x_i^t|\theta_{st})$ is then again estimated using the t-distribution, from the set of data points $\{x_i^{s_1} - x_i^{t_1}, ..., x_i^{s_n} - x_i^{t_n}\}$. We similarly estimate $f(\mathbf{t} - \mathbf{s}|\delta_s, \delta_t)$ as $\prod_{i=1}^m f(x_i^s - x_i^t|\delta_s, \delta_t)$. The mean of $f(x_i^s - x_i^t|\delta_s, \delta_t)$ is the same as the mean of $f(x_i^s - x_i^t|\theta_{st})$, but the variance is estimated from the differences $x_i^{s_l} - x_i^{t_k}$ corresponding to n randomly sampled pairs (s_l, t_k), where s_l and t_k are respectively sampled from the source and target words occurring in the training examples. Note that if the assumption that the considered relation corresponds to a translation is wrong, we can expect the variance of $f(x_i^s - x_i^t|\theta_{st})$ and $f(x_i^s - x_i^t|\delta_s, \delta_t)$ to be similar, in which case the last factor in the evaluation of $P(\theta_{st}|\mathbf{s}, \mathbf{t})$ will be approximately 1. In other words, the model implicitly takes into account how much the translation assumption appears to be satisfied.

3.2 Regression Model

Intuition. While there are several relations that can be approximately modelled as vector translations, there are many other relations for which this is not the case. To illustrate this, Figure 1b has been constructed in the same way as Figure 1a, but using the instances of the meronymy relation from the DiffVec dataset (Gladkova et al., 2016). This figure clearly suggests that we cannot expect an approach that models meronymy in terms of translations to perform well (for this word embedding). As an alternative, in this section we propose a model which weakens the translation assumption, and merely assumes that there is a linear mapping from source to target words. We can expect that the resulting model should perform well for a broader set of relations; for example, as we will see in Section 4, the meronymy relation can be modelled rather accurately in this way. The most important drawback of this approach is that we need more training examples to reliably estimate a linear mapping than to estimate a translation. In fact, while a translation can be estimated from a single example, we can only learn a linear mapping if the number of training examples is higher than the number of dimensions. We address this issue by reducing the number of dimensions of the source space, based on the available number of training examples.

Model description. We now estimate the probability that (s, t) is a valid instance of the considered relation as follows:

$$\begin{aligned} P(\theta_{st}|\mathbf{s}, \mathbf{t}) &\propto \frac{f(\mathbf{s}|\delta_s)}{f(\mathbf{s})} \cdot \frac{f(\mathbf{t}|\delta_t)}{f(\mathbf{t})} \cdot \frac{f(\mathbf{t}|\mathbf{s}, \theta_{st})}{f(\mathbf{t}|\mathbf{s}, \delta_s, \delta_t)} \\ &= \frac{f(\mathbf{s}|\delta_s)}{f(\mathbf{s})} \cdot \frac{f(\mathbf{t}|\delta_t)}{f(\mathbf{t})} \cdot \frac{f(\mathbf{t}|\mathbf{s}, \theta_{st})}{f(\mathbf{t}|\delta_t)} \\ &= \frac{f(\mathbf{s}|\delta_s)}{f(\mathbf{s})} \cdot \frac{f(\mathbf{t}|\mathbf{s}, \theta_{st})}{f(\mathbf{t})} \end{aligned}$$

The densities $f(\mathbf{s}|\delta_s)$, $f(\mathbf{s})$ and $f(\mathbf{t})$ are estimated as before. We estimate $f(\mathbf{t}|\mathbf{s}, \theta_{st})$ as $\prod_{i=1}^m f(x_i^t|\mathbf{s}, \theta_{st})$, where x_i^t is again the i^{th} coordinate of $\mathbf{t}$. Each univariate density $f(x_i^t|\mathbf{s}, \theta_{st})$ is estimated using a Bayesian linear regression model that predicts the possible representations of the target word from $\mathbf{s}$. However, this is only feasible if $\mathbf{s}$ has at most $n - 2$ coordinates. Therefore, we use a low-rank approximation of the source word representations, as follows.

Let A be a matrix whose rows are the vectors $\mathbf{s_1}, ..., \mathbf{s_n}$ and let $A = U\Sigma V^T$ be the SVD decomposition of A. Let $\mathbf{v_1}, ..., \mathbf{v_k}$ be the first k row vectors of V, for some $k < n - 1$. For a given vector $\mathbf{p}$, we can think of $\mathbf{p}^S = (\mathbf{p} \cdot \mathbf{v_1}, ..., \mathbf{p} \cdot \mathbf{v_k})$ as the representation of $\mathbf{p}$ in the source subspace. Given that we typically need far fewer dimensions to represent the source space than the total number of dimensions in the word embedding, we should be able to predict the target word from $\mathbf{s}^S$, even for relatively small values of k. In any case, the choice of k represents a trade-off: the lower the value of k, the better we can characterize the uncertainty underlying our predictions, but the less information we have for making predictions. In the experiments, we have used $k = \frac{n-1}{2}$. We estimate $f(x_i^t|\mathbf{s}, \theta_{st})$ as follows:

$$\begin{aligned} \int & G(x_i^t; \mathbf{s}^*\beta, \sigma^2) \cdot \\ & G(\beta; (X^T X)^{-1} X^T \mathbf{b^i}, (X^T X)^{-1}\sigma^2) \cdot \\ & NI\chi^2(\sigma^2|\nu_0, \sigma_0^2) d\beta d\sigma \end{aligned}$$

where $\mathbf{b^i} = (x_i^{t_1}, ..., x_i^{t_n})$, X is composed of the first k columns of $U\Sigma$ (with U and Σ the matrices from the SVD decomposition of A) with an additional 1 appended at the end of each row for the bias term, and $\mathbf{s}^*$ is the vector $\mathbf{s}^S$ with an additional 1 appended. Assuming a flat prior on the residual variance σ^2, the parameters ν_0 and σ_0^2 can be estimated from the training data as:

$$\nu_0 = n - k - 1$$
$$\sigma_0^2 = \frac{1}{n-k-1}(\mathbf{b^i} - X\hat{\beta})^T(\mathbf{b^i} - X\hat{\beta})$$

with $\hat{\beta}$ the least squares solution.

4 Evaluation

In this section, we experimentally compare the two proposed models with a number of baseline methods from the literature. The relations we consider are taken from three standard benchmark datasets, each containing a mixture of syntactic and semantic relationships: (i) the Google Analogy Test Set (Google), which contains 14 types of relations with a varying number of instances per relation (Mikolov et al., 2013a), (ii) the Bigger Analogy Test Set (BATS), which contains 40 relations with 50 instances per relation (Gladkova et al., 2016), and (iii) the DiffVec Test Set (DV), which contains 36 relations with a varying number of instances per relation (Vylomova et al., 2016). We report results for two embeddings that have been learned using Skip-gram, one from the Wikipedia dump of 2 November 2015 (SG-Wiki) and one from a 100B words Google News data set[3] (SG-GN). We also use two embeddings that have been learned with GloVe, one from the same english Wikipedia dump (GloVe-Wiki) and one from the 840B words Common Crawl data set[4] (GloVe-CC).

For relations with at least 10 instances, we use 10-fold cross validation, whereas for relations with less than 10 instances, we use a leave-one-out evaluation. Note that the considered datasets only contain positive examples. To generate negative test examples, we use four strategies. First, for each pair (s, t) in the test fold, we add (t, s) as a negative example. Second, for each source word s in the test fold, we randomly sample two target words from the test fold (provided that the test fold contains enough pairs), which do not occur together with s, and for each such target word t, we add (s, t) as a negative example. Third, for each positive example, we randomly select a pair from the other relations. Finally, for each positive example, we generate a random word pair from the words available in the dataset. This ensures that the evaluation involves negative examples that consist of related words, as well as negative examples that consist of unrelated words.

If we consider the task as a classification task, i.e. deciding for an unseen pair (s, t) whether it has the considered relation, we need to select a threshold, as the considered methods only produce a confidence score. To choose this threshold, we randomly select 10% of the 9 training folds as validation data. In the results below, we separately report precision, recall and F1. We can also evaluate this task as a ranking problem, where we merely evaluate to what extent each method assigns the highest score to the correct pairs. In that case, we use mean average precision (MAP).

4.1 Baselines

The first baseline we consider is the 3CosAvg (or 3CA) method proposed in (Drozd et al., 2016), which essentially treats the relation induction problem like an analogy completion problem, where we use the average translation vector across all pairs (s_i, t_i) from the training data. In particular, this method assigns the following score to the test pair (s, t):

$$\textit{score}_{\textit{3CA}}(t, s) = \cos\left(\mathbf{s} + \frac{\sum_i \mathbf{t_i} - \mathbf{s_i}}{n}, \mathbf{t}\right)$$

Another method proposed in (Drozd et al., 2016), called LRCos (or LRC), is based on the assumption that (s, t) is likely correct if $\cos(\mathbf{s}, \mathbf{t})$ is high and t is of the correct type, where a logistic regression

[3] https://code.google.com/archive/p/word2vec/

[4] https://nlp.stanford.edu/projects/glove/

Table 2: Results of the relation induction experiments (macro-averages).

		SG-Wiki			GloVe-Wiki			SG-GN			GloVe-CC		
		Google	BATS	DV	Google	BATS	DV	Google	BATS	DV	Google	BATS	DV
3CA	Pr	0.651	0.433	0.210	0.510	0.330	0.205	0.466	0.377	0.214	0.642	0.428	0.225
3CA	Rec	0.602	0.620	0.491	0.541	0.635	0.478	0.592	0.619	0.507	0.665	0.609	0.506
3CA	F1	0.626	0.510	0.294	0.525	0.434	0.287	0.522	0.469	0.301	0.653	0.503	0.311
3CA	MAP	0.736	0.463	0.274	0.663	0.376	0.261	0.615	0.424	0.273	0.741	0.480	0.288
LRC	Pr	0.516	0.475	0.374	0.366	0.256	0.166	0.488	0.486	0.389	0.427	0.383	0.257
LRC	Rec	0.646	0.672	0.527	0.527	0.577	0.439	0.670	0.646	0.570	0.659	0.596	0.474
LRC	F1	0.573	0.557	0.437	0.432	0.355	0.241	0.565	0.555	0.462	0.518	0.466	0.333
LRC	MAP	0.710	0.580	0.519	0.508	0.322	0.265	0.713	0.614	0.545	0.628	0.481	0.389
SVM	Pr	0.407	0.336	0.198	0.383	0.365	0.215	0.464	0.398	0.276	0.407	0.381	0.225
SVM	Rec	0.680	0.417	0.412	0.628	0.461	0.376	0.646	0.531	0.384	0.671	0.501	0.408
SVM	F1	0.509	0.372	0.267	0.476	0.408	0.274	0.540	0.455	0.321	0.507	0.433	0.290
SVM	MAP	0.494	0.366	0.283	0.502	0.404	0.298	0.611	0.467	0.366	0.502	0.425	0.296
Trans	Pr	0.794	0.627	0.449	0.635	0.445	0.284	0.741	0.660	0.498	0.744	0.571	0.378
Trans	Rec	0.649	0.708	0.563	0.618	0.620	0.446	0.771	0.705	0.604	0.713	0.689	0.552
Trans	F1	0.714	0.665	0.500	0.626	0.518	0.347	0.756	0.682	0.546	0.728	0.624	0.449
Trans	MAP	0.906	0.729	0.596	0.791	0.541	0.387	0.890	0.773	0.635	0.898	0.678	0.520
Regr	Pr	0.668	0.474	0.410	0.536	0.281	0.259	0.627	0.476	0.469	0.613	0.401	0.357
Regr	Rec	0.603	0.470	0.471	0.580	0.403	0.422	0.665	0.449	0.537	0.646	0.439	0.467
Regr	F1	0.634	0.472	0.439	0.557	0.331	0.321	0.646	0.462	0.501	0.629	0.419	0.404
Regr	MAP	0.834	0.618	0.570	0.741	0.434	0.381	0.801	0.639	0.621	0.793	0.549	0.506

classifier was trained on the target words $\{t_1, ..., t_n\}$ to predict the probability that t is a valid 'target word'. To adapt this method to our setting, we also need to consider the probability that s is a valid 'source word' (which is not needed in the analogy completion setting considered in (Drozd et al., 2016), since s is always given as a valid source word). To allow for a more direct comparison with our methods, instead of using a logistic regression classifier, we will use our Bayesian estimation for the probability that s and t are of the correct type. In particular, we use the score $\mathit{score}_{\mathit{LRC}}(t, s)$ defined as follows:

$$\frac{P(\mathbf{s}|\delta_s)}{P(\mathbf{s})} \cdot \frac{P(\mathbf{t}|\delta_t)}{P(\mathbf{t})} \cdot \cos(\mathbf{s}, \mathbf{t})$$

As our final baseline, we train a linear SVM classifier using the training pairs $(s_1, t_1), ..., (s_n, t_n)$ as positive examples. Following (Vylomova et al., 2016), we use negative examples of the form (t_i, s_i), obtained by swapping the position of source and target word, as well as negative examples of the form (s_i, t_j), obtained by swapping t_i by the target word of another instance (while ensuring that (s_i, t_j) does not appear in the training data as well). Finally, we also add n random word pairs as negative examples. The C parameter (i.e. the cost parameter for mislabeled examples) of the SVM is tuned for each relation separately (choosing values from $\{0.01, 0.1, 1, 10, 100\}$), using the same validation data that is used for selecting the thresholds in the other models. To address class imbalance, negative examples were weighted by the ratio of positive to negative examples.

4.2 Results

The results are summarized in Table 2. As can be observed, our translation model consistently outperforms all other methods in both MAP and F1 score. Moreover, the regression model consistently outperforms the baselines in terms of MAP score, and outperforms the baselines for the Google and DV test sets in terms of F1 score (with the exception of the Glove-CC embedding, where 3CA has a better F1 score for the Google test set). The performance of the baselines varies, with 3CA generally performing best for the Google test set and LRC generally performing best for DiffVec.

To compare the performance of the methods across different types of relations, Table 3 contains the MAP scores for a number of selected relations from the DiffVec and BATS test sets, for the SG-GN word embedding. For the BATS dataset, the translation model consistently outperforms the baseline across all relations (including the relations that are not shown in the table). In the case of DiffVec there are a few exceptions, as can be seen in Table 3, but in such cases the differences with the translation model are small. The regression model also outperforms the baselines in most cases, but there are a few

Table 3: MAP scores for selected relations (based on the SG-GN word embedding).

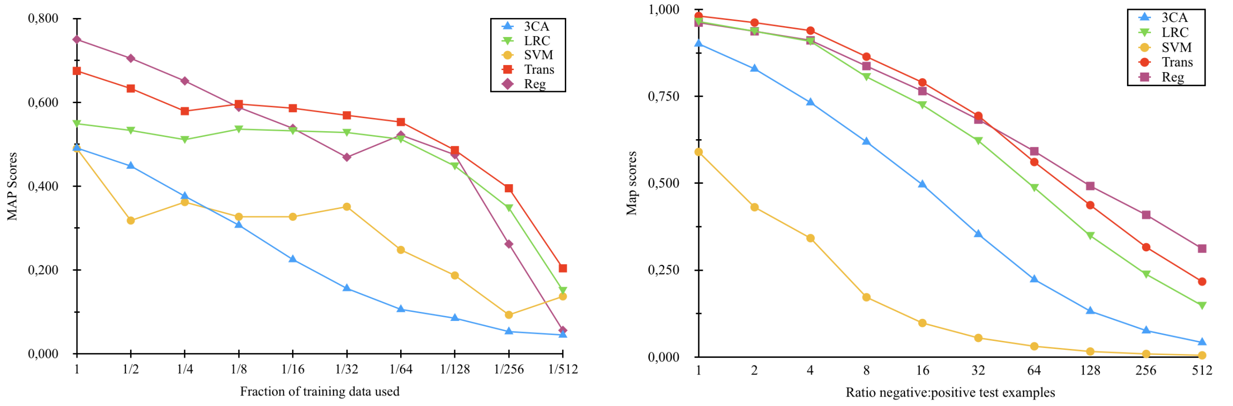
DiffVec	3CA	LRC	SVM	Trans	Regr
Object:State	0.11	**0.59**	0.27	0.57	0.50
CompensatoryAction	0.08	0.58	0.41	0.62	**0.66**
IntendedAction	0.14	0.53	0.35	**0.62**	0.58
Prevention	0.17	0.66	0.55	**0.71**	0.62
Collective noun	0.37	0.58	0.39	0.56	**0.69**
Event	0.61	0.72	0.40	0.74	**0.94**
Hyper	0.48	0.55	0.39	0.75	**0.91**
Lvc	0.11	0.71	**0.77**	0.74	0.22
Mero	0.49	0.55	0.40	0.67	**0.83**
Prefix re	0.40	0.49	0.30	**0.72**	0.68
Expression	0.14	0.82	0.51	0.81	**0.82**
Knowledge	0.14	0.69	0.51	**0.72**	0.70
Plan	0.08	0.55	0.29	0.57	**0.62**
Representation	0.09	**0.50**	0.40	0.49	0.39
Contiguity	0.11	0.53	0.30	**0.63**	0.61
Sign:Significant	0.08	0.38	0.30	**0.39**	0.38
Loc:Action/Activity	0.22	0.73	0.51	**0.75**	0.72
Loc:Process/Product	0.13	0.41	0.57	0.48	**0.66**
Verb 3rd	0.96	0.61	0.40	**0.98**	0.96
Verb Past	0.93	0.64	0.32	**0.99**	0.90

BATS	3CA	LRC	SVM	Trans	Regr
Regular plurals	0.83	0.59	0.40	**0.88**	0.79
Comparative degree	0.93	0.65	0.47	**0.96**	0.88
Superlative degree	0.87	0.71	0.61	**0.93**	0.87
Infinitive: past	0.78	0.58	0.46	**0.96**	0.72
3Ps.Sg: past	0.72	0.70	0.56	**0.98**	0.95
Noun+less	0.38	0.58	0.43	0.62	**0.63**
Un+adj	0.30	0.55	0.35	**0.77**	0.69
Over+adh./Ved	0.20	0.63	0.38	0.74	**0.77**
Re+verb	0.39	0.66	0.37	**0.82**	0.74
Verb+able	0.21	0.63	0.57	0.76	**0.76**
Verb+ment	0.34	0.54	0.47	**0.78**	0.69
Hypernyms animals	0.43	0.71	0.64	**0.85**	0.28
Hypernyms misc	0.35	0.64	0.54	**0.77**	0.23
Meronyms substance	0.23	0.51	0.36	**0.61**	0.27
Synonyms intensity	0.34	0.50	0.29	**0.67**	0.23
Synonyms exact	0.28	0.45	0.26	**0.51**	0.19
Antonyms binary	0.22	0.44	0.31	**0.50**	0.24
Capitals	0.25	0.68	0.47	**0.75**	0.74
Country:language	0.19	0.64	0.52	0.66	**0.71**
Nationalities	0.22	0.77	0.60	**0.85**	0.63
Animals sounds	0.25	0.65	0.45	0.66	**0.77**
thing:color	0.44	0.77	0.66	0.76	**0.79**
Male:female	0.61	0.61	0.47	**0.84**	0.72

exceptions where it performs much worse (e.g. *Lvc* for DiffVec, and *Hypernyms-animals*, *Meronyms-substance*, *Synonyms-intensity* and *Antonyms-binary* in the case of BATS). While the regression model is outperformed by the translation model on average, there are several cases where it performs better. For relations such as *Event*, *Hyper* and *Mero* from DiffVec, where the number of examples is rather large (resp. 3583, 1173, 2825), we can see that the regression model actually substantially outperforms the translation model. The main weakness of the regression model is that it needs more training data: while a vector translation can be estimated from a single training example, learning an arbitrary linear mapping requires the number of training examples to be larger than the number of dimensions. While this can be addressed by using a low-dimensional approximation of the source word, information is lost in this way.

The impact of the amount of training data on the relative performance of the regression model is further analyzed in Figure 2a, taking the Mero relation from DiffVec as an example (for SG-GN). While the regression model performs best if all the training data is used, the translation model is less sensitive to the amount of training data, and starts outperforming the regression model if less than 1/8th of the training data is used.

We also noticed that the performance of the methods crucially depends on the number of negative test



(a) MAP scores for the Mero relation, in function of training data (as a fraction of the total amount).

(b) MAP scores for the Mero relation, in function of the ratio of negative to positive test examples.

Figure 2: MAP scores for the Mero relation, in function of training data/ratio of negative to positive test examples.

examples that are considered, even if these negative examples are randomly chosen word pairs. This is illustrated in Figure 2b, which shows the MAP scores for the Mero relation from DiffVec (for SG-GN), for different numbers of negative examples. For these experiments, as negative examples, we only considered random word pairs. The total number of such negative examples was varied from 1 times the number of positive examples, which is equal to the number of positive examples, to 512 times the number of positive examples. While the performance of each of the methods is affected by the number of such negative examples, the performance of the baseline models drops more quickly. Moreover, the regression model is more robust than the translation model, and starts outperforming it if the ratio of negative examples to positive examples is higher than 32:1.

5 Conclusions

We have proposed two probabilistic models for identifying word pairs that are in a given relation. The first model is based on the common assumption that lexical relations correspond to vector translations in a word embedding. The other model is based on linear regression, relying on the weaker assumption that there is a linear relationship between the source and target words of the considered relation. Both models implicitly factor in whether their underlying assumption is satisfied, and could thus easily be used in combination with each other, or with additional models. In our experimental evaluation, we have found both models to outperform existing approaches, with the translation model outperforming the regression model on average. However, in cases where sufficient training data is available, the regression model tends to perform better. We have also found some evidence that the regression model is better able to handle cases of extreme imbalance between positive and negative examples.

There are several interesting avenues for future work. First, a number of variants of the proposed models can be developed. For example, a model based on vector concatenations could intuitively model similar kinds of relationships as the regression model. However, in the case of vector concatenations, we can no longer use a diagonal covariance matrix, as that would mean that no interactions between source and target words are being captured. One solution could be to use a low-rank approximation of the vector concatenations and estimate full covariance matrices in a lower-dimensional space. Another interesting option to explore would be to estimate prior probabilities from coarser grained relations for which more training data is available. For example, we could learn a generic model for causal relations, and use that as a prior for the specific types of causal relationships that are considered in the DiffVec test set. It may even be useful to learn priors capturing e.g. syntactic relations, which would intuitively amount to finding a subspace of the embedding that relates to syntactic features.

Acknowledgments

This work was supported by ERC Starting Grant 637277.

References

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 European Chapter of the Association for Computational Linguistics*, pages 23–32.

A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, pages 2787–2795.

J. Derrac and S. Schockaert. 2015. Inducing semantic relations from conceptual spaces: a data-driven approach to plausible reasoning. *Artificial Intelligence*, pages 74–105.

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. 2016. Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 3519–3530.

Katrin Erk. 2009. Representing words as regions in vector space. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 57–65.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the Student Research Workshop at NAACL 2016*, pages 8–15.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. 2015. Distributional vectors encode referential attributes. In *Proc. EMNLP*, pages 12–21.

Shoaib Jameel and Steven Schockaert. 2017. Modeling context words as regions: An ordinal regression approach to word embedding. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pages 123–133.

Joo-Kyung Kim and Marie-Catherine de Marneffe. 2013. Deriving adjectival scales from continuous space word representations. In *Proc. EMNLP*, pages 1625–1630.

Stanley Kok and Pedro Domingos. 2007. Statistical predicate invention. In *Proc. ICML*, pages 433–440.

Ni Lao, Tom Mitchell, and William W Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In *Proc. EMNLP*, pages 529–539.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*.

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

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the ACL*, pages 1003–1011.

Kevin Murphy. 2007. Conjugate Bayesian analysis of the Gaussian distribution. Technical report, University of British Columbia.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2012. Factorizing YAGO: Scalable machine learning for linked data. In *Proceedings of the 21st International Conference on World Wide Web*, pages 271–280.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proc. ECML/PKDD*, pages 148–163.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proc. HLT-NAACL*, pages 74–84.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proc. COLING*, pages 1025–1036.

Sascha Rothe and Hinrich Schütze. 2016. Word embedding calculus in meaningful ultradense subspaces. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 512–517.

Stefan Schoenmackers, Jesse Davis, Oren Etzioni, and Daniel S. Weld. 2010. Learning first-order horn clauses from web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1088–1098.

Robert Speer, Catherine Havasi, and Henry Lieberman. 2008. Analogyspace: reducing the dimensionality of common sense knowledge. In *Proceedings of the 23rd AAAI Conference on Artificial intelligence*, pages 548–553.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465.

Luke Vilnis and Andrew McCallum. 2015. Word representations via gaussian embedding. In *Proceedings of the International Conference on Learning Representations*.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2016. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pages 1112–1119.

William Yang Wang, Kathryn Mazaitis, Ni Lao, and William W. Cohen. 2015. Efficient inference and learning in a large knowledge base - reasoning with extracted information using a locally groundable first-order probabilistic logic. *Machine Learning*, 100(1):101–126.

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*, pages 2249–2259.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the International Conference on Learning Representations*.