

BagPack: A general framework to represent semantic relations

Amaç Herdağdelen
CIMEC, University of Trento
Rovereto, Italy
amac@herdagdelen.com

Marco Baroni
CIMEC, University of Trento
Rovereto, Italy
marco.baroni@unitn.it

Abstract

We introduce a way to represent word pairs instantiating arbitrary semantic relations that keeps track of the contexts in which the words in the pair occur both together and independently. The resulting features are of sufficient generality to allow us, with the help of a standard supervised machine learning algorithm, to tackle a variety of unrelated semantic tasks with good results and almost no task-specific tailoring.

1 Introduction

Co-occurrence statistics extracted from corpora lead to good performance on a wide range of tasks that involve the identification of the semantic relation between two words or concepts (Sahlgren, 2006; Turney, 2006). However, the difficulty of such tasks and the fact that they are apparently unrelated has led to the development of largely *ad-hoc* solutions, tuned to specific challenges. For many practical applications, this is a drawback: Given the large number of semantic relations that might be relevant to one or the other task, we need a multi-purpose approach that, given an appropriate representation and training examples instantiating an arbitrary target relation, can automatically mine new pairs characterized by the same relation. Building on a recent proposal in this direction by Turney (2008), we propose a generic method of this sort, and we test it on a set of unrelated tasks, reporting good performance across the board with very little task-specific tweaking.

There has been much previous work on corpus-based models to extract broad classes of related words. The literature on word space models (Sahlgren, 2006) has focused on taxonomic similarity (synonyms, antonyms, co-hyponyms...) and general association (e.g., finding topically related words), exploiting the idea that taxonomically or associated words will tend to occur in similar contexts, and thus share a vector of co-occurring words. The literature on relational similarity, on the other hand, has focused on *pairs* of words, devising various methods to compare how similar the contexts in which target pairs appear are to the contexts of other pairs that instantiate a relation of interest (Turney, 2006; Pantel and Pennacchiotti, 2006). Beyond

these domains, purely corpus-based methods play an increasingly important role in modeling constraints on composition of words, in particular verbal selectional preferences – finding out that, say, children are more likely to eat than apples, whereas the latter are more likely to be eaten (Erk, 2007; Padó et al., 2007). Tasks of this sort differ from relation extraction in that we need to capture *productive* patterns: we want to find out that shabu shabu (a Japanese meat dish) is eaten whereas ink is not, even if in our corpus neither noun is attested in proximity to forms of the verb *to eat*.

Turney (2008) is the first, to the best of our knowledge, to raise the issue of a unified approach. In particular, he treats synonymy and association as special cases of relational similarity: in the same way in which we might be able to tell that hands and arms are in a part-of relation by comparing the contexts in which they co-occur to the contexts of known part-of pairs, we can guess that cars and automobiles are synonyms by comparing the contexts in which they co-occur to the contexts linking known synonym pairs.

Here, we build on Turney’s work, adding two main methodological innovations that allow us further generalization. First, merging classic approaches to taxonomic and relational similarity, we represent concept pairs by a vector that concatenates information about the contexts in which the two words occur independently, and the contexts in which they co-occur (Mirkin et al. 2006 also integrate information from the lexical patterns in which two words co-occur and similarity of the contexts in which each word occurs on its own, to improve performance in lexical entailment acquisition). Second, we represent contexts as bag of words and bigrams, rather than strings of words (“patterns”) of arbitrary length: we leave it to the machine learning algorithm to zero in on the most interesting words/bigrams.

Thanks to the concatenated vector, we can tackle tasks in which the two words are not expected to co-occur even in very large corpora (such as selectional preference). Concatenation, together with unigram/bigram representation of context, allows us to scale down the approach to smaller training corpora (Turney used a corpus of more than 50 billion words), since we do not need to see the words directly co-occurring, and the unigram/bigram dimensions of the

vectors are less sparse than dimensions based on longer strings of words. We show that our method produces reasonable results also on a corpus of 2 billion words, with many unseen pairs. Moreover, our bigram and unigram representation is general enough that we do not need to extract separate statistics nor perform *ad-hoc* feature selection for each task: we build the co-occurrence matrix once, and use the same matrix in all experiments. The bag-of-words assumption also makes for faster and more compact model building, since the number of features we extract from a context is linear in the number of words in the context, whereas it is exponential for Turney. On the other hand, our method is currently lagging behind Turney’s in terms of performance, suggesting that at least some task-specific tuning will be necessary.

Following Turney, we focus on devising a suitably general featural representation, and we see the specific machine learning algorithm employed to perform the various tasks as a parameter. Here, we use Support Vector Machines since they are a particularly effective general-purpose method. In terms of empirical evaluation of the model, besides experimenting with the “classic” SAT and TOEFL datasets, we show how our algorithm can tackle the selectional preference task proposed in Padó (2007) – a regression task – and we introduce to the corpus-based semantics community a challenge from the ConceptNet repository of common-sense knowledge (extending such repository by automated means is the original motivation of our project).

In the next section, we will present our proposed method along with the corpora and model parameter choices used in the implementation. In Section 3, we describe the tasks that we use to evaluate the model. Results are reported in Section 4 and we conclude in Section 5, with a brief overview of the contributions of this paper.

2 Methodology

2.1 Model

The central idea in BagPack (**B**ag-of-words representation of **P**aired concept **k**nowledge) is to construct a vector-based representation of a pair of words in such a way that the vector represents both the contexts where the two words co-occur and the contexts where the single words occur on their own. A straightforward approach is to construct three different sub-vectors, one for the first word, one for the second word, and one for the co-occurring pair. The concatenation of these three sub-vectors is the final vector that represents the pair.

This approach provides us a graceful fall back mechanism in case of data scarcity. Even if the two words are not observed co-occurring in the corpus – no syntagmatic information about the pair –, the corresponding vector will still represent the individual contexts where the words are observed on their own. Our hypothesis (and hope) is that this information will be representative of the semantic relation between the pair, in the

sense that, given pairs characterized by same relation, there should be paradigmatic similarity across the first, resp. second elements of the pairs (e.g., if the relation is between professionals and the typical tool of their trade, it is reasonable to expect that both professionals and tools will tend to share similar contexts).

Before going into further details, we need to describe what a “co-occurrence” precisely means, define the notion of context, and determine how to structure our vector. For a single word W , the following pseudo regular expression identifies an observation of *occurrence*:

$$“C W D” \quad (1)$$

where C and D can be empty strings or concatenations of up to 4 words separated by whitespace (i.e. C_1, \dots, C_i and D_1, \dots, D_j where $i, j \leq 4$). Each observation of this pattern constitutes a *single context* of W . The pattern is matched with the longest possible substring without crossing sentence boundaries.

Let (W_1, W_2) denote an ordered pair of words W_1 and W_2 . We say the two words *occur as a pair* whenever one of the following pseudo regular expressions is observed in the corpus:

$$“C W_1 D W_2 E” \quad (2)$$

$$“C W_2 D W_1 E” \quad (3)$$

where C and E can be empty strings or concatenations of up to 2 words and similarly, D can be either an empty string or concatenation of up to 5 words (i.e. $C_1, \dots, C_i, D_1, \dots, D_j$, and E_1, \dots, E_k where $i, j \leq 2$ and $k \leq 5$). Together, patterns 2 and 3 constitute the *pair context* for W_1 and W_2 . The pattern is matched with the longest possible substring while making sure that D does not contain neither W_1 nor W_2 .

The number of context words allowed before, after, and between the targets are actually model parameters but for the experiments reported in this study, we used the aforementioned values with no attempt at tuning.

The vector representing (W_1, W_2) is a concatenation $\mathbf{v}_1 \mathbf{v}_2 \mathbf{v}_{1,2}$, where, the sub-vectors \mathbf{v}_1 and \mathbf{v}_2 are constructed by using the single contexts of W_1 and W_2 correspondingly (i.e. by pattern 1) and the sub-vector $\mathbf{v}_{1,2}$ is built by using the pair contexts identified by the patterns 2 and 3. We refer to the components as *single-occurrence vectors* and *pair-occurrence vector* respectively.

The population of BagPack starts by identifying the b most frequent unigrams and the b most frequent bigrams as *basis terms*. Let T denote a basis term. For the construction of \mathbf{v}_1 , we create two features for each term T : t_{pre} corresponds to the number of observations of T in the single contexts of W_1 occurring before W_1 and t_{post} corresponds to the number of observations of T in the single occurrence of W_1 where T occurs after W_1 (i.e. number of observations of the pattern 1 where $T \in C$ and $T \in D$ correspondingly). The construction of \mathbf{v}_2 is identical except that this time the features

correspond to the number of times the basis term is observed before and after the target word W_2 in single contexts. The construction of the pair-occurrence sub-vector $\mathbf{v}_{1,2}$ proceeds in a similar fashion but in addition, we incorporate also the order of W_1 and W_2 as they co-occur in the pair context: The number of observations of the pair contexts where W_1 occurs before W_2 and T precedes (follows) the pair, are represented by feature t_{+pre} (t_{+post}). The number of cases where the basis term is in between the target words is represented by t_{+betw} . The number of cases where W_2 occurs before W_1 and T precedes the pair is represented by the feature t_{-pre} . Similarly the number of cases where T follows (is in between) the pair is represented by the feature t_{-post} (t_{-betw}).

Assume that the words "only" and "that" are our basis terms and consider the following context for the word pair ("cat", "lion"): "Lion is the only cat that lives in large social groups." The observation of the basis terms should contribute to the pair-occurrence sub-vector $\mathbf{v}_{1,2}$ and since the target words occur in reverse order, this context results in the incrementation of the features *only_betw* and *that_post* by one.

To sum up, we have $2b$ basis terms (b unigrams and b bigrams). Each of the single-occurrence sub-vectors \mathbf{v}_1 and \mathbf{v}_2 consists of $4b$ features: Each basis term gives rise to 2 features incorporating the relative position of basis term with respect to the single word. The pair-occurrence sub-vector, $\mathbf{v}_{1,2}$, consists of $12b$ features: Each basis term gives rise to 6 new features; $\times 3$ for possible relative positions of the basis term with respect to the pair and $\times 2$ for the order of the words. Importantly, the $2b$ basis terms are picked only once, and the overall co-occurrence matrix is built once and for all for *all* the tasks: unlike Turney, we do not need to go back to the corpus to pick basis terms and collect separate statistics for different tasks.

The specifics of the adaptation to each task will be detailed in Section 3. For the moment, it should suffice to note that the vectors \mathbf{v}_1 and \mathbf{v}_2 represent the contexts in which the two words occur on their own, thus encode paradigmatic information. However, $\mathbf{v}_{1,2}$ represents the contexts in which the two words co-occur, thus encode syntagmatic information.

The model training and evaluation is done in a 10-fold cross-validation setting whenever applicable. The reported performance measures are the averages over all folds and the confidence intervals are calculated by using the distribution of fold-specific results. The only exception to this setting is the SAT analogy questions task simply because we consider each question as a separate mini dataset as described in Section 3.

2.2 Source Corpora

We carried out our tests on two different corpora: ukWaC, a Web-derived, POS-tagged and lemmatized collection of about 2 billion tokens,¹ and the Yahoo!

¹<http://wacky.sslmit.unibo.it>

database queried via the BOSS service.² We will refer to these corpora as ukWaC and Yahoo from now on.

In ukWaC, we limited the number of occurrence and co-occurrence queries to the first 5000 observations for computational efficiency. Since we collect corpus statistics at the lemma level, we construct Yahoo! queries using disjunctions of inflected forms that were automatically generated with the NodeBox Linguistics library.³ For example, the query to look for "lion" and "cat" with 4 words in the middle is: "(lion OR lions) * * * (cat OR cats OR catting OR catted)". Each pair requires 14 Yahoo! queries (one for W_1 , one for W_2 , 6 for (W_1, W_2) , in that order, with 0-to-5 intervening words, 6 analogous queries for (W_2, W_1)). Yahoo! returns maximally 1,000 snippets per query, and the latter are lemmatized with the TreeTagger⁴ before feature extraction.

2.3 Model implementation

We did not carry out a search for "good" parameter values. Instead, the model parameters are generally picked at convenience to ease memory requirements and computational efficiency. For instance, in all experiments, b is set to 1500 unless noted otherwise in order to fit the vectors of all pairs at our hand into the computer memory.

Once we construct the vectors for a set of word pairs, we get a *co-occurrence matrix* with pairs on the rows and the features on the columns. In all of our experiments, the same normalization method and classification algorithm is used with the default parameters: First, a TF-IDF feature weighting is applied to the co-occurrence matrix (Salton and Buckley, 1988). Then following the suggestion of Hsu and Chang (2003), each feature t 's $[\hat{\mu}_t - 2\hat{\sigma}_t, \hat{\mu}_t + 2\hat{\sigma}_t]$ interval is scaled to $[0, 1]$, trimming the exceeding values from upper and lower bounds (the symbols $\hat{\mu}_t$ and $\hat{\sigma}_t$ denote the average and standard deviation of the feature values respectively). For the classification algorithm, we use the C-SVM classifier and for regression the ϵ -SVM regressor, both implemented in the Matlab toolbox of Canu et al. (2005). We employed a linear kernel. The cost parameter C is set to 1 for all experiments; for the regressor, $\epsilon = 0.2$. For other pattern recognition related coding (e.g., cross validation, scaling, etc.) we made use of the Matlab PRTTools (Duin, 2001).

For each task that will be defined in the next section, we evaluated our algorithm on the following representations: 1) Single-occurrence vectors ($\mathbf{v}_1 \mathbf{v}_2$ condition) 2) Pair-occurrence vectors ($\mathbf{v}_{1,2}$ condition) 3) Entire co-occurrence matrix ($\mathbf{v}_1 \mathbf{v}_2 \mathbf{v}_{1,2}$ condition).

²<http://developer.yahoo.com/search/boss/>

³<http://nodebox.net/code/index.php/Linguistics>

⁴<http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>

3 Tasks

3.1 SAT Analogy Questions

The first task we evaluated our algorithm on is the SAT analogy questions task introduced by Turney et al. (2003). In this task, there are 374 multiple choice questions with a pair of related words like (lion,cat) as the *stem* and 5 other pairs as the *choices*. The correct answer is the choice pair which has the relationship most similar to that in the stem pair.

We adopt a similar approach to the one used in Turney (2008) and consider each question as a separate binary classification problem with one positive training instance and 5 unknown pairs. For a question, we pick a pair at random from the stems of other questions as a pseudo negative instance and train our classifier on this two-instance training set. Then the trained classifier is evaluated on the choice pairs and the pair with the highest posterior probability for the positive class is called the *winner*. The procedure is repeated 10 times picking a different pseudo-negative instance each time and the choice pair which is selected as the winner most often is taken as the answer to that question. The performance measure on this task is defined as the percentage of correctly answered questions. The mean score and confidence intervals are calculated over the performance scores obtained for all folds.

3.2 TOEFL Synonym Questions

This task, introduced by Landauer and Dumais (1997), consists of 80 multiple choice questions in which a word is given as the stem and the correct choice is the word which has the closest meaning to that of the stem, among 4 candidates. To fit the task into our framework, we pair each choice with the stem word and obtain 4 word pairs for each question. The word pair constructed with the stem and the correct choice is labeled as positive and the other pairs are labeled as negative. We consider all 320 pairs constructed for all 80 questions as our dataset. Thus, the problem is turned into a binary classification problem where the task is to discriminate the synonymous word pairs (i.e. positive class) from the other pairs (i.e. negative class). We made sure that the pairs constructed for the same question were never split between training and test set, so that no question-specific learning is performed. The reason for this precaution is that the evaluation is done on a per-question basis. The estimated posterior class probabilities of the pairs constructed for the same question are compared to each other and the pair with the highest probability for the positive class is selected as the answer for the question. By keeping the pairs of a question in the same set we make sure their posteriors are calculated by the same trained classifier. The performance measure is the percentage of correctly answered questions and we report the mean performance over all 10 folds.

3.3 Selectional Preference Judgments

Linguists have long been interested in the semantic constraints that verbs impose on their arguments, a broad area that has also attracted computational modeling, with increasing interest in purely corpus-based methods (Erk, 2007; Padó et al., 2007). This task is of particular interest to us as an example of a broader class of linguistic problems that involve *productive* constraints on composition. As has been stressed at least since Chomsky’s early work (Chomsky, 1957), no matter how large a corpus is, if a phenomenon is productive there will always be new well-formed instances that are not in the corpus. In the domain of selectional restrictions this is particularly obvious: we would not say that an algorithm learned the constraints on the possible objects/patients of eating simply by producing the list of all the attested objects of this verb in a very large corpus; the interesting issue is whether the algorithm can detect if an unseen object is or is not a plausible “eatee”, like humans do without problems. Specifically, we test selectional preferences on the dataset constructed by Padó (2007), that collects average plausibility judgments (from 20 speakers) for nouns as either subjects or objects of verbs (211 noun-verb pairs).

We formulate this task as a regression problem. We train the ϵ -SVM regressor with 18-fold cross validation: Since the pair instances are not independent but grouped according to the verbs, one fold is constructed for each of the 18 verbs used in the dataset. In each fold, all instances sharing the corresponding verb are left out as the test set. The performance measure for this task is the Spearman correlation between the human judgments and our algorithm’s estimates. There are two possible ways to calculate this measure. One is to get the overall correlation between the human judgments and our estimates obtained by concatenating the output of each cross-validation fold. That measure allows us to compare our method with the previously reported results. However, it cannot control for a possible verb-effect on the human judgment values: If the average judgment values of the pairs associated with a specific verb is significantly higher (or lower) than the average of the pairs associated with another verb, then any regressor which simply learns to assign the average value to all pairs associated with that verb (regardless of whether there is a patient or agent relation between the pairs) will still get a reasonably high correlation because of the variation of judgment scores across the verbs. To control for this effect, we also calculated the correlation between the human judgments and our estimates for each verb’s plausibility values separately, and we report averages across these separate correlations (the “mean” results reported below).

3.4 Common-sense Relations from ConceptNet

Open Mind Common Sense⁵ is an ongoing project of acquisition of common-sense knowledge from ordinary

⁵<http://commons.media.mit.edu/en/>

Relation	Pairs	Relation	Pairs
IsA	316	PartOf	139
UsedFor	198	LocationOf	1379
CapableOf	228	Total	1943

Table 1: ConceptNet relations after filtering.

people by letting them carry out simple semantic and linguistics tasks. An end result of the project is ConceptNet 3, a large scale semantic network consisting of relations between concept pairs (Havasi et al., 2007). It is possible to view this network as a collection of semantic assertions, each of which can be represented by a triple involving two concepts and a relation between them, e.g. *UsedFor(piccolo, make music)*. One motivation for this project is the fact that common-sense knowledge is assumed to be known by both parties in a communication setting and usually is not expressed explicitly. Thus, corpus-based approaches may have serious difficulties in capturing these relations (Havasi et al., 2007), but there are reasons to believe that they could still be useful: Eslick (2006) uses the assertions of ConceptNet as seeds to parse Web search results and augment ConceptNet by new candidate relations.

We use the ConceptNet snapshot released in June 2008, containing more than 200,000 assertions with around 20 semantic relations like *UsedFor*, *Desirious-EffectOf*, or *SubEventOf*. Each assertion has a confidence rating based on the number of people who expressed or confirmed that assertion. For simplicity we limited ourselves to single word concepts and the relations between them. Furthermore, we eliminated the assertions with a confidence score lower than 3 in an attempt to increase the "quality" of the assertions and focused on the most populated 5 relations of the remaining set, as given in Table 3.4. There may be more than one relation between a pair of concepts, so the total number is less than the sum of the size of the individual relation sets.

4 Results

For the multiple choice question tasks (i.e. SAT and TOEFL), we say a question is *complete* when all of the related pairs (stem and choice) are represented by vectors with at least one non-zero component. If a question has at least one pair represented by a zero-vector (*missing pairs*), then we say that the question is *partial*. For these tasks, we report the worst-case performance scores where we assume that a random guessing performance is obtained on the partial questions. This is a strict lower bound because it discards all information we have about a partial question even if it has only one missing pair. We define *coverage* as the percentage of complete questions.

4.1 SAT

In Yahoo, the coverage is quite high. In the $v_{1,2}$ only condition, 4 questions had at least some choice/stem

pairs with all zero components. In all other cases, all of the pairs were represented by vectors with at least one non-zero component. The highest score is obtained for the $v_1 v_2 v_{1,2}$ condition with a 44.1% of correct questions, that is not significantly above the 42.5% performance of $v_{1,2}$ (paired t-test, $\alpha = 0.05$). The $v_1 v_2$ only condition results in a poorer performance of 33.9% correct questions, statistically lower than the former two conditions.

For ukWaC, the $v_{1,2}$ only condition provides a relatively low coverage. Only 238 questions out of 374 were complete. For the other conditions, we get a complete coverage. The performances are statistically indistinguishable from each other and are 38.0%, 38.2%, and 39.6% for $v_{1,2}$, $v_1 v_2$, and $v_1 v_2 v_{1,2}$ respectively.

Condition	Yahoo	ukWaC
$v_{1,2}$	42.5%	38.0%
$v_1 v_2$	33.9%	38.2%
$v_1 v_2 v_{1,2}$	44.1%	39.6%

Table 2: Percentage of correctly answered questions in SAT analogy task, worst-case scenario.

In Fig. 1, the best performances we get for Yahoo and ukWaC are compared to previous studies with 95% binomial confidence intervals plotted. The reported values are taken from the ACL wiki page on the state of the art for SAT analogy questions⁶. The algorithm proposed by Turney (2008) is labeled as Turney-PairClass.

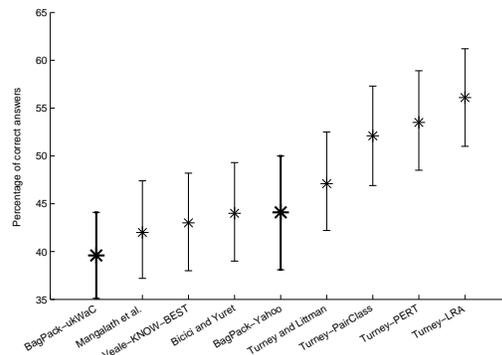


Figure 1: Comparison with previous algorithms on SAT analogy questions.

Overall, the performance of BagPack is not at the level of the state of the art but still provides a reasonable level even in the $v_1 v_2$ only condition for which we do not utilize the contexts where the two words co-occur. This aspect is most striking for ukWaC where the coverage is low and by only utilizing the single-occurrence sub-vectors we obtain a performance of 38.2% correct answers (the comparable "attributional" models re-

⁶See <http://aclweb.org/aclwiki/> for further information and references

ported in Turney, 2006, have an average performance of 31%).

4.2 TOEFL

For the $\mathbf{v}_{1,2}$ sub-vector calculated for Yahoo, we have two partial questions out of 80 and the system answers 80.0% of the questions correctly. The single occurrence case $\mathbf{v}_1\mathbf{v}_2$ instead provides a correct percentage of 41.2% which is significantly above the random performance of 25% but still very poor. The combined case $\mathbf{v}_1\mathbf{v}_2\mathbf{v}_{1,2}$ provides a score of 75.0% with no statistically significant difference from the $\mathbf{v}_{1,2}$ case. The reason of the low performance for $\mathbf{v}_1\mathbf{v}_2$ is an open question.

For ukWaC, the coverage for the $\mathbf{v}_1\mathbf{v}_2$ case is pretty low. Out of 320 pairs, 70 were represented by zero-vectors, resulting in 34 partial questions out of 80. The performance is at 33.8%. The $\mathbf{v}_1\mathbf{v}_2$ case on its own does not lead to a performance better than random guessing (27.5%) but the combined case $\mathbf{v}_1\mathbf{v}_2\mathbf{v}_{1,2}$ provides the highest ukWaC score of 42.5%.

Condition	Yahoo	ukWaC
$\mathbf{v}_{1,2}$	80.0%	33.8%
$\mathbf{v}_1\mathbf{v}_2$	41.2%	27.5%
$\mathbf{v}_1\mathbf{v}_2\mathbf{v}_{1,2}$	75.0%	42.5%

Table 3: Percentage of correctly answered questions in TOEFL synonym task, worst-case scenario.

To our knowledge, the best performance with a purely corpus-based approach is that of Rapp (2003) who obtained a score of 92.5% with SVD. Fig. 2 reports our results and a list of other corpus-based systems which achieve scores higher than 70%, along with 95% confidence interval values. The results are taken from the ACL wiki page on the state of the art for TOEFL synonym questions.

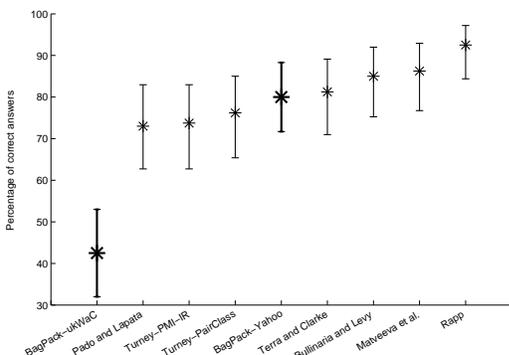


Figure 2: Comparison with previous algorithms on TOEFL synonym questions with 95% confidence intervals.

We note that our results obtained for Yahoo are comparable to the results of Turney but even the best results obtained for ukWaC and the Yahoo’s results for

$\mathbf{v}_1\mathbf{v}_2$ only condition are very poor. Whether this is because of the inability of the sub-vectors to capture synonymity or because the default parameter values of SVM are not adequate is an open question. Notice that our concatenated $\mathbf{v}_1\mathbf{v}_2$ vector does not exploit information about the similarity of \mathbf{v}_1 to \mathbf{v}_2 , that, presumably, should be of great help in solving the synonym task.

4.3 Selectional Preference

The coverage for this dataset is quite high. All pairs were represented by non-zero vectors for Yahoo while only two pairs had zero-vectors for ukWaC. The two pairs are discarded in our experiments. For Yahoo, the best results are obtained for the $\mathbf{v}_{1,2}$ case. The single-occurrence case, $\mathbf{v}_1\mathbf{v}_2$, provides an overall correlation of 0.36 and mean correlation of 0.26. However low, in case of rarely co-occurring word pairs this data could be the only data we have in our hands and it is important that it provides reasonable judgment estimates.

For the ukWaC corpus, the best results we get are an overall correlation of 0.60 and a mean correlation of 0.52 for the combined case $\mathbf{v}_1\mathbf{v}_2\mathbf{v}_{1,2}$. The results for $\mathbf{v}_{1,2}$ and $\mathbf{v}_1\mathbf{v}_2\mathbf{v}_{1,2}$ are statistically indistinguishable.

Condition	Yahoo		ukWaC	
	Overall	Mean	Overall	Mean
$\mathbf{v}_{1,2}$	0.60	0.45	0.58	0.48
$\mathbf{v}_1\mathbf{v}_2$	0.36	0.26	0.33	0.22
$\mathbf{v}_1\mathbf{v}_2\mathbf{v}_{1,2}$	0.55	0.42	0.60	0.52

Table 4: Spearman correlations between the targets and estimations for selectional preference task.

In Fig. 3, we present a comparison of our results with some previous studies reported in Padó et al. (2007). The best result reported so far is a correlation of 0.52. Our results for Yahoo and ukWaC are currently the highest correlation values reported. Even the verb-effect-controlled correlations achieve competitive performance.

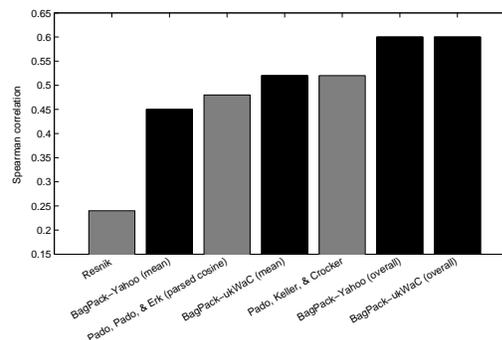


Figure 3: Comparison of algorithms on selectional preference task.

4.4 ConceptNet

Only for this task, (because of practical memory limitations) we reduced the model parameter b to 500, which means we used the 500 most frequent unigrams and 500 most frequent bigrams as our basis terms. For each of the 5 relations at our hand, we trained a different SVM classifier by labeling the pairs with the corresponding relation as positive and the rest as negative. To eliminate the issue of unbalanced number of negative and positive instances we randomly down-sampled the positive or negative instances set (whichever is larger). For the *IsA*, *UsedFor*, *CapableOf*, and *PartOf* relations, the down-sampling procedure means keeping some of the negative instances out of the training and test sets while for the *LocationOf* relation it means keeping a subset of the positive instances out. We performed 5 iterations of the down-sampling procedure and for each iteration we carried out a 10-fold cross-validation to train and test our classifier. The results are test set averages over all iterations and folds. The performance measure we use is the area under the *receiver operating characteristic* (AUC in short for area under the curve). The AUC of a classifier is the area under the curve defined by the corresponding true positive rate and false positive rate values obtained for varying the threshold of the classifier to accept an instance as positive. Intuitively, AUC is the probability that a randomly picked positive instance’s estimated posterior probability is higher than a randomly picked negative instance’s estimated posterior probability (Fawcett, 2006).

The coverage is quite high for both corpora: Out of 1943 pairs, only 3 were represented by a zero-vector in Yahoo while in ukWaC this number is 68. For simplicity, we discarded missing pairs from our analysis. We report only the results obtained for the entire co-occurrence matrix. The results are virtually identical for the other conditions too: Both for Yahoo and ukWaC, almost all of the AUC values obtained for all relations and for all conditions are above 95%. Only the *PartOf* relation has AUC values above 90% (which is still a very good result).

Relation	Yahoo	ukWaC
IsA	99.0%	98.0%
UsedFor	98.2%	98.5%
CapableOf	98.9%	99.1%
PartOf	97.6%	95.0%
LocationOf	99.0%	98.8%

Table 5: AUC scores for 5 relations of ConceptNet, classifier trained for $v_1 v_2 v_{1,2}$ condition.

The very high performance we observe for the ConceptNet task is surprising when compared to the moderate performance we observe for other tasks. Our extensive filtering of the assertions could have resulted in a biased dataset which might have made the job of the classifier easy while reducing its generalization ca-

capacity. To investigate this, we decided to use the pairs coming from the SAT task as a validation set.

Again, we trained an SVM classifier on the ConceptNet data for each of the 5 relations like we did previously, but this time without cross-validation (i.e. after the down-sampling, we used the entire set as the training dataset in each iteration). Then we evaluated the classifiers on the 2224 pairs of the SAT analogy task (removing pairs that were in the training data) and averaged the posterior probability reported by each SVM over each down-sampling iteration. The 5 pairs which are assigned the highest posterior probability for each relation are reported in Table 6. We have not yet quantified the performance of BagPack in this task but the preliminary results in this table are, qualitatively, exceptionally good.

5 Conclusions

We presented a general way to build a vector-based space to represent the semantic relations between word pairs and showed how that representation can be used to solve various tasks involving semantic similarity. For SAT and TOEFL, we obtained reasonable performances comparable to the state of the art. For the estimation of selective preference judgments about verb-noun pairs, we achieved state of the art performance. Perhaps more importantly, our representation format allows us to provide meaningful estimates even when the verb and noun are not observed co-occurring in the corpus – which is an obvious advantage over the models which rely on syntagmatic contexts alone and cannot provide estimates for word pairs that are not seen directly co-occurring. We also obtained very promising results for the automated augmentation of ConceptNet.

The generality of the proposed method is also reflected in the fact that we built a single feature space based on frequent basis terms and used the same features for all pairs coming from different tasks. The use of the same feature set for all pairs makes it possible to build a single database of word-pair vectors. For example, we were able to re-use the vectors constructed for SAT pairs as a validation set in the ConceptNet task. Furthermore, the results reported here are obtained for the same machine learning model (SVM) without any parameter tweaking, which renders them very strict lower bounds.

Another contribution is that the proposed method provides a way to represent the relations between words even if they are not observed co-occurring in the corpus. Employing a larger corpus can be an alternative solution for some cases but this is not always possible and some tasks, like estimating selectional preference judgments, inherently call for a method that does not exclusively depend on paired co-occurrence observations.

Finally, we introduced ConceptNet, a common-sense semantic network, to the corpus-based semantics community, both as a new challenge and as a repository we

Rank	IsA	UsedFor	PartOf	CapableOf	LocationOf
1	watch,timepiece	pencil,draw	vehicle,wheel	motorist,drive	spectator,arena
2	emerald,gem	blueprint,build	spider,leg	volatile,vaporize	water,riverbed
3	cherry,fruit	detergent,clean	keyboard,finger	concrete,hard	bovine,pasture
4	dinosaur,reptile	guard,protect	train,caboose	parasite,contribute	benediction,church
5	ostrich,bird	buttress,support	hub,wheel	immature,develop	byline,newspaper

Table 6: Top 5 SAT pairs classified as positive for ConceptNet relations, classifier trained for $v_1 v_2 v_{1,2}$ condition.

can benefit from.

In future work, one of the most pressing issue we want to explore is how to better exploit the information in the single occurrence vectors: currently, we do not make any use of the *overlap* between v_1 and v_2 . In this way, we are missing the classic intuition that taxonomically similar words tend to occur in similar contexts, and it is thus not surprising that $v_1 v_2$ flunks the TOEFL. We are currently looking at ways to augment our concatenated vector with “meta-information” about vector overlap.

References

- S. Canu, Y. Grandvalet, V. Guigue and A. Rakotomamonjy. 2005. *SVM and Kernel Methods Matlab Toolbox*, Perception Systèmes et Information, INSA de Rouen, Rouen, France
- N. Chomsky. 1957. *Syntactic structures*. Mouton, The Hague.
- R. P. W. Duin. 2001. PRTOOLD (Version 3.1.7), A Matlab toolbox for pattern recognition. Pattern Recognition Group, Delft University of Technology.
- K. Erk. 2007. A simple, similarity-based model for selectional preferences. *Proceedings of ACL 2007*.
- K. Erk and S. Padó. 2008. A structured vector space model for word meaning in context. *Proceedings of EMNLP 2008*.
- I. Eslick. 2006. Searching for commonsense. Master’s thesis, Massachusetts Institute of Technology.
- T. Fawcett. 2006. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874.
- C. Havasi, R. Speer and J. Alonso. 2007. Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*, Borovets, Bulgaria, September.
- C.-W. Hsu, C.-C Chang. 2003. *A practical guide to support vector classification*. Technical report, Department of Computer Science, National Taiwan University.
- T.K. Landauer and S.T. Dumais. 1997. A solution to Plato’s problem: The Latent Semantic Analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2): 211–240.
- H. Liu and P. Singh. 2004. ConceptNet — A practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4) 211–226.
- S. Mirkin, I. Dagan and M. Geffet. 2006. Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. *Proceedings of COLING/ACL 2006*, 579–586.
- S. Padó, S. Padó and K. Erk. 2007. Flexible, corpus-based modelling of human plausibility judgements. *Proceedings EMNLP 2007*, 400–409.
- U. Padó. 2007. *The Integration of Syntax and Semantic Plausibility in a Wide-Coverage Model of Sentence Processing*. Ph.D. thesis, Saarland University.
- P. Pantel and M. Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. *Proceedings of COLING/ACL 2006*, 113–120.
- R. Rapp. 2003. Word sense discovery based on sense descriptor dissimilarity. *Proceedings of MT Summit IX*: 315–322.
- M. Sahlgren. 2006. *The Word-space model*. Ph.D. dissertation, Stockholm University, Stockholm.
- G. Salton and C. Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5): 513–523.
- R. Speer, C. Havasi and H. Lieberman. 2008. Analogyspace: Reducing the dimensionality of common sense knowledge. In Dieter Fox and Carla P. Gomes, editors, *AAAI*, pages 548–553. AAAI Press.
- P. Turney. 2006. Similarity of semantic relations. *Computational Linguistics*, 32(3): 379–416.
- P. Turney. 2008. A uniform approach to analogies, synonyms, antonyms and associations. *Proceedings of COLING 2008*, 905–912.