Exploring the Vector Space Model for Finding Verb Synonyms in Portuguese

Luís Sarmento[1], Paula Carvalho[2] and Eugénio Oliveira[1]
[1]Faculdade de Engenharia da Universidade do Porto - DEI - LIACC
Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal
[2]Faculdade de Ciências da Universidade de Lisboa - DI - XLDB
Campo Grande, 1749-016 Lisboa, Portugal
las@fe.up.pt, *pcc@xldb.di.fc.ul.pt* and *eco@fe.up.pt*

Abstract

We explore the performance of the Vector Space Model (VSM) in finding verb synonyms in Portuguese by analyzing the impact of three operating parameters: (i) the weighting function, (ii) the context window used for automatically extracting features, and (iii) the minimum number of vector features. We rely on distributional statistics taken from a large n-gram database to build feature vectors, using minimal linguistic pre-processing. Automatic evaluation of synonym candidates using gold-standard information from the OpenOffice and Wiktionary thesaurus shows that low frequency features carry most information regarding verb similarity, and that a [0, +2] window carries more information than a [-2, 0] window. We show that satisfactory precision levels require vectors with 50 or more non-nil components. Manual evaluation over a set of *declarative verbs* and *psychological verbs* show that VSM-based approaches achieve good precision in finding verb synonyms for Portuguese, even when using minimal linguistic knowledge. This lead us to proposing a performance baseline for this task.

Keywords

Vector Space Model, Semantics, Relation Extraction, Statistical Methods, Language Resources, Evaluation

1 Introduction

Large-coverage and fine-grained linguistic resources are crucial for the majority of the applications in natural language processing, but they are still scarce and, in most cases, they do not satisfy every particular information need. Manual creation of linguistic resources is time-consuming and requires linguistic expertise. Therefore, there is a rising interest in developing automatic or semi-automatic methods and techniques for building language resources with minimal human intervention. However, automatic methods usually involve a large set of parameters, whose impact on final results is difficult to assess, and thus to optimize. In this paper, we address the task of automatically creating a lexicon of verb synonyms for Portuguese using the Vector Space Model (VSM), and we explore the impact of three of its core parameters: (i) the *context* used for extracting vector features, (ii) the function used for weighting features, and (iii) the *cut-off threshold* for removing vectors with insufficient feature information. We rely on n-gram information collected from a large dump of the Portuguese web, in order to obtain distributional statistics for verb lemmas. For performing parameter exploration, we evaluate results automatically using gold-standard information extracted from the OpenOffice thesaurus and from Wiktionary. Fine-grained evaluation was achieved by manually assessing the synonym candidates obtained for a sample of two syntactic-semantic classes of verbs: *psychological verbs* and *declarative verbs*. We chose these two specific verb classes for two reasons. First, they exhibit different syntactic and semantic behavior, and thus present different challenges for the task of synonymy finding. Psychological verbs do not have a prototypical syntactic structure and they usually convey a plurality of meanings, which can only be disambiguated in context. In contrast, declarative verbs are less ambiguous and the syntactic structure where they occur is better defined. Second, these two verb classes are crucial in several information extraction task, such as for example quotation extraction from news or opinion mining, so it is particularly interesting to evaluate the performance over them for practical reasons.

To the best of our knowledge, this study is pioneer for Portuguese. Since our approach relies only on minimal linguistic processing, the results presented can be considered a *baseline* for other methods that try to perform the same task, using additional linguistic information.

2 Related Work

Curran [5] follows an experimental methodology for testing several parameters of the VSM in the process of automatically computing a language thesaurus – the context for extracting features, functions for weighting those features, functions for computing vector similarity, cut-off thresholds for input data and algorithms for computing pairwise vector similarity. The author performs large scale experimentation on the parameter space and evaluates results automatically by computing precision at several ranks, inverse ranks (InvR) and direct comparison with a gold standard built by aggregating 5 thesauri: the *Roget's Thesaurus*, the *New Roget's Thesaurus*, the *Moby Thesaurus*, the *New Oxford Thesaurus of English* and the *Macquire Encyclopedic Thesaurus*. WordNet was also used to automatically check if results on synonymy are contaminated with antonyms, hyponyms or meronyms. Detailed error analysis was performed for a sample of 300 words. Results show that when the number of features associated to vector drops below 1000, or for words with frequencies below 5000, performance decays significantly. Additionally, direct comparison and InvR measures tend to increase for words with multiple senses with larger number of senses while the precision measures are fairly stable. Results also demonstrate that it is more difficult to find synonyms for words related with certain Wordnet classes such as *entities* and *abstractions*.

Sahlgren [11] builds vector spaces for capturing either *paradigmatic* or *syntagmatic* relations, and tests how such spaces can then be used for different tasks – thesaurus generation, synonym finding, antonym detection and POS guessing. The author evaluates the impact of several VSM parameters such as (i) the context (paradigmatic vs. syntagmatic), size of the context window (narrow vs. wide and small vs. large), the weighting of the windows (constant vs. aggressive decay) feature weighting functions (raw frequency vs. binary vs. tf-idf vs. logarithmic). For the specific task of finding synonyms the author concludes that spaces built using paradigmatic contexts clearly outperform those built using syntagmatic contexts. Additionally, vectors built by extracting word features from *narrow windows* (with two or three context words around the headword) lead to better performance. Interestingly, wide windows lead to better results for the task of finding *antonyms*.

In im Walde [9], a set of experiments on clustering German verbs (by synonymy) is presented. Verbs are described by vectors whose features are extracted from 3 types of contexts with increasing levels of semantic information: (i) syntactical relations (from a set of 38 possible frames); (ii) syntactical relations + information about prepositional preferences, and (iii) 15 possible semantic categories of the verb arguments (mostly nouns and noun phrases) taken from GermaNet. The author concludes that the addition of more informative features – from (i) to (iii) – has a positive effect on clustering results. Also, they observe that (a) similarity metrics such as the Kullback-Liebler and its variants tended to produce better results in larger data-sets, and (b) low-frequency verbs had a negative impact in the quality of the clusters. More importantly, the authors conclude that the choice of features and the overall success of the clustering approach greatly depends on definition of *verb group* one wishes to replicate automatically.

The work by Chklovski and Pantel [2] also focus on finding semantic relations between verbs, namely *similarity*, *strength*, *antonymy*, *enablement* and *happens-before*. The procedure involves querying a search engine for co-occurrences of pairs of verbs in specific lexical-syntactic patterns that indicate that the verbs might establish one of such relations. Results were evaluated by human assessors. Lin [10] uses a broad-coverage parser to obtain grammatical relationships between pairs of words. Each word is then represented by a vector whose features are derived from the set grammatical relations it establishes with other words. Raw frequency values are weighted using a variation of the Mutual Information function. Pairs-wise similarity between nouns, verbs and adjectives/adverbs that occurred at least 100 times was computed, using several similarity metrics. Then for each word, a thesaurus entry was created using the top most similar words. Evaluation was performed using WordNet and the Roget Thesaurus.

Related work on VSM generally takes advantage of significant linguistic information, usually extracted from *annotated* corpora. In this study, we rely mostly on the information directly derived from data, in particular, on raw n-grams statistics taken from a large non-annotated collection of web documents. Apart from dictionary-based filtering and lemmatization, no additional linguistic processing (e.g. POS annotation, word-sense disambiguation) is used. Given the increasing availability of large databases of n-grams computed from non-annotated terabyte web collections (e.g. Goolge's N-gram database) and the lack of publicly available resources for Portuguese with refined semantic information, we believe that this is an interesting approach.

3 VSM Parameters

The Vector Space Model provides a convenient framework for finding semantic similarities between words, because it allows to express a strong intuition regarding semantic similarity: the *Distributional Hypothesis* [8]. There are several parameters related to the VSM, the most crucial being perhaps the choice of the appropriate *context* for extracting features capable of leading to *meaningful* vector representations of words. Usually, relevant features can be found at lexical level (by exploring the lexical surroundings of words) or at syntactical level (by exploring syntactic relations between words and constituents in a sentence, such as "subject-predicate" relation). The choice of a specific feature context has a huge impact on the information that is *transferred* to the Vector Space, thus directly affecting the notion of "similarity" that may be inferred from feature vectors (see [11]). There are also several *cutoff thresholds* used for limiting the feature vectors included in the space. These are important for cases where there might not be enough empirical evidence associated with the corresponding words. Such vectors might lead to noisy associations.

Another important parameter in the VSM is the choice of the *feature weighting function*, such as tf-idf [12], Mutual Information [3] and the Log-Likelihood Ratio [6]. Different weighting functions tend to promote (or demote) different sections of the feature spectrum, so choosing the appropriate weighting function for a specific word comparison task might have a deep impact in the final results (e.g. should "idiosyncratic" features be considered more important?). A closely related question is the choice of a *distance metric* for comparing the (weighted) vectors. Global performance of VSM approaches depends on the combination of a specific weighting function and a specific distance metric, and there is usually an optimal combination for different tasks (see [5]). However, in this work we will not explore this parameter in order to avoid dealing with additional complexity for now. Thus, in all our experiments we will keep the same metric (i.e. the cosine)

4 VSM for Verb Synonyms

As mentioned before, we wish to investigate the impact of considering a restricted set of lexical units that co-occur with a particular verb, in the specific task of synonymy detection. Concretely, we confined the context window to the *four* words around the verb, i.e. a [-2 : +2] window. Since Portuguese is an SVO language, we believe that such context contains, in the majority of the cases, relevant information about *verb-object* and *subject-verb* relations[1]. The right and the left contexts are specially important for the case of *transitive* and *intransitive verbs*, respectively. We also assume that features extracted from such contexts might be compiled independently, so that feature vectors can be created by aggregating the two sources of statistical evidence.

For obtaining verb context information we used a database of n-gram statistics compiled from a dump of the Portuguese web, totalling about 1000 million words. We scanned 3-gram information of the form (w_1, w_2, w_3, f) for cases where either w_1 or w_3 were verbs. N-gram information in this collection is *not* POS-tagged. Nevertheless, since the majority of verb forms are inflected, they can be unambiguously recognized using a simple dictionary (at least for the vast majority of possible forms). Hence, we used a dictionary to filter out ambiguous verb forms – i.e. those that could not be uniquely assigned to an unique (verb) lemma – so that only the 3-grams matching either of the two following selection patterns were chosen (v_{uf} = unambiguous verb form):

- *Pattern 1* = [$w_1 = v_{uf}$ & w_2 = * & w_3 = *]
- *Pattern 2* = [w_1 = * & w_2 = * & $w_3 = v_{uf}$]

Verb forms (at w_1 or at w_3) are lemmatized in order to obtain feature tuples of the form (verb lemma, "X w_2 w_3", frequency) and (verb lemma, "w_1 w_2 X", frequency), with *X* signalling the original position of the verb in relation to the extracted features. Feature information extracted for the various forms of the same lemma is merged so to that a single feature vector is obtained for each verb lemma. At this point, feature vectors contain raw frequency information regarding features extracted from the two words before the verb and from the two words after the verb. Features can then be weighted according to given weighting function to produce *weighted feature vectors*, which should be able to reflect more faithfully the association between verbs and features. Next, weighted feature vectors are compared so that we obtain all pairwise similarities. Synonyms for verb v_i are obtained among the other verbs, v_j, whose feature vectors $[V_j]$ are more similar to $[V_i]$. By this procedure, we are not producing *closed sets* of verb synonyms: we are building a network of similarities which enables a verb to be synonym of many other verbs, depending on the different senses it conveys.

However, we know in advance that the chosen context scope will not allow to differentiate between synonyms and antonyms. Opposite sense verbs tend to occur in the *same contexts*, since they usually select identical arguments and allow the same modifiers (e.g. "Please, open the door!" and "Please, close the door!"). Nevertheless, we decided to analyze how VSM performs in the detection of synonyms in Portuguese and assess the true impact of this limitation. Furthermore, we assume that antonyms could be identified in a subsequent *post-processing* step by using techniques such as the ones described in [2].

5 Evaluating Verb Synonyms

We used a publicly available resource as a gold-standard for automatic evaluation: the OpenOffice thesaurus for Portuguese[2]. From the OpenOffice thesaurus we collected (verb → list of synonyms) mappings for 2,783 verbs, each having 3.83 synonyms in average. However, this information refers only to about 50% of the verb lemmas one can find in standard on-line dictionaries for Portuguese (e.g. [1]). More important, there are serious *recall* problems for the mappings collected. For example, many high-frequency verbs have *only one* synonym in OpenOffice thesaurus: "ganhar" (to "win") → "poupar" ("to save");"afirmar" ("to state") → "declarar" ("to declare"); "chamar" ("to call") → "invocar" ("to invoke"), among many others. In order to minimize this problem, we extracted additional verb synonym information from the Portuguese version of the Wiktionary project[3]. We thus obtained additional (verb → list of synonyms) mappings for 2,171 verbs, each having in average 1.95 synonyms. By merging mappings extracted from both resources we obtained a larger gold-standard covering 3,423 verbs, with 4.53 synonyms per verb. This larger gold-standard still has coverage and recall problems, but we believe that it provides a good solution for the purpose of performing *parameter exploration.*

Nevertheless, we chose to perform a more thorough evaluation by manually analyzing results obtained two subclasses of verbs. We selected two groups of verbs with different syntactic and semantic properties (see Table 1). The first group includes 25 *declarative verbs*, such as "dizer" ("to say") or "mencionar" ("to mention"), and will be referred as V_{com}. The second group includes 25 *psychological verbs*, such as "gostar" ("to like") and "envergonhar" ("to shame"), and will be mentioned as V_{emo}. V_{emo} are related to the expression of a sentiment or an emotion, which can be experienced

[1] It should be stressed, however, that this context window is not sufficient for all cases, namely when there is a modifier between the verb and one of its arguments.

[2] Available from http://openthesaurus.caixamagica.pt/. The most recent version is dated from 2006-08-17.

[3] Available at http://download.wikimedia.org/ptwiktionary/

by the human noun occupying the subject or the complement position, according to the verb at stake. The level of polysemy of verbs in V_{com} is relatively low. On the other hand, verbs in V_{emo} are highly polysemous. This fact is somehow reflected by the vast list of possible antonyms, with various degrees of strength, that can be associated to verbs in V_{emo}. Sets V_{com} and V_{emo} can be placed in opposite ends of the spectrum regarding the performance that one expects to achieve in the task of synonym finding: performance for V_{com} should be higher than for V_{emo}.

	Verbs
V_{com}	acrescentar, adiantar, afirmar, alertar, anunciar, avisar, comunicar, confessar, contar, comentar, declarar, defender, destacar, dizer, esclarecer, explicar, frisar, indicar, mencionar, nomear, responder, referir, revelar, salientar, sublinhar
V_{emo}	aborrecer, adorar, agradar, amar, angustiar, assustar, atemorizar, chatear, decepcionar, detestar, emocionar, enternecer, entristecer, entusiasmar, envergonhar, fascinar, gostar, humilhar, impressionar, intimidar, irritar, lisonjear, orgulhar, preocupar, ridicularizar

Table 1: *Verb groups chosen for manual evaluation.*

Performance Metrics

Let V_{gold} be the set of verb entries in the *gold standard verb thesaurus*, and let V_{auto} be the set of verb entries for which synonyms mappings were obtained by the automatic method. Also, let $S_{gold}(v_i)$ be the set of verb synonyms defined for entry v_i in the gold standard thesaurus (i.e. the "true" synonyms), and $S_{auto}(v_i)$ be the set of synonyms inferred automatically for v_i. As a result of the automatic process, elements in $S_{auto}(v_i)$ are ranked according to the degree of synonymy they have with v_i. Thus, traditional metrics used in information retrieval can be used for evaluating the ranked sets of verb synonyms $S_{auto}(v_i)$ against those in $S_{gold}(v_i)$. Because verb mappings contained in the gold standard are far from being complete, we will not compute recall figures and we will mainly focus on evaluating *precision*.

More specifically, for each verb entry $v_i \in (V_{auto} \cap V_{gold})$, we will compute three precision figures. The first is *Precision at Rank 1*, $P_{@}(v_i, 1)$. The second is *Precision at Rank* $N_{gold}(v_i)$, $P_{@}(v_i, N_{gold}(i))$, with $N_{gold}(v_i)$ being the number of true synonyms contained in $S_{gold}(v_i)$. The third is *Average Precision*, $AP(v_i)$, which gives a global view of the precision by combining the values of the precision at various ranks:

$$AP(v_i) = \frac{\sum_{r=1}^{N_{gold}(i)} P_{@}(v_i, r) \times rl_{@}(v_i, r)}{N_{gold}(i)} \qquad (1)$$

with $N_{gold}(i)$ being the number of elements in $S_{gold}(v_i)$, and $rl_{@}(v_i, r)$ a binary function indicating if the element of $S_{auto}(v_i)$ at rank r is element of $S_{gold}(v_i)$ (1) or not (0).

Global performance figures can be obtained by averaging $P_{@}(v_i, 1)$, $P_{@}(v_i, N_{gold}(v_i))$ and $AP(v_i)$ over all entries for which evaluation was possible, i.e for $v_i \in (V_{auto} \cap V_{gold})$. This allows us to compute three global precision figures: $P_{@}^{avg}(1)$, $P_{@}^{avg}(N)$ and MAP. A global *coverage* figure, $\mathcal{C}$, can be computed by dividing the number of entries evaluated by the total number of entries in the gold standard thesaurus: $\mathcal{C} = |V_{auto} \cap V_{gold}|/|V_{gold}|$. For manual evaluation, we are no longer limited by the number of "true" synonyms contained in the gold standard for a given entry, so we can compute the value of precision at several ranks up to a reasonable value (although we still can not list all possible synonyms of a verb). We chose to compute precision at ranks 1, 5, 10 and 20, which will be represented by $P_{@}^{man}(v_i, n)$, with $n \in \{1, 5, 10, 20\}$.

6 Experimental Setup

We wish to test the impact of three VSM parameters on the overall quality of the automatically generated synonymy mappings. First, for assessing the impact of different *weighting functions* (*Experiment Set 1*) we will run the complete procedure for automatically generating synonym mappings iteratively times, keeping the same context scope - a window of $[-2, +2]$ words - while using different feature weighting functions. We will try several well-documented (and frequently used) weighting functions, namely: tf-idf [12], Log-Likelihood Ratio (LL) [6], Z-Score [14], Pearson's χ^2 test [7], Student's T test [7], Mutual Information (MI) [3], Mutual Dependency (MD) [15] and ϕ^2 test [4]. We also run the complete experiment using no weighting function, i.e. using raw frequencies. For this set of experiments, we arbitrarily set the cutoff threshold on the minimum number of features to 1. Additionally, pairs with cosine similarity lower than 0.1 will be excluded (which can lead to different coverage values).

The second parameter to be explored is the context window used for extracting features. *Experiment Set 2* will consist in executing the complete synonymy finding procedure using only features extracted from a $[-2, 0]$ window (i.e. the two words preceding the verb) and from a $[0, +2]$ window (i.e. the two words following the verb). These experiment will be run using the *best performing* weighting function found in the previous experiment. The third parameter we wish to investigate is the *cutoff threshold* to be applied to raw frequency feature vectors based on the number of non-null features. In *Experiment Set 3* we will select the *best performing* weighting function found in Experiment Set 1, and repeat the complete synonym finding process with increasing cutoff thresholds. We expect to obtain increasing precision values, while coverage should slowly decrease.

Finally, for refining the figures obtained by automatic evaluation, we will manually evaluate two subsets of verbs that lie on the opposite ends of the spectrum in what performance is concerned. The main purpose is to define a possible baseline for the task of automatic synonym finding, knowing in advance that the VSM approach we used is almost purely lexical (it relies on a minimal set of linguistic features) and does not try to address issues related with antonymy and ambiguity. We will chose the best performing configuration, in terms of $P^{avg}_{@}1$ found in Experiment 3 and manually evaluate candidate synonyms found for 25 verbs V_{com} (related to *communication*) and 25 verbs V_{emo} (related the *expression of emotion*). Results for verbs in V_{emo} are expected to be substantially lower than those for V_{com}.

Feature information was obtained from our n-gram database ([13]). There are 173,607,555 distinct 3-grams available in the database. Selection Pattern 1 allowed collecting feature information for 4,972 verbs, described in a space with 2,002,571 dimensions. Selection pattern 2 allowed to collect feature information for 4,962 verbs over 2,066,282. Globally, by aggregating information from both patterns we were able to collect information for 5,025 verbs in a space with 4,068,853 dimensions. Table 2 presents an histogram regarding the number of word vectors and number of features.

# feat.	# vec.	# feat.	# vec.
< 10	541	200 - 499	777
10 - 19	220	500 - 1k	580
20 - 29	145	1k - 2k	456
30 - 39	136	2k - 5k	497
40 - 49	112	5k - 10k	306
50 - 99	353	10k - 50k	382
100 - 199	471	≥ 50k	49

Table 2: *Number of vectors per number features*

7 Results and Analysis

Global precision figures $P^{avg}_{@}1$, $P^{avg}_{@}N$ and MAP (mean average precision) for Experiment Sets 1, 2 and 3 (automatic evaluation) are presented in Tables 3, 4 and 5. Results of manually evaluating synonym identification for the 25 verbs related to *communication*, V_{com}, and the 25 verbs related the *expression of emotion* V_{emo} are presented in Table 6 (synonym candidates were obtained by setting the cutoff threshold to 200, i.e. best $P_{@}1$ found in Experiment Set 3). The most relevant, yet expected, fact regarding results from automatic evaluation is that precision values are all quite low, even for the best configurations (< 0.30). This is not surprising since the gold standard used has serious recall gaps, so it is possible that many correct top found synonyms can be evaluated, thus decreasing precision figures. In [11], even lower precision figures are reported. Also, we knew in advance that the context chosen for generating feature vectors does not allow to effectively differentiate between a verb and its possible opposite senses. Still, performance values obtained can be interpreted from a relative point of view.

Results presented in Table 3 confirm that the impact of the weighting function is very relevant. The best performing weighting function (Mutual Information) leads to a Mean Average Precision figure that outperforms the one obtained using the worst performing weighting function with comparable coverage (Log-Likelihood) by over 300%. Notably, the two best performing weighting functions are Mutual Information and Mutual Dependency, both grounded in information theoretic concepts (the two metrics are actually very similar). A well-known effect of these type of metrics is that they tend to asymptotically over-promote rare features. This suggests that rare features might be of crucial value in the task of finding semantically similar verbs. It is also quite surprising to see that most weighting functions score worse than performing not weighting at all (*raw*). This is so even in the case of popular weighting functions such as tf-idf. One possible reason for this is having set the cut-off threshold on the minimum number of non-nil features to 1, which resulted in considering many verb vectors with insufficient statistical information (see Table 2). Some of the weighting functions used might be particularly sensitive to this effect, and actually lead to worse results than performing no weighting at all. Another observation is that by imposing a minimum cosine similarity threshold of 0.1 the coverage obtained using the weighting functions χ^2 and ϕ^2 was approximately half of that obtained for the others. This confirms that there is a considerable interaction between the choice of the weighting function and the similarity metrics used.

Weighting	$P^{avg}_{@}1$	$P^{avg}_{@}N$	MAP	$\mathcal{C}$
MI	0.221	0.121	0.125	0.800
MD	0.164	0.083	0.083	0.800
Z	0.134	0.096	0.067	0.712
χ^2	0.087	0.075	0.030	0.392
ϕ^2	0.084	0.075	0.027	0.375
raw	0.083	0.041	0.043	0.798
tf-idf	0.076	0.038	0.039	0.800
T	0.073	0.040	0.040	0.800
LL	0.059	0.034	0.037	0.796

Table 3: *Experiment Set 1: context window = [-2, + 2] and cutoff threshold = 1*

.

Window	$P^{avg}_{@}1$	$P^{avg}_{@}N$	MAP	$\mathcal{C}$
[-2, 0]	0.136	0.078	0.079	0.779
[0, +2]	0.196	0.107	0.111	0.798
[-2, +2]	0.221	0.121	0.125	0.800

Table 4: *Experiment Set 2: weighting function = mutual information and cutoff threshold = 1*

.

Results from the Experiment Set 2 (Table 4) show that using feature information from *both* the left and right the verb lead to better results that using any of the two sides individually. From a relative point of view, the two words following the verb (i.e. context [0, +2]) appear to carry more information regarding verb synonymy than the two previous words (i.e. context [-2, 0]), which seems quite natural since most verbs are transitive.

As for Experiment Set 3, results shown in Table 5 confirm expectation: increasing the cutoff threshold lead to better precisions values, at the cost of reducing coverage. However, if threshold is set too high (≥ 200), values of precision do not increase anymore, while the global coverage figure falls continually. For even higher thresholds (≥ 500) precision figures actually drop, since by excluding word vectors below

cut.	$P^{avg}_{@}1$	$P^{avg}_{@}N$	MAP	$\mathcal{C}$
1	0.221	0.121	0.125	0.800
10	0.251	0.136	0.136	0.783
20	0.263	0.142	0.141	0.767
50	0.277	0.149	0.149	0.736
100	0.288	0.154	0.154	0.695
200	0.297	0.155	0.155	0.632
500	0.297	0.146	0.146	0.507
1000	0.290	0.141	0.141	0.398
2000	0.294	0.140	0.141	0.300

Table 5: *Experiment Set 3: weighting function = mutual information and context window [-2, +2]*

.

the threshold we are also removing correct word synonyms of verbs that were not filtered out, leading to a decrease in precision values for these more frequent verbs.

Group	$P^{man}_{@}1$	$P^{man}_{@}5$	$P^{man}_{@}10$	$P^{man}_{@}20$
V_{com}	0.88	0.71	0.56	0.44
V_{emo}	0.60	0.44	0.37	0.27

Table 6: *Manual evaluation of sets V_{com} and V_{emo}*

Results shown in Table 6 suggest that automatic evaluation underestimates performance. This is due mostly to the low *recall* of the gold-standard used. Also performance achieved for V_{com} is very high. Top ranked synonyms found for V_{com} are correct most of the times. More specifically, the values of $P^{man}_{@}1$ (0.88) and $P^{man}_{@}5$ (0.71) confirm that antonyms seem not to represent such a severe problem for the case of V_{com}. On the other hand, for verbs in V_{emo} antonyms populate the top ranked positions, and in some cases are best ranked candidate. An interesting case in V_{emo} is the verb "gostar" ("to like"), which scored 0, or close to 0 precision, at all ranks tested despite being a very frequent verb. As expected, performance figures obtained for V_{emo} are much lower than those obtained for V_{com}. Due to the simplicity of the VSM approach we followed, the figures obtained for V_{emo} can be considered baseline values for other automatic approaches aiming at finding verb synonyms for Portuguese.

8 Conclusions

We confirmed that the weighting function chosen has a crucial impact on the performance obtained when using the VSM for finding verb synonyms in Portuguese. Results achieved by combining the cosine distance with the Mutual Information weighting function suggest the low frequency features carry most of the information regarding verb similarity. We showed that information obtained from both sides of the verb is important for identifying possible synonyms, but the two following words seem to carry more information than the two preceding words. Also, we showed that it is beneficial to exclude word vectors with less than 50 non-nil features, but when the cutoff threshold is set too high both precision and coverage figures will be affected. Manual evaluation showed that the performance obtained by the VSM approach varies greatly depending of the linguistic and semantic properties of the verbs at stake. Results for verbs related with communication show that the VSM approach can potentially lead to very high performance figures. Results with the much more complex class of psychological verbs related with the expression of emotion exposed the limitations of this method in coping with antonymy. Because of the almost absence of linguistic pre-processing of our approach, such results – specially $P_{@}1 \simeq 0.60$ and $P_{@}5 \simeq 0.45$ – can be seen as *baseline* values for the task of automatically finding verb synonyms for Portuguese.

Acknowledgments

This work was partially supported by grants SFRH/BD/23590/2005 and SFRH/BPD/45416/2008 FCT-Portugal, co-financed by POSI.

References

[1] J. Almeida and U. Pinto. Jspell – um módulo para analise lexica genérica de linguagem natural. In *Actas do X Encontro da Associação Portuguesa de Linguística*, pages 1–15, Évora 1994, 1995.

[2] T. Chklovski and P. Pantel. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In *In Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, Barcelona, Spain, 2004.

[3] K. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

[4] K. W. Church and W. A. Gale. Concordances for parallel text. In *Proceedings of the Seventh Annual Conference of the UW Center for the New OED and Text Research*, pages 40–62, September 1991.

[5] J. R. Curran. From distributional to semantic similarity. Technical report, PhD. Thesis, University of Edinburgh. College of Science, 2004.

[6] T. E. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[7] S. Evert. *The statistics of word cooccurrences : word pairs and collocations.* PhD thesis, Institut fr maschinelle Sprachverarbeitung, Universitt Stuttgart, 2005.

[8] Z. S. Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

[9] S. S. im Walde. Experiments on the automatic induction of german semantic verb classes. *Computational Linguistics*, 32(2):159–194, 2006.

[10] D. Lin. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of COLING-ACL 1998*, volume 2, page 768?773, Montreal, 1998.

[11] M. Sahlgren. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high dimensional vector spaces.* Sics dissertation series 44, Stockholm University, Sweden, 2006.

[12] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA, 1986.

[13] L. Sarmento. BACO - A large database of text and cooccurrences. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*, pages 1787–1790, Genoa, Italy, 22-28 May 2006.

[14] F. Smadja. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19:143–177, 1993.

[15] A. Thanopoulos, N. Fakotakis, and G. Kokkinakis. Comparative evaluation of collocation extraction metrics. In *In Proceedings of the 3rd Language Resources Evaluation Conference*, pages 620–625, 2002.