

Using graph-kernels to represent semantic information in text classification

Teresa Gonçalves and Paulo Quaresma

Departamento de Informática, Universidade de Évora
7000-671 Évora, Portugal
{tcg,pq}@di.uevora.pt

Abstract. Most text classification systems use bag-of-words representation of documents to find the classification target function. Linguistic structures such as morphology, syntax and semantic are completely neglected in the learning process.

This paper proposes a new document representation that, while including its context independent sentence meaning, is able to be used by a structured kernel function, namely the direct product kernel.

The proposal is evaluated using a dataset of articles from a Portuguese daily newspaper and classifiers are built using the SVM algorithm. The results show that this structured representation, while only partially describing document's significance has the same discriminative power over classes as the traditional bag-of-words approach.

1 Introduction

Current Information Technologies and Web-based services need to manage, select and filter increasing amounts of textual information. Text classification allows users, through navigation on class hierarchies, to browse more easily the texts of their interests. This paradigm is very effective both in filtering information as in the development of online end-user services.

As the number of documents involved in these applications is large, efficient and automatic approaches are necessary for classification. A Machine Learning approach can be used to automatically build the classifiers. The construction process can be seen as a problem of supervised learning: the algorithm receives a relatively small set of labelled documents and generates the classifier.

However, as learning algorithms do not directly interpret digital documents, it is required to get a compact representation of their content. The most common approach, called bag-of-words, uses a statistical representation of the document, counting, in any way, its words. Language structures (such as syntax and semantic) typical of natural language documents are completely neglected.

To assess the semantic information value in text classification we developed a structured document representation that includes its logical form (context independent sentence meaning) and is capable of being used by kernel methods.

The semantic information is represented in a graph structure using Discourse Representation Structures (DRS) from the Discourse Representation Theory [13]

(DRT). Learners are built using the Support Vector Machine (SVM) algorithm since it supports structured representations and is known to produce good results on text classification tasks [12]. This setup is applied to dataset of articles from a Portuguese newspaper and the results are compared to common document representation techniques that use the bag-of-words approach.

This paper is organised as follows: Section 2 presents the concepts related to linguistic information and automatic text classification, Section 3 introduces the novel document representation, Section 4 describes the experiments and displays the results while Section 5 evaluates them. Conclusions and future work are pointed out, respectively, on Sections 6 and 7.

2 Concepts and Tools

This section introduces the concepts employed in this work: the representation of linguistic information and kernel functions for structured data. It concludes by presenting the used software tools.

2.1 Linguistic information

The Portuguese language is morphological rich: while nouns and adjectives have 4 forms (two *genders* – masculine and feminine and two *numbers* – singular and plural), a regular verb has 66 different forms (two *numbers*, three *persons* – 1st, 2nd and 3rd and five *modes* – indicative, conjunctive, conditional, imperative and infinitive, each with different number of *tenses* ranging from 1 to 5).

Representation. Morphological information includes word’s stem and its morphological features, like grammatical class and flexion. While some natural language processing tasks use word’s stem, others use its lemma.

Most syntactic language representations are based on the context-free grammar (CFG) formalism introduced by [4] and, independently, by [1]: given a sentence, it generates the corresponding syntactic structure. It is usually represented by a tree structure, known as sentence’s *parse tree*, that contains its constituents structure (such as noun and verb phrases) and words’ grammatical class.

On the other way, some semantic information can be obtained by context independent sentence meaning by examining words’ meaning and combining them. This information can be produced directly from sentence’s syntactic structure, and is named sentence’s *logical form*.

Discourse Representation Theory [13] (DRT) is a dynamic semantic theory that uses a language over Discourse Representation Structures (DRS) to represent dependent context meaning. A simple DRS, is a pair of a set of discourse referents U known as its universe, and a set of conditions C . In its general form, a DRS is represented as a pair $\langle U, C \rangle$. Intuitively the universe collects the discourse entities, while the conditions express entity restrictions (properties, relations). Figure 1 shows the DRS graphical notation for the sentence “Mother observes her daughter playing with the doll”. There are three referents, x , y and z , and

five conditions over them: z refers the mother, y the daughter, and z the doll while the other conditions represent the action **observe** (being x the subject and y the object) and **play**.

x	y	z
mother(x)		
observe(x, y)		
daughter(y)		
play(y, z)		
doll(z)		

Fig. 1. DRS for the sentence “Mother observes her daughter playing with the doll”.

2.2 Kernel functions

Most approaches to text classification use the basic vector space model (VSM) to represent documents. The simplest measure that takes into account words frequency in each document can be naturally reinterpreted as a kernel method [12]. Normalisation and term reduction approaches can also be interpreted as kernel functions [20] and other standard kernels (like the polynomial one) apply non linear transformations to the usual VSM approach.

The *convolution kernel* [9] is the most well-known kernel for structured objects. A structured object is an object formed by the composition of simpler components; frequently, these components are, recursively, simpler objects of the same type. It’s the case of strings, trees or graphs. The convolution kernel definition is based on kernels defined over structure’s components.

Tree kernels. For tree structured objects, the feature space is indexed by subtrees and similarity is based on counting common subtrees. Examples of tree kernels are the subtree kernel [21] and the subset tree kernel [5]. Intuitively, the first counts all common n -descendants until the leaves (being n the root node) and the second adds to that counting all trees considering as leaves all internal nodes. Using some syntactic structures (and possibly some background knowledge) these kernels have produced good results on parse tree ranking [5], predicate argument classification [3] and question classification [23].

Graph kernels. The application of kernels to graph structured data was introduced independently by [7] and [14]. Conceptually they are based on measures over graphs’ walks with common labels: the first counts walks with initial and final common labels and the last calculates probabilities of equal label sequences on random walks.

Gartner *et al.* [8] prove that the computation of a kernel function able to completely recognise graph structure is NP-hard and introduce a walk based kernel function that computes in polynomial time including both previous kernels

as special cases. This kernel, known as *product graph kernel* is based on the concept of the direct product graph counting walks on that graph. Product graphs are a discrete mathematics tool [10] and the direct product graph is between the four most important ones.

2.3 Tools

Next we briefly present the linguistic information extraction tools and learning software used in this study.

Information extraction. POLARIS, a lexical database [15], was used to retrieve Portuguese words' lemma and PALAVRAS [2] parser, developed in the context of the VISL project by the Institute of Language and Communication of the University of Southern Denmark, was used to obtain sentences parse tree.

SIN2SEM [18] tool transforms PALAVRAS parse tree into a DRS revealing sentence logical form. The structure is represented by a two term Prolog predicate: the referents and a set of conditions over them. Although the representation theory considers sentence's meaning in the context where it is produced, SIN2SEM tool builds DRSs considering, only, each sentence independently.

Learning software. For the traditional bag-of-words experiments we used WEKA that implements the sequential minimal optimisation algorithm [17]. WEKA [22] is a software package, developed in New Zealand's Waikato University, implementing a large collection of Machine Learning algorithms.

For the semantic information experiments we used SVM^{light}-TK [16]. This software is an extension of SVM^{light} [11], that uses convolution kernels to represent tree structures.

3 Document's logical form representation

In order to use a kernel method like the SVM algorithm, it is necessary to define a specific kernel or to adapt the problem representation to an existing one. This section describes DRS adaptation to a directed graph enabling the use of document's logical form by kernel methods through the direct product kernel.

3.1 DRS as a directed graph

A DRS can be transformed into a directed graph by mapping DRS's basic units (referents and conditions) into graph's basic ones (vertices and edges) and defining how they interconnect:

- a referent becomes a vertex labelled by referent's name;
- a condition becomes a vertex labelled by condition's name and a set of edges that represent referents' restrictions. Edges are directed from the condition to the restricted referents vertices and are labelled with a number indicating referent's position in the condition.

Using this map, DRS from Figure 1 becomes the directed graph depicted in Figure 2.

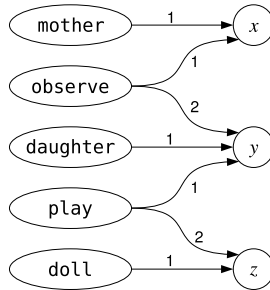


Fig. 2. Directed graph representation of a DRS.

Nevertheless, this direct mapping does not allow the use of the direct product kernel, since SIN2SEM analyses each sentence individually (generating different referents for each one) and does not make any posterior unification. From the definition of the direct product graph:

- its vertex set comprises only common vertices. Using SIN2SEM, it will only be made up of those vertices that represent common conditions;
- its edges set comprises common edges between vertices from the direct product graph. Since all edges connect condition vertices to referent vertices, and there are no common referents between two different DRSs, there are no edges in the direct product graph.

Using the definition of the adjacency matrix, for a direct product graph composed only by a set of vertices, it would always be the zero matrix.

3.2 Referent substitution

Although not being able to use the direct product kernel with the direct mapping, by making the appropriate kind of referent unification/substitution this restriction disappears. The substitutions are related to conditions associated with proper nouns and properties (atomic conditions).

Proper nouns. A proper noun y associated with an entity x is represented by a condition $\text{name}(x, y)$. By replacing all referent instances constrained by the name y (and removing the name condition) we obtain a corpus referent unification connecting all referents that refer to the same proper noun y . Formally, and using the general DRS representation this replacement is given by

$$\langle U, C \rangle \longrightarrow \langle U', C' \rangle$$

where

$$\begin{aligned}
U' &= U - \{x : \exists y \text{ name}(x, y) \in C\} \\
C' &= C - \{\text{name}(x, y) : \text{name}(x, y) \in C\} \\
&\quad - \{c : c \in C \wedge (\exists \text{name}(x, y) \in C : x \in \text{args}(c))\} \\
&\quad \cup \{d : (\exists c \in C, \exists \text{name}(x, y) \in C : x \in \text{args}(c) \\
&\quad \quad \wedge \text{pred}(d) = \text{pred}(c) \wedge \text{args}(d) = \text{subst}(\text{args}(c), x, y))\}
\end{aligned}$$

where $\text{pred}(c)$ returns the condition name c , $\text{args}(c)$ returns c referents list and $\text{subst}(l, a, b)$ is the function that substitutes in l all occurrences of a by b .

Figure 3 shows an example with two DRSs corresponding to sentences “Mother observes her daughter Carlota” and “Daughter Carlota plays with the doll”.

$w \quad x$ mother(w) daughter(x) name(x , Carlota) observe(w , x)	$y \quad z$ daughter(y) name(y , Carlota) doll(z) play(y , z)
↓	↓
w mother(w) daughter(Carlota) observe(w , Carlota)	z daughter(Carlota) doll(z) play(Carlota, z)

Fig. 3. DRS representation before and after proper noun replacement.

Properties. A property associated with an entity x is represented by an atomic condition $\text{prop}(x)$. By replacing all this referent instances by prop (and removing the prop condition) we obtain a sentence unification. Since the same referent can be restricted by more than one property, we should build a list of referent properties and use it in the replacement. Formally, and using the general DRS representation this replacement is given by

$$\langle U, C \rangle \longrightarrow \langle U', C' \rangle$$

where

$$\begin{aligned}
U' &= U - \{x : (\exists c \in C : \text{args}(c) = [x])\} \\
C' &= C - \{c : c \in C \wedge \#\text{args}(c) = 1\} \\
&\quad - \{d : d \in C \wedge (\exists x \in U, \exists c \in C : \text{args}(c) = [x] \wedge x \in \text{args}(d))\} \\
&\quad \cup \{e : (\exists d \in C, \exists x \in U, \exists c \in C : \text{args}(c) = [x] \\
&\quad \quad \wedge x \in \text{args}(d) \wedge \text{pred}(e) = \text{pred}(d) \\
&\quad \quad \wedge \text{args}(e) = \text{subst}(\text{args}(d), x, y) \wedge \text{pred}(c) \in y)\}
\end{aligned}$$

where $pred(c)$ returns the condition name c , $args(c)$ and $\#args(c)$ return, respectively, c referents list and number of referents and $subst(l, a, b)$ is the function that substitutes in l all occurrences of a by b .

This replacement can only be made at sentence level. Figure 4 shows that for the sentence “Mother observes her daughter playing with the doll”.

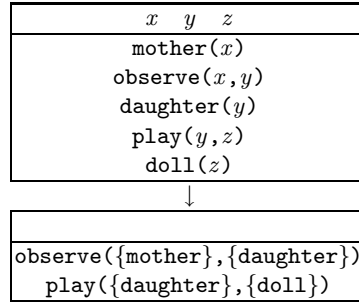


Fig. 4. DRS representation before and after property replacement.

Proper noun and Property. Since the same referent x can be restricted by $name(x, y)$ and $prop(x)$ conditions, it is always necessary to replace a referent by a list. Figure 5 displays the DRS replacement when applying proper noun and property substitutions to the sentence “Mother observes her daughter Carlota that plays with the doll”.

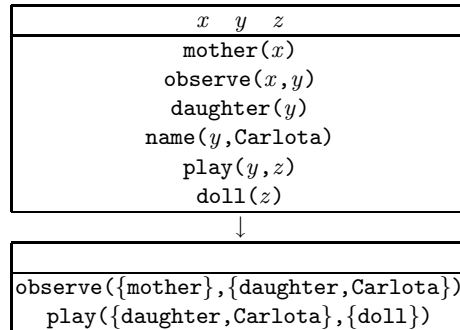


Fig. 5. DRS representation before and after proper noun and property replacement.

3.3 DRS representation

The replacements just described allows the use of the direct product kernel. With it, DRS basic units (referents and conditions) are transformed into graph’s basic ones (vertices and edges) as follows:

- a condition becomes a vertex labelled by condition’s name and a set of edges. Edges are directed from the condition to its referent vertices. Edges are labelled with a number indicating referent’s position in the condition.
- a referent becomes
 - a vertex labelled by referent’s name and an out-degree equal to zero;
 - a sub-graph comprised by a vertex with a proper label (equal for all subgraphs that represent a referent) and a set of edges connecting that vertex to the each element of the substitution set.
- each element of the substitution set becomes a vertex labelled by the property or proper noun identifier that restrict the referent represented by the substitution set. These vertices have out-degree equal to zero.

Figure 6 illustrates the graph representation of the DRS depicted on Figure 5.

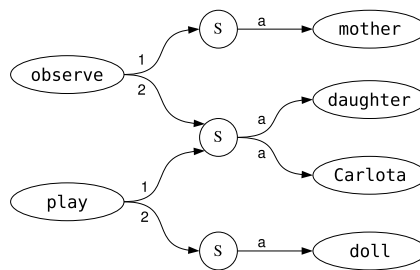


Fig. 6. Directed graph representation of a DRS with referent substitution.

3.4 Document representation

Using DRT, document’s logical form is given by the DRSs set that describes it. In this way, document’s representation is given by the graph constituted by its DRSs. Nevertheless, since some substitutions are made at sentence level, it is possible to have some sub-graph replication in different DRSs. For the proper use of the direct product kernel, it is necessary to remove those replications by changing edges that come from different DRSs to the same subgraph (and removing the replications).

4 Experiments

This section introduces the used dataset, describes the experimental setup and presents the obtained results for the traditional bag-of-words and semantic representations.

4.1 Dataset description

Público is a Portuguese daily newspaper and **Publico** corpus contains its 1994 and 1995 news taken from 9 different sections (used as semantic classes). It totals 101646 documents, where there are 282657 distinct words, and, on average, 512 running words (tokens) and 254 unique words (types) per document.

For the semantic information experiments, a subset of this corpus with the October 1995 news was used. **Pub9510** has 4290 documents, with 70743 distinct words, and, on average, 215 tokens and 124 types per document. Table 1 shows the semantic classes and proportion of documents for each dataset.

section	Publico	Pub9510
	doc %	doc %
ciências, tecnologia e educação (science, technology, education)	6.2	6.7
cultura (culture)	15.5	14.5
desporto (sports)	9.9	10.3
diversos (diverse)	8.2	8.1
economia (economy)	13.3	10.5
local (local)	17.2	21.3
mundo (world)	9.4	9.3
nacional (national)	9.2	10.3
sociedade (society)	11.2	9.1

Table 1. **Publico** and **Pub9510** corpora: classes and proportion of documents.

4.2 Experimental setup

Traditional bag-of-words representations used a linear kernel while the semantic information ones were run with the subset tree kernel. WEKA was run with default parameters (normalised training data and $c=1$, the trade-off between training error and margin) and SVM^{light} -TK was run with $L=0.001$ (decay factor) and $c=10$ (trade-off between training error and margin). A train-and-test procedure was applied with 33% of documents used for testing.

Learner’s performance was analysed through precision (π), recall (ρ) and F_1 (f_1) measures [19] of each category (obtained from classification’s contingency table: prediction *vs.* manual classification). For each one, we calculated the micro- ($^{\mu}$) and macro-averages (M) and made significance tests regarding a 95% confidence level.

4.3 Bag-of-words representation

First we considered the traditional bag-of-words representation. It’s the typical representation used in Information Retrieval techniques and it serves as a baseline experiment. We used the word and its lemma (**1m**) as indexing terms and considered several filtering and weighting techniques. Word’s selection was made

using three classes of experiments: stopword elimination (**st**), a filtering function (word frequency – **fr** and mutual information – **mi**) and a threshold value (**τ**). To weight the selected terms we used the three usual components: document (term frequency – **t**), collection (no component – **x**, and inverse term frequency – **f**) and normalisation (co-sin – **c**). All these options can be graphically represented in a tri-dimensional space with normalisation, selection and weighting axes. In turn, selection and weighting techniques can also be represented in other three-dimensional spaces. Figure 7 illustrates this, with the marking point corresponding to the experiment with stopword elimination, using lemma as the indexing term, mutual information as the filtering function with threshold equal to one and **tfidf** as the weighting technique.

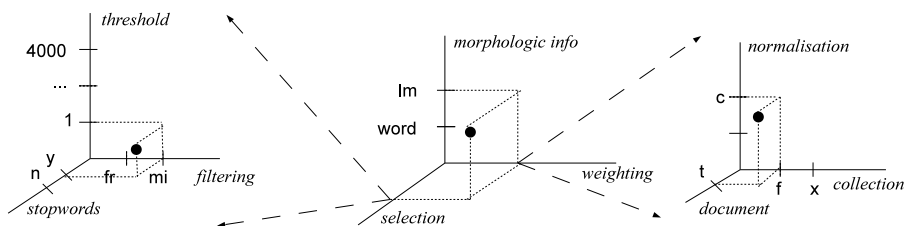


Fig. 7. Tree-dimensional spaces for normalisation, selection and weighting options.

Results. We made experiments with all combinations of options described above and threshold values ranging from 1 to 4000 (this value indicates the smallest frequency above which the term is selected), in a total of 88 different runs. Table 2 shows the minimum, maximum, average and standard deviation values for the micro- and macro averages of the performance measures.

	π^μ	ρ^μ	f_1^μ	π^M	ρ^M	f_1^M
min	.787	.787	.787	.779	.770	.774
max	.843	.843	.843	.842	.831	.836
avg	.824	.824	.824	.821	.810	.815
std	.012	.012	.012	.013	.013	.013

Table 2. Publico min, max, avg and std values for micro- and macro averages.

For each performance measure, we searched those experiments with values having no significant difference with the maximum. There were 5 experiments with all 6 performance measures in that set:

- word lemmatisation (**lm**), with threshold value 1 (**τ**₁);
- stopword elimination (**st**), term frequency filtering function (**fr**) and threshold value 50 (**τ**₅₀);

- stopword elimination, mutual information filtering function (**mi**), **tfidf** weighting technique (**tfc**) and threshold value 50 (τ_{50}).

Table 3 displays those performance measures. Boldface values have no significant difference between them.

	π^μ	ρ^μ	f_1^μ	π^M	ρ^M	f_1^M
lm.fr.txc.l ₁	.840	.840	.840	.839	.828	.833
lm.fr.tfc.l ₁	.843	.843	.843	.842	.831	.836
st.fr.txc.l ₅₀	.839	.839	.839	.837	.826	.831
st.fr.tfc.l ₅₀	.840	.840	.840	.838	.828	.832
st.mi.tfc.l ₅₀	.840	.840	.840	.839	.828	.833

Table 3. Publico performance values with bag-of-words representation.

4.4 Semantic representation

One property of the graph that represents sentence’s logical form is that it has no cycles, allowing the transformation of this structure into a tree, a less complex one. The use of tree structures enables the application of tree kernels that, in turn, are more performant than the direct product kernel.

This transformation comprises two steps:

- the creation of a new vertex (with equal label for all documents) that constitutes the tree’s root and a set of directed edges from it to each vertex that identifies DRS conditions;
- the replication of DRS conditions’ sub-graphs, when two or more conditions are connected to the same vertex (in-degree greater than one).

The obtained tree has the following characteristics: the root represents the document, nodes at depth one represent DRS conditions and each of their child represent the restricted referents. For referents substituted by proper nouns and/or properties another level exists for representing the substitution set.

Since it is necessary to point referent’s position inside the condition the tree should be ordered. On the other hand, since DRS conditions constitute a set, it is necessary to order them alphabetically; the same kind of ordering needs to be done over the vertices that constitute a substitution set.

This document representation that incorporates its logical form was named *discourse-structure representation*. The different kind of substitutions proposed on Section 3.2 gives rise to distinct discourse-structure representations.

Figure 8 illustrates a two sentence document, the SIN2SEM output and the discourse-structure representation with both substitutions. This discourse-structure representation corresponds to the graph illustrated on Figure 3.

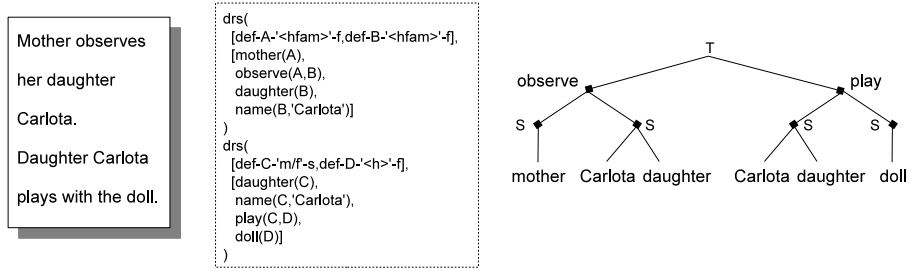


Fig. 8. Original document, SIN2SEM output and discourse-structure representation.

Results. For the discourse-structure representation (**dis**) we considered two different kinds of referent substitution: proper nouns (**noun**) substitution and proper nouns and property substitution (**noun+pro**). For each one we also tried to use the first n DRSs of each document with $n \in \{1, 3, 5, 10\}$.

Aiming to access the structure influence on the classification process, we also considered a bag-of-words representation with the words extracted from the discourse-structure representation. Table 4 shows the obtained performance measures (presenting in boldface the values with no significant difference when compared with best value obtained for each measure).

		π^μ	ρ^μ	f_1^μ	π^M	ρ^M	f_1^M
noun	dis	.655	.655	.655	.732	.599	.623
	dis ₁	.364	.364	.364	.533	.278	.288
	dis ₃	.484	.484	.484	.604	.418	.451
	dis ₅	.545	.545	.545	.660	.481	.510
	dis ₁₀	.593	.593	.593	.692	.538	.567
	bag	.821	.821	.821	.816	.808	.810
noun+pro	dis	.833	.833	.833	.831	.817	.820
	dis ₁	.471	.471	.471	.484	.437	.445
	dis ₃	.679	.679	.679	.671	.645	.651
	dis ₅	.740	.740	.740	.735	.710	.717
	dis ₁₀	.787	.787	.787	.780	.772	.773
	bag	.814	.814	.814	.822	.788	.788

Table 4. Pub9510 performance values with semantic information.

5 Evaluation

Looking at traditional bag-of-words approach (Table 3) one can say that it was possible to reduce the number of attributes (τ_{50}) without compromising performance. However, these values were achieved only for experiments with the original words and not with lemmatisation. It also seems that the mutual information filtering function should be used with the **tfidf** weighting one, while when filtering by the term frequency, the weighting function seems indifferent.

On the other hand, when using document’s semantic information (Table 4), a structured representation with the proper noun and property substitutions (`noun+pro`) seems to add valuable information when compared to the corresponding bag-of-words representation (it has better macro- recall and f_1 values).

For comparing both linguistic levels, one using morphological and the other using semantic information, we elected a “best” experiment for each level. For the morphological level we chose the bag-of-words representation using word’s lemma and `tfidf` weighting measure with co-sin normalisation (`lm.fr.tfc.t1`), while for the semantic one we chose a discourse-structure representation with proper nouns and property conditions substitutions (`dis.noun+pro`).

Since discourse-structure experiments used a subset of `Publico` corpus, a SVM was run on this subset (`Pub9510`) for the chosen bag-of-words representation. Table 5 displays the performance values for each linguistic information level (once again, boldface values points to values with no significant differences).

	π^μ	ρ^μ	f_1^μ	π^M	ρ^M	f_1^M
Morphological	.855	.855	.855	.854	.840	.844
Semantic	.833	.833	.833	.831	.817	.820

Table 5. Pub95 performance values for both linguistic information levels.

Although the attainment of the semantic representation is more time consuming than the morphological one, it is possible to say that both representations have the same discriminative power over classes, since there is no significant difference between all performance values. Moreover, we can also say that semantic information uses a valid form of attribute selection since it has 46186 types while the traditional bag-of-words approach has 70743.

6 Conclusions

This paper presents a series of experiments aiming at comparing our proposal of incorporating linguistic information using structured representations with the usual methods adopted on text classification problems. The experiments were conducted on a Portuguese written dataset.

Concerning morphological information, results show that, when properly combined, word normalisation, filtering and weighting functions and threshold values can sharpen performance. Comparing `Publico` dataset results with previous work on text classification with Portuguese written documents [6], one can conclude that the best combination depends on the dataset (or its domain).

The results also show that, when using semantic information, the discourse-structure representation with proper noun and property substitutions presents at least the same discriminative power as the non-structured representation.

Moreover, the traditional bag-of-words approach (morphological information) and the proposed use of semantic information, show equivalent performance values. This statement demonstrates that both representations, one based on statistics over words and other based on document’s meaning, are valid.

Considering the number of types used by morphological and semantic representations, it is possible to say that documents' logical form performs a valid form of attribute selection: an about 30% reduction was accomplished.

Finally, one can conclude that the proposed discourse-structure representation is able to contain document's logical form and seems promising since at this time it only describes document's meaning partially. We believe that by perfecting document's logical form, the semantic representation performance will be higher than the morphological one.

7 Future work

Regarding future work, we intend to perform further tests on different collections/domains and languages. It will be important to evaluate if these results are bound to the Portuguese language and/or the kind of the dataset domain. We also want to verify the validity of this approach using the graph kernel directly.

On the other way, it is possible to obtain document's semantic representation closer to its real meaning by eliminating some of the known limitations of the used natural language tools. Although always generating an output, even in presence of incorrect or incomplete sentences, in some situations (for example, in presence of interrogative sentences) PALAVRAS generates incorrect parse trees. These errors are then propagated to DRS generation since parse trees become the input of SIN2SEM.

SIN2SEM can also be refined by removing some of its limitations. For example, it does not implement verb sub-categorisation, being unable to add specific verb properties that would otherwise give a closer sentence's meaning. It also does not give the proper output for noun clauses with more than one entity.

Finally, the DRSs would translate more accurately the true meaning of documents whith the incorporation of other linguistic information like synonymous, anaphora resolution and identification of named entities.

References

1. J. Backus. The syntax and semantics of the proposed international algebraic of the Zurich ACM-GAMM Conference. In *Proceedings of the International Conference on Information Processing – IFIP Congress*, pages 125–132. UNESCO, Paris, 1959.
2. E. Bick. *The Parsing System PALAVRAS – Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus University Press, 2000.
3. S. Bloehdorn and A. Moschitti. Combined syntactic and semantic kernels for text classification. In *ECIR-07, 29th European Conference on Information Retrieval*, pages 307–318. Springer, 2007.
4. N. Chomsky. Three models for the description of language. *IRI Transactions on Information Theory*, 2(3):113–124, 1956.
5. M. Collins and N. Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *ACL-02, 30th Annual Meeting of the Association for Computational Linguistics*, pages 263–270, 2002.

6. T. Gonçalves, C. Silva, P. Quaresma, and R. Vieira. Analysing part-of-speech for Portuguese text classification. In *CICLing-06, 7th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 551–562, Mexico City, MX, February 2006. Springer.
7. T. Gärtner. Exponential and geometric kernels for graphs. In *NIPS-02, 16th Neural Information Processing Systems – Workshop on Unreal Data: Principles of Modeling Nonvectorial Data*, 2002.
8. T. Gärtner, P.A. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. In B. Schölkopf and M.K. Warmuth, editors, *COLT-03, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop*, LNAI 2777, pages 129–143. Springer Berlin Heidelberg, 2003.
9. D. Haussler. Convolution kernels on discrete structures. Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.
10. W. Imrich and S. Klavzar. *Product Graphs: Structure and Recognition*. John Wiley, 2000.
11. T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.
12. T. Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer Academic Publishers, 2002.
13. H. Kamp and U. Reyle. *From Discourse to Logic: An Introduction to Model Theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, volume 42 of *Studies in Linguistics and Philosophy*. Springer, 1993.
14. H. Kashima and A. Inokuchi. Kernels for graph classification. In *ICDM-02, IEEE International Conference on Data Mining – Workshop on Active Mining*, 2002.
15. J.G. Lopes, N.C. Marques, and V.J. Rocio. Polaris: PORTuguese Lexicon Acquisition and Retrieval Interactive System. In *The Practical Applications of Prolog*, page 665. Royal Society of Arts, 1994.
16. A. Moschitti. A study on convolution kernels for shallow semantic parsing. In *ACL-04, 42nd Annual Meeting on Association for Computational Linguistics*, pages 335–342, Barcelona, SP, 2004.
17. J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, , and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 185–208. MIT Press, 1999.
18. P. Quaresma, L. Quintano, I. Rodrigues, and P. Salgueiro. The University of Évora approach to QA@CLEF-2004. In *Multilingual Information Access for Text, Speech and Images*, LNCS 3491, pages 534–543. Springer-, 2005.
19. G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
20. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
21. S.V. Vishwanathan and A.J. Smola. Fast kernels on strings and trees. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 569–576. MIT Press, Cambridge, MA, 2003.
22. I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, US, 2nd edition, 2005.
23. D. Zhang and W.S. Lee. Question classification using support vector machines. In *SIGIR-03, 26th ACM International Conference on Research and Development in Information Retrieval*, pages 26–32, 2003.