Noname manuscript No.
(will be inserted by the editor)

Classifying unlabeled short texts using a fuzzy declarative approach

Francisco P. Romero · Pascual Julian-Iranzo · Andres Soto · Mateus Ferreira-Satler · Juan Gallardo-Casero

the date of receipt and acceptance should be inserted later

Abstract Web 2.0 provides user-friendly tools that allow persons to create and publish content online. User generated content often takes the form of short texts (e.g., blog posts, news feeds, snippets, etc). This has motivated an increasing interest on the analysis of short texts and, specifically, on their categorisation. Text categorisation is the task of classifying documents into a certain number of predefined categories. Traditional text classification techniques are mainly based on word frequency statistical analysis and have been proved inadequate for the classification of short texts where word occurrence is too small. On the other hand, the classic approach to text categorization is based on a learning process that requires a large number of labeled training texts to achieve an accurate performance. However labeled documents might not be available, when unlabeled documents can be easily collected.

This paper presents an approach to text categorisation which does not need a preclassified set of training documents. The proposed method only requires the category names as user input. Each one of these categories is defined by means of an ontology of terms modelled by a set of what we call *proximity equations*. Hence, our method is not category occurrence frequency based, but highly depends on the definition of that category and how the text fits that definition. Therefore, the proposed approach is an appropriate method for short text classification where the frequency of occurrence of a category is very small or even zero. Another feature of our method is that the classification process is based on the ability of an extension of the standard Prolog language, named Bousi∼Prolog, for flexible matching and knowledge representation. This declarative approach provides a text classifier which is quick and easy to build, and a classification process which is easy for the user to understand. The results of experiments showed that the proposed method achieved a reasonably useful performance.

Keywords Text Categorization · Ontologies · Thesauri · Unlabeled Short Texts

Francisco P. Romero · Pascual Julian-Iranzo · Mateus Ferreira-Satler · Juan Gallardo-Casero
Department of Information Technologies and Systems, University of Castilla La Mancha,
Paseo de la Universidad, 4, 13071 - Ciudad Real,Spain
E-mail: FranciscoP.Romero@uclm.es

Andres Soto
Department of Computer Science, Universidad Autònoma del Carmen,
CP 24160, Ciudad del Carmen, Campeche, Mèxico

1 Introduction

User generated content has been a major aspect of Web 2.0 era. Often these contents are formed by short texts which are created on daily basis as on-line evaluations of commercial products, posts of blogs or comments in social networks, news feeds, web pages titles, snippets, etc. In March 2011, Blogpulse Stats[1] shown that the total amount of identified blogs was greater than 158 million and more than one million blog posts were indexed a day. This proliferation of contents has motivated that in recent years, the computational linguistics community shown an increasing interest on the efficient analysis of short texts. Moreover, classification of short text messages is one of the most useful method to avoid becoming overwhelmed by the raw data.

Classification or categorisation is the task of assigning objects to one of several predefined categories. Text categorisation (also known as text classification) is the task of automatically sorting a set of documents into categories from a predefined set (Sebastiani, 2002). In automatic text categorisation, the decision criterion of the text classifier is usually learned from a set of training documents, labelled for each class (Meretakis et al, 2000). This type of learning is called supervised learning because a supervisor, the human who defines the classes and labels training documents, serves as a teacher directing the learning process (Manning et al, 2008).

The text categorisation process is usually split into several steps. In the first place, a set of previously classified documents is assumed to be available. Those documents, which are labelled for each class, constitute the training document set. Using a learning algorithm, the decision criterion of the text classifier is learned automatically from the training document set by an induction process. A set of rules which describe the different categories is obtained after this step. This description will be used later to classify new documents, not included in the training set. Therefore, the degree of precision of the classifying method heavily depends on the decision criterion; that is the set of rules previously mentioned.

Traditional text classification techniques, mainly based in word frequency statistical analysis, work well when the word frequency is high enough to capture the semantics of the document. However, when dealing with shorter text messages, traditional techniques will not perform as well as they would have performed on larger texts (Sriram et al, 2010). This behavior conforms with our initial intuition, since the word occurrence is too small, these word frequency based techniques do not provide sufficient knowledge about the text itself, what prevents a correct classification. Thus, short text categorization cannot be carried out only relying in statistical methods, it is also necessary to exploit the semantic relationships between words.

Another problem with these methods is related with the exponential growth of the number of labeled document training sets required as the desired precision degree of the method increases. This way, the time and effort required for collecting and preparing an adequate training set could be a restriction, and probably, prohibitive. This is an important issue in order to classify short texts, because there are a lot of available short texts, but the majority of them are unlabeled.

There are several approaches to address the classification problems induced by short texts. Faguo et al (2010) proposed a novel method for short text classification based on statistics and rules. Their proposal achieve a high performance in terms of precission and recall, but requires the participation of a human agent. So it is not an

[1] `http://blogpulse.com/`

automatic classification method. Liu et al (2008) use short snippets of blogger's posts to user modeling. The proposal is based on a two-layer classification model, one for the probability of a snippet belonging to each category and another for feature selection. That approach needs a huge volume of blog posts to train the first layer classifiers for blog snippets. In order to reduce the user participation, a method that combine labelled and unlabelled documents is presented in (Zelikovitz and Hirsh, 2000). That method for classifying short texts uses a combination of labeled training data plus a secondary corpus of unlabeled but related longer documents.

Another relevant approach is the one proposed by (Boutari et al, 2010), a deep study about the use of five term concept association measures to drive text expansion prior to performing classification and clustering of short texts. That work investigates a term expansion approach based on analyzing the relationships between the term concepts present in the concept lattice associated with a document corpus.

On the other hand, the approach of categorising texts based on lists of categories and unlabelled documents has been attempted previously in the literature: A generalised bootstrapping algorithm for text categorisation is proposed in (Gliozzo et al, 2005). In that paper, the categories are described by relevant seed features. Its main contributions are the introduction of two unsupervised steps in order to improve the initial categorisation step of the bootstrapping scheme. Gliozzo's approach (Gliozzo et al, 2005) has been improved in recent papers. In (Barak et al, 2009) the words that are likely to refer specifically to the meaning of the category name are extracted from WordNet (Fellbaum, 1998) and Wikipedia [2]. The final definition of each category is obtained through a disambiguation process of the extracted words using an LSA model. The results obtained by this approach increase the classification precision of the previously related works.

Another approach to unsupervised text categorisation is the one proposed in (Ko and Seo, 2009), which is an extension of a previously related work developed by the same authors (Ko and Seo, 2004). In this paper, the text classifier is built by using only unlabelled documents and the label of each category. The learning method is based on a bootstrapping algorithm and feature projection techniques. The proposed method can also be used as an assistant tool for easily creating training data for supervised methods. The achieved results are reasonably useful compared to supervised methods.

Previously mentioned papers use an unsupervised learning approach for training the document classifier. Although they do not need previously classified documents, all of them require a training phase. During this phase, they apply different methods of analysis, extraction, and knowledge representation to the document collection which should be classified.

The approach introduced in this paper for text categorisation does not require any previously classified collection or training phase. Knowledge required for text categorisation is obtained from thesauri and ontologies like WordNet (Fellbaum, 1998) or ConceptNet (Liu and Singh, 2004) by measuring the semantic closeness between concepts. As a first stage, a proximity relation is generated. Afterwards, the classifying process uses it in combination with a flexible search method implemented by an extension of the Prolog programming language. Moreover, our method is not being directly based on an analysis of the frequency of occurrence of a certain category, but depends highly on the definition of that class (through an ontology) and how the text fits that

[2] http://www.wikipedia.org

definition. Therefore, the proposed approach is an appropriate method for short text classification where the frequency of occurrence of a category is very small or even zero.

Under this novel approach, the only input required is the list of category names. These names are used to retrieve the semantic descriptions of the concepts involved with each one of the categories from the previously mentioned thesauri and ontologies. In this way, the category names are transformed into a set of concept descriptions. More precisely, each one of these categories is defined by means of an ontology or thesaurus of terms modelled by a set of proximity equations. These category descriptions are the main input of the classification process. At this point, it is worth noting that text categorisation based on concepts is an approach to overcome the main difficulties inherent to classification based only on lexical aspects (Garcés et al, 2006), as long as accurate and explicit concept definitions are available

As was just mentioned, in our method the classification process is based on the abilities of an extension of the standard Prolog language, called Bousi∼Prolog for flexible matching and knowledge representation. This extension offers a fuzzy unification mechanism based on proximity relations which allow the flexible search of concepts in documents (Iranzo et al, 2009). Hence, our method implements a clean separation between knowledge (refined by an ontology), logic (expressed by rules) and control of the underlying programming language. The combination of these components provides a declarative approach to text classification, where a text classifier is easier to build than usually it is. At the same time, the classification process became more understandable for the user, since it mainly relies on an ontology description. While most of the work in classification nowadays is founded on statistical methods, this paper takes a Semantic Web and Soft-Computing approach using thesauri as a source of domain knowledge

In order to evaluate the performance of the classification method, four distinct text categorization tasks have been carried out. In each case, the examples are short texts that have been from the World Wide Web (snippets, newswires, web titles, RSS feeds). The experimental results show that different types of proximity relations, used as input for the classification process, produce diverse results. Although some difficulties exist with some of the input relations, accurate results have been generally obtained by our method, equivalent to the ones referenced in literature (as will be shown in Section 4).

The paper is organised as follows: Section 2 includes a concise description of proximity relations between concepts and the Bousi∼Prolog language which offers the required mechanisms to implement a text categorisation method using a declarative approach. Section 3 describes our method in detail including an explanatory example. Section 4 explains the experiments and the results obtained in order to verify how good the solution is. Finally, our conclusions and future work are outlined in Section 5.

2 Background

In this section, the fundamental concepts supporting our approach to text classification are explained beginning with the notion of proximity relation. These relations are used to express the semantic proximity between concepts included in ontologies and thesauri. Four relevant relations appropriate for the approach are introduced. Later, the main features of the Bousi∼Prolog programming language are explained. Bousi∼Prolog can be considered a Prolog extension which implements proximity-based fuzzy unification. Thus, it is a declarative programming language, well suited to flexible query answering.

2.1 Proximity Relations between concepts

Binary fuzzy relations were introduced by Zadeh in (Zadeh, 1965). Formally, a *binary fuzzy relation* on a set U is a fuzzy subset on $U \times U$ (that is, a mapping $U \times U \longrightarrow [0, 1]$). Given a and b two elements in U, an *entry* of a fuzzy relation will be denoted as $\mathcal{R}(a, b) = \alpha$, being α its *relationship* degree. A binary fuzzy relation $\mathcal{R}$ is said to be a *proximity relation* if it fulfills the *reflexive* property (i.e. $\mathcal{R}(x, x) = 1$ for any $x \in U$) and the *symmetric* property (i.e. $\mathcal{R}(x, y) = \mathcal{R}(y, x)$ for any $x, y \in U$). A proximity relation which in addition fulfills the *transitive* property (i.e., $\mathcal{R}(x, z) \geq \mathcal{R}(x, y) \triangle \mathcal{R}(y, z)$, for any $x, y, z \in U$) is said to be a *similarity relation*. The operator '$\triangle$' is an arbitrary t-norm. The notion of transitivity above is $\triangle$-transitive. If the operator $\triangle = \wedge$ (that is, it is the minimum of two elements), we speak of *mim*-transitive or $\wedge$-transitive. This is the standard notion of transitivity used in this paper.

According to the approach introduced in this paper, concepts with a positive closeness relation to a category name should be identified, including the degree of the relationship, and formalised into a fuzzy relation. For this purpose, knowledge bases, ontologies and thesauri are used in order to estimate the relationship degree between two concepts, i.e. how semantically similar or close they are. Several methods have been proposed in the literature to compute semantic closeness. Le et al. (Le and Goh, 2007) presented a survey of these methods. More specifically, they explored the existing techniques to calculate semantic resemblance and highlight the advantages and disadvantages of each one. A taxonomy of methods to measure concept resemblance is shown in Figure 1:

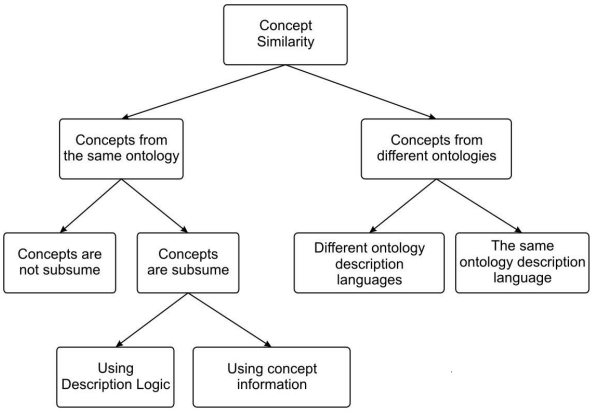

Fig. 1 A taxonomy of approaches to compute semantical relationships between concepts.

Specifically, in order to calculate the semantic closeness between concepts, we use different conceptual relations included in ontologies, thesauri and dictionaries like Concept Net and WordNet. In the following paragraphs we summarise the main conceptual relations used in this paper:

Structural Analogy ConceptNet (Liu and Singh, 2004), is a freely available common sense know-ledge base and natural language processing toolkit[3]. ConceptNet is constructed as a network of semi-structured natural language fragments. The ConceptNet Java API has a `GetAnalogousConcepts()` function that returns a list of structurally analogous concepts, given a source concept. The degree of structural analogy between these terms and the source concept is also provided by ConceptNet. Then, for each element b in the list of structurally analogous concepts to a source concept a and their degree of relationship α, we build an entry $\mathcal{R}(a, b) = \alpha$ of a fuzzy relation.

Example 1 The set of entries shown below is a partial view of the original output obtained by the `GetAnalogousConcepts()` function and the source concept "wheat".

$$\begin{array}{ll} \mathcal{R}(wheat, bean) = 0.315, & \mathcal{R}(wheat, horse) = 0.32, \\ \mathcal{R}(wheat, corn) = 0.32, & \mathcal{R}(wheat, human) = 0.2. \\ \mathcal{R}(wheat, grass) = 0.315, & \end{array}$$

It is important to note that structural analogy is not strictly a semantic measure but a resemblance degree according to the characteristics of the represented concepts. In ConceptNet, two nodes are analogous if their sets of incoming edges overlap. Two concepts with a positive degree of structural analogy share similar properties and have similar functions. For example, "scissors", "razor", "nail clipper", and "sword" are perhaps like a "knife" because they are all "sharp", and can be used to "cut something".

Contextual neighborhood In ConceptNet (Liu and Singh, 2004) the contextual neighbourhood around a concept is found by performing spreading activation from that source concept, radiating outwardly to include other concepts. The relatedness of any particular concept with some other concept is a function of the number of links and the number of paths between them, and the directionality of the edges. In addition, pairwise resemblance of concepts indicates the mutual information shared between two concepts, allowing similar nodes to be aggregated, leading to a more accurate estimation of contextual neighbourhood. For example, concepts like "menu", "order food" or "waiter" are in the contextual neighborhood of the source concept "restaurant".

WordNet Similarity WordNet is another possible source of knowledge to be used. WordNet (Fellbaum, 1998) iis a freely available software package that offers an implementation of six measures of semantic resemblance and three measures of relatedness between pairs of concepts (or word senses), all of which are based on the WordNet lexical database. Three resemblance measures are based on path lengths between concepts, and the three remaining resemblance measures are based on information content, which is a corpus-based measure of the specificity of a concept. Finally, one of the three measures of relatedness is path based, and classifies relations in WordNet as having a direction, and the last two measures incorporate information from WordNet glosses as a unique representation for the underlying concept. An option to define closeness relations could be the combined use of a dictionary that provides a definition of a word and the WordNetSimilarity API [4] that provides some relationship degrees between two words.

[3] Available at `http://conceptnet.media.mit.edu/`.

[4] http://wn-similarity.sourceforge.net

Example 2 The definition of "wheat" can be extracted from WordNet ("annual or biennial grass having erect flower spikes and light brown grains...") and used to compute the closeness of the terms expressed in the definition using the WUP measure (Wu and Palmer, 1994) provided by WordNetSimilarity.

$$\begin{array}{ll} \mathcal{R}(wheat, annual) = 0.696, & \mathcal{R}(wheat, spike) = 0.174, \\ \mathcal{R}(wheat, biennial) = 0.696, & \mathcal{R}(wheat, light) = 0.231, \\ \mathcal{R}(wheat, grass) = 0.923, & \mathcal{R}(wheat, brown) = 0.174, \\ \mathcal{R}(wheat, flower) = 0.692, & \mathcal{R}(wheat, grain) = 0.273. \end{array}$$

Synonymy-based similarity Wordnet is a thesaurus but is also an ontology (Fellbaum, 1998). It groups English words into sets of synonyms called synsets. It also provides short, general definitions, and records the various semantic relations between these synonym sets. Thus, it is possible to know the meaning of a word and, at the same time, to associate it with other words using ontological relations like synonymy, antonymy, hyperonymy, hyponymy, meronymy, etc. The semantic relationships and the synsets can be used to obtain a degree of closeness between two words and thus, the set of words related to another word.

In (Soto et al, 2008) a formula is introduced in order to calculate concept resemblance based on the synonymy degree between those concepts. The degree of relationship between two concepts depends on the number of WordNet meanings that they share. More formally, we can define a semantic relation based on the meaning shared by two words. Let $M(t)$ be the set of different meanings associated with a certain term t and $|M(t)|$ the number of meanings of the term t, then the fuzzy relation $\mathcal{R}$ between two terms t_1,t_2, defined in WordNet, expressing the degree of proximity between both terms, is defined as:

$$\mathcal{R}(t_1, t_2) = \frac{|M(t_1) \cap M(t_2)|}{|M(t_1)|} \tag{1}$$

Before ending this subsection, it is important to mention that all of the above detailed methods build a partial view of a fuzzy relation (certainly, only the entries connecting a category with a set of related terms are produced). Therefore, some post-processing of that partial relation may be needed depending on the features of the semantic relationship that we wish to establish. If we need to work with a proximity relation, it is necessary to build the reflexive, symmetrical closure of the partial relation. On the other hand, if the desired relation is a similarity, we need to build the reflexive, symmetrical and transitive closure of the partial relation. Fortunately, Bousi∼Prolog gives automatic support for the generation of these kinds of closures, as will be commented in the next subsection.

2.2 Bousi∼Prolog and flexible search

Bousi∼Prolog Iranzo and Rubio-Manzano (2009) Julián-Iranzo and Rubio-Manzano (2009),Iranzo et al (2009) is a fuzzy logic programming language whose main objective is to make the query answering process flexible and to manage the vagueness occurring in the real world by using declarative techniques. Its design has been conceived to make a clean separation between Logic, Vague Knowledge and Control. In a Bousi∼Prolog program Logic is specified by a set of Prolog facts and rules, Vague Knowledge is mainly specified by a set of, what we call, *proximity equations*, defining a fuzzy binary relation (expressing how close two concepts are), and Control is let automatic to the system,

through a "weak" SLD resolution operational mechanism. *Weak SLD resolution* is an enhancement of the SLD resolution principle where the classical syntactic unification procedure is replaced by a fuzzy unification algorithm based on proximity relations defined on a syntactic domain. Informally, this *weak unification* algorithm states that two terms $f(t_1, ..., t_n)$ and $g(s_1, ..., s_n)$ weakly unify if the root symbols f and g are close, with a certain degree, and each of their arguments t_i and s_i weakly unify. Therefore, the weak unification algorithm does not produce a failure if there is a clash of two syntactical distinct symbols, whenever they are approximate, but a success with a certain approximation degree. Hence, Bousi∼Prolog computes substitutions as well as approximation degrees.

Bousi∼Prolog is implemented as an extension of the standard Prolog language. It is publicly available and can be executed via web [5]. Currently it is delivered in two implementation formats: a high level and a low level implementation. The high level implementation Iranzo et al (2009) is written in Prolog through a meta-interpreter. One step further, in Julián-Iranzo and Rubio-Manzano (2009) we presented the structure and main features of a low level implementation for Bousi∼Prolog, consisting in a compiler and an enlargement of the Warren Abstract Machine able to incorporate fuzzy unification and to execute BPL programs efficiently.

The Bousi∼Prolog syntax is mainly the Prolog syntax but enriched with a built-in symbol "∼" used for describing proximity relations [6] by means of what we call a "proximity equation". *Proximity equations* are expressions of the form:

```
<symbol> ~ <symbol> = <proximity degree>.
```

Although, a proximity equation represents an entry of an arbitrary fuzzy binary relation, its intuitive reading is that two constants, n-ary function symbols or n-ary predicate symbols are approximate or similar with a certain degree. That is, a proximity equation $a \sim b = \alpha$ can be understood in both directions: a is approximate/similar to b and b is approximate/similar to a with degree α. Therefore, a Bousi∼Prolog program is a sequence of Prolog facts and rules followed by a sequence of proximity equations. The following example illustrates both the syntax and some features of the weak resolution semantics.

Example 3 Assume a fragment of a deductive database that stores information about people and their preferences on teaching.

```
% PROXIMITY EQUATIONS  % FACTS
chemistry~math=0.6.    likes_teaching(john,physics).   has_degree(john,physics).
physics~math=0.8.      likes_teaching(mary,chemistry). has_degree(mary,chemistry).
physics~chemistry=0.8.

% RULES
can_teach(X,M):-has_degree(X, M), likes_teaching(X, M).
```

In a standard Prolog system, if we ask about who can teach mathematics, launching the goal "`?-can_teach(X,math).`", the system do not produce any answer. However the Bousi∼Prolog system answers "`X=john with 0.8`" and "`X=mary with 0.6`". In order to understand this behavior, it is interested to reproduce the different steps that the Bousi∼Prolog system follows to obtain these answers:

[5] http://dectau.uclm.es/bousi

[6] Actually, fuzzy binary relations which are automatically converted into proximity or similarity relations.

1. At compiling time, the proximity equations (jointly with the rest of the program code) are translated to a internal representation. Here, for our explanatory purposes, the important thing is that they are interpreted as defining a fuzzy relation characterized by the set of entries: $\{\mathcal{R}(physics, math) = 0.8, \mathcal{R}(physics, chemistry) = 0.8, \mathcal{R}(chemistry, math) = 0.6\}$. Then, the reflexive, symmetric closure of this fuzzy relation is generated, constructing a proximity relation[7]. In other words, for the proximity equation "`physics~math=0.8`", the following entries are produced: $\{\mathcal{R}(physics, physics) = 1, \mathcal{R}(physics, math) = 0.8, \mathcal{R}(math, physics) = 0.8, \mathcal{R}(math, math) = 1\}$.
2. At running time, the goal is solved by weak SLD resolution. The operational mechanism of the Bousi∼Prolog system tries to unify the current goal "`can_teach(X,math)`" and the head of a rule, in this case: `can_teach(X1,M1)`. The result of this first step is a partial answer [8] `({X=X1, M1= math}, 1)` and a new goal to resolve: `has_degree(X1, math), likes_teaching(X1, math)`. Then the subgoal "`has_degree(X1, math)`" is selected and the resolution process continues. It tries to unify this subgoal with the fact "`has_degree(john, physics)`". Because there exists the entry "$\mathcal{R}(physics, math) = 0.8$" in the constructed proximity relation (that is, `physics` is close to `mathematics`, with approximation degree 0.8), the unification process succeeds leading to the partial answer "`({X1=john}, 0.8)`" and a new goal: `likes_teaching(john, math)`. Similarly, this goal weakly unifies with the fact "`likes_teaching(john, physics)`" leading to the empty clause (that is, a refutation). The final answer results from the composition of the partial substitutions and the minimum of the approximation degrees obtained in the previous steps: `({X=john}, 0.8)`.

The nondeterministic operational mechanism of the language also computes a second successfully derivation leading to the answer: `({X=mary}, 0.6)`. In this case, the clue is the existence of the entry "$\mathcal{R}(math, chemistry) = 0.6$" in the considered proximity relation.

On the other hand, Bousi∼Prolog implements a number of remarkable features, such as the inclusion of fuzzy sets in the core of the language Julián-Iranzo and Rubio-Manzano (2010) or the automatic support for generating some standard closures of a fuzzy relation Julián-Iranzo (2008). The last one is intensively used in our proposal of categorization through the internal operational mechanism of Bousi∼Prolog. Due to the importance of this last feature, ending this subsection, we light up its fundamentals and some implementation details.

Given a finite set A of cardinality n and assuming that we list the elements of A on an arbitrary sequence $\{a_1, a_2, \ldots, a_n\}$. Then a fuzzy binary relation $\mathcal{R}$ on A can be represented by a matrix $M = [m_{ij}]$ such that $m_{ij} = \mathcal{R}(a_i, a_j)$. Sometimes we say that m_{ij} is the *entry* $\langle i, j\rangle$ of M, which is called the *adjacency matrix* of $\mathcal{R}$. Note that, because we work with finite alphabets, fuzzy binary relations on a syntactic domain can be represented by adjacency matrices. In order to build the reflexive, symmetric and transitive closures of a fuzzy relation we proceed as follows:

Building the reflexive closure of $\mathcal{R}$: for each entry $\langle i, i\rangle$ in M do $m_{ii} := 1$;
Building the symmetric closure of $\mathcal{R}$: for each entry $\langle i, j\rangle$ in M, such that $m_{ij} \neq 0$, do $m_{ji} := m_{ij}$;
Building the transitive closure of R: for each column k and entry $\langle i, j\rangle$ in M do $m_{ij} :=$

[7] This is the default behavior. See later, at the end of this subsection, for more information.
[8] That is, a pair `(substitution, approximation_degree)`.

$m_{ij} \vee (m_{ik} \wedge m_{kj})$; where "$\vee$" and "$\wedge$" are, respectively, the maximum and the minimum operators;

Note that, for computing the transitive closure of a relation, we use a direct extension of the wellknown Warshall's algorithm (Warshall, 1962), where the classical *meet* and *joint* operators on the set $\{0, 1\}$ have been changed by the *maximum* and the *minimum* operators on the real interval $[0, 1]$ respectively. A fact that makes this Warshall-like's algorithm attractive is that it computes the transitive closure in only one pass over M (in the sense that each element is tested once), a fact that is not obvious (Warshall, 1962). Another interesting property is that it preserves the approximation degrees provided with the original relation $\mathcal{R}$[9].

Corresponding to any fuzzy binary relation R on A and its adjacency matrix representation M, there is a labeled *directed graph* (or *digraph*) G whose *nodes* (or *vertices*) are the members of the domain of R and whose labeled arcs are the triples $a_i \xrightarrow{\alpha_{ij}} a_j$ for which $R(a_i, a_j) = \alpha_{ij}$. Hence we can see these procedures as processes that complete the original relation with new (direct) labeled arcs. In the case of the transitive closure, these new (direct) labeled arcs are storing information on the existence of a path between two elements. Moreover, the path stored is the one with the minimum approximation degree, being a lower bound of the existing relationship of those connected elements.

At this point, it is important to underline that closure construction is done at compiling time, so it has not a harmful effect on the execution efficiency of a program. Quite the opposite, we think it contributes to its efficiency (e.g. avoiding the search of path connexions among elements in order to establish their closeness). Also, closure construction provides the programmer with great freedom to define the fuzzy binary relation he wants to work. Certainly, he can supply to the system a partial specification of the relation, given an initial subset of relation entries represented by proximity equations. Then, by default, the system automatically generates a reflexive, symmetric closure in order to build a proximity relation, completing the partially specified relation. On the oder hand, if the BPL directive "`:- transitivity(yes)`" is included in a BPL program, the transitive closure is also computed, leading to a similarity relation. Note however that, it is not easy (for the programmer itself) to define a similarity relation on a set of entries due to the transitivity constrains, which may contradict the initial approximation degrees. Therefore, this is a highly valuable feature also by this reason.

3 Text Classification Proposal

Bousi~Prolog allows us to implement a declarative approach to text categorisation using flexible matching and knowledge representation by means of an ontology of terms modelled by a set of proximity equations. The following sections show how proximity equations can be used as a fuzzy model for text categorisation where the knowledge base is selected from an ontology; that is, a structured collection of terms that formally defines the relations among them (Gruber, 1995). This is an useful application for the Semantic Web (Shadbolt et al, 2006), where people are exposed to great amounts of (textual) information.

[9] Whenever the elements of the initial matrix fulfill the so called "transitivity property" (Julián-Iranzo, 2008).

The objective of any process of classification of documents is to assign one or more predetermined categories to classify each one of the documents. In our approach, the availability of a set of labelled documents or a training process is not necessary; only background knowledge is used to classify the documents. The proposed method consists of the following steps or phases:

1. **Knowledge Base Building**: It is necessary to build the definition of the categories by using proximity relations extracted from thesauri and ontologies.
2. **Document Processing**: The input documents are processed using classical techniques of natural language processing like stop word removal and stemming.
3. **Flexible Search and Computing Occurrence Degrees**: Bousi∼Prolog is used to search into each document content, the terms close to a category in order to classify them , obtaining their degrees of occurrence. An *occurrence degree* of a term is an aggregation of the number of occurrences of the term (in a document) and the approximation degree, with regard to the analysed category.
4. **Computing Document Compatibility Degrees**: The compatibility degrees of the documents with regard to a category are computed using a certain compatibility measure. A *compatibility measure* is an operation which uses the occurrence degrees of the terms close to a category to calculate a document compatibility degree, that is, an index of how compatible the document is with regard to the analysed category.
5. **Classification Process**: Each document is classified as pertaining to the category or categories that reach a higher compatibility degree.

In order to describe the proposed classification method effectively and to detail the phases above enumerated, let us consider a running example that will bee l developed throughout this section. We are going to consider the problem of classifying a short text with regard to a set of categories, and to describe the results produced when they are processed by the proposed method.

Example 4 Consider a set with four categories `air`, `agriculture`, `water` and `transportati-on` jointly with the following document extracted from the English version of EnviWeb Portal[10] —and stored in a file named "`runningEX`"—.

```
Urban Rivals. That biocide pollution of agricultural Pesticides. Pesticides and Biocides can
Cause Serious  Harm to aquatic ecosystems. A study by Swiss researchers found that has
the levels of some Common Ppsticides and biocides entering wastewater and rivers ..
```

First, we want to link one or several categories with the document, since this is the essence of a classification process. On the other hand, note that the "Enviweb expert" classified the document inside the `water` categorie. Categories like `air` or `agriculture` are also possible but the expert didn't choose them. Therefore, we would like to identify what is the knowledge that the expert used to classify the document in these categories.

3.1 Knowledge Base Building

The first step in classifying a document, with regard to a set of categories without any prior training process, is to define each of these categories to use them as accurately as possible. The starting point of this definition is the concept related to the category name. Therefore, concepts like "water" or "air", that are examples of categories in the EnviWeb Portal, will be the source of the background knowledge.

[10] http://www.enviweb.cz

The definition of a concept is built from the set of concepts that are semantically close to it. These semantic relationships are extracted from some kind of controlled vocabulary or thesaurus which is relevant to a certain domain of knowledge, like economics. More precisely, we follow the techniques described at the end of Section 2.1 to construct a fuzzy relation.

In this context, it is important to choose the correct meaning of a word for the classification process. If the work domain is known, it is necessary to use a process of disambiguation in order to realise the definition of polysemous words. For example, in the economic and financial domain of the Reuters collection, the category "interest" is related to "loan" or "debt" but not to "curiosity".

For our running example (Example 4), the knowledge is extracted from a thesaurus. More precisely, the sources are Wordnet related terms and WordNetSimilarity. A fragment of the generated proximity relation is as follows:

$\mathcal{R}(air, wind) = 0.68$, $\mathcal{R}(agriculture, food) = 0.23$,
$\mathcal{R}(air, carbon) = 0.32$, $\mathcal{R}(agriculture, biocide) = 0.13$,
$\mathcal{R}(air, pollution) = 0.13$, $\mathcal{R}(transportation, car) = 0.$,
$\mathcal{R}(air, oxygene) = 0.68$, $\mathcal{R}(transportation, pollution) = 0.09$,
$\mathcal{R}(agriculture, pesticide) = 0.11$, $\mathcal{R}(transportation, vehicle) = 0.3$
$\mathcal{R}(agriculture, fertilizer) = 0.09$, $\mathcal{R}(transportation, ship) = 0.57$
$\mathcal{R}(water, river) = 0.40$, $\mathcal{R}(water, wastewater) = 0.35$,
$\mathcal{R}(water, aquatic) = 0.50$, $\mathcal{R}(water, ocean) = 0.35$.

Afterwards, these entries are represented as a set of proximity equations:

```
air~wind=0.68       water~wastewater=0.35       transportation~vehicle=0.3
air~carbon=0.32     agriculture~food=0.23       transportation~ship=0.57
air~pollution=0.13  agriculture~biocide=0.53    transportation~car=0.46
air~oxygene=0.68    agriculture~pesticide=0.11  transportation~pollution=0.09
water~river=0.45    agriculture~fertilizer=0.09
water~aquatic=0.5   water~ocean=0.35
```

Once the proximity equations are established, they are loaded into the Bousi∼Prolog system in order to serve as a knowledge base for the flexible search and classification process. We recall that, by default, Bousi∼Prolog compiles proximity equations into a proximity relation. That is, it automatically generates the reflexive and symmetric closures of the original (partial) relation. Additionally, if the transitivity flag is enabled (by means of the `transitivity/1` directive), the transitive closure is also generated, producing a similarity relation.

3.2 Document Processing

The first stage in processing the document is a linguistic pre-process that consists of removing stop words, performing a stemming process based on WordNet and grouping meaningful couples of words. For our running example, the text obtained after this process is the following:

```
urban rival biocide pollution agricultural pesticide pesticide biocide
cause serious harm aquatic ecosystem study swiss researcher found level
common pesticide biocides enter wastewater river
```

This pre-processed text acts as one of the inputs for the next step, mainly consisting of a flexible search of terms which are close to one of the considered categories.

At this point, it is important to note that the following phases of the classification process are managed by the Bousi∼Prolog system. An application program, named `inspect.bpl`, drives the rest of the process. The program `inspect.bpl` inspects a sequence of documents stored in a file whose internal structure is consistent with the SMART standard format (see Figure 2). It takes advantages from the remarkable features of Bousi∼Prolog language in order to search for flexible solutions. This program includes an ontology of terms, modeled by proximity equations, and a set of more than 49 predicates and 720 lines of code.

```
.I 1                        .
.W                          .
<Document 1>                .
.I 2                  .I N
.W                    .W
<Document 2>          <Document N>
```

Fig. 2 SMART document structure.

3.3 Flexible Search and Computing Occurrence Degrees

As was just noted, the essence of this phase is searching for the terms which are close to one of the considered categories and computing the occurrence degrees that will provide the necessary results for selecting the category or categories that must be assigned to a document.

The content of a file is read, word by word, by a predicate called `inspect/3` looking for those words that are close (according to the proximity equations) to a term, that is one of the pre-established categories that may be assigned to a document. As a result of the inspection, a record with statistical data is returned with the following structure:

```
[[texNumber(1)|L1], [texNumber(2)|L2], ...]
```

There is a sublist for each document `i` stored in the file `Filename`. Each sublist `Li` stores a sequence of triples `t(X, N, D)`, where `X` is a term close or similar to the term `Keyword`, with degree `D`, which occurs `N` times in the text `texNumber(i)`. In order to search for words close or similar to a given one, this predicate relies on the fuzzy unification mechanism implemented in the core of the Bousi∼Prolog language. More specifically, it uses a *weak unification operator*, also denoted by ∼, which is the fuzzy counterpart of the syntactic unification operator present in the standard Prolog language.

Coming back to our running example, after inspecting the text `runningEX` for the category "`water`", by using the predicate `inspect/3`, the system offers the following output:

```
BPL> inspect(runningEX, water, R).
Processing file, runningEX. This may take a while..
End of file reached.
  R = [[texNumber(1), t(aquatic,1,0.5), t(wastewater,1,0.35), t(river,1,0.45)]]
    With approximation degree: 1.0
```

The result shows the number of times that the words "`aquatic`", "`wastewater`" and "`river`" occur in the text (i.e. only once) and the degree of relation between these words and the category "water" (`0.5`, `0.35` and `0.45`, respectively)[11].

[11] Observe that, the hole predicate `inspect/3` is a crisp predicate (that is, it only returns "yes", with approximation degree 1.0, or "no") because the weak unification operator was

Table 1 Compatibility Measures

Operator	Description	Formula
`mAx`	Maximum	$CD_i = max\{D_{i_1}, \ldots, D_{i_n}\}$
`sum`	Sum of the occurrences degrees	$CD_i = \sum_{k=1}^{n} \left(N_{i_k} * D_{i_k}\right)$
`wa`	Weighted average	$CD_i = \frac{\sum_{k=1}^{n} \left(N_{i_k} * D_{i_k}\right)}{\sum_{k=1}^{n} \left(N_{i_k}\right)}$

3.4 Computing Document Compatibility Degrees

In order to estimate the degree of compatibility between a category and the document contents, it is necessary to execute the predicate `compDegree/4` which is a higher order predicate based on `inspect/3`. It takes a file, named `File`, a category, `Category`, a compatibility measure operation, named `Operator`, and returns a document compatibility degree account list, named `DCD_Account`.

```
compDegree(File, Category, Operator, DCD_Account):-
          inspect(File, Category, DataAccount),
          applyTo(Operator, [DataAccount, DCD_Account]).
```

The predicate `compDegree/4`, after calling the predicate `inspect/3`, compresses the `DataAccount` list into a document compatibility degree, using the operation `Operator`. In more detail, the predicate `applyTo/2` constructs the expression "`Operator(DataAccount, DCD_Account)`" and launches it as a goal. Then, for each sublist `[texNumber(i), t(Ti1,Ni1,Di1), ..., t(Tin,Nin,Din)]` of the `DataAccount` list, the former expression computes a new sublist `[texNumber(i), CDi]`, where `CDi` is the compatibility degree of the category `Category` for the document `i`. It is possible the use of several formulae to obtain these compatibility degrees. Table 1 summarises a set of sensible options.

For our running example, using the compatibility measure operator `sum` (*sum of the occurrence degrees*), defined in Table 1, we obtain a 1.3 compatibility degree of the category `water` for the considered text.

```
BPL> compDegree(runningEX, water, sum, R).
  R = [[texNumber(1), 1.3]]
    With approximation degree: 1.0
```

The last step in the computation of the compatibility degrees is driven by the predicate `seqInspect/4`. This predicate takes as input a file, `File`, a list of categories, `CategoryList`, to be inspected and a compatibility measure operator, `Operator`, returning a document compatibility list, named `CompList`. Roughly speaking, it consists of the sequential execution of `compDegree/4`, looking for words that are related to each one of the categories which exist in `CategoryList` and computing a document compatibility degree for these categories. Each category reaches a compatibility degree within each document in the text file. More precisely, the list `CompList`, returned by `seqInspect/4` with statistical data, has the following structure:

```
[r(Category1, [DCD_Account11,...,DCD_Account1n1)]),
 r(Category2, [DCD_Account21,...,DCD_Account2n2)]),
      . . .
 r(CategoryN, [DCD_AccountN1,...,DCD_AccountNnN)])]
```

designed as a crisp operator (a term is either close or similar to another one or it is not). Hence, the approximation degree for the hole goal is 1.0 in this example, since there are positive answers (three words close or similar to "water" were found in the file `runningEX`).

where every DCD_Account_ij is a document compatibility degree list computed by the predicate compDegree for a Category_i and a document j. For our running example, the execution of this predicate produces the following output:

```
BPL> seqInspect(runningEX, [air,agriculture,transportation, water], sum, R).
Processing file, runningEX. This may take a while..
End of file reached.
R = [r(air, [[texNumber(1), 0.13]]), r(agriculture, [[texNumber(1), 0.86]]),
     r(transportation, [[texNumber(1), 0.09]]),  r(water, [[texNumber(1), 1.3]])]
```

3.5 Classification Process

The procedure for the classification of documents is very simple: the categories with the higher compatibility degree are selected as the "winners". A predicate classify/2 gets the document compatibility list, CompList, obtained in the previous phase and produces a list with the following structure:

```
[[texNumber(1)|WinTopics_1]...[texNumber(N)|WinTopics_N]]
```

where WinTopics_k is the list of categories assigned to the document k in the text file. The list WinTopics_k may contain one or several categories or it may be empty. In the last case, the document can not be classified and we say that it is *unclassified* with regard to the list CategoryList of input categories.

For our running example, the category water is the winner with a compatibility degree of 1.3. It is clear that the category water should be selected as a winner because the semantic closeness relations maintained with water and the words aquatic, river and wasterwater. The word agriculture does not occur in the text but the semantic closeness between agriculture, biocide and pesticide provides a high compatibility degree between the text and this category. This reasoning scheme would be, more or less, the procedure that the expert could have followed to classify the document based on the ontological/semantic knowledge represented by the proximity equations.

4 Experiments

In this section, the performance of the proposed classification method is evaluated in terms of the classification accuracy.

4.1 Test Data Collections

The proposed classification approach has been tested on four distinct text-categorization tasks that we have taken from the World Wide Web. Table 2 shows the name of the data set, number of samples, total number of categories, and average length of the samples.

1. *News Snippets*: 1160 news has been extracted from the English Version of the Environmental Web-portal (EnviWeb[12]) (Hrebicek and Kubasek, 2004). The primary purpose of EnviWeb is to provide public environmental information and enable freedom access to environmental information. This complex environmental web

[12] http://www.enviweb.cz

Table 2 Details of the Data Sets

Data Set	Samples	Categories	Average Length (chars.)
News Snippets	1160	8	217
Web Snippets	115	10	187
Blog Posts	814	7	396
NewsWires	267	10	140

portal has grown up into one of the most visited portals in this branch. One of the most important parts of this portal is the archive of articles. The dataset has been divided into the following overlapped categories: air, agriculture, atmosphere, climate change, health, policies, transportation and water. Only the news title and the snippet has been considered as text source (around 200 characters).

2. *Web Snippets*: ODP-239 (C. and G., 2009) is a collection of web snippets. Each one of its elements includes the URL, the title, and the snippet of one web site extracted from the Open Directory Proyect[13]. This dataset is specially designed for evaluating subtopic information retrieval. The topic "Bussiness" and the subtopic "Energy" have been randomly selected. The obtained 115 documents are divided in ten categories: oil and gas, renewable, utilities, electricity, consulting, management, employment, fuel cells, associations, hydrogen.
3. *Blog Posts*: A collection of recent blog posts from several sources related to the topic of climate change[14]. The final collection is composed of 814 documents divided into the following categories: water, air, emissions, impacts, hazard, economy and technology. These documents were manually labeled with the best matching category.
4. *NewsWires*: Our experiment consists of classifying a set of short texts (news limited up to 160 characters long) selected from Reuters-21578 [15], the most widely used test collection for text categorization research. This experiment has been inspired by the available sources of newswires on the Web [16]. The test data set contains 267 pre-classified articles corresponding to ten overlapped categories (earn, acq, interest, money, grain, oilseed, rapeseed, copper, ship, wheat).

4.2 Perfomance Measures

As performance measures, we followed the standard definition of recall, precision, and F measure (the harmonic mean between precision and recall) (Van Rijsbergen, 1979). For the evaluation of performance average across categories, we used the micro-averaging method (Yang and Liu, 1999).

Given a certain known category C, possibly assigned to a document by an expert, and the classifier decided category ζ, the precision (P), recall (R) and their F measure (F) are calculated by the following formulas.

$$P(C,\zeta) = \frac{|C \bigcap \zeta|}{|C|} \qquad (2) \qquad\qquad R(C,\zeta) = \frac{|C \bigcap \zeta|}{|\zeta|} \qquad (3)$$

[13] http://www.dmoz.org/
[14] for example http://climateprogress.org/
[15] http://www.daviddlewis.com/resources/testcollections/reuters21578/
[16] http://www.euronews.net/newswires/

$$F(C,\zeta) = \frac{2 * P(C,\zeta) * R(C,\zeta)}{P(C,\zeta) + R(C,\zeta)} \qquad (4)$$

where $|\zeta|$ denotes the number of documents which are classified as corresponding to category ζ; analogously, $|C|$ denotes the number of documents whose assigned category is C and $|C \bigcap \zeta|$ is a rough notation for the number of documents which are classified into a correct category.

4.3 Proximity Relations

As previously explained, several proximity equations are needed in order to develop the classification process. These equations are extracted from the previously mentioned knowledge bases(see Section 2.1).

Different types of closeness relations have been used in order to develop the knowledge base required for the classification process. They can be grouped as:

- *Structural Analogy*: The structural analogy between two terms extracted from Concept Net 2.1 are used in this experiment.
- *Contextual Neighborhood*: The contextual neighbourhood extracted from Concept Net 2.1 is also used in this experiment.
- *WordNetSimilarity*:
  - WordNet: Using the definition of the each one of the categories in WordNet, a proximity degree has been estimated by means of the measures included in WordNetSimilarity. The best results were obtained by using Vector measure (Schütze, 1998).
  - *Wikipedia*: Like in the previous case, using the definition of the each one of the categories in Wikipedia, a proximity degree was estimated by means of the metrics included in WordNetSimilarity. The best results, once more again, were obtained by using Vector measure (Schütze, 1998).
- *Synonymy-based Similarity*. The degree of proximity between category names is calculated taking into account those terms included in common synsets and then applying the proximity measure defined in (Soto et al, 2008) (see Section 2.1).

The baseline is represented by the use of the syntatic equality (i.e., a category is represented only by its name).

4.4 Experiment Process

In order to classify the document collection using the conceptual proximity relations previously described, the following two sets are defined:

1. The set T contains the labels which represent each one of the categories. Each one of the labels is denoted by t_i.
2. The set D contains the documents to be classified. Each one of the documents will be identified as d_j, while t_{ij} denotes the theoretical label (maybe assigned by an expert) corresponding to document d_j and t'_{ij} denotes the label obtained by the flexible search procedure.

Once those sets are computed, the classification process is launched for each one of the proximity relations defined in the previous section. The steps of the process are described as follows:

1. The proximity degrees are calculated according to the selected relation for each one of the predefined categories and represented by means of proximity equations.
2. The obtained equations are loaded into the Bousi∼Prolog system and are used by the flexible search algorithm which allows the classification of the documents.
3. The compatibility degree between each one of the labels t_i and each document d_j is calculated. Those degrees are denoted by CD_{ij} and computed according to the predicate `compDegree` described in section 3.4 which allows to find t_i in d_j by means of a flexible search.
4. The chosen label t'_{ij} is the one with the compatibility degree ($t'_{ij} = arg(max(CD_{ij}))$). Then it is compared with the theoretical category (t_{ij}) assigned by an expert. If they match, the classification is considered as right. The percentage of right classifications defines the degree of fitness obtained by the ontology in the classification process.

All the experiments are carried out by using the predicate `experiment/3` that is defined in the following way:

```
experiment(FileName, CategoryList, Process):-
          seqInspect(FileName, CategoryList, Process, ResultList),
          zipResultList(ResultList, ZipList),   classify(ZipList, TexCatalog),
          concat_atom([FileName,'.',exp], FileName_exp), see(FileName_exp),
          write('Reading file, '), write(FileName_exp),
          write(' with an expert classification...'), nl,
          read(ExpTexCatalog), seen, compareW(TexCatalog, ExpTexCatalog).
```

`FileName` is the name of the file in which the documents to be classified are stored, `CategoryList` is list of topics, in this case the predefined ones were used (see Table 2). Finally, `Process` gives the aggregation function to compute the compatibility degree between a category and a concept. In our case, the sum of occurence degress (`sum`) is specified.

Essentially, as explained in Section 3, each category is sequentially searched by `seqInspect/4` in each one of the documents and the compatibility degree between each pair category-document is obtained. Then, using `classify/2`, the resulting categories are classified according to their compatibility degree and those ones with the higher degree are selected by obtaining a list of selected categories by document. Finally, using `compareW/2`, the selected categories are compared with the categories chosen by the expert in order to estimate the degree of fitness of the classification process.

4.5 Experiment Results

During the analysis of the experiment results, while comparing the percentage of correct classifications (C) (the document is classified correctly at least in one of this categories) with the "incorrect" ones, it is important to distinguish between those produced by wrong classifications (W) and by unclassified documents (U). The first case, wrong classifications, implies a contradiction with the knowledge used by the expert for classifying. In the second case, unclassified documents, means that one or more definitions are absent from the knowledge base, which should (or could) be completed in a subsequent phase.

The classification process was carried out with each one of the semantic relations previously defined. Classification results are shown in the following tables.

Table 3 News Snippets Experiment Results)

Proximity Relation	C	W	U	P	R	F
Structural Analogy	26%	33%	41%	33%	24%	28%
Contextual Neighborhood	61%	**18%**	21%	21%	59%	31%
Wikipedia/Vector	**66%**	21%	**14%**	**61%**	**71%**	**65%**
WordNet/Vector	51%	35%	14%	58%	46%	51%
WordNet/Synonymy	28%	6%	66%	53%	25%	34%
BaseLine	22%	6%	72%	78%	20%	32%

Table 4 Web Snippets Experiment Results

Proximity Relation	C	W	U	P	R	F
Structural Analogy	52%	36%	12%	33%	52%	40%
Contextual Neighborhood	54%	43%	3%	38%	54%	45%
Wikipedia/Vector	**78%**	**21%**	**1%**	**63%**	**78%**	**69%**
WordNet/Vector	51%	48%	1%	40%	51%	45%
WordNet/Synonymy	49%	48%	3%	18%	49%	26%
BaseLine	10%	17%	74%	37%	10%	16%

Table 5 Blog posts Experiment Results

Proximity Relation	C	W	U	P	R	F
Structural Analogy	36%	14%	50%	52%	29%	37%
Contextual Proximity	45%	14%	41%	66%	40%	50%
Wikipedia/Vector	**75%**	25%	**0%**	**77%**	**73%**	**75**
WordNet/Vector	72%	**20%**	8%	73%	72%	72%
WordNet/Synonymy	54%	25%	21%	74%	52%	61%
BaseLine	7%	26%	67%	42%	8%	13%

Table 6 NewsWires Experiment Results

Proximity Relation	C	W	U	P	R	F
Structural Analogy	17%	13%	69%	42%	16%	23%
Contextual Neighborhood	61%	11%	27%	64%	45%	53%
Wikipedia/Vector	79%	14%	7%	71%	63%	67%
WordNet/Vector	72%	10%	17%	69%	53%	60%
WordNet/Synonymy	41%	41%	18%	50%	37%	43%
BaseLine	10%	2%	88%	49%	5%	9%

The results obtained by using the proximity relations based on Concept Net are not acceptable, only the contextual neighboorhood achieve good results in certain classification tasks.

Comparing the proximity relations used, it is clear that the results is greatly improved when the definition of each category is complete enough. The best results were obtained by using the combination of Wikipedia and WordnetSimilarity, which brings a more complete concept definition of the categories in all the experiments. In many cases, the precision is poor when there are a high percentage of wrong calification. On the other hand, the recall is poor when there are many short texts not assigned to any category.

A good result obtained by the use of an specific ontology means that it has a definition of the concepts with a higher quality and completeness than the rest. However, the effectiveness of the method greatly depends on a good pairing of the problem with the background knowledge, and WordNet and Concept Net are not specific sources of domain knowledge. The selection of an appropriate specific ontology in a certain knowledge domain can provide better results. Another relevant result is the considerable improvement achieved on the effectiveness of the classification process by using semantic relations. The number of documents correctly classified is significantly better than the one obtained when the syntactic equality is used exclusively.

Although results according to F-Measure are not the best compared to other methods, those results could be considered acceptable specially taking into account that the best results were obtained using a transformed combination of knowledge bases (Barak et al, 2009). The approach proposed here is a classification method with limited complexity and a high dependency on the knowledge base used. Therefore, the obtained results promise great possibilities because they are better than those results obtained by (Barak et al, 2009) using WordNet and Wikipedia or those obtained by (Gliozzo et al, 2005) using context information.

5 Conclusions and Future Work

One key difficulty with current text classification learning algorithms is that they require a large, often prohibitive, number of labeled training examples to learn accurately. Labeling must often be done by a person, a tedious and time-consuming process. In this paper, a declarative text categorization approach, which does not employ a training process, has been presented. The method proposed is based on semantic relations (in particular, proximity relations) between concepts which describe each one of the categories.

One of the main strengths of this approach is the possibility of classifying documents without having some pre-classified training set of documents and even without a training process. In this way, starting from a list of category names, a classification mechanism could be set out without requiring additional treatments. Since text classification is a task based on the pre-defined categories, then categories for classifying documents should be known. This way there is no need of training the software over the document collection but to increase the knowledge about the tag collection, which is supposed to be known a priori. Also, it should be possible to apply the method with different document collections while keeping the same tag set without training the software once more. Within this approach, the category names are defined by means of an ontology of terms modeled by a set of proximity equations. The definition of a concept is built from the set of concepts that are semantically close to it. These semantic relationships are extracted from controlled vocabularies and thesauri which are relevant to a certain domain of knowledge.

Using these descriptions, the Bousi∼Prolog logic programming language allows to perform a flexible search of the concepts represented by those categories inside the documents. Once the proximity equations are established, they are loaded into the Bousi∼Prolog system as a knowledge base for the flexible search and classification process. By default, proximity equations are compiled into a proximity relation, generating the reflexive and symmetric closures of the original relation. Optionally, also it is possible to generate the transitive closure, leading to a similarity relation. This last ability has been used in this paper to model ontologies which are structural analogies.

The logic of the proposed classification mechanism is independent from the knowledge base used, providing a declarative approach to text classification where these main components are treated separately. The knowledge used for the classification process could be obtained from generic thesauri and expressed in a way which is understandable for any non-expert user. Thus the classification process is more comprehensible to the user than other approaches like Bayesian classifiers. Moreover, the knowledge to be used could be general or domain specific in order to classify the document according to certain pre-established categories.

The main problem of this method is that its performance depends on the quality of the category definitions (represented by proximity equations). If a category name is not well defined, the classification process performance achieved will be comparatively poor. There are several options for improving our approach in the near future. The first is to improve the knowledge bases used in the classification process and to incorporate other new ones as well. The second is the application of more complex aggregation formulae in order to determine the occurrence degree of a term in a document and/or the compatibility degree between a category and a document.

Acknowledgments

This research was partially supported by the Spanish Ministry of Science and Innovation (MEC) under TIN2007-67494 and TIN2010-20395 projects and by the Regional Government of Castilla-La Mancha under PEIC09-0196-3018, POII10-0133-3516 and PII1I09-0117-4481 projects.

References

Barak L, Dagan I, Shnarch E (2009) Text categorization from category name via lexical reference. In: Proceedings of the NAACL '09, Association for Computational Linguistics, Morristown, NJ, USA, pp 33–36

Boutari AM, Carpineto C, Nicolussi R (2010) Evaluating term concept association measures for short text expansion: two case studies of classification and clustering. In: Proceedings of the Seventh International Conference on Concept Lattices and their Applications (CLA 2010), pp 162–174

C C, G R (2009) Odp239 dataset. http://credo.fub.it/odp239/ . Last Visit March 2011

Faguo Z, Fan Z, Bingru Y, Xingang Y (2010) Research on short text classification algorithm based on statistics and rules. In: Proceedings of the 2010 Third International Symposium on Electronic Commerce and Security, IEEE Computer Society, Washington, DC, USA, ISECS '10, pp 3–7

Fellbaum C (1998) WordNet: An Electronic Lexical Database. MIT Press

Garcés P, Olivas J, Romero F (2006) Concept-matching IR systems versus word-matching information retrieval systems: Considering fuzzy interrelations for indexing web pages. J Am Soc Inf Sci Technol 57(4):564–576

Gliozzo A, Strapparava C, Dagan I (2005) Investigating unsupervised learning for text categorization bootstrapping. In: Proceedings of the Conference on HLT '05, Association for Computational Linguistics, Morristown, NJ, USA, pp 129–136, DOI http://dx.doi.org/10.3115/1220575.1220592

Gruber TR (1995) Toward principles for the design of ontologies used for knowledge sharing. Int J Hum-Comput Stud 43(5-6):907–928, DOI http://dx.doi.org/10.1006/ijhc.1995.1081

Hrebicek J, Kubasek M (2004) EnviWeb and Environmental Web Services: Case Study of an Environmental Web Portal, Springer, London, pp 21–24

Iranzo PJ, Rubio-Manzano C (2009) A Declarative Semantics for Bousi∼Prolog. In: Porto A, López-Fraguas FJ (eds) Proceedings of the 11th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming, September 7-9, 2009, Coimbra, Portugal (PPDP'2009), ACM, pp 149–160

Iranzo PJ, Rubio-Manzano C, Gallardo-Casero J (2009) Bousi∼Prolog: a Prolog Extension Language for Flexible Query Answering. Electronic Notes in Theoretical Computer Science 248:131–147

Julián-Iranzo P (2008) A procedure for the construction of a similarity relation. In: L Magdalena JV M Ojeda-Aciego (ed) In Proc. of the IPMU 2008, June 22-27, 2008, Torremolinos (Málaga), Spain, U. Málaga, pp 489–496

Julián-Iranzo P, Rubio-Manzano C (2009) A Similarity-Based WAM for Bousi∼Prolog. In: et al JC (ed) Bio-Inspired Systems: Computational and Ambient Intelligence, 10th International Work-Conference on Artificial Neural Networks, IWANN 2009, Salamanca, Spain, June 10-12, 2009, Proceedings, Part I, Springer, Lecture Notes in Computer Science, vol 5517, pp 245–252

Julián-Iranzo P, Rubio-Manzano C (2010) An efficient fuzzy unification method and its implementation into the bousi∼prolog system. In: 2010 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010)

Ko Y, Seo J (2004) Learning with unlabeled data for text categorization using bootstrapping and feature projection techniques. In: Proceedings ACL Workshop for Unsupervised Learning in Natural Language Processing, Association for Computational Linguistics

Ko Y, Seo J (2009) Text classification from unlabeled documents with bootstrapping and feature projection techniques. Inf Process Manage 45(1):70–83

Le DN, Goh AES (2007) Current Practices in Measuring Ontological Concept Similarity. DOI http://dx.doi.org/10.1109/SKG.2007.217

Liu H, Singh P (2004) Commonsense reasoning in and over natural language. Proc of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES-2004) pp 293–306

Liu J, Birnbaum L, Pardo B (2008) Categorizing blogger's interests based on short snippets of blog posts. In: Proceeding of the 17th ACM conference on Information and knowledge management, ACM, New York, NY, USA, CIKM '08, pp 1525–1526

Manning CD, Raghavan P, Schutze H (2008) An Introduction to Information Retrieval. Cambridge University Press, Cambridge, England

Meretakis D, Fragoudis D, Lu D, Likothanassis S (2000) Scalable association based text classiffication. In: Proc. 9th ACM Int. Conf. Information and Knowledge Management, Washington, USA, pp 5–11

Schütze H (1998) Automatic word sense discrimination. Comput Linguist 24(1):97–123

Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv 34(1):1–47, DOI http://doi.acm.org/10.1145/505282.505283

Shadbolt N, Berners-Lee T, Hall W (2006) The semantic web revisited. IEEE Intelligent Systems 21(3):96–101, DOI http://dx.doi.org/10.1109/MIS.2006.62

Soto A, Olivas JA, Prieto M (2008) Fuzzy approach of synonymy and polysemy for information retrieval. Studies in Fuzziness and Soft Computing 224:179–198

Sriram B, Fuhry D, Demir E, Ferhatosmanoglu H, Demirbas M (2010) Short text classification in twitter to improve information filtering. In: Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, SIGIR '10, pp 841–842

Van Rijsbergen C (1979) Information Retrieval. Butterworth, London, UK

Warshall S (1962) A Theorem on Boolean Matrices. Journal of ACM (9):11–12

Wu Z, Palmer M (1994) Verbs semantics and lexical selection. In: Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics, Morristown, NJ, USA, pp 133–138, DOI http://dx.doi.org/10.3115/981732.981751

Yang Y, Liu X (1999) A re-examination of text categorization methods. In: SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, pp 42–49, DOI http://doi.acm.org/10.1145/312624.312647

Zadeh L (1965) Fuzzy sets. Information and Control 8:338–353

Zelikovitz S, Hirsh H (2000) Improving short text classification using unlabeled background knowledge to assess document similarity. In: In Proceedings of the Seventeenth International Conference on Machine Learning, pp 1183–1190