Merging Structural and Taxonomic Similarity for Text Retrieval Using Relational Descriptions

Stefano Ferilli[1,3], Marenglen Biba[2], Nicola Di Mauro[1,3], Teresa M.A. Basile[1], and Floriana Esposito[1,3]

[1] Dipartimento di Informatica, Università di Bari
via E. Orabona, 4 - 70125 Bari, Italia
{ferilli,ndm,basile,esposito}@di.uniba.it
[2] Computer Science Department, University of New York, Tirana
Rr. "Komuna e Parisit", Tirana, Albania
marenglenbiba@unyt.edu.al
[3] Centro Interdipartimentale per la Logica e sue Applicazioni
Università di Bari
via E. Orabona, 4 - 70125 Bari, Italia

Abstract. Information retrieval effectiveness has become a crucial issue with the enormous growth of available digital documents and the spread of Digital Libraries. Search and retrieval are mostly carried out on the textual content of documents, and traditionally only at the lexical level. However, pure term-based queries are very limited because most of the information in natural language is carried by the syntactic and logic structure of sentences. To take into account such a structure, powerful relational languages, such as first-order logic, must be exploited. However, logic formulæ constituents are typically uninterpreted (they are considered as purely syntactic entities), whereas words in natural language express underlying concepts that involve several implicit relationships, as those expressed in a taxonomy. This problem can be tackled by providing the logic interpreter with suitable taxonomic knowledge.

This work proposes the exploitation of a similarity framework that includes both structural and taxonomic features to assess the similarity between First-Order Logic (Horn clause) descriptions of texts in natural language, in order to support more sophisticated information retrieval approaches than simple term-based queries. Evaluation on a sample case shows the viability of the solution, although further work is still needed to study the framework more deeply and to further refine it.

1 Introduction

The spread of digital technologies has caused a dramatic growth in the availability of documents in digital format, due to the easy creation and transmission thereof using networked computer systems. Hence, the birth of several repositories, aimed at storing and providing such documents to interested final users. The shortcoming of this scenario is in the problem of finding useful documents

M. Agosti, F. Esposito, and C. Thanos (Eds.): IRCDL 2010, CCIS 91, pp. 149–160, 2010.

that can satisfy an information need (the so-called *information overload* problem). Indeed, in legacy environments, few selected publications were available, and librarians could properly assess their content and consequently tag them for subsequent retrieval. Now, the amount of available documents is so huge that manual evaluation and tagging is infeasible. This represented a significant motivation for the invention of proper Information Retrieval techniques, that could rely on automatic techniques for document indexing and retrieval. More precisely, documents are almost always indexed based on their textual content, and are searched for by expressing textual queries. Due to the inborn complexity of Natural Language, information retrieval techniques have typically focussed their attention on the lexical level, that seemed a good tradeoff between computational requirements and outcome effectiveness. The text is seen as a sequence of un-related words (*bag-of-words*) and the query is expressed as a set of terms that are to be found in the documents. The weakness of such approaches, however, is that the syntactic and logical structure underlying the sentences is completely lost. Unfortunately, the real meaning of a sentence is mostly determined just by that level, and hence the quality of the term-based retrieval outcomes can be significantly affected by such a lack. Indeed, although much more computationally demanding than simple bag-of-word approaches traditionally exploited in the literature, techniques that take into account the syntactic structure of sentences are very important to fully capture the information they convey. Reporters know very well that, swapping the subject and the object in a sentence like "The dog bit the man", results in very different interest of the underlying news.

The landscape has slightly changed recently, due to the improved computational capabilities of current computer machines. Thus, considering the structural level of natural language sentences in text processing is no more technically infeasible, although still hard. However, handling the structural aspects in Natural Language Processing (*NLP* for short) cannot be reduced to just building syntactic parsers for the various languages. Sophisticated techniques for representing and handling this kind of information are needed, as well. First-Order Logic (or *FOL*) is a powerful representation language that allows to express relationships among objects, which is often an unnegligible requirement in real-world and complex domains. Logic Programming [12] is a computer programming framework based on a FOL sub-language, which allows to perform reasoning on knowledge expressed in the form of Horn clauses. Inductive Logic Programming (ILP) [14] aims at learning automatically logic programs from known examples of behaviour, and has proven to be a successful Machine Learning approach in domains where relations among objects must be expressed to fully capture the relevant information. One of the main reasons why FOL is a particularly complex framework compared to simple propositional or attribute-value ones relates to the problem of *indeterminacy*, meaning that different portions of one formula can be mapped in (often many) different ways onto portions of another.

An obstacle towards fruitful application of FOL to NLP is the fact that, in the traditional FOL approach, predicates that make up the description language are defined by the knowledge engineer that is in charge of setting up the reasoning

or learning problem, and are uninterpreted by the systems. Conversely, some kinds of information need to be interpreted in order to be fully exploited, which requires a proper background knowledge to be set up. For instance, numeric information must be exploited referring to mathematical concepts such as number ordering relationships and arithmetic operations. Analogously, descriptions of natural language sentences obviously include words of the vocabulary, that are the expression of underlying concepts among which many implicit relationships exist that can be captured by a taxonomy. Being able to properly consider and handle such an information is crucial for any successful application of pure FOL techniques to NLP [7].

An advantage of the logic framework is that a background knowledge can be defined and provided to help improving performance or effectiveness of the results. In the above case, a taxonomic background knowledge is needed. Unfortunately, unless the problem domain is very limited, natural language typically requires huge taxonomic information, and the problems of synonimy and polisemy introduce further complexity. In these cases, the use of existing state-of-the-art taxonomies can be a definite advantage. This work proposes the use of a framework for similarity assessment between FOL Horn clauses, enhanced to properly take into account also a taxonomic background knowledge, in order to find documents whose textual content is similar to a prototype sentence representing the query of the user. The basic similarity framework is borrowed from [6], while the taxonomic information is provided by WordNet.

The next section shows how natural language can be described in FOL, and how the introduction of taxonomic background knowledge can support the exploitation of implicit relationships between the concepts underlying the descriptions. Then, Section 3 describes the similarity formula and framework for structural and taxonomic similarity assessment. Section 4 shows experiments that suggest the effectiveness of the proposed approach. Lastly, Section 5 concludes the paper and outlines future work directions.

2 NLP = FOL + Taxonomies

The considerations reported in the previous section motivate the adoption of both FOL and taxonomic information for the description of natural language sentences. To give an idea, consider the following sentences:

1 - “The boy wants a small dog”
2 - “The girl desires a yellow canary”
3 - “The hammer hits a small nail”

They structurally exhibit the same grammatical pattern, thus no hint is available to assess which is more similar to which. Going more in depth, at the lexical level, the only common word (‘small’) appears in sentences 1 and 3, which would suggest they are closer to each other than to sentence 2. However, it becomes clear that the first two are conceptually the most similar to each other as long as one knows and considers that ‘boy’ and ‘girl’ are two young persons, ‘to want’ and

Table 1. First-order logic language for structural description of sentences

`subj(X,Y)`	Y is the subject of sentence X
`pred(X,Y)`	Y is the predicate of sentence X
`dir_obj(X,Y)`	Y is the direct object of sentence X
`ind_obj(X,Y)`	Y is the in direct object of sentence X
`noun(X,Y)`	Y is a noun appearing in component X of the sentence
`verb(X,Y)`	Y is a verb appearing in component X of the sentence
`adj(X,Y)`	Y is an adjective appearing in component X of the sentence
`adv(X,Y)`	Y is an adverb appearing in component X of the sentence
`prep(X,Y)`	Y is a preposition appearing in component X of the sentence
`sing(X)`	the number of noun X is singular
`pl(X)`	the number of noun X is plural
`past(X)`	the tense of verb X is past
`pres(X)`	the tense of verb X is present
`fut(X)`	the tense of verb X is future

'to desire' are synonyms and 'dog' and 'canary' are two pets. In an information retrieval perspective, if 1 represents the user query, and {2,3} are the documents in the repository, we would like 2 to be returned first, while pure syntactic or lexical techniques would return 3 as the best matching solution. Note that the interesting case is that of sentences that are very close or identical grammatically, but very different in meaning; indeed, for sentences that differ already at the grammatical level the basic structural similarity framework described in [6] would be enough for assessing their degree of distance.

There are several levels of the grammatical structure that can be exploited to describe natural language sentences at different levels of abstraction. Of course, the deeper the level, the more complex the description and the more computational demanding its processing. The best grain-size to be exploited depends on the particular situation, and should represent a suitable tradeoff between expressive power and complexity. For demonstration purposes, in the following let us consider the very simple sentence structural description language reported in Table 1. Additionally, each noun, verb, adjective or adverb is described by the corresponding concept (or word) in the sentence, that is to be interpreted according to the taxonomy. To specify which literals are to be interpreted, suppose that they are enclosed as arguments of a `tax/1` predicate. This yields, for the previous three sentences, the following descriptions:

```
s1 = sentence(s1) :- subj(s1,ss1), pred(s1,ps1), dir_obj(s1,ds1),
     noun(ss1,nss1), sing(nss1), tax(boy(nss1)),
     verb(ps1,vps1), pres(ps1), tax(want(vps1)),
     adj(ds1,ads1), tax(small(ads1)),
     noun(ds1,nds1), sing(nds1), tax(dog(nds1)).
s2 = sentence(s2) :- subj(s2,ss2), pred(s2,ps2), dir_obj(s2,ds2),
     noun(ss2,nss2), sing(nss2), tax(girl(nss2)),
     verb(ps2,vps2), pres(ps2), tax(desire(vps2)),
     adj(ds2,ads2), tax(yellow(ads2)),
     noun(ds2,nds2), sing(nds2), tax(canary(nds2)).
```

```
s3 = sentence(s3) :- subj(s3,ss3), pred(s3,ps3), dir_obj(s3,ds3),
     noun(ss3,nss3), sing(nss3), tax(hammer(nss3)),
     verb(ps3,vps3), pres(ps3), tax(hit(vps3)),
     adj(ds3,ads3), tax(small(ads3)),
     noun(ds3,nds3), sing(nds3), tax(nail(nds3)).
```

As already pointed out, setting up a general taxonomy is a hard work, for which reason the availability of an already existing resource can be a valuable help in carrying out the task. In this example we will refer to the most famous taxonomy available nowadays, WordNet (WN) [13], that provides both the conceptual and the lexical level. Note that, if the concepts are not explicitly referenced in the description, but common words in natural language are used instead, due to the problem of polysemy (a word may correspond to many concepts), their similarity must somehow combine the similarities between each pair of concepts underlying the words. Such a combination can consist, for instance, in the average or maximum similarity among such pairs, or more sensibly can exploit the domain of discourse. A distance between groups of words (if necessary) can be obtained by couplewise working on the closest (i.e., taxonomically most similar) words in each group.

3 Similarity Framework

Many AI tasks can take advantage from techniques for descriptions comparison: subsumption procedures (to converge more quickly), flexible matching, instance-based classification techniques or clustering, generalization procedures (to focus on the components that are more likely to correspond to each other). Here, we are interested in the assessment of similarity between two natural language texts described by both lexical/syntactic features and by taxonomic references.

Due to its complexity, few works exist on FOL descriptions comparison. In [6], a framework for computing the similarity between two Datalog Horn clauses has been provided, which is summarized in the following. Let us preliminary recall some basic notions involved in Logic Programming. The *arity* of a predicate is the number of arguments it takes. A *literal* is an n-ary predicate, applied to n terms, possibly negated. *Horn clauses* are logical formulæ usually represented in Prolog style as $l_0 \text{ :- } l_1, \ldots, l_n$ where the l_i's are *literals*. It corresponds to an implication $l_1 \wedge \cdots \wedge l_n \Rightarrow l_0$ to be interpreted as "l_0 (called *head* of the clause) is true, provided that l_1 and ... and l_n (called *body* of the clause) are all true". Datalog [3] is, at least syntactically, a restriction of Prolog in which, without loss of generality [16], only variables and constants (i.e., no functions) are allowed as terms. A set of literals is *linked* if and only if each literal in the set has at least one term in common with another literal in the set. We will deal with the case of linked Datalog clauses. In the following, we will call *compatible* two sets or sequences of literals that can be mapped onto each other without yielding inconsistent term associations (i.e., a term in one formula cannot correspond to different terms in the other formula).

Intuitively, the evaluation of similarity between two items i' and i'' might be based both on parameters expressing the amounts of common features, which should concur in a positive way to the similarity evaluation, and of the features of each item that are not owned by the other (defined as the *residual* of the former with respect to the latter), which should concur negatively to the whole similarity value assigned to them [11]:

n , the number of features owned by i' but not by i'' (*residual* of i' wrt i'');
l , the number of features owned both by i' and by i'';
m , the number of features owned by i'' but not by i' (*residual* of i'' wrt i').

A similarity function that expresses the degree of similarity between i' and i'' based on the above parameters, and that has a better behaviour than other formulæ in the literature in cases in which any of the parameters is 0, is [6]:

$$sf(i', i'') = \mathrm{sf}(n, l, m) = 0.5\frac{l+1}{l+n+2} + 0.5\frac{l+1}{l+m+2} \tag{1}$$

It takes values in $]0, 1[$, which resembles the theory of probability and hence can help human interpretation of the resulting value. When $n = m = 0$ it tends to the limit of 1 as long as the number of common features grows. The full-similarity value 1 is never reached, being reserved to two items that are exactly the same ($i' = i''$), which can be checked in advance. Consistently with the intuition that there is no limit to the number of different features owned by the two descriptions, which contribute to make them ever different, it is also always strictly greater than 0, and will tend to such a value as long as the number of non-shared features grows. Moreover, for $n = l = m = 0$ the function evaluates to 0.5, which can be considered intuitively correct for a case of maximum uncertainty. Note that each of the two terms refers specifically to one of the two items under comparison, and hence they could be weighted to reflect their importance.

In FOL representations, usually terms denote objects, unary predicates represent object properties and n-ary predicates express relationships between objects; hence, the overall similarity must consider and properly mix all such components. The similarity between two clauses C' and C'' is guided by the similarity between their structural parts, expressed by the n-ary literals in their bodies, and is a function of the number of common and different objects and relationships between them, as provided by their least general generalization $C = l_0 \text{ :- } l_1, \ldots, l_k$. Specifically, we refer to the θ_{OI} generalization model [5]. The resulting formula is the following:

$$\mathrm{fs}(C', C'') = \mathrm{sf}(k' - k, k, k'' - k) \cdot \mathrm{sf}(o' - o, o, o'' - o) + \mathrm{avg}(\{\mathrm{sf}_s(l'_i, l''_i)\}_{i=1,\ldots,k})$$

where k' is the number of literals and o' the number of terms in C', k'' is the number of literals and o'' the number of terms in C'', o is the number of terms in C and $l'_i \in C'$ and $l''_i \in C''$ are generalized by l_i for $i = 1, \ldots, k$. The similarity of the literals is smoothed by adding the overall similarity in the number of overlapping and different literals and terms.

The similarity between two compatible n-ary literals l' and l'', in turn, depends on the multisets of n-ary predicates corresponding to the literals directly linked to them (a predicate can appear in multiple instantiations among these literals), called *star*, and on the similarity of their arguments:

$$\mathrm{sf}_s(l', l'') = \mathrm{sf}(n_s, l_s, m_s) + \mathrm{avg}\{\mathrm{sf}_o(t', t'')\}_{t'/t'' \in \theta}$$

where θ is the set of term associations that map l' onto l'' and S' and S'' are the stars of l' and l'', respectively:

$$n_s = |S' \setminus S''| \qquad l_s = |S' \cap S''| \qquad m_s = |S'' \setminus S'|$$

Lastly, the similarity between two terms t' and t'' is computed as follows:

$$\mathrm{sf}_o(t', t'') = \mathrm{sf}(n_c, l_c, m_c) + \mathrm{sf}(n_r, l_r, m_r)$$

where the former component takes into account the sets of properties (unary predicates) P' and P'' referred to t' and t'', respectively:

$$n_c = |P' \setminus P''| \qquad l_c = |P' \cap P''| \qquad m_c = |P'' \setminus P'|$$

and the latter component takes into account how many times the two objects play the same or different roles in the n-ary predicates; in this case, since an object might play the same role in many instances of the same relation, the *multi*sets R' and R'' of roles played by t' and t'', respectively, are to be considered:

$$n_r = |R' \setminus R''| \qquad l_r = |R' \cap R''| \qquad m_r = |R'' \setminus R'|$$

Now, since the taxonomic predicates represent further information about the objects involved in a description, in addition to their properties and roles, term similarity is the component where the corresponding similarity can be introduced in the overall framework. Hence, the similarity between two terms becomes:

$$\mathrm{sf}_o(t', t'') = \mathrm{sf}(n_c, l_c, m_c) + \mathrm{sf}(n_r, l_r, m_r) + \mathrm{sf}(n_t, l_t, m_t)$$

where the additional component refers to the similarity between the taxonomic information associated to the two terms t' and t''. In particular, it suffices providing a way to assess the similarity between two concepts. Then, in case the taxonomic information is expressed in the form of words instead of concepts, either a Word Sense Disambiguation [8] technique is exploited to identify the single intended concept for polysemous words, or, according to the one-domain-per-discourse assumption, the similarity between two words can be referred to the closest pair of concepts associated to those words. In principle, in case of synonymy or polysemy, assuming consistency of domain among the words used in a same context [10], the similarity measure, by couplewise comparing all concepts underlying two words, can also suggest a ranking of which are the most probable senses for each, this way serving as a simple Word Sense Disambiguation procedure, or as a support to a more elaborate one.

To assess the similarity between concepts in a given taxonomy, and indirectly the similarity between the words that express those concepts, (1) can be applied directly on the taxonomic relations. The most important and significant relationship among concepts expressed in any taxonomy is the generalization/specialization one, relating concepts or classes to their super- and sub-concepts or classes, respectively. According to the definition in [2], this yields a similarity measure rather than

a full semantic relatedness measure, but we are currently working to extend it by taking into account relations other than hyponimy as well. Intuitively, the closer a common ancestor of two concepts c' and c'', the more they can be considered as similar to each other, and various distance measure proposed in the literature are based on the length of the paths that link the concepts to be compared to their closest common ancestor. In our case, (1) requires three parameters: one expressing the common information between the two objects to be compared, and the others expressing the information carried by each of the two but not by the other.

If the taxonomy is a hierarchy, and hence can be represented as a tree, this ensures that the path connecting each node (concept) to the root (the most general concept) is unique: let us call $< p'_1, \ldots, p'_{n'} >$ the path related to c', and $< p''_1, \ldots, p''_{n''} >$ the path related to c''. Thus, given any two concepts, their closest common ancestor is uniquely identified, as the last element in common in the two paths: suppose this is the k-th element (i.e., $\forall i = 1, \ldots, k : p'_i = p''_i = p_i$). Consequently, three sub-paths are induced: the sub-path in common, going from the root to such a common ancestor ($< p_1, \ldots, p_k >$), and the two trailing sub-paths ($< p'_{k+1}, \ldots, p'_{n'} >$ and $< p''_{k+1}, \ldots, p''_{n''} >$). Now, the former can be interpreted as the common information, and the latter as the residuals, and hence their lengths $(n'-k, k, n''-k)$ can serve as arguments (n, l, m) to apply the similarity formula. This represents a novelty with respect to other approaches in the literature, where only one or both of the trailing parts are typically exploited, and is also very intuitive, since the longest the path from the top concept to the common ancestor, the more they have in common, and the higher the returned similarity value.

Actually, in real-world domains the taxonomy is not just a hierarchy, but rather it is a heterarchy, meaning that multiple inheritance must be taken into account and hence a concept can specialize many other concepts. This is very relevant as regards the similarity criterion above stated, since in a heterarchy the closest common ancestor and the paths linking two nodes are not unique, hence many incomparable common ancestors and paths between concepts can be found, and going to the single common one would very often result in overgeneralization. Our solution to this problem is computing the whole set of ancestors of either concept, and then considering as common information the intersection of such sets, and as residuals the two symmetric differences. Again this is fairly intuitive, since the number of common ancestors can be considered a good indicator of the common information and features between the two concepts, just as the number of different ancestors can provide a reasonable estimation of the different information and features they own.

4 Evaluation

To assess the effectiveness of the proposed technique, two separate evaluations must be carried out. First of all, the taxonomic similarity measure alone must be proved effective in returning sensible similarity values for couples of concepts. Then, the overall similarity framework integrating both structural and taxonomic similarity assessment must be proved effective in evaluating the similarity

Table 2. Sample similarity values between WordNet words/concepts

Concept	Concept	Similarity
cat (wild) [102127808]	tiger (animal) [102129604]	0.910
cat (pet) [102121620]	tiger (animal) [102129604]	0.849
mouse (animal) [102330245]	cat (pet) [102121620]	0.775
mouse (device) [103793489]	computer (device) [103082979]	0.727
cat (pet) [102121620]	dog (pet) [102084071]	0.627
cat (wild) [102127808]	dog (pet) [102084071]	0.627
dog (pet) [102084071]	horse (domestic) [102374451]	0.542
mouse (animal) [102330245]	computer (device) [103082979]	0.394
mouse (animal) [102330245]	mouse (device) [103793489]	0.394
mouse (device) [103793489]	cat (pet) [102121620]	0.384
cat (domestic) [102121620]	computer (device) [103082979]	0.384
horse (domestic) [102374451]	horse (chess) [103624767]	0.339

between complex description. In the case of NLP, the latter must additionally show its ability in overcoming problems of interpretation due to the presence of polysemous words. We will show with some examples that both these requirements can be satisfied by the proposed approach.

As to the taxonomic similarity assessment alone, consider the following words and concepts, and some of the corresponding similarity values reported in Table 2:

102330245 *mouse* (animal) : 'any of numerous small rodents typically resembling diminutive rats having pointed snouts and small ears on elongated bodies with slender usually hairless tails'
103793489 *mouse* (device) : 'a hand-operated electronic device that controls the coordinates of a cursor on your computer screen as you move it around on a pad; on the bottom of the device is a ball that rolls on the surface of the pad'
103082979 *computer* (device) : 'a machine for performing calculations automatically' calculating machines)'
102121620 *cat* (pet) : 'feline mammal usually having thick soft fur and no ability to roar: domestic cats; wildcats'
102127808 *cat* (wild) : 'any of several large cats typically able to roar and living in the wild'
102129604 *tiger* (animal) : 'large feline of forests in most of Asia having a tawny coat with black stripes; endangered'
102084071 *dog* (pet) : 'a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds'
102374451 *horse* (animal) : 'solid-hoofed herbivorous quadruped domesticated since prehistoric times'
103624767 *horse* (chess) : 'a chessman shaped to resemble the head of a horse; can move two squares horizontally and one vertically (or vice versa)'

At the level of concepts, it is possible to note that the similarity ranking is quite intuitive, in that less related concepts receive a lower value. The closest pairs are

'wild cat'-'tiger' and 'pet cat'-'tiger', followed by 'mouse (animal)'-'cat (pet)', then by 'mouse (device)'-'computer (device)', by 'cat (pet)'-'dog (pet)' and by 'dog (pet)'-'horse (animal)', all with similarity values above 0.5. Conversely, all odd pairs, mixing animals and devices or objects (including polysemic words), get very low values, below 0.4.

Then, for checking the overall structural and taxonomic similarity assessment capability, let us go back to the sample sentences in the Introduction for an application of the proposed taxonomically-enhanced similarity framework to descriptions of sentences written in natural language.

As a first step, the similarity between single words must be assessed. Applying the proposed procedure, the similarity values are as follows:

boy-girl = 0.75	boy-hammer = 0.435
girl-hammer = 0.435	want-desire = 0.826
want-hit = 0.361	desire-hit = 0.375
yellow-small = 0.562	small-small = 1
dog-canary = 0.667	dog-nail = 0.75
canary-nail = 0.386	

It is possible to note that all similarities agree with the intuition, except the pair dog-nail that gets a higher similarity value than dog-canary, due to the interpretations of 'dog' as 'a hinged catch that fits into a notch of a ratchet to move a wheel forward or prevent it from moving backward' and 'nail' as 'a thin pointed piece of metal that is hammered into materials as a fastener'.

Without considering taxonomic information, the generalization between s1 and s2 and between s2 and s3 is:

```
sentence(X) :- subj(X,Y), pred(X,W), dir_obj(X,Z),
           noun(Y,Y1), sing(Y1), verb(W,W1), prest(W1),
           adj(Z,Z1), noun(Z,Z2), sing(Z2).
```

while the generalization between s1 and s3 is:

```
sentence(X) :- subj(X,Y), pred(X,W), dir_obj(X,Z),
           noun(Y,Y1), sing(Y1), verb(W,W1), pres(W1),
           adj(Z,Z1), tax(small(Z1)), noun(Z,Z2), sing(Z2).
```

so that the latter, having an additional literal with respect to the former, will take a greater similarity value due to just the structural similarity of the two sentences, in spite of the very different content. Conversely, by considering the taxonomic similarity among words, the comparisons become:

$$\text{fs(s1,s2)} = 2.444 \qquad \text{fs(s1,s3)} = 2.420 \qquad \text{fs(s2,s3)} = 2.318$$

where, indeed, the first two sentences neatly get the largest similarity value with respect to the other combinations. Notwithstanding the 'dog-nail' ambiguity, the overall correct similarity ranking between sentences is correct.

In order to better understand the effect of the approach on the similarity values, other tests were performed on the following sample sentences:

1- “the boy buys a jewel for a girl”
2- “the girl receives a jewel from a boy”
3- “a young man purchases a gem for a woman”
4- “a young man purchases a precious stone for a woman”

having a different structure (e.g., transitive vs intransitive in 1-2), containing synonyms that could belong to different synsets (boy vs ’young man’ in 1-3), or made up of multiple words instead of a single one (e.g., gem vs precious stone in 1-4). For these sentences the similarity values obtained were:

$$\text{fs}(1,2) = 2.383 \qquad \text{fs}(1,3) = 2.484 \qquad \text{fs}(1,4) = 2.511$$

5 Conclusions

Information retrieval effectiveness has become a crucial issue with the enormous growth of available digital documents and the spread of Digital Libraries. Search and retrieval are mostly carried out on the textual content of documents, and traditionally only at the lexical level. However, pure term-based queries are very limited because most of the information in natural language is carried by the syntactic and logic structure of sentences. To take into account such a structure, powerful relational languages, such as first-order logic, must be exploited. However, logic formulæ constituents are typically uninterpreted (they are considered as purely syntactic entities), whereas words in natural language express underlying concepts that involve several implicit relationships, as those expressed in a taxonomy. This problem can be tackled by providing the logic interpreter with suitable taxonomic background knowledge.

This work proposed the exploitation of a similarity framework that includes both structural and taxonomic features to assess the similarity between First-Order Logic (Horn clause) descriptions of texts in natural language, in order to support more sophisticated information retrieval approaches than simple term-based queries. Although the proposed framework applies to any kind of structural description and taxonomy, being able to reuse an already existing taxonomy would be of great help. For this reason, the examples reported in this paper exploited the WordNet (WN) database, that can be naturally embedded in the proposed framework. Other works exist in the literature that combine in various shapes and for different purposes structural (and possibly logical) descriptions of sentences, some kind of similarity and WN. Some concern Question Answering [17], others Textual Entailment [9, 4, 1, 15]. However, the taxonomy relationships exploited in these works, or the way in which the structure of sentences is handled, makes them useless to our purpose.

Evaluation on a sample case shows the viability of the solution, and its robustness with respect to problems due to lexical ambiguity and polysemy. Several non-organized small experiments on tens of sentences having different length have been carried out so far, confirming the sample results, but revealing large computational times (1-2 min for long sentences). Thus, a first direction for future work will concern efficiency improvement in order to make it scalable. After that, more

thorough experimentation and fine-tuning of the taxonomic similarity computation methodology by exploiting other relationships represented in WordNet will make sense. Also application of the proposed similarity framework to other problems, such as Word Sense Disambiguation in phrase structure analysis would be interesting directions deserving further investigation.

References

[1] Agichtein, E., Askew, W., Liu, Y.: Combining lexical, syntactic, and semantic evidence for textual entailment classification. In: Proc. 1st Text Analysis Conference, TAC (2008)
[2] Budanitsky, A., Hirst, G.: Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In: Proc. Workshop on WordNet and Other Lexical Resources, 2nd meeting of the North American Chapter of the Association for Computational Linguistics, Pittsburgh (2001)
[3] Ceri, S., Gottlöb, G., Tanca, L.: Logic Programming and Databases. Springer, Heidelberg (1990)
[4] Clark, P., Harrison, P.: Recognizing textual entailment with logical inference. In: Proc. 1st Text Analysis Conference, TAC (2008)
[5] Esposito, F., Fanizzi, N., Ferilli, S., Semeraro, G.: A generalization model based on oi-implication for ideal theory refinement. Fundamenta Informaticæ 47(1-2), 15–33 (2001)
[6] Ferilli, S., Basile, T.M.A., Biba, M., Di Mauro, N., Esposito, F.: A general similarity framework for horn clause logic. Fundamenta Informaticæ 90(1-2), 43–46 (2009)
[7] Ferilli, S., Fanizzi, N., Semeraro, G.: Learning logic models for automated text categorization. In: AI*IA 2001: Advances in Artificial Intelligence. Springer, Heidelberg (2001)
[8] Ide, N., Véronis, J.: Word sense disambiguation: The state of the art. Computational Linguistics 24, 1–40 (1998)
[9] Inkpen, D., Kipp, D., Nastase, V.: Machine learning experiments for textual entailment. In: Proc. 2nd PASCAL Recognising Textual Entailment Challenge, RTE-2 (2006)
[10] Krovetz, R.: More than one sense per discourse. In: NEC Princeton NJ Labs., Research Memorandum (1998)
[11] Lin, D.: An information-theoretic definition of similarity. In: Proc. 15th International Conf. on Machine Learning, pp. 296–304. Morgan Kaufmann, San Francisco (1998)
[12] Lloyd, J.W.: Foundations of Logic Programming, 2nd edn. Springer, Berlin (1987)
[13] Miller, G.A.: Wordnet: A lexical database for English. Communications of the ACM 38(11), 39–41 (1995)
[14] Muggleton, S.: Inductive logic programming. New Generation Computing 8(4), 295–318 (1991)
[15] Pennacchiotti, M., Zanzotto, F.M.: Learning shallow semantic rules for textual entailment. In: Proc. International Conference on Recent Advances in Natural Language Processing, RANLP 2007 (2007)
[16] Rouveirol, C.: Extensions of inversion of resolution applied to theory completion. In: Inductive Logic Programming, pp. 64–90. Academic Press, London (1992)
[17] Vargas-Vera, M., Motta, E.: An ontology-driven similarity algorithm. Tech. Report kmi-04-16. Knowledge Media Institute (KMi), The Open University, UK (July 2004)