The Semantic Wildcard

Rada F. MIHALCEA

University of Texas at Dallas
Richardson, Texas, 75083-0688
rada@utdallas.edu

Abstract

The IRSLO (Information Retrieval using Semantic and Lexical Operators) project aims at integrating semantic and lexical information into the retrieval process, in order to overcome some of the impediments currently encountered with today's information retrieval systems. This paper introduces the semantic wildcard, one of the most powerful operators implemented in IRSLO, which allows for searches along general-specific lines. The semantic wildcard, denoted with #, acts in a manner similar with the lexical wildcard, but at semantic levels, enabling the retrieval of subsumed concepts. For instance, a search for *animal#* will match any concept that is of type *animal*, including *dog*, *goat* and so forth, thereby going beyond the explicit knowledge stated in texts. This operator, together with a lexical locality operator that enables the retrieval of paragraphs rather than entire documents, have been both implemented in the IRSLO system and tested on requests of information run against an index of 130,000 documents. Significant improvement was observed over classic keyword-based retrieval systems in terms of precision, recall and success rate.

1. Introduction

As the amount of information continues to increase, there must be new ways to retrieve and deliver information. Information is of no use if it cannot be located and the key to information location is a retrieval system. Traditionally, information retrieval systems use keywords for indexing and retrieving documents. These systems end up retrieving a lot of irrelevant information along with some useful information that the query/question was intended to elicit. Moreover, implicit knowledge makes often the bridge between a question and a document, and classic retrieval systems do not have the capability of going beyond explicit knowledge embedded in texts, thereby missing the answers to such queries.

To overcome some of the impediments currently encountered with today's information retrieval systems, we have started the IRSLO (Information Retrieval using Semantic and Lexical Operators) project that aims at integrating semantic and lexical information into the retrieval process, to the end of obtaining improved precision and recall. This paper introduces the *semantic wildcard*, one of the most powerful operators implemented in IRSLO.

Users' information needs are most of the times expressed along general-specific lines, and this paper provides analytical support towards this fact. *What sport*, *What animal*, *What body part*, are all examples of question types that require implicit knowledge about what constitutes a *sport*, *animal*, or *body-part*. The *semantic wildcard*, denoted with #, is designed to retrieve subsumed concepts. For instance, a search for *animal#* will match any concept that is of type *animal*, thereby going beyond the explicit knowledge stated in texts.

The *semantic wildcard*, together with a lexical locality operator previously introduced that enables the retrieval of paragraphs rather than entire documents (Mihalcea, 1999), were implemented in the IRSLO system and tested on requests of information run against an index of 130,000 documents. Significant improvement was observed over classic retrieval systems, in terms of precision, recall and success rate.

The paper is organized as follows. First, we present an analysis of questions asked by real time users, bringing evidence towards the fact that information need is most of the times expressed along general-specific lines. Next, we show how a novel encoding scheme - referred to as *DD-encoding* - can be applied to WordNet, in order to exploit the general-specific relations encoded in this semantic net. We then present the architecture of IRSLO, with emphasis on the *semantic wildcard* operator and the *paragraph operator*, together with experiments, results and walk through examples.

2. Defining Information Need

In order to define users' information need and assess the role that may be played by semantics in an information retrieval environment, we have performed a qualitative and quantitative analysis of information requests expressed by users in the form of natural language questions. Two sets of data are used during the experiments: (1) the Excite question log, for a total of 68,631 questions asked by the users of a search engine and (2) the TREC-8, TREC-9 and TREC-10 questions, for a total of 1,393 questions.

The noisy Excite log was cleaned up with two filters. First, we extracted only those lines containing one of the keywords *Where*, *When*, *What*, *Which*, *Why*, *Who*, *How*, *Why* or *Name*. Next, we eliminated the lines containing the phrase *"find information"* to avoid the bias towards Web searching questions. [1]

From the total of 25,272 Excite *What* questions[2] we have randomly selected a subset of 5,000 questions that were manually analyzed and classified. The decision of what question type to assign to a particular question was

[1] To our knowledge, only one other large scale question analysis is mentioned in the literature (Hovy et al., 2001).

[2] We emphasize the experiments involving *What* questions, since they provide the largest coverage and are considered to be the most ambiguous types of questions. Similar analyses were performed for the other types of questions, but are not reported here due to lack of space.

merely based on the possibility of implementing a procedure that would make use of this question type in the process of finding relevant information. For instance, a question like *What does Acupril treat?* expects a DISEASE as answer, which is doable in the sense that an ontology like WordNet does have a disease node with pointers to a large number of disease names. On the other hand, *What about this Synthyroid class action?* does not require a specific answer, but rather information related to a topic, and therefore no question type is assigned to this question (the type NONE is used instead). For the entire set of 5,000 questions, 361 categories are extracted.

2.1. Quantitative Analysis

To the end of observing the behavior and learning rate associated with question types, subsets of different sizes were created and the number of question types was determined for each subset. The measurements were performed using a 10-fold cross validation scheme on randomly selected samples of data.

Figure 1 plots the distribution of question types with respect to the subset size. It turns out that the number of question types grows sublinearly with the number of questions. Moreover, we noticed a behavior of the curve similar with *Heaps' Law* (Heaps, 1978), which relates the number of words in a text with the text size. *Heaps' Law* states that the size of the vocabulary for a text of size n is $V = Kn^\beta = O(n^\beta)$.

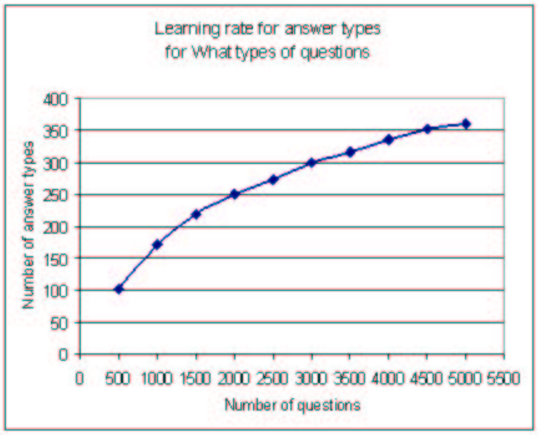

Figure 1: Number of question types vs. number of questions for *What* questions in the Excite log.

Denoting the number of question types with T_q and the number of questions with N_q, it follows:

$$\mathrm{T}_q = KN_q{}^\beta \qquad (1)$$

The equation is solved by taking the log in both sides. For the Excite *What* set, it results a value of $K = 5.18$, respectively $\beta = 0.50$. The values of the two parameters are changed in the TREC *What* set: $K = 3.89$ and $\beta = 0.54$, which illustrates the difference in question types distribution for the uniform TREC set versus the noisy Excite set.

This is an interesting result, as it defines the behavior of question types with respect to the number of questions. Moreover, it gives us the capability of making estimates on what is the expected number of question types for N_q given questions. For instance, 10,000 questions will result in about 518 question types, 100,000 in about 1,638 question types, and so forth.

2.2. Qualitative Analysis

The qualitative analysis brings evidence for the organization of question types in semantic hierarchies, and supports the idea of incorporating semantics into information retrieval.

An analysis of the questions benchmarks suggested that the majority of question types are found in a general-specific (ISA) relation. This hypothesis is sustained by empirical evidence. We classified the questions into four categories as listed in Table 1[3]. It turns out that on average about 60% of the questions are clear general-specific questions. It is debatable whether or not the DEFINITION types of questions can be classified as general-specific questions or not. It is often the case that a definition requires a more general concept to explain an unknown entity (Prager et al., 2001), and therefore it could be considered as a general-specific information request. Under this hypothesis, it results an average of 80% of information requests being expressed along general-specific lines.

Information type	Frequency
Excite questions	
GENERAL-SPECIFIC	54.6%
DEFINITION	19.6%
NONE	14.8%
OTHER	10.8%
TREC questions	
GENERAL-SPECIFIC	65.0%
DEFINITION	20.9%
NONE	6.6%
OTHER	7.4%

Table 1: Information requests along general-specific lines

Figure 2 shows examples of annotated questions extracted from the Excite log, mapped on an *animal* hierarchy of question types.

The conclusion of these experiments is that the majority of information requests are expressed along general-specific lines, and therefore a semantic based retrieval system that exploits these relations would possibly increase the quality of the information retrieved. This idea was also expressed by (Berners-Lee et al., 2001) in the context of Semantic Web.

3. Conversion of WordNet to DD-encoding

On the one side, we have the users' information need expressed most of the times as a general-specific request.

[3] The OTHER category includes questions that require an answer that cannot be obtained by following a general-specific line. Examples of such question types are CAUSE, EFFECT, QUOTE|ALBUM, QUOTE|MOVIE, WORD-TRANSLATION, etc.

- What is the largest DINOSAUR of all times?
- What is Connecticut state FISH?
- What SHARK lives off th coast of Georgia?
- What is a good family DOG?
- What are some INSECTS in South Carolina?
- What is the world largest LIZARD?
- What is the largest MAMMAL that is currently living?
- What is an endangered REPTILE?
- What is the state BIRD of Colorado?

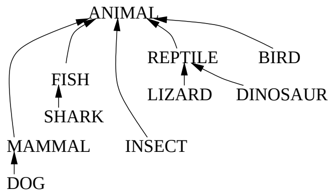

Figure 2: Question types mapped onto the *animal* hierarchy.

On the other side, we have WordNet (Miller, 1995) as the largest general purpose semantic network available today, which encodes about 86,605 general-specific (ISA) relations. We want to exploit as much as possible the semantic network structure of WordNet. To this end, we propose in this section a new encoding to be used for WordNet entries that would enable more efficient semantic searches. The so called *DD-encoding* was inspired by the Dewey Decimal code scheme used by librarians.

There are many times when keywords in a query are used with "generic" meanings and they are intended as representatives for entire categories of objects. *Foxes eat hens* is a statement that can be evaluated as a good match for *Animals eat meat*. Unfortunately, with current indexing and retrieval techniques this is not possible, unless both *animal* and *meat* are expanded with their subsumed concepts, which may sometimes become a tedious process. For this particular example, WordNet defines 7,980 concepts underneath *animal*, and there are 199 entries that inherit from *meat*, and therefore we end up with more than 1,500,000 (7,980 x 199) queries to cover the entire range of possibilities. Alternatively, if boolean queries are allowed and the OR operator is available, a query with 8,179 (7,980 + 199) terms can be used. None of these solutions seems acceptable and this is why none of them have been used so far.

We would like to find a way such that *fox* matches *animal* and we propose the employment of matching codes as an elegant solution to accomplish this task.

Finding the means that would allow for this type of matches is a problem of central interest for retrieval applications, as most information requests are expressed along general-specific lines. We want to retrieve documents containing *cat* in return to a search for *animal*, and retrieve *dachshund* and do not retrieve *cat* as the result of a search for *dog*.

To enable this type of general-specific searches and at the same time take advantage of the semantic structure already encoded in WordNet, we propose the employment of a codification scheme similar with the one used in librarian systems, and associate a code to each entry in WordNet.

The role of this code is to make evident to an external tool, such as an indexing or retrieval process, the relation that exists between inter-connected concepts. No information can be drawn from the simple reading of the *animal* and *dog* strings. Things are completely different when we look at *13.1* and *13.1.7*: the *implicit* relation between the two tokens has now been turned into an *explicit* one.

A code is assigned to each WordNet entry such that it replicates its parent code, and adds a unique identifier. For instance, if *animal* has code *13.1*, then *chordate*, which is a directly subsumed concept, has code *13.1.29*, *vertebrate* has code *13.1.29.3*, and so forth. Figure 3 illustrates a snapshot from the noun WordNet hierarchy and shows the *DD-codes* attached to each node. This encoding creates the grounds for matching at semantic levels in a manner similar with the lexical matches already employed by several information retrieval systems.

To our knowledge, this is a completely new approach taken towards the goal of making possible searches at semantic levels. The idea underneath this encoding is very simple but it allows for a powerful operator: the *semantic wildcard*.

3.1. Technical Issues

There are several implementation issues encountered during WordNet transformation, and we shall address them in this section.

Specifically, the new encoding is created using the following algorithm:

```
1. Start with the top of WordNet hierarchies. For each top, load its hyponyms, and for each hyponym go to step 2.
2. Execute the following steps:
    2.1. Assign to the current synset the DD-code of its parent plus an unique identifier that is generated as a number in a successive series.
    2.2. If the current synset has been already assigned a DD-code, then generate a special link between its parent and the current synset itself.
    2.3. Load all hyponyms of current synset and go to step 2.
```

The algorithm performs a recursive traversal of the entire WordNet hierarchy and generates codes. A code is associated with a synset, and we created a list of pairs containing a synset offset (the current WordNet encoding) and a *DD-code*.

It is worth mentioning the case of multiple inheritance, handled by the Dewey classification system as an addition made for a particular category. For instance, 675+678 means *leather and rubber*. This solution is not satisfactory for our purpose, since it may result in very long codes. Instead, a list of *special links* (generated in step 2b) is created, containing all the links between a *second parent* and a child. For example, if *house* inherits from both *domicile* and *building*, we have the code 1.2.1.32.12.23 for *house*, 1.2.1.32.28.6 for *domicile* and 1.2.1.32.12 for *building*, and in addition a special link is generated to indicate that *domicile* is the parent of *house* even if no direct matching can be performed.

For the entire noun hierarchy in WordNet, 74,488 *DD-codes* were generated. In addition, 4,280 multiple inheritance links were created. The average length of a code is 16 characters. Given the fact that disk space is a cheap re-

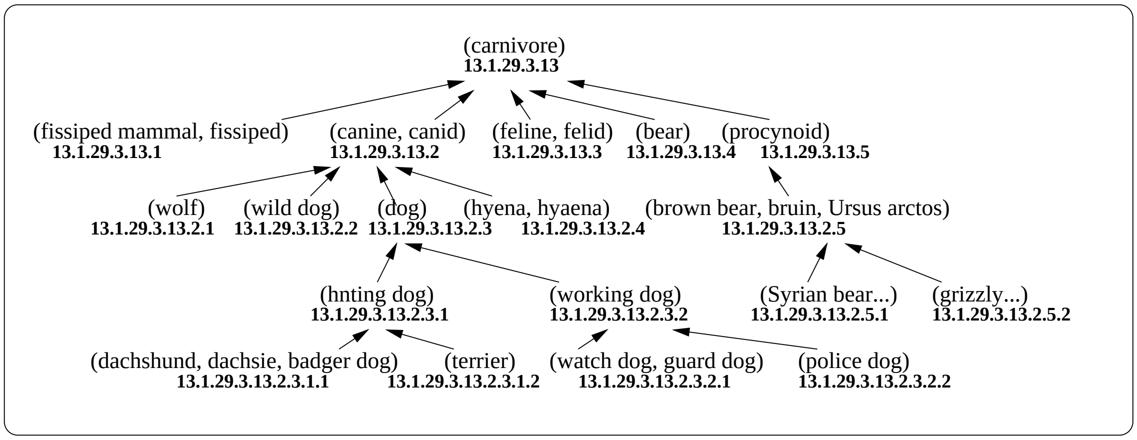

Figure 3: *DD-codes* assigned to a sample of the WordNet hierarchy

source, the length of the codes does not represent a real disadvantage of the proposed approach. Moreover, one should take into consideration that no optimizations were sought in the process of code generation. A simple strategy, like the usage of all 256 ASCII characters instead of using only the 1-9 digits, can shorten significantly the length of the codes (e.g. 1.2.1.32.12.23 changes into 1.2.1.z.b.f). Approaches like Huffman code or other compression methods can be as well exploited for this purpose, but we will not consider these issues here.

4. The IRSLO System

Our improved semantic based information retrieval system comprises the same main components as found in any other retrieval system.

4.1. Question/Query Processing

This stage usually includes a keyword selection process. It may sometimes imply keyword stemming or other processing, and in most cases keywords to be employed in the retrieval stage are selected based on weights, frequencies and stop-words lists.

In IRSLO, we start this stage with a simple tokenization and part of speech tagging using Brill tagger (Brill, 1995). Next, collocations are identified based on WordNet definitions. We also identify the baseform of each word.

Depending on the notation employed by the user, we distinguish three keyword types. (1) Words with a semantic wildcard, denoted with #. (2) Words to be searched by their *DD-code*, denoted with @ (synonymy marker). (3) Words with no special notation, to be sought in the index in their given form. By default, we assume a # assigned to the answer type word, and no other notation for the rest of the words. All words that are denoted with # or @ are passed on to a word sense disambiguation component that solves their semantic ambiguity. Alternatively, this step can be skipped and a default sense of one with respect to WordNet is assigned, with reasonable precision (over 75% as measured on SemCor). The results reported in this paper are based on a simplified implementation that considers the second alternative. Next, *DD-codes* are assigned to words in text and subsequently used in the retrieval process. *DD-codes* are currently assigned only to nouns, considered to be the most informative words. See section 3. for more details regarding *DD-encoding*.

We also face the task of identifying relevant keywords to be included in a query. Extensive analysis of keywords identification was previously reported in (Pasca, 2001). We use a simplified keywords identification procedure, based on the following rules:

1. Use all proper nouns and quoted words.
2. Use all nouns.
3. Use all adjectives in superlative form.
4. Use all numbers (cardinals).
5. If more than 200 documents are returned, use the adjectives modifying the first noun phrase.
6. If no documents are returned, drop the nouns acting as modifiers. Particular attention is paid to abstract nouns, such as type, kind, name, where the importance of the roles played by a head and a modifier in a noun phrase are interchanged.

Any of these keywords may be expressed using its corresponding *DD-code*. The answer type word is also important. It practically denotes the type of information sought, whether is a *country*, an *animal*, a *fish*, etc. We use a simple approach that selects the answer type as the head of the first noun phrase. There are few exceptions from this rule, consisting of the cases where the head is an abstract noun like *name*, *type*, *variety* and so forth, and in such cases we select its modifier. If the answer detected is of a generic type, such as *person*, *location*, *organization*, then we replace it with the corresponding named entity tag. Otherwise, the answer type word is assigned a # semantic wildcard. Notice that the answer type selection process is invoked only if there is no word a priori denoted with #.

After all these processing steps, we end up with a query in IRSLO format. The words that were assigned a semantic wildcard # are now represented as *DD-code**. The words with a synonymy marker are simply replaced with their *DD-code* (thereby allowing for the retrieval of synonym words in addition to the word itself). The other words are replaced with their baseform. See Section 5.4. for representation examples.

4.2. Document Processing

Typically, documents are simply tokenized and terms are extracted, in preparation for the indexing phase. Optionally, stop-words are eliminated and words are stemmed prior to indexing.

In IRSLO, documents are processed following similar steps to question processing. First, the text is tokenized and part of speech tagged. We have an additional component that involves named entity recognition (Lin, 1994). Next, we identify compound words, apply a disambiguation algorithm or, alternatively, assign to each word its default sense from WordNet. Finally we assign to each noun its corresponding *DD-code*.

At this stage, we also identify paragraphs and store them as one paragraph per line. This helps improving efficiency during paragraph retrieval.

4.3. Indexing and Retrieval

The indexing process is not different in any ways with respect to a classic information retrieval system. A TF/IDF weight is assigned to each term. We index complex terms, including the *DD-codes* attached to each noun and the named entity tags, when available. No additional stemming or stop-words elimination is performed. The retrieval system allows for flexible searches, including regular expressions. Based on *DD-codes*, we have the capability of using the *semantic wildcard* operator, in addition to the lexical wildcard. We also have the capability of retrieving named entities of a certain type (e.g. perform a search for *person*). Moreover, we allow for boolean operators and for the new *paragraph operator* for a more focused search. Documents are ranked using the TF/IDF weight associated with each keyword.

5. Experiments with IRSLO

This section focuses on the application of the *semantic wildcard* and *paragraph operator* within the IRSLO system. First, the semantic wildcard enables searches for information along general-specific lines. Second, the paragraph indexing component limits the scope of keywords search to a single paragraph, rather than an entire document.

5.1. Experimental Setup

Several standard text collections are made available through the Information Retrieval community. For our experiments, we have selected the *L.A. Times* collection, which includes a fairly large number of documents. There are more than 130,000 documents adding up to 500MB of text. *L.A. Times* is part of the TREC (Text REtrieval Conference) collections.

The main advantage of standard text collections is the fact that question sets and relevance judgments are usually provided in association with the document collection.

About 1,393 questions have been released during the TREC-8, TREC-9 and TREC-10 Q&A TREC competitions. Relevance judgments are provided for the first two competitions, i.e. for 893 questions. From the 893 questions, we selected only the *What* type of questions, as being the most ambiguous types of questions and the best candidates for the semantic wildcard operator. Subsequently, we identified those questions known to have an answer in the *L.A. Times* collection[4], and out of these 75 questions were randomly selected for further tests.

For this question set, we have the knowledge about the information expected in response to each question (answer patterns provided by the TREC community). We also have a list of *docid*-s pointing to documents containing the answer for each question (list of documents judged to contain a correct answer by TREC assessors). This information helps us measure *precision* and *recall*.

5.2. Evaluating Retrieval Effectiveness

A common methodology in evaluating information retrieval systems consists in measuring *precision* and *recall*. *Precision* is defined as the number of relevant documents retrieved over the total number of documents retrieved. *Recall* is defined as the number of relevant documents retrieved over the total number of relevant documents found in the collection. Additionally, the *F-measure* proposed in (Van Rijsbergen, 1979) provides the means for combining recall and precision into one single formula, using relative weights.

$$F_{measure} = \frac{(\beta^2 + 1.0) * P * R}{(\beta^2 * P) + R}$$

where P is precision, R is recall and β is the relative importance given to recall over precision. During the system evaluations reported here, we considered both precision and recall of equal importance, and therefore β is set to 1.

Moreover, we employ the *success rate* measure (Woods, 1997) as an indicative of how many questions were answered by the system. The *success rate* for a question/query is 1 if relevant documents/answers are found, and 0 otherwise.

Finally, we evaluate IRSLO results using the TREC Q&A score, with a different mark assigned to an answer depending on its position within the final rank. A correct answer on the first position results in a maximum score of 1.00. The second position gets 0.50, the third position is scored with 0.33, the fourth with 0.25 and the fifth and last one acceptable receives 0.20 points.

5.3. Experiments

Three types of experiments were performed, to evaluate the performance of the new *semantic wildcard* and *paragraph operator*.

Experiment 1. Extract the keywords[5] from each question and run the queries formed in this way against a classic index created with the *L.A. Times* collection. The purpose of this experiment is to simulate classic keyword-based retrieval systems. The ranking is provided through a TF/IDF weighting scheme.

Experiment 2. Extract the keywords from each question and run the queries against the paragraph index. In paragraph

[4] The set of 893 questions was devised to ensure an answer in the entire TREC collection, including 2.5GB of text in addition to the *LA Times* collection that we employ in our experiments

[5] See Section 4.1. for the keywords selection procedure

indexing, we use a boolean model that includes the *paragraph operator*, plus a measure that determines the closeness among keywords to rank the paragraphs.

Experiment 3. Again, extract keywords from questions and run them against the paragraph index. Additionally, we allow the *semantic wildcard* (including named entity tags) to be specified in the keywords.

The results of experiments 1 and 2 are compared, to show the power of paragraph indexing. Experiments 2 and 3 provide comparative results to support the use of semantics, specifically the *semantic wildcard*.

The first experiment represents a classic keyword-based information retrieval run, and therefore we evaluate it in terms of *precision*, *recall* and *F-measure*. The second and third experiments are also evaluated in terms of *precision*, *recall* and *F-measure*. Additionally, we use the *success rate* and *TREC score*.

5.4. Walk-through Examples

This section gives several running examples of the IRSLO system, using the *semantic wildcard* and *paragraph operator*.

Example 1. What is the brightest star visible from Earth?
Relevant paragraph. In the year 296036 , Voyager 2 will make its closest approach to Sirius , the brightest star visible from Earth .
Comments. The query formed in this case is star# AND bright AND Earth. Only two answers are found by the system, and the one listed above, which is the correct one, is ranked on the first position. Sirius is defined in WordNet as a star, and consequently was annotated as such in the text.

Example 2. What kind of sports team is the Buffalo Sabres?
Relevant paragraph. Another religious broadcasting company , Tri - State Christian TV Inc. of Marion , Ill. , which was set up with the help of loan guarantees from Trinity , announced recently that it has purchased WNYB Channel 49 in Buffalo , N.Y. , from the Buffalo Sabres hockey team for $2.5 million .
Comments. The query employed is team# AND Buffalo AND Sabres. The original query team# AND sport AND Buffalo AND Sabres did not return any answers, and consequently the back off scheme was invoked and dropped noun modifiers. A total of six paragraphs are found in return to this question, all of them correct.

Example 3. What U.S. Government agency registers trademarks?
Relevant paragraph. After your application arrives at the Patent Office , it is turned over to an attorney who determines whether there is anything " confusingly similar "between your trademark and others [...]
Comments. Patent Office is a type of Government agency, and therefore the query U.S. AND government_agency# AND trademark leads to the correct answer.

Example 4. What cancer is commonly associated with AIDS?
Relevant paragraph. A team of transplant specialists at City of Hope National Medical Center in Duarte is among several groups nationwide that plan to test the experimental procedure on a small number of patients with AIDS - related lymphomas , or tumors of the lymph nodes .
Comments. The query employed is cancer# AND AIDS. The answer was found at rank 4, and it seems that none of the teams in the TREC competition identified this answer, because there is no direct reference in the text to cancer, but only a hidden relation from lymphomas to cancer. Our semantic model has the capacity to detect such non-explicit relations.

5.5. Results

Tests were performed using the benchmark of 75 questions. For each question, we run three experiments, as mentioned earlier. (1) Keyword-based information retrieval using a TF/IDF scheme. (2) Paragraph indexing and retrieval (i.e. enable the paragraph operator). (3) An experiment that involves both paragraph operator and semantic wildcard.

Precision, *recall* and *F-measure* are determined for all these experiments. We have also determined *success rate* and *TREC score*.

Ten sample requests of information are presented below, with their evaluations shown in Table 2. The following notations are used: P = *precision*, R = *recall*, F = *F-measure*, SR = *Success Rate*, TS = *TREC score*.

1. What American composer wrote the music for "West Side Story"?
2. What U.S. Government agency registers trademarks?
3. What U.S. state's motto is "Live free or Die"?
4. What actor first portrayed James Bond?
5. What animal do buffalo wings come from?
6. What cancer is commonly associated with AIDS?
7. What city does McCarren Airport serve?
8. What instrument does Ray Charles play?
9. What is the population of Japan?
10. What is the tallest building in Japan?

Cumulative results for all 75 questions are compared in Table 2. It turns out that the *F-measure* doubles when paragraph indexing is used with respect to document indexing, with increased *precision* and lower *recall*, as expected. The *success rate* is determined for the second and third experiments to evaluate the effect of the *semantic wildcard* over simple paragraph indexing, and an increase of 17% is observed. As of the *TREC score*, the additional use of semantics brings a gain of 34% with respect to simple paragraph indexing.

These results are very encouraging, and in agreement with the suggestions made in (Light et al., 2002) that query expansion and semantic relations are essential for increased performance, for information retrieval in general and Q&A systems in particular.

6. Related Work

Significant work has been performed in the field of semantics applied to information retrieval. The most important directions include: (1) query expansion (Voorhees, 1998), (2) phrase indexing (Strzalkowski et al., 1996), (3) conceptual indexing (Woods, 1997), (4) semantic indexing (Sussna, 1993), (Krovetz, 1997). In addition, the Semantic Web is a new field that considers the use of semantics for Web applications (Berners-Lee et al., 2001).

7. Conclusion

This paper has introduced the *semantic wildcard*, a novel operator that enables the use of semantics in information retrieval applications. The *semantic wildcard*, together with the new *paragraph operator*, were implemented in the IRSLO system. Experiments were performed on a collection of 130,000 documents with 75 *What*-questions extracted from the questions released during TREC competitions. Three experiments were performed. (1) One that

	Experiment												
Question	1. Classic IR			2. Par.op.					3. Sem.wildcard + par.op.				
number	P	R	F	P	R	F	SR	TS	P	R	F	SR	TS
1	0.14	0.21	0.17	0.50	0.07	0.12	1	1.00	0.75	0.86	0.80	1	1.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0	0.00	1.00	1.00	1.00	1	1.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0	0.00	1.00	0.67	0.80	1	1.00
4	0.25	0.44	0.32	0.43	0.17	0.24	1	1.00	0.16	1.00	0.27	1	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0	0.00	0.43	1.00	0.60	1	0.33
6	0.08	0.84	0.14	0.03	1.00	0.03	1	0.00	0.37	0.74	0.49	1	0.25
7	1.00	1.00	1.00	1.00	1.00	1.00	1	1.00	1.00	1.00	1.00	1	1.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0	0.00	0.38	1.00	0.55	1	0.50
9	0.03	1.00	0.05	0.04	0.50	0.07	1	0.00	0.08	0.33	0.13	1	1.00
10	0.03	0.50	0.06	0.40	0.50	0.44	1	1.00	1.00	1.00	1.00	1	1.00

Table 2: Precision, recall, F-measure, success rate and TREC score for 10 sample requests of information

	Experiment		
Measure	1. Classic IR	2. Par.op.	3. Sem.wildcard. + par.op.
Precision	0.05	0.12	0.12
Recall	0.66	0.57	0.61
F-measure	0.092	0.19	0.20
Success rate	-	66.0%	77.3%
TREC score	-	43.4%	58.3%

Table 3: Comparative results for (1) keyword-based information retrieval (2) paragraph operator and (3) paragraph operator + semantic wildcard

simulates classic keyword-based information retrieval with a TF/IDF weighting scheme. (2) A second experiment that implements the *paragraph operator*. (3) Finally, a third experiment where both *semantic wildcard* and *paragraph operator* are employed. Various measures were used to evaluate the performance attained during these experiments, and all measures have proved the efficiency of our *semantic wildcard* operator, respectively the *paragraph operator*, over keyword-based retrieval techniques. As a follow-up analysis, it would be interesting to determine the *min* and *max* bounds proposed in (Light et al., 2002) for the precision achievable on a question set when the semantic wildcard is enabled.

8. References

T. Berners-Lee, J. Hendler, and O. Lassila. 2001. The Semantic Web. *Scientific American*, 1(501), May.

E. Brill. 1995. Transformation-based error driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–566, December.

H.S. Heaps. 1978. *Information Retrieval, Computational and Theoretical Aspects*. Academic Press.

E. Hovy, L. Gerber, U. Hermjakob, C.-Y. Lin, and D. Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the Human Language Technology Conference, HLT 2001*, San Diego, CA.

R. Krovetz. 1997. Homonymy and polysemy in information retrieval. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97)*, pages 72–79.

M. Light, G.S. Mann, E. Riloff, and E. Breck. 2002. Analyses for elucidating current question answering technology. *Journal of Natural Language Engineering (forthcoming)*.

D. Lin. 1994. Principar - an efficient, broad-coverage, principle-based parser. In *In Proceedings of the Fifteenth International Conference on Computational Linguistics COLING-ACL '94*, pages 42–48, Kyoto, Japan.

R. Mihalcea. 1999. Word sense disambiguation and its application to the Internet search. Master's thesis, Southern Methodist University.

G. Miller. 1995. Wordnet: A lexical database. *Communication of the ACM*, 38(11):39–41.

M. Pasca. 2001. *High performance question answering from large text collections*. Ph.D. thesis, Southern Methodist University.

J. Prager, D. Radev, and K. Czuba. 2001. Answering what-is questions by virtual annotation. In *Proceedings of the Human Language Technology Conference, HLT 2001*, San Diego, CA.

T. Strzalkowski, L. Guthrie, J. Karigren, J. Leistensnider, F. Lin, J. Perez-Caballo, T. Straszheim, J. Wang, and J. Wilding. 1996. Natural language information retrieval, TREC-5 report. In *Proceedings of the 5th Text Retrieval Conference (TREC-5)*, pages 291–314, Gaithersburg, Maryland, November.

M. Sussna. 1993. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the second international conference on Information and knowledge management CIKM '93*, pages 67–74, Washington, November.

C.J. Van Rijsbergen. 1979. *Information Retrieval*. London: Butterworths. available on-line at http://www.dcs.gla.ac.uk/Keith/Preface.html.

E.M. Voorhees. 1998. Using WordNet for text retrieval. In *WordNet, An Electronic Lexical Database*, pages 285–303. The MIT Press.

W.A. Woods. 1997. Conceptual indexing: A better way to organize knowledge. Technical Report SMLI TR-97-61, Sun Microsystems Laboratories, April. available online at: *http://www.sun.com/ research/techrep/ 1997/abstract-61.html*.