Retrieving Descriptive Phrases from Large Amounts of Free Text

Hideo Joho
University of Sheffield
Western Bank, Sheffield
S10 2TN, UK
+44 (0)114 222 2675
h.joho@sheffield.ac.uk

Mark Sanderson
University of Sheffield
Western Bank, Sheffield
S10 2TN, UK
+44 (0)114 222 2630
m.sanderson@sheffield.ac.uk

ABSTRACT

This paper presents a system that retrieves descriptive phrases of proper nouns from free text. Sentences holding the specified noun are ranked using a technique based on pattern matching, word counting, and sentence location. No domain specific knowledge is used. Experiments show the system able to rank highly those sentences that contain phrases describing or defining the query noun. In contrast to existing methods, this system does not use parsing techniques but still achieves high levels of accuracy. From the results of a large-scale experiment, it is speculated that the success of this simpler method is due to the high quantities of free text being searched. Parallels between this work and recent findings in the very large corpus track of TREC are drawn.

Keywords

Information retrieval, descriptive phrase, large corpora.

1. INTRODUCTION

The opportunities to use online text databases for the mining of valuable information are great. As these stores increase in size, the possibility of accurately extracting that information using increasingly simpler techniques seems to also increase. This principle was demonstrated in the results of the Very Large Collection (VLC) track of TREC-6 [4]. In the track the same topics as those used in the ad hoc task were applied to a 20Gb collection, which is a superset of the standard collection (2Gb). When comparing the effectiveness of systems retrieving on the two collections, it was noted that precision measured at rank position twenty was consistently higher for the systems searching the larger VLC. The reason for this was not explained by differences in retrieval techniques between the two runs, but that in the VLC, there were simply more relevant documents that held a high percentage of the specified query terms. In other words, because the collection was larger, users had a better chance of finding relevant documents that used the same combination of words and phrases as found in their query[1]. This effect only occurs with high precision measures: when considering all relevant documents in the VLC, the retrieval systems were not performing better, just a greater fraction of relevant documents were appearing in the top ranked positions.

The result from VLC implies that for retrieval tasks where finding a small fraction of relevant items is more important than finding them all, use of large corpora and simplistic matching techniques is likely to be a promising approach. Retrieving the descriptive or defining phrases of a proper noun is one such task.

Discovering the meaning of a particular word or phrase can be vital to the understanding a text. The conventional source of such a meaning (a dictionary) is often inadequate when the word in question is a proper noun. Other locations of reference information such as encyclopædia or online services, may be not easily accessible, have wide enough coverage, or be sufficiently up to date. Locating definitions within free text documents is an alternative approach[2].

A noun phrase defining or describing another noun within the same sentence is known as an *appositio*n. For example in the paragraph above, "encyclopædia" is described in the same sentence by the phrase "locations of reference information". While not a perfect or complete definition, it nevertheless provides some information on the meaning of the term. Finding this information within free text may at first seem to be a hard problem, however, work in a related area has shown that descriptions worded in a certain way can be located.

Hearst studied the problem of locating the IS-A lexical relationship within corpora [6][3] (taking an example from her paper, a broken bone *is an* injury). She showed that a word and its hypernym (the more general term) were often found together in sentences linked by common phrases; she manually and then semi-automatically located patterns that were reliable indicators of

[1] This will only work if the query does not match well to non-relevant documents.

[2] Motivation for this work came after one of the authors moved to the US and encountered a large number of references, in documents or in conversations, to proper nouns specific to US culture, which were not easily found in reference works.

[3] Hearst originally reported this work in an earlier publication [5].

the IS-A relation. These *key phrases* were "such as", "and other", "or other", "especially", and "including". Hearst looked for these phrases in the following patterns.

- *(dp* such | such *d*p) as *qn*
    - e.g. "...injuries such as broken bones"
- *qn* (and | or) other *dp*
    - e.g. " broken bones and other injuries..."
- *dp* especially *qn*
    - e.g. "... injuries especially broken bones"
- *dp* including *qn*

Here, *qn* (a noun later referred to as the *query nou*n) is the hyponym, and *dp* (a noun phrase later referred to as the *descriptive phras*e) is the hypernym.

Hearst reported on the accuracy of the phrases and discussed using the large number of "IS-A" relations listed in WordNet [8] to try to find other such indicative phrases. However, she was unable to find a fully automatic method for locating them.

Hearst stated that her technique "…is meant to be useful as an … aid to lexicographers…". Her work, however, has a broader applicability: the hypernym of a word appearing in the same sentence of that word is an apposition. Therefore, the key phrase method can be used as a starting point for building a system to locate descriptions of nouns. The coverage of such a method is likely to be poor, as the key phrases are relatively rare. Users of such a system are unlikely to want to see all descriptions (preferring high precision to high recall), as long as a few are found relatively accurately, most will be satisfied. Similar to the VLC track of TREC, as long as the corpus being searched is large enough, the likelihood of locating a description of the query noun in a "key phrase form" will hopefully be high.

It is with a strategy based in part on Hearst's lexical relation method that a descriptive phrase retrieval system was created. Starting with a review of previous work in the area, the rest of this paper describes the system's design, building, and testing. This is followed by speculation on possible future work and the paper closes with conclusions.

2. Previous work

The descriptive phrase retrieval system has similarities to some of the MUC (Message Understanding Conference) tasks and to the field of Question Answering (QA).

The annual MUC conferences of the 1990s tested research groups' abilities at various Information Extraction techniques [2]. The basic task was to fill a template with information extracted from a stream of documents. One of the slots in the template was used to hold any extracted description information of the entities identified in the texts. This task has clear similarities to the problem described in this paper. However, the methods used to process the documents in MUC were usually specialised to a particular domain making use of parsing technologies. Therefore, the solutions proposed were of less use as the aim of the work in this paper was to have a system as widely applicable as possible.

Interest in QA has long been active. Cooper described what he called a "fact retrieval system", which would search for text fragments in a small document collection which confirmed or denied a query statement [3]. Using a hand built parser working over a small set of sentences, Cooper reported some experimental success on his limited domain system. With a more general approach, Kupiec described a system that searched an online encyclopædia for answers to a set of closed class questions (in this work only "Who..." and "What..." questions were fully implemented, e.g. "Who's won the most Oscars for costume design?") [7]. The system used a parser to locate and type the important phrases within a question. From this information Boolean queries were constructed to search for sentences in the encyclopædia. These sentences were themselves parsed to try to find potential answer phrases. Secondary queries were then constructed to try to confirm which, if any, of these identified phrases was a valid answer. For example, "Who..." questions were expected to have a person's name as an answer, for these, the secondary queries were used to confirm a potential answer was actually a name. Kupiec tested his system on seventy questions taken from the board game Trivial Pursuit. The highest ranked sentence returned by the QA system for each query was correct 53% of the time. Within the top five sentences, the correct answer was found in 74% of the questions. More recently, there has been a growth of interest in QA with the Question Answering track of TREC. According to [13], two of the better performing systems in the TREC-8 evaluation were from [9] and [12], both of who made extensive use of parsers, existing knowledge bases and pre-calculated question templates. The differences between the work of this paper and QA are described in the next Section (2.1).

Work on explicitly extracting descriptive phrases was recently conducted by Radev [11]. His system was presented with a user specified query noun and it would locate and return a list of descriptive phrases of that noun extracted from a database of news web sites. Although it is not described in detail, it would appear that the system used an grammar to locate one of two basic syntactic patterns and their variants:

- *dp qn*
    - "Politician Tony Blair ..."
- *q*n, *d*p, or *d*p, *q*n
    - "Tony Blair, politician, ..."

The system was also capable of typing the descriptive phrases, deciding for example if the phrase was a location, an occupation, an age, etc. After manually examining 611 descriptions identified by the system, Radev found that they were correct 90% of the time. No results on the accuracy of the typing of descriptions were reported.

2.1 Design of the system

Question Answering is a more general problem than the locating of descriptive phrases. A system performing this more restricted task can be thought of as a specialised QA tool capable of answering the questions "Who is *q*n?" and "What is *q*n?". There are advantages to be had by concentrating on this smaller problem. First, as will be seen below, solutions to this particular sub-problem of the QA task perform well without use of specialised domain knowledge or language tools and so can be expected to operate in a wide range of domains with little or no adjustment. The second advantage stems from the answers expected for these particular problems. In the wider QA task, one cannot assume how often the answer will occur in the collection to be searched. When searching for the descriptive phrases of a query noun, however, it is believed that there is a greater likelihood of the descriptions appearing many times across many documents and, as will be seen, this abundance of answers can be exploited in ways perhaps less used in QA.

The design of the system was as follows, given a query noun *(q*n), all documents holding it were retrieved and from them all sentences containing *qn* were extracted. These were ranked based on a series of criteria described below. We evaluated the top five and top ten highest ranked sentences for relevance. The system was judged successful for a particular query if at least one sentence in the ranked list contained information answering, at least in part, the who or what question. It may seem that this is a rather low measure of success, however, it is believed that in this task, users will be more than capable of locating the real description and ignoring the other non-relevant sentences.

Three criteria were used to rank the sentences:

- presence of a key phrase in the sentence,
- a high number of common terms,
- and the position of the sentence as found in the document.

Each of these features is now described, followed by the means of their combination.

2.2 Key phrases

Key phrases were used as the basis of the detection system. Using the phrases already listed in Section 1, three more were added: one to find acronyms, one to find "is a" type descriptions and another to locate appositions parenthesised by commas (similar to [11]). The patterns were defined as follows

- *qn (d*p) or *(d*p) *qn*
  - e.g. "MP (Member of Parliament)"
- *qn* (is | was | are | were) (a | an | the) *dp*
  - e.g. "Tony Blair is a politician..."
- *q*n, (a | an | the) *d*p
  - e.g. "Tony Blair, the politician, ..."
- *q*n, which (is | was | are | were) *d*p,
- *q*n, (a | an | the) *dp* (. | ? | !)
- *q*n, *d*p, (is | was | are | were)

The system to match these patterns required approximately seventy lines of Perl script. In contrast, both [6] and [11] used parsers to process candidate sentences. In this work, however, it was decided to avoid the use of these more complex tools so as to examine how successful a technique based on simple pattern matching would be. If it proved to be just as effective, this approach would in all likelihood be preferable to a parser based method because of its speed, simplicity, and potential applicability to a wide domain.

In the system for this paper, it was judged that a set of ranked sentences should always be returned to the user regardless of the success of the pattern matching. It was quite possible that descriptions of the query noun were present but had not been found through a mistake or lack of coverage in the patterns being matched. Therefore two additional more general criteria were included. It was anticipated that they would act as both a fallback when no patterns matched and as a way of ranking sentences found to match a pattern hopefully ranking better descriptions higher. The criteria were based on the information retrieval (IR) related techniques of location within a document and cross-document term weighting and are now described.

2.3 Sentence position

It seemed reasonable to expect that if a noun was used a number of times within a document, then any accompanying description of it was going to be found nearer the start than the end. Therefore the ordinal position of sentences containing the query noun (e.g. 1st, 2nd, 3rd, etc) was noted and used in the ranking calculation. The earlier sentences were given a higher score.

2.4 Word counting method

If a query noun was described in one document, it was assumed that it was likely to be described in others. It was anticipated that this repetition of the same or similar descriptions across documents could be exploited. A simple word counting technique was devised to examine all sentences retrieved in response to a query noun and to find words co-occurring with the noun that commonly co-occurred across documents. A number of methods were tried, the one found to be most successful (evaluated on a test collection described below) involved examining in each document (containing *q*n) only the first sentence that *qn* occurred in. The case of the words in these sentences was normalised, stop words were removed, and a stemmer was applied [10]. The frequency of occurrence of all remaining terms in the sentences was calculated and the twenty most frequent were noted. When ranking all matching sentences, each was assigned a score based on the number of the top twenty terms present. Those containing more of these common terms were given a higher score.

2.5 Tuning and combining the criteria

Before any evaluation of the system could take place, it was necessary to tune it to try to get an optimal performance from the three sentence ranking criteria. Therefore, a descriptive phrase test collection was created half of which (the training set) was used for tuning the system, the other half (the test set) used later for evaluation. This section describing the building of the collection and its use in system tuning.

2.5.1 Building the collection

The document collection to be searched was a set of LA Times articles from 1989 & 1990 (475Mb, taken from TREC). The advantage of using the TREC data was the relative ease of access to it. The disadvantage was that the authors were left with the challenge of thinking of a large number of query nouns that might have been described ten years ago. Seventy six such queries were thought by the authors (or suggested by colleagues) of which ten were not present in the collection and a further sixteen that only occurred a small number of times (<20). These final sixteen were removed, as it was felt that there was little challenge in finding those sentences that described them as a user would probably be willing to read through all the sentences retrieved by those queries. The remaining fifty queries were used in the test collection. They occurred in 16,111 sentences; each of these was assessed for relevance. As stated above, a sentence was judged relevant to a query if it contained information that would help a user understand more about the noun they queried on. As with all relevance judgements there were some sentences that were hard to decide on. For example in one sentence containing the query "Adolf Hitler", he is described as a person.

"...not only for Kraft but for people such as Adolf Hitler and Adolf Eichmann..."

Although this is a valid description the sentence was judged to be not relevant. It is hard to imagine someone not knowing that a person's name refers to a person. This type of problem was, however, the exception and for most sentences it was clear if it contained a description or not.

Note that judging relevance was on sentences and not on extracted descriptions or some other unit of text smaller than a sentence. This made it possible to automatically process the results of the descriptive phrase system in a similar manner to traditional IR evaluation. This contrasts with the evaluation method used in the QA track of TREC, where judges had to examine each individual answer from each run of each participating system [13].

Evaluation was measured using TREC-like measures concentrating on the rank-based ones: precision at ranks 1, 5, and 10. In addition the number of queries for which at least one correct answer was found within those rank positions was also calculated.

2.5.2 Tuning key phrases

While writing the key phrase pattern matcher, it was clear that some of the patterns were going to be better at locating descriptive phrases than others. Therefore, the training set was examined to measure the accuracy of the patterns. Table 1 shows this along with their coverage.

As can be seen, all the patterns are relatively rare (compare with numbers for no patterns), though the comma parenthesised apposition and "such as" were the most used, "and other" proved to be most accurate. These figures were used in the ranking of sentences with those containing the more accurate key phrases getting a higher score.

2.5.3 Combining the criteria

A series of tests were run on different combinations of the scores gained from the key phrases, the sentence location, and co-occurring word counts. A weighted sum of the scores was found to work best

$$aKPW + bWC + c(d - SN)$$

where *KPW* is the key phrase accuracy weight taken from above, *WC* is the co-occurring word count, *SN* is the sentence number (1st, 2nd, etc sentences occurring earlier in documents got a higher score). The values of a, b, c, *d* (tuning constants) were set to 2000, 1, 75, and 500 respectively after a series of trials on the training set. In the trials, different combinations of the constants were examined, each time measuring the effectiveness of the system using the performance measures outlined in Section 4.

Table 1. Accuracy of key phrase pattern matcher

	Not rel	**Rel**	**Total**	**Accuracy**
No pattern	6424	872	7296	12.0%
especially	0	0	0	0.0%
qn, dp,	89	63	152	41.4%
is a	23	18	41	43.9%
including	20	17	37	45.9%
or other	1	1	2	50.0%
such as	59	59	118	50.0%
acronym	14	23	37	62.2%
and other	9	23	32	71.9%

3. The system in action

Working with the training set seemed to show that the system was producing reasonable results. To illustrate the following are the manually extracted descriptions taken from a few high ranking relevant sentences selected at random from the query set.

- John Lennon - '60s rock artist
- Tofu - [no phrase found in top 10]
- Hitachi - top manufacturer
- NEC - established portable computer company
- Nintendo - a 99-year-old Japanese firm
- Star Wars - defence program

As with the Hitachi query, sometimes the description is too general. But for the most part, these descriptions seem to be reasonable. If someone knew nothing about these query nouns, the descriptions here would give that person more information than they had before.

The type of description found within a corpus clearly depends on the audience that the corpus was written for and how much it is thought that they already know. The documents used in this work were ten-year old US newspaper articles. Unless the queries being searched have an American or international significance, it is unlikely they will be found in the corpus.

The shortest description found was one word describing Bob Dylan as an "artist", one of the longest was of US TV news presenter Diane Sawyer described as "the Grace Kelly of television - the perfectly groomed Ice Queen whose every gesture seems scripted".

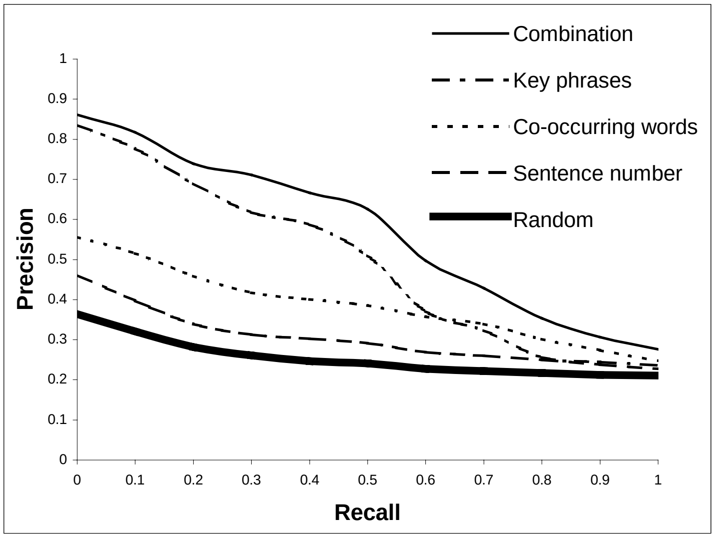

Figure 1. P-R graph of four strategies

4. Evaluation

Now that the system had been tuned, it was evaluated on the test set. Two experiments were conducted, the first was a test of the effect of collection size on the key phrase matching system, the second was a full evaluation of the system.

4.1 Collection size

In this experiment random samples of the test set were taken and the effectiveness of the key phrase criteria was measured for each of these samples. Results from this experiment are shown in Table 2. Samples taken were for 10%, 25%, 50%, 75% and 100% of the test set.

The results of this test show that as the collection got smaller, the effectiveness of the system reduced. This is of course because the likelihood of the system finding a sentence holding one of the key phrases reduced as the collection got smaller, therefore, the precision of the system in the top ranked sentences fell. This result echoes that obtained on the VLC collection described in Section 1.

Table 2. Precision (Key phrase) and collection size

P. at rank	**100%**	**75%**	**50%**	**25%**	**10%**
1	0.75	0.78	0.69	0.63	0.62
5	0.51	0.51	0.46	0.38	0.32
10	0.42	0.40	0.35	0.28	0.24

4.2 Testing the system

The effectiveness of each of the individual sentence ranking criteria was tested along with combination formula derived above. Unlike most IR test collections, the ratio of relevant to non-relevant was relatively high. Therefore, it was important to establish a random baseline as well. To achieve this, for each of the fifty queries, 100 random orderings of the sentence collection (in the test set) were generated and the average effectiveness of these cumulative runs was measured.

A precision-recall graph was plotted showing the effectiveness of the four strategies plus random retrieval (see Figure 1). As can be seen, the three criteria and their combination do better than random retrieval[4]. A sentence ranking based purely on the key phrase weights was extremely effective, except for high recall situations where the co-occurring word counting method was better. The most effective technique for finding descriptive phrases, however, was the combination formula, which, through a t-test, was found to be significantly better than any of the other methods, including key phrases. The difference between the combination and key phrase methods was found to be significant at $p<0.05$ for recall levels 0.1 and 0.2 and significant at $p<0.01$ for all higher values of recall. When evaluated with precision oriented measures a similar picture emerged. Table 3 shows precision

Table 3. Precision of each strategy

P. at rank	**Comb.**	**Key phr.**	**Co-occ. Words**	**Sent. No.**	**Rand.**
1	0.76	0.75	0.37	0.25	0.20
5	0.57	0.51	0.35	0.27	0.20
10	0.46	0.42	0.35	0.27	0.20
15	0.42	0.36	0.33	0.24	0.20
20	0.38	0.32	0.32	0.23	0.20
30	0.32	0.26	0.28	0.22	0.19
100	0.17	0.15	0.16	0.15	0.14

[4] The monotonically decreasing line of the random system is an artefact of the standard interpolation used when measuring precision at fixed recall levels. Although the relevant documents along the ranking are evenly (randomly) distributed, when precision is measured on this distribution the line shown in the figure is the result.

measured at rank positions ranging from one to one hundred.

As can be seen, the combination method is consistently higher than the key phrase. Like the precision recall graph, significance testing was performed: the combination method was found to be significantly better than key phrase for all rank positions from five through to 100 (p<0.01). As stated at the end of Section 2.5.1, the percentage of queries with at least one correct answer in the top *n* was also calculated. Here *n* was chosen to be 5 and 10 as it was felt that a user would be willing to look through this number of sentences. For the best performing method (combination) 90% of the queries had a correct answer in the top 5 (compared with 22% for random) and 94% correct in the top 10 (c.f. random 31%).

5. Conclusion

This paper has presented a means of locating descriptive phrases of a user specified query noun. A method designed to locate lexical relations within text, using key phrases, was applied to this new task. It was adapted by expanding the number of key phrases and by incorporating additional within document and cross-document information. More complex linguistic processing and reliance on lexical resources was avoided.

Through large-scale experimental testing, results showed the system was successful. In tests on a collection of over 8,000 sentences (the test set), the system was capable of ranking a description-bearing sentence within the top ten for 94% of the tested queries: a level of accuracy anticipated to be acceptable to most users.

The experiment confirmed earlier results from the TREC VLC track that showed that simple methods searching on a large corpus can produce accurate results.

6. Future Work

There are a number of possible areas of further work.

6.1 Extracting descriptions

The descriptions shown in Section 3 were manually extracted from the sentences. Currently the system can only present whole sentences to the user. Although the query noun and some of the words of the descriptive phrase can be highlighted, it would be preferable for the phrase to be automatically extracted. For sentences containing key phrases, a prototype extraction system has already been written. Although it has not been formally tested, it does appear to work well. No automatic method for extracting phrases from sentences where no key phrase was found has been created and this is something we plan to pursue.

6.2 Managing ambiguity and time

In Section 3 the descriptive phrase for the query "Star Wars" illustrates the ubiquitous problem of ambiguity (the term refers equally well to the defence program as it does to the science fiction film). Methods to classify a word into its different senses have been well researched and we plan to apply some of these techniques for this problem.

Related to ambiguity is the issue of time, the descriptions of things will alter: people will change their jobs for example. A means of detecting and presenting this change to users will also be explored.

6.3 Generality of descriptive phrases

From the examples shown in Section 3, it is clear that there are different levels of descriptions ranging from the general to the specific. We believe that it will be possible to estimate the generality or specificity of a description through use of a range of simplistic methods: using the description's inverse document frequency (or the idf of its component words) may provide an estimate of the specificity of the phrase. Use of this simple statistic has been used for this purpose before [1]. It may also be possible to examine the range of proper nouns that a particular description has been applied to and use this as a means of estimating generality of the phrase.

6.4 The web

Given that the system is designed to work best on very large corpora, the obvious VLC on which to apply the phrase description system to is the Web. We plan to use our system as a front end for an existing search engine (e.g. AltaVista) using the engine to retrieve relevant documents and then using our system to locate the descriptive phrases. We anticipate that this will further improve the accuracy of our simple yet effective system.

7. References

[1] Caraballo, S.A., Charniak, E., "Determining the specificity of nouns from text", *Proc. the joint SIGDAT conference on empirical methods in natural language processing (EMNLP) and very large corpora (VLC)*, 63-70, (1999).

[2] Chinchor, N.A., "Overview of MUC-7/MET-2", *Proc. the Message Understanding Conference Proceedings MUC-7*, (1998).

[3] Cooper, W.S., "Fact Retrieval and Deductive Question-Answering Information Retrieval Systems", *Journal of the AC*M, ACM Press, 11(2), 117-137, (1964).

[4] Hawking, D., Thistlewaite, P., "Overview of TREC-6 Very Large Collection Track", *NIST Special Publication 500-240: The Sixth Text REtrieval Conference (TREC 6*), E.M. Voorhees, D.K. Harman (eds.), 93-106, (1997).

[5] Hearst, M., "Automatic Acquisition of Hyponyms from Large Text Corpora", *Proc. the 14th International Conference on Computational Linguistics (COLING 92*), 539-545, (1992).

[6] Hearst, M.A., "Automated Discovery of WordNet Relations", *WordNet: an electronic lexical databas*e, C. Fellbaum (ed.), MIT Press, (1998).

[7] Kupiec, J., "MURAX: A Robust Linguistic Approach For Question Answering Using An On-Line Encyclopedia", *Proc. the 16th annual international ACM SIGIR conference on Research and Development in Information Retrieva*l, 181-190, (1993).

[8] Miller, G.A., "WordNet: A lexical database for English", *Communications of the AC*M, 38(11), 39- 41, (1995).

[9] Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Goodrum, R., Girju, R., Rus, V., "Lasso: A Tool for Surfing the Answer Net", *NIST Special Publication XXX-XXX: The 8th Text REtrieval Conference (TREC 8)*, (1999).

[10] Porter, M.F., "An algorithm for suffix stripping", *Program - automated library and information system*s, 14(3), 130-137, (1980).

[11] Radev, D.R., McKeown, K.R., "Building a Generation Knowledge Source using Internet-Accessible Newswire", *Proc. the 5th Conference on Applied Natural Language Processing (ANLP*), 221-228, (1997).

[12] Singhal, A., Abney, S., Bacchiani, M., Collins, M., Hindle, D., Pereira, F., "AT&T at TREC-8", *NIST Special Publication XXX-XXX: The 8th Text REtrieval Conference (TREC 8)*, (1999).

[13] Voorhees, E.M., Tice, D.M., "The TREC-8 Question Answering Track Evaluation", *NIST Special Publication XXX-XXX: The 8th Text REtrieval Conference (TREC 8)*, (1999).