

# The PHASAR Search Engine

Presented at NLDB2006. Copyright Springer

Cornelis H.A. Koster, Olaf Seibert and Marc Seutter

Dept. Comp. Sci.  
Radboud University Nijmegen  
The Netherlands.

**Abstract.** This article describes the rationale behind the PHASAR system (Phrase-based Accurate Search And Retrieval), a professional Information Retrieval and Text Mining system under development for the collection of information about metabolites from the biological literature. The system is generic in nature and applicable (given suitable linguistic resources and thesauri) to many other forms of professional search. Instead of keywords, the PHASAR search engine uses Dependency Triples as terms. Both the documents and the queries are parsed, transduced to Dependency Triples and lemmatized. Queries consist of a set of Dependency Triples, whose elements may be generalized or specialized in order to achieve the desired precision and recall. In order to help in interactive exploration, the search process is supported by document frequency information from the index, both for terms from the query and for terms from the thesaurus.

The professional retrieval process as found e.g. in Bio-Informatics can be distinguished into two phases, viz. Search and Analysis. The *search* phase is needed to find the relevant documents (achieving recall), the *analysis* phase for identifying the relevant information (achieving precision). Analysis can be performed visually and manually, partly automatically and partly interactively.

Both processes are complicated by the fact that the searcher has to guess what words will be used in the documents to express the relevant facts about the topic of interest. For the search process, many online databases are available, but plowing through the many hits provided by a word-based search system is hard work. Analysis is a heuristic process, demanding all the searcher's knowledge, experience and skills, for which presently only primitive keyword- or pattern-based support is available. The searcher may become very skilled in search and analysis, but the process is only partially automated and when searching for another topic the whole manual process must be repeated.

In this paper we describe the ideas behind a novel search engine based on linguistically derived phrases, designed to support the search and analysis processes involved in retrieving information from the Biomedical literature. In section 1 we describe our approach to searching, which combines linguistic techniques with Information Retrieval. In section 2 we illustrate a possible way-of-working by

means of an example. In section 3 we briefly describe the status of the implementation of PHASAR and the Medline/P collection used in the project. Finally, in section 4 we reflect on what has been achieved and what challenges are still ahead.

## 1 The PHASAR approach

Many years of TREC challenges and evaluations have produced diminishing improvements in the precision and recall from best-practice query-based retrieval systems. In particular, the use of phrases in query-based retrieval has never led to the hoped-for breakthrough in accuracy. Due to the ubiquity of word-based Search Engines like Lycos and Google, searchers have come to consider a combination of a few words as a natural and precise way to formulate a query, and they have learnt to cope with the deluge of hits that the query may cause by looking at only a few of them.

In designing the PHASAR system, we tried to “rethink” the support of phrase-based search and retrieval and the corresponding way-of-working.

### 1.1 Professional search

*Professional search* may be distinguished from what could be termed *incidental search* by the following characteristics:

1. the search is performed by professionals, in their own area of competence
2. it is worth investing some (expensive) time and effort
3. the search is over a very large collection of documents, including many which may be relevant
4. the information need is clear but complex, the user can recognize relevant answers
5. the information need may have to be answered by gathering (passages from) many documents
6. repetitions of the search process with small modifications in the query are routine.

Contrast this with incidental search, which may have a vicarious and opportunistic character; where the searcher may easily be side-tracked; where out of a million hits only the first ten are ever considered; where the main problem is in formulating the information need, which can often be answered by a single document.

### 1.2 From specifying the question to specifying the answer

In the traditional view of Information Retrieval, the searcher has to formulate his information need, and the task of the search algorithm is to find those documents (best) answering the information need. Taking this approach too literally may lead to a quest for complete, consistent and formal specifications of the

information need, which are not only hard to construct but for which also no effective search algorithm exists short of a reasoning agent with a complete model of the world. This works well only for severely limited application areas (no world model needed) and for limited query formalisms (where a weak but efficient inference mechanism suffices, like the Vector Space Model).

We take a different point of view: The searcher has to indicate what formulations in the documents are expected to be relevant to his information need. He has to guess what the answers will look like, rather than the question (a form of query-by-example).

### 1.3 Dependency Triples as terms

The Information Retrieval community has for a long time had high expectations of the use of phrases as index terms instead of keywords, but in practice the benefits were found hard to realize (see [Strzalkowski, 1999], in particular [Sparck Jones, 1999]).

Rather than single words, word pairs [Fagan, 1988, Strzalkowski, 1995] or sequences of words (word n-grams; chunks; complete noun phrases) we shall use *dependency triples* as terms in the document representation and the search process. A dependency triple (DT) is a pair of (lemmatized) words (the *head* and the *modifier*) occurring in the document in a certain *relation*, which we denote by [head RELATION modifier].

Dependency triples (aka *dependency relations*) have already been used successfully in Question Answering [Bouma et al, 2005, Cui et al, 2005] for the precise matching of answers from a database to questions, but we propose to use DT's directly in a search interface querying large collections of free text without special preprocessing. They are closely related to the Index Expressions of [Grootjen and van der Weide, 2004].

The following table illustrates the major syntactic dependency relations.

relation	examples
subject relation	[penicillin SUBJ reduce] [it SUBJ improve]
object relation	[cause OBJ reduction] [interrupt OBJ it]
attributive relation	[cancer ATTR lung] [cancer ATTR alveolar]
preposition relation (various prepositions)	[interact WITH medium] [interaction WITH medium] [relative TO period]

In order to obtain this representation, both documents and queries are syntactically analyzed (including NER and robust lexicalization) and transduced to dependency trees that are then unnested to dependency triples in which all word forms are lemmatized. In the transduction, all elements of the text that do not

contribute to its *aboutness* (see [Bruza and Huibers, 1996, Arampatzis et al., 2000]) are elided: determiners, quantifiers, auxiliary verbs.

A query matches any document that contains *all of the words and triples derived from the query*. Besides this *by document* matching, it is also possible to invoke stricter matching rules, matching by passage, sentence or phrase, which raise precision and reduce recall. The most precise matching strategy, phrase matching, amounts to the matching of *factoids*, taking the non-composed phrases of the documents as factoids.

#### 1.4 Linguistic Normalization

Natural language admits great variability in the expression of a given piece of information. A writer or speaker hates to use the same formulation every time when referring to some person, object or fact, and will go out of his way to bring variation to his formulations. This holds not only in literature, but also for journalistic and even for scientific texts. This linguistic variability has to be countered in some way in full-text retrieval.

Rather than letting the searcher predict all possible variations in the form of the answer, we systematically apply normalizations to reduce the linguistic variation in both the documents and the queries:

- morphological normalization by lemmatization
- syntactical normalizations, e.g. standardizing the word-order and transforming passive formulations into active ones
- lexico-semantic normalization using multiple thesauri
- elimination of all elements from the text that do not contribute to the aboutness, such as time, degree and modality.

Ideally, *all phrases that are equivalent formulations of the same factoid should be conflated to one same phrase*.

#### 1.5 The fruit machine model

Although our intended users are professionals, they are in most cases not professional linguists. The user interface should therefore be such that they are not burdened with arcane linguistic notions or notations. On the other hand, they can be expected to have a general understanding of basic linguistic notions, like the structure and function of subject, verb, object and complements in a sentence.

In the PHASAR system, a query is in its most extensive form a basic sentence in natural language, in English an SVOC sentence. The user interface expects any or all of its four elements (subject, verb, object and eventual complement) in a box of its own on the screen, resulting in a display of four boxes that looks like a *fruit machine*, a Las Vegas slot machine.<sup>1</sup>

---

<sup>1</sup> All screen shots presented here are taken from the prototype version 0.8 of the PHASAR system.

medline contains 16819607 documents.

Subject	Verb	Object	Complement
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Search"/>	<input type="button" value="Clear"/>	Selection:	<input type="text" value="Thesaurus"/>

No documents found.

The query can be a word, a collocation, a phrase or a sentence, that has to be distributed appropriately over the four boxes. Any box can be left blank, and at most one box (under certain restrictions) may contain a question mark (the *joker*, see later).

The query in the fruit machine can be modified by modifying its elements. Furthermore, there are explicit mechanisms to generalize or specialize the query (next sections). The user can return to a previous query by using the backward and forward buttons on the browser.

### 1.6 Generalizing the query

Although phrases are highly precise indexing terms, they also have a very low probability of occurrence, and therefore a low recall – the problem of *sparseness*. In order to increase recall when the query is too restrictive, the query can be generalized by replacing some of its elements by a more general one:

- a list of terms joined by |
- a semantic type (SEM-type) taken from one of the thesauri (see 1.9), standing for all terms belonging to that type.

In this way, queries may be generalized from individual phrases to phrase patterns, which may match many different phrases in the documents. Furthermore, queries can be combined disjunctively in a profile (see 1.10) to raise recall, or conjunctively to raise precision.

### 1.7 Specializing the query

In order to raise precision when there are too many irrelevant hits, the query can be specialized by specializing some of its elements. A phrase can be made as precise as you wish. Phrases consist of words, joined together in linguistically meaningful ways to achieve more precision. Each word in a phrase helps to disambiguate the others.

- adding another element, e.g. a preposition phrase as complement, or
- specializing an element, e.g. by adding a modifier to a noun (an adjective or another noun), and adverb to a verb.

This specialization process is supported by information from the index.

## 1.8 Support from the index

The searcher has to know what is actually said in the documents, and how often, in order to get nearer to his answer. It is inefficient to get this information by browsing the documents, which takes too much time and effort. Therefore this information must be presented at a higher level of aggregation. Quantitative information about the documents is gathered by indexing after parsing, producing an index of words (lemmatized word forms) and DT's giving for each element the dependency relations in which it occurs in the text. As an example, the DT index of Medline for 'aspirin' starts with

triples with "aspirin" as head		triples with "aspirin" as modifier	
829	[aspirin SUBJ induce]	2065	[effect OF aspirin]
778	[aspirin SUBJ inhibit]	951	[dose OF aspirin]
532	[aspirin SUBJ reduce]	916	[therapy ATTR aspirin]
522	[aspirin SUBJ have]	781	[administration OF aspirin]
327	[aspirin PRED effective]	679	[treatment ATTR aspirin]
327	[aspirin SUBJ prevent]	453	[combination OF aspirin]
232	[aspirin SUBJ cause]	424	[kg OF aspirin]
216	[aspirin SUBJ treat]	383	[ingestion OF aspirin]
197	[aspirin SUBJ decrease]	250	[trial OF aspirin]
187	[aspirin SUBJ increase]	234	[acetylsalicylic_acid PRED aspirin]
158	[aspirin SUBJ produce]	225	[administer OBJ aspirin]

## 1.9 Support from thesauri

An important mechanism for generalizing from single words to collections of words is furnished by domain-specific thesauri providing synonyms and hypernyms. We see such a thesaurus purely as a device for imposing a hierarchical *categorization* of semantical types (SEM-types) on the words occurring in the documents. Therefore we need shallow and broad hierarchies whose categories are of obvious importance to a certain domain, rather than deep and detailed semantical categorizations. Our use of thesauri is syntactical (what can be written?) rather than semantical (what does it mean?). Looking up the word 'aspirin' in a simplified UMLS thesaurus (reduced to four levels) provides the following information:

1. [109] aspirin [84], asa [25],
  - \* ORGANIC-CHEMICAL: [185638] go
    - o CHEMICALS-&-DRUGS: [766632] go
      - + ROOT: [2422116] go
2. [109] aspirin [84], asa [25],
  - \* PHARMACOLOGIC-SUBSTANCE: [249944] go
    - o CHEMICALS-&-DRUGS: [766632] go
      - + ROOT: [2422116] go

The numbers given in square brackets indicate how many Medline records contain a word with this SEM-type. On the basis of this combined qualitative and quantitative information, the searcher may choose to generalize the word ‘aspirin’ to one of the SEM-types ‘ORGANIC-CHEMICAL:’ or ‘PHARMACOLOGIC-SUBSTANCE:’; or he may click on ‘go’ to inspect the thesaurus entry for one of the hypernyms of ‘aspirin’.

The professional user knows the semantic ontology of his domain and need not be reminded of it, but he needs a way to specify easily whether the subject or object of his verb is a DRUG:, ILLNESS:, GENE:, PATHOGEN: or PERSON:. In this light, the reduced UMLS thesaurus is not particularly helpful.

A word or collocation may occur more than once in a thesaurus, or in multiple thesauri. We do not distinguish word senses, but we exploit the fact that the head of a triple tends to disambiguate the modifier and vice versa. We shall also need domain-dependent thesauri of verbs (CURE:, CAUSE:, etc.).

### 1.10 From queries to profiles

Experiences in Automatic Document Classification show that a profile, in the sense of a *weighted collection of terms* can give surprisingly high precision and recall. In fact, many successful classification techniques (for an overview see [Sebastiani, 2002]) result in a linear classifier, which is just such a weighted collection of terms. A profile of words can be learned automatically (provided enough training examples are available) but a profile of linguistically motivated search patterns can also be built by hand, (witness the work of Riloff, e.g. [Furnkranz et al, 1998, Riloff and Lorenzen, 1999]), which is the case in PHASAR.

As a side effect of the search process, we will allow the user to build a complex combination of search patterns, a *profile*, in which search patterns may be combined by *discourse operators*: not only logical operators like AND, OR and NOT but also textual operators like BEFORE and more subtle ones that will have to be the subject of further research. Furthermore we shall investigate how to use information from the browsing activities to assign weights to the patterns, so that we can impose a ranking on hits for browsing.

### 1.11 Browsing

Browsing long documents to inspect relevant phrases is tedious and time consuming. That is why some search machines supply each search result with one or two sentences extracted from the document, which helps greatly in determining its relevance. This problem also motivates much present research on summarization techniques and passage retrieval.

In the PHASAR browser, all terms from the query are weakly highlighted in the document, and sentences that match the complete query are strongly highlighted. Furthermore, the unit of viewing can be chosen by the user, being either the whole document or only the passages or sentences matching the query. Highlighting and condensed viewing significantly improve the efficiency of browsing.

## 2 An example search

PHASAR is intended for professional search in a wide sense. Since it differs greatly from traditional search engines, a whole new way-of-working will have to be found, which must still be shaped in trial applications, along with its support tools. In this section we illustrate a proposed way-of-working by means of an example which was contrived to explain the interface rather than its application.

Assume that some doctor or journalist wishes to investigate the knowledge expressed in Medline abstracts about cancer, its causes and effects. He can start the search by supplying either a (noun)phrase or a sentence. For the heck of it he types ‘aspirin causes cancer’ into the fruit machine and hits the search button.

medline contains 16819607 documents.

Subject	Verb	Object	Complement
aspirin	causes	cancer	

Search Clear Selection:  Thesaurus

[N:aspirin]	[V:cause]	[N:cancer]	[N:aspirin,V:cause]	[V:cause,N:cancer]
20452	755656	441958	232	1365

No documents found.

Of course there are zero hits. Nobody in his right mind would write something like that in a Medline abstract. It is an example of a *hyper-precise query*. But it will serve to start the search.

The fruit machine is presented showing this sentence, with under it the elements of the query, each with its document frequency in the collection. Clicking on one of the elements under the fruit machine will cause the relevant part of the pair indices to be shown, as was done for ‘aspirin’ in 1.8.

The user replaces the word ‘aspirin’ by a more likely choice, namely ‘solvents’.

medline contains 16819607 documents.

Subject	Verb	Object	Complement
solvents	cause	cancer	

Search Clear Selection:  Thesaurus

[N:solvent]	[V:cause]	[N:cancer]	[N:solvent,V:cause]	[V:cause,N:cancer]
26675	755656	441958	138	1365

No documents found.

Again no hits are obtained, although the elements of the query all appear in the Medline collection. He now realizes that he is not so much interested in general remarks containing the word ‘solvent’ but in the names of all solvents that are said in Medline to cause cancer. The term ‘solvent’ must be generalized to all terms in the semantic category of solvents, which unfortunately does not occur in the UMLS thesaurus.

Going on another tack, the searcher decides to find out what specific substances are said in Medline to cause cancer. He replaces the word ‘solvents’ in the subject field of the fruit machine by a *joker* (in the form of a question mark). Pressing the search-button gets him a list of the left-relations of the verb ‘cause’ occurring in the hits of the query ‘causes cancer’ (this sounds quite complicated but is actually quite natural: The query determines the subset of documents to

look in, and the joker in the subject box indicates that left-relations of the verb are wanted). Through another use of the joker it is possible to obtain the right-relations of the object 'cancer'. The lists are shown together in the following table:

78	[it SUBJ cause OBJ cancer]	104	[cause OBJ cancer ATTR cervical]
72	[mutation SUBJ cause OBJ cancer]	74	[cause OBJ cancer ATTR liver]
63	[exposure SUBJ cause OBJ cancer]	54	[cause OBJ cancer ATTR bladder]
45	[infection SUBJ cause OBJ cancer]	42	[cause OBJ cancer ATTR prostate]
34	[smoking SUBJ cause OBJ cancer]	30	[cause OBJ cancer ATTR colorectal]
32	[agent SUBJ cause OBJ cancer]	25	[cause OBJ cancer OF tract]
29	[factor SUBJ cause OBJ cancer]	24	[cause OBJ cancer ATTR certain]
26	[gene SUBJ cause OBJ cancer]	24	[cause OBJ cancer ATTR stomach]
26	[virus SUBJ cause OBJ cancer]	21	[cause OBJ cancer SUBJ occur]
24	[type SUBJ cause OBJ cancer]	18	[cause OBJ cancer ATTR cervix]
21	[hpv SUBJ cause OBJ cancer]	16	[cause OBJ cancer ATTR breast]
19	[they SUBJ cause OBJ cancer]	15	[cause OBJ cancer ATTR kidney]

In the left list, the personal pronouns in first and last line are place holders for anaphora, the others are reasonable (although not very novel) causes of cancer. Maybe lower down the list is more surprising. The right list mainly enumerates forms of cancer. The searcher can again replace the joker by a word (or more generally a list of terms) and maybe try another joker to find the most promising directions for further specialization.

Returning to the query 'aspirin causes cancer', the searcher would at this point like to generalize the verb form 'causes' to a wider class of verb forms 'CAUSES:' but we do not as yet dispose of a suitable thesaurus of verbs to make this generalization. That is future work.

The searcher modifies the query into 'cancer causes pain' and submits it. This time he gets some hits:

medline contains 16819607 documents.

Subject	Verb	Object	Complement
cancer	causes	pain	
<input type="text"/>			
<input type="button" value="Search"/> <input type="button" value="Clear"/> Selection: <input type="text"/> <input type="button" value="Thesaurus"/>			
[N:cancer]	[V:cause]	[N:pain]	[N:cancer.V:cause]
441958	755656	187382	928
			[V:cause.N:pain]
			3277

40 documents found.

[First document](#)
[Next document](#)
[Previous document](#)
[Random document](#)

Browsing through the documents found, he arrives at the following document, which is shown with deep highlighting for the sentence matched and, with less color difference, for the other matches with words from the query. Notice the effect of the passive-to-active normalization.

Abstract: Hundred patients with far-advanced **cancer** and **pain** were interviewed within a few days of admission to a special care unit. Eighty had more than one **pain**; 34 had four or more. A total of 303 anatomically distinct **pains** were recorded. Ninety-one patients had **pain** caused by the **cancer** itself. Twelve had treatment-related **pain**; and 19 had **pain** related to chronic disease or debility ('associated **pain**'). Thirty-nine patients had one or more **pains** unrelated to **cancer** or treatment; the most common of these was myofascial **pain**. In 41 patients only was all the **pain** caused directly by the **cancer**. Bone involvement and nerve compression were the most common forms of **cancer**-related **pain**; soft tissue and visceral **pains** also occurred frequently. Fifty-seven patients had **pain** for more than 4 months.

It is also possible to browse by passage and even by sentence, which makes good sense for larger documents than those in Medline.

Rather than starting with a hyper-precise query, the search will usually start with a single word, e.g. 'aspirin' or 'cancer', or (in other applications) with a proper name. The appropriate verbs may then be found by means of a joker.

The searcher continues to generalize or specialize the elements of the query until he is satisfied with the accuracy of the query pattern. He can combine this query pattern with others into a profile, and in various ways ply the tools of his trade.

### 3 Status of the project

In this section we briefly describe the application for which the system was implemented, the implementation tools used and the present status.

The mining of information about metabolites from the medical literature provides a difficult test for the PHASAR technology. It is definitely professional search, satisfying the criteria given in section 1.1. Metabolites are roughly those side- and end products of biochemical processes which are not considered important enough to write papers about them, but that may be mentioned in passing. Therefore it is hard to find them in the literature by traditional means.

In the course of an interactive analysis of selected documents mentioning certain processes, reagents or metabolites, a profile is generated, consisting of (possibly weighted) query patterns joined by discourse operators, which can then be used for an exhaustive automatic analysis of a large body of literature. The profiles can afterwards be re-used for other metabolites or even to find (as yet unknown) candidate metabolite names. For each metabolite, all passages referring to its source, effects and interactions can be retrieved.

As a first corpus for experiments with PHASAR, a snapshot of Medline containing all abstracts which were on-line as of September 25, 2005 has been parsed, transduced to dependency trees, unnested into dependency triples and lemmatized (more than 16 million documents, 2.4 Giga words). The resulting terms were indexed using the indexing system MRS [Hekkelman and Vriend, 2005] which is in the public domain. The whole process took 792 CPU-hours on the LISA system of SARA <sup>2</sup> in Amsterdam. The next step will be to analyze and index all openly accessible full-text papers and journals.

The parser used was generated using the AGFL-system [Koster and Verbruggen, 2002] from the EP4IR grammar and lexicon of English, which are freely available in the public domain <sup>3</sup>. The EP4IR grammar version 2.4 was used as provided, but

<sup>2</sup> [www.sara.nl](http://www.sara.nl)

<sup>3</sup> <http://www.cs.ru.nl/agfl>

in order to adapt the lexicon to Biomedical applications it was extended with material from the Unified Medical Language System (UMLS) lexicon <sup>4</sup>, which is also freely available.

The current version of the PHASAR system is a thin client/server system built using HTML, Javascript, Perl and C++. It is a prototype system available only to project members, intended for development of the interface, search tactics and thesaurus resources. It is running on one of the mirror servers of Medline.

## 4 Conclusion

The PHASAR system is a new kind of search engine, developed for supporting professional search processes in the BioMedical field. Although the present implementation of PHASAR is a prototype and comparative evaluation still has to start, the ideas behind PHASAR should be of interest to the whole Information Retrieval community. The system is now being benchmarked on tasks from BioInformatics and from other areas. We expect the system to perform more accurate searching than conventional search engines, at least in the hands of professionals, because of its use of phrases and linguistic techniques. Replacing browsing over documents by passage browsing and browsing over the index makes it an efficient tool for *exploratory search* over very large document collections. Its ability to reduce documents to sentences matching a pattern makes it a suitable tool for full-text mining.

Due to its generic character, it is applicable other application areas, such as:

- professional search by patent clerks and documentalists
- text-based research by linguists and musicologists
- forensic search and chat monitoring, and
- opinion mining and media analysis on the Internet.

A lot of work will have to be done in our project to develop PHASAR into a mature tool for BioInformatics:

- improving the linguistic resources: increasing the coverage of the lexicon and the accuracy of the grammar, implementing anaphora resolution
- constructing a number of comprehensive flat thesauri for the application domain, largely by conversion of existing resources
- elaborating the user interface and the way-of-working on the basis of user experiences
- testing and evaluating the technology in various benchmarks, and improving it wherever possible.

The PHASAR system can search or mine enormous collections of documents. As section 3 shows, the technology exists for parsing very many documents (in English) at reasonable speed. Although for parsing and indexing the present

---

<sup>4</sup> <http://www.nlm.nih.gov/research/umls/>

Medline collection a few hundred computers were used in parallel for a few hours, a single computer can keep up with the weekly increments in Medline.

The Medline collection contains about 17M (short) documents, less than one promille of the size of the Internet measured in web-pages<sup>5</sup>. But the parsing and indexing technology we have used can certainly be scaled up by three orders of magnitude. It is therefore feasible to construct and maintain a syntactically analyzed and indexed version of all documents in English accessible on the Internet and search them with PHASAR.

## References

- [Arampatzis et al., 2000] A. Arampatzis, Th.P. van der Weide, C.H.A. Koster, P. van Bommel (2000), An Evaluation of Linguistically-motivated Indexing Schemes. *Proceedings of BCS-IRSG, 22nd Annual Colloquium on IR Research*, pg 34-45.
- [Bouma et al, 2005] G. Bouma, J. Mur, G. van Noord, L. van der Plas, and J. Tiedemann (2005), Question Answering for Dutch using Dependency Relations. *Proceedings of the CLEF 2005 Workshop*.
- [Bruza and Huibers, 1996] P. Bruza and T.W.C. Huibers, (1996), A Study of Aboutness in Information Retrieval. *Artificial Intelligence Review*, 10, p 1-27.
- [Cui et al, 2005] Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan and Tat-Seng Chua (2005), Question Answering Passage Retrieval Using Dependency Relations, In *Proceedings SIGIR 2005*.
- [Fagan, 1988] J.L. Fagan (1988), *Experiments in automatic phrase indexing for document retrieval: a comparison of syntactic and non-syntactic methods*, PhD Thesis, Cornell University.
- [Furnkranz et al, 1998] Furnkranz, J., Mitchell, T., and Riloff E., (1998), Case Study in Using Linguistic Phrases for Text Categorization on the WWW, *AAAI/ICML Workshop on Learning for Text Categorization*.
- [Grootjen and van der Weide, 2004] F. A. Grootjen and T. P. van der Weide (2004), Effectiveness of Index Expressions. *NLDB 2004, Springer LNCS 3136* pp. 171-181
- [Hekkelman and Vriend, 2005] M.L. Hekkelman and G. Vriend (2005), MRS: A fast and compact retrieval system for biological data. *Nucleic Acids Res.* 2005 July 1; 33(Web Server issue): W766W769.// Also <http://mrs.cmbi.ru.nl/>.
- [Koster and Verbruggen, 2002] C.H.A. Koster and E. Verbruggen (2002), The AGFL Grammar Work Lab, *Proceedings FREENIX/Usenix 2002*, pp 13-18.
- [Melčuk, 1988] I. A. Melčuk. *Dependency Syntax: Theory and Practice*. State University of New York Press, Albany, NY, 1988.
- [Riloff and Lorenzen, 1999] Riloff, E. and Lorenzen, J., (1999), Extraction-based Text Categorization: Generating Domain-specific Role Relationships Automatically, In [Strzalkowski, 1999].
- [Sebastiani, 2002] F. Sebastiani (2002), Machine learning in automated text categorization, *ACM Computing Surveys*, Vol 34 no 1, 2002, pp. 1-47.
- [Sparck Jones, 1999] K. Sparck Jones (1999), The role of NLP in Text Retrieval. In: [Strzalkowski, 1999] pp. 1-24.
- [Strzalkowski, 1995] T. Strzalkowski (1995), Natural Language Information Retrieval, *Information Processing and Management*, 31 (3), pp. 397-417.
- [Strzalkowski, 1999] T. Strzalkowski, editor (1999), *Natural Language Information Retrieval*, Kluwer Academic Publishers, ISBN 0-7923-5685-3.

---

<sup>5</sup> according to <http://www.metamend.com/internet-growth.html>