

Head/Modifier Frames for Information Retrieval

C.H.A. Koster

Computing Science Institute,
University of Nijmegen,
The Netherlands,
E-mail: kees@cs.kun.nl

Abstract. We describe a principled method for representing documents by phrases abstracted into Head/Modifier pairs. First the notion of aboutness and the characterization of full-text documents by HM pairs is discussed. Based on linguistic arguments, a taxonomy of HM pairs is derived. We briefly describe the EP4IR parser/transducer of English and present some statistics of the distribution of HM pairs in newspaper text. Based on the HM pairs generated, a new technique to measure the accuracy of a parser is introduced, and applied to the EP4IR grammar of English. Finally we discuss the merits of HM pairs and HM trees as a document representation.

1 Introduction

The Information Retrieval community has for a long time held high hopes concerning the value of linguistic techniques. However, the improvements in precision and/or recall expected from the use of phrases in retrieval and in text categorization have repeatedly been found disappointing [22].

Although the use of simple noun phrases as indexing terms is now commonly accepted, practical Information Retrieval systems using phrases like the CLARIT system [7] do not appear to perform consistently better than those based on keywords. There is a growing conviction that the value of Natural Language Processing to IR is dubious, even among people who tried hard to make linguistically-based IR work [15, 20]. The predominant feeling, as voiced in [18], is that only ‘shallow’ linguistic techniques like the use of stop lists and lemmatization are of any use to IR, the rest is a question of using the right statistical techniques.

In spite of these negative experiences, we are trying to improve the accuracy of automatic document classification techniques by using (abstractions of) phrases as terms. In this paper we shall first discuss the notion of *aboutness*, which plays a central role in Information Retrieval. We then introduce Head/Modifier (HM) pairs as an abstraction of phrases preserving their aboutness, and give a taxonomy of HM pairs based on the intra-sentence relations they represent. We describe the EP4IR grammar, in which the transduction of English text to HM pairs is realized, and which is now available in the public domain.

We introduce a new technique to measure the accuracy of a parser/transducer, based on the HM pairs generated, and apply it to the EP4IR grammar. We give some experimental results about the distribution of HM pairs, and report our experiences in using HM pairs as indexing terms in Text Categorization. Finally we discuss the strengths and limitations of the HM pair representation.

2 Aboutness

The notion of *aboutness* is highly central to Information Retrieval: the user of a retrieval system expects the system, in response to a query, to supply a list of documents which are *about* that query. Practical retrieval systems using words as terms are based on an extremely simpleminded notion of aboutness:

If the word x occurs in the document then the document is *about* x .

This notion can be refined by introducing a measure for the *similarity* between the query and the document. For phrases, aboutness can be defined in the same way:

If the phrase x occurs in the document then the document is *about* x .

Although intuitively it seems likely that phrases provide a more informative document representation than keywords, the above formulation is not helpful in deciding what phrases to choose, which parts to eliminate and how to represent them. Certainly, taking literal phrases as terms may lead to very low Recall, because of the human preference for morphological, syntactical and semantical variation in formulating texts [1]. Furthermore, it considers a phrase as a monolithic term, disregarding the elements out of which it is composed. A model-theoretic basis for the notion of aboutness was described in [2]:

An information carrier i will be said to be *about* information carrier j if the information borne by j holds in i

The rather abstract sounding notion of “information carrier” can denote a single term, but also a composition of terms into a structured query or a document.

In other retrieval models (Boolean, vector space, logic-based or probabilistic) the notion of aboutness can be defined analogously (see [3]). *The problem with all these definitions is that they are not concrete enough to use them in reasoning about document representations.*

Our treatment of phrases as indexing terms is based on the following premises:

- The representation of a phrase as an indexing term is composed of words extracted from the phrase occurring in a linguistically meaningful relation
- words that have no classificatory value as keywords (by themselves) can be omitted.

These thoughts are elaborated further in the following sections.

3 Linguistic Phrases as indexing terms

The use of Linguistically Motivated Indexing terms (some abstraction of linguistically derived phrases) has always fascinated researchers in IR (for an overview see [19]). Even our particular choice of abstraction, using HM pairs as indexing terms is not new: it has been made previously by many researchers like [8, 21] and recently [9]. In particular the Noun Phrase enjoys popularity as an indexing term, because on the one hand it obviously carries a lot of information, and on the other hand it is relatively easy to extract.

3.1 Noun Phrases as indexing terms

According to [23], a Noun Phrase is to be considered as a *reference to, or description of a complicated concept*. It is interesting to note that the preferred form of informative titles of articles in the exact sciences appears to be a (complicated) noun phrase. As a query, a noun phrase is definitely more precise than the bag of its constituent words.

The use of simple NP's as indexing terms is now common practice in many Information Retrieval systems. NP's can be extracted from a text using a chunker or shallow parser (e.g. [6, 10]) which is easier to construct than a full-blown grammar-based parser.

3.2 Verb Phrases as indexing terms

The verb phrase comprizes a verb group together with its complements. Semantically, a Verb Phrase can be seen as the *description of a fact, event or process*. It describes something dynamic, in contrast to the noun phrase which describes something static. For the aboutness of the phrase, only the main verb is of importance, because the auxiliaries serve to indicate time, modality, emotion. They will be elided during the transduction. The relations between the main verb and its complements however, including the subject, are essential for the aboutness of the phrase.

3.3 Phrase normalization

Phrases used as terms are very precise, much more precise than single words. The reason why it is hard to gain by using them, is the very fact that they are so precise: the probability for a specific phrase to (re)occur in a document is much smaller than for a specific word. Thus, the term space for phrases is much more sparse than that of words. Using phrases instead of keywords, we may gain Precision but we loose Recall.

In writing a text, people prefer to avoid literal repetition, they will go out of their way to choose another word, another turn of phrase, using anaphora, synonymy, hypernymy, periphrasis, metaphor and hyperbole in order to avoid the literal repetition of something they have already said or written before. From

a literary standpoint this is wonderful, but it gives complications in IR: we have to compensate for this human penchant for variety. Essentially, we must try to conflate all semantically equivalent forms of a phrase, for instance by mapping them onto one same form.

Rather than using literal phrases taken from the text as terms, we shall therefore reduce them to a normal form which expresses only the bare bones of their aboutness. We must eliminate all dispensable ornaments and undo all morphological, syntactic and semantic variation we can. In this way, we strive to regain Recall while surrendering little Precision.

4 HM trees and HM pairs

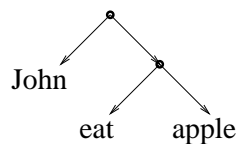
In the context of document classification, we shall represent each document by a bag of terms, where the terms are *Head/Modifier pairs*, derived from the phrases in the document by a *transduction* process. Each phrase is first transduced to a HM tree and then *unnested* to one or more HM pairs.

4.1 HM trees

A Head/Modifier tree or HM tree is a form of (binary) dependency tree, denoted as a recursive structure over pairs of the form

[head, modifier]

where both the head and the modifier consist of a sequence of zero or more words and (nested) HM trees. As an example, the HM tree



is denoted by [John, [eat, apple]].

4.2 HM pairs

A pair [head, modifier] corresponds to a flat HM tree in which both the head and the modifier are not nested, i.e. consist only of zero or more words. A HM tree will be unnested to the set of HM pairs which are contained in it and we shall use a bag-of-HM pairs representation for documents.

The intuition behind using HM pairs as terms is that the modifier is joined to the head in order to make it more precise, i.e. to distinguish the pair from another pair with the same head, in particular to distinguish between different meanings of a polysemous head. Thus, there are many forms of engineering, but we may focus on [engineering, software].

It may be asked why we do not simply use *collocations* as terms, frequent multi-word combinations, such as *software engineering*. Apart from the necessity for conflation of equivalent terms, for which HM pairs give more scope, it should be pointed out that there is more information in the above HM pair than in the collocation: in expressing the relevance of a term to a class c we can use $P(c|head, modifier)$ in both cases, but $P(c|head, \overline{modifier})$ only for the HM pair – in this case the information that the document is about *another* form of engineering.

4.3 Unnesting

As produced by the transduction, trees may be nested, i.e. contain embedded trees.

```
[[mower, lawn], large]
 [operation, [consuming, time]]
```

We want to use pairs without nesting as terms, possibly in combination with single words. A nested tree F may be unnested into a set of pairs S by repeatedly taking some embedded tree without nesting from F while replacing it by its head, and adding the corresponding pair to the set S , until F is empty. Some (artificial) examples:

```
[a, b] ==> [a, b]
[a, [b, c]] ==> [b, c][a, b]
[[a, b], c] ==> [a, b][a, c]
```

In the unnesting process, the head of a tree serves as an abstraction for that tree: a lawn mower is a (some kind of) mower, a large mower is a mower, etc.

```
[[mower, lawn], large] ==> [mower, lawn][mower, large]
```

```
[operation, [consuming, time]] ==> [operation, consuming][consuming, time]
```

We also allow the |-sign as an or-operator:

```
[a, b | c ] ==> [a, b][a, c]
[a | b, c ] ==> [a, c][b, c]
```

A tree with an empty head, typically occurring nested in another one, obtains an empty abstraction, e.g.

```
[I, [see, man [, [you, [give, book|to man]]]]]
 ==> [I, see][see, man][you, give][give, book][give, to man]
```

is transduced from the sentence I saw the man to whom you gave a book.

5 A Taxonomy of HM pairs

Head/Modifier pairs should not be arbitrary combinations of words from the text, but they should represent some *linguistically meaningful relation* between components (words and collocations) extracted from the text. This is the basis for the following taxonomy.

5.1 Word types

It is to be expected that the words (and collocations) occurring in the HM pairs should include the words that as keywords serve best to characterize the document; the pairs only add precision. Traditionally, nouns are considered the most important keywords, and function words are eliminated by the stop list. In [1] the relative contribution of words with different parts of speech to the accuracy in automatic document classification was investigated. It was found that eliminating all terms except nouns, adjectives and verbs gives no loss (sometimes a small gain) in classification accuracy. These words appear to carry all of the aboutness to be found in single keywords.

Our HM pairs will therefore be composed primarily out of nouns, adjectives and verbs. Furthermore we shall include prepositions forming part of a preposition phrase (PP). Lastly, we include the pronouns, not because they are by themselves very informative, but because they often appear as place holders. All other words like adverbs, auxiliaries, quantifiers, determiners etc will be elided.

5.2 Word relations

The set of all possible pairwise combinations of the four word types can be analysed as follows (see Fig. 1):

head	modifier				
	V	N	P	A	PP
V	–	object relation		–	yes
N, P	subject relation	predicative/attributive relation			yes
A	–	–	–	–	yes

Fig. 1. Relations realized by HM pairs

1. verb in modifier position

A pair of the form [N:, V:] or [P:, V:] represents the *subject relation*. The type [A:, V:] cannot occur, since an adjective in that position would be promoted to a noun ([N:A:, V:]).
2. verb in head position

a pair of the form [V:, N:] or [V:, P:] where the verb should be transitive represents the *object relation*. The type [V:, A:] will not be generated: In a sentence like his nose turned red the verb does not provide any additional information beyond [N:nose, A:red].
3. no verb
 - (a) noun as head

A noun can be modified by another noun [N:, N:], an adjective [N:, A:] or a (possessive) pronoun [N:, P:], all expressing the *attributive relation* (software engineering, the red car, my car). The same pairs arise

from the *predicative* relation (the car is a lemon, the car is red, the car is mine).

(b) adjective as head

Will not occur, since an adjective can only be modified by an adverb, which is elided.

(c) a pronoun in head position is treated like a noun.

4. verbs in both positions

Although a sentence like I like to walk might suggest a pair [V:like,V:walk], this sentence has the same aboutness as I walk. Similarly, he decided to leave the country can be argued to have the same aboutness as he left the country. But this is a moot point, since we have no good notion of aboutness.

To complicate matters, we have also to represent indirect objects, preposition complements and other adjuncts. For that purpose, we allow a verb, noun or adjective to have a modifier consisting of a preposition followed by a noun or pronoun which stands as the abstraction of an NP, for example

I give you a knife [P:I,[V:give,knife|to P:you]]

a cry for freedom [N:cry,for N:freedom]

open to suspicion [A:open,to N:suspicion]

Our HM trees include the *index expressions* of [4] as a subset, but they are richer because we also express the main verbs.

Other relationships (negation, quantification, auxiliary verbs, adverbial modifiers) are not expressed in the transduction, even though they are recognized by the grammar, because there is some evidence that they are not important for the intended application in text categorization [1].

Notice that the *order* of the constituents in a pair is important for all the relations described above. We shall exploit the *polarity* in the pairs to distinguish a pair [a, b] from a pair [b, a]. Thus, we shall not consider [air, pollution] as equivalent to [pollution, air].

Expressed in the framework of HM trees, the structure of a simple sentence will be:

[subject, [verb, object|other complement]]

where some of the elements may be missing. By unnesting, this structure yields the atomic HM pairs

[subject, verb], [verb, object] and [verb, other complement]

5.3 Morphological normalization

All words or collocations occurring in the pairs will be morphologically normalized by lemmatization. The goal is to map all different forms of the same lemma onto one representative form. The lemmatization process is aided by the word type supplied in the transduction.

As an example, the pair [N:man, A:V:sneezing] may be obtained from both the attributive the sneezing man and the predicative the man is sneezing. It will be normalized to [man, sneeze]. The nested tree [N:man, [V:sneezed,

]] obtained from this man sneezed has the same result, just like all its variants in time and modality.

All pronouns are mapped onto *it*, which can be seen as a place holder element, in particular for anaphora resolution (not yet implemented).

5.4 Syntactic normalization

As indicated above, all elements which are deemed not to contribute to the aboutness will be elided during transduction: articles, quantifiers, adverbs, connectors - which is much like applying a stop list. We shall have to determine experimentally which elements may be elided and what information should be expressed in the pairs (e.g, time and modality). In our classification context, it is quite feasible to investigate for any specific feature whether its inclusion or exclusion would measurably influence the classification result. At a later time we will also investigate the use of HM trees rather than HM pairs, by dispensing with the unnesting.

For each construct described by the grammar, its transduction is described wholly by the grammar (as part of the syntax rule for the construct). Thus, the transductions of complicated constructs are expressed compositionally in terms of those of their components. In the process, elements are elided or re-ordered and additional symbols injected (like the [, , and]) in order to express uniformly the four relations described above.

The syntactic normalizations implemented in this way include *de-passivation*:
the train was driven by a clockwork engine
is, by unnesting and morphological normalization, turned into
[N:engine,N:clockwork] [N:engine,V:drive] [V:drive,N:train]

6 Extracting HM pairs

In this section discuss the resources presently available for obtaining HM pairs from English text. We briefly describe the EP4IR grammar, introduce a technique for measuring the accuracy of a grammar in terms of the HM pairs produced, and apply it to the EP4IR grammar. We give some experimental results concerning the distribution of HM phrases in full text, and discuss the limitations of the HM pair approach.

6.1 The EP4IR grammar

The ‘English Phrases for IR’ (EP4IR) grammar of English was developed for investigations into the effective use of phrasal document representations in Information Retrieval applications like document classification, filtering and routing. It is available in the public domain, together with its lexicon and the AGFL parser generator system [13].

The grammar is written in the AGFL formalism for the syntactic description of natural languages. An Affix Grammar over a Finite Lattice (AGFL) can

be seen as a CF grammar extended with set-valued features (called affixes or attributes), where the features express finite syntactic and semantic categories like number, person, time, subcategorization, etc.

Since the grammar is not the subject of this paper, we will only sketch its main properties. It gives a robust description of the structure of the Noun Phrase and the Verb Phrase, including their transduction to Head/Modifier trees. It has an extensive lexicon (309007 entries including collocations) and uses various techniques to resolve ambiguity (subcategorization, penalties).

6.2 Some statistics

In order to get an impression of the distribution of the various HM pair types, we have parsed the EPO1A corpus [14, 12], using a parser/transducer generated from the EP4IR grammar and lexicon. The corpus contains 16000 abstracts of patent applications from the European Patent Office (2 Million words, totalling 12.4 Mbytes). This corpus yielded 727363 HM pairs (excluding adverbs, quantities etc).

Fig. 2 shows the relative distribution of the various types of HM pairs in the EPO1A corpus, which should be roughly similar for other corpora. As a rule of thumb, one HM pair is produced for every 2-3 words.

head	modifier				
	V	N	P	A	PP
V	12998	112557	2098	10263	58325
N	91648	150332	4000	152995	95484
P	30627	718	18	1735	1163
A					2368

Fig. 2. HM pairs in EPO1A

6.3 Measuring grammar accuracy

In developing a grammar, an objective measure is needed for the accuracy of the grammar, in order to do regression testing after each major modification to the grammar, and to assess the suitability of the parsers resulting from it for their intended purpose.

We can not use the ubiquitous Bracket Crossing (BC) measure [17]. To begin with, it has a large number of weakness and shortcomings [5]: it is not suited for partial parses, only useful for constituency based parsers, not fine grained enough for some specific syntactic phenomena, and there is no clear agreement on the granularity of bracketing. According to [16], in BC a mis-attachment can be punished more than once, so that a shallow parse with less syntactic information scores better than a “richer” analysis. Furthermore, the bracket-crossing approach needs an extensive syntactically analyzed corpus, which causes

a chicken-and-egg problem, and it is very much oriented to parse trees, whereas we are interested in HM pairs.

The HM pairs generated from a text represent precisely the relations that we are interested in, and closely reflect the dependency relations in the text. If we can extract the right HM pairs from a sentence, we can also derive the complete sentence structure. Therefore we propose to express the accuracy of the grammar/transducer in terms of the HM pairs produced by it when applied to a reference corpus for which the HM pairs are known.

This HMP-annotated reference corpus is based on a collection of sentences which is representative for the intended application domain and for the syntactic constructions occurring in it. This collection need not be very big (in comparison with a modern treebank) but it must be expected to generate enough HM pairs to allow a measurement at the required granularity - say two thousand HM pairs if we want 3 decimals of accuracy, just a few hundred sentences.

The manual annotation of the reference corpus with the correct HM pairs is tedious and errorprone, but it can be performed in interaction with the parser to be tested, presenting the sentences in larger or smaller fragments and verifying the results by inspection. By skillfully exploiting the compositional character of grammar, this can be done in an efficient and reliable way.

Two persons may mark the same corpus and discuss the points of difference in order to guarantee the correctness of the reference corpus. The measurement procedure is as follows:

1. generate a parser/transducer from the grammar and its lexicon
2. collect the reference corpus
3. manually derive the HM pairs to be generated from it
4. let the parser generate HM pairs from it
5. compare these, computing precision and recall in the usual way.

6.4 The accuracy of EP4IR

In order to get an indication of the accuracy (Precision and Recall) of EP4IR, we have constructed a small reference corpus, consisting of 26 sentences from the OHSUMED collection and 113 sentences from the EPO1A corpus, totalling 3458 words. By semi-automatic analysis, 1529 HM pairs were found in the test set.

We also analyzed the same test set automatically. As can be seen from Fig. 3, the overall Precision and Recall were 66.6% and 64.5%, giving an F1-value of 0.65. The traditional NN and NA combinations have the highest accuracy. The letter Z in this table stands for a PP.

The breakdown of Precision and Recall per type of HM pair allows us to focus on areas of improvement. There are some dubious combinations, like VV (also in the test set). PP attachment and the assignation of subject and object must be improved. The EP4IR grammar is not yet very accurate and there are still some inconsistencies in the transduction. The grammar should be made probabilistic, so that it becomes better at finding the most likely analysis. Furthermore, it is

HM pair	present	found	correct	recall	precision
AZ	2	3	0	0.000	0.000
NA	293	291	219	0.747	0.753
NN	379	376	254	0.670	0.676
NP	15	18	15	1.000	0.833
NV	190	174	107	0.563	0.615
NZ	147	174	83	0.565	0.477
PA	4	2	2	0.500	1.000
PN	1	4	0	0.000	0.000
PV	88	70	58	0.659	0.829
PZ	1	0	0	0.000	0.000
VA	8	15	1	0.125	0.067
VN	262	222	154	0.588	0.694
VP	4	5	2	0.500	0.400
VV	0	19	0	0.000	0.000
VZ	135	105	60	0.444	0.571
total	1529	1478	986	0.645	0.666

Fig. 3. HM pairs in mixed test set

weak in its treatment of coordination and some special constructs. But in the mean time it is available to the IR community.

6.5 Limitations of HM pairs

The choice of HM pairs as a realizations of phrases has its limitations from a linguistic point of view:

- the many pairs involving a personal pronoun (PV and VP) show the need for anaphora resolution
- it is hard to normalize pairs representing periphrastic constructions (make a comparison between ... and ... for compare ... with ..., a number of cars vs many cars), especially when they border on idiom (a gaggle of geese)
- specialized lexical resources are needed for e.g. HM pair synonymy
- ternary (and higher) collocations are not expressible as a pair (software engineering conference; transverse collating bin)
- similar problem with composed words, especially in languages like German and Dutch (but also in English, e.g. well known, well-known and wellknown).

In fact, the problems with collocations and composed words can be solved by dealing with HM trees of order higher than one without unnesting. But there is at present no theory of language modeling based on pairs, let alone based on trees. Both practically and theoretically, much remains to be done.

7 Conclusion

We have introduced HM trees and their unnesting to HM pairs, describing their use for document representation. We have introduced a linguistically motivated

taxonomy of HM pairs allowing to capture the aboutness (as opposed to the constituency structure) of both the verb phrase and the noun phrase while rigorously eliminating non-informative elements.

We have introduced an accuracy measure for grammars transducing to HM pairs and used it to measure the Precision and Recall achieved by the EP4IR parser/transducer on a small test corpus. There is obviously room for improvement of the grammar, and we are working on it.

For experimental results using HM pairs in Text Categorization, the reader is referred to [12], which is concerned with the classification of the EPO1A corpus of patent abstracts. The results are (still) disappointing but it is argued that HM pairs may be better suited for query-based retrieval and Question Answering than for categorization, due to the strongly statistical character of the latter.

Further research on HM pairs is needed:

- to determine experimentally which elements of a sentence contribute most to the aboutness of a document
- to investigate theoretically how to make optimal use of structured terms such as HM pairs in text categorization
- to investigate anaphora resolution for HM pairs, selective clustering of terms and fuzzy semantic matching
- to investigate the possibility to dispense with unnesting, using arbitrarily complicated HM trees as terms.

Many improvements in theory, techniques and resources are still needed to reach a situation where phrases make an important improvement to Information Retrieval.

8 Acknowledgements

My sincere thanks go out to all participants in the DORO and PEKING projects, and in particular to T. Verhoeven who elaborated the HM pair-based grammar evaluation technique.

References

1. A. Arampatzis, Th.P. van der Weide, C.H.A. Koster, P. van Bommel (2000), An Evaluation of Linguistically-motivated Indexing Schemes. Proceedings BCS-IRSG 2000 Colloquium on IR Research, Cambridge, England.
2. P. Bruza and T.W.C. Huibers, (1994), Investigating Aboutness Axioms using Information Fields. Proceedings SIGIR 94, pp. 112-121.
3. P. Bruza and T.W.C. Huibers, (1996), A Study of Aboutness in Information Retrieval. *Artificial Intelligence Review*, 10, p 1-27.
4. Peter Bruza and Theo P. van der Weide (1991), The Modelling and Retrieval of Documents Using Index Expressions, SIGIR Forum vol 25 no 2, pp. 91-103.
5. J. Carroll, M. Guido and E. Briscoe (1999) Corpus Annotation for Parser Evaluation. *Proceedings of the EACL workshop on Linguistically Interpreted Corpora (LINC)*, 1999.

6. W. Daelemans, S. Buchholz and J. Veenstra (1999), Memory-based shallow parsing, proceedings CoNLL, Bergen, Norway.
7. D.A. Evans, R.G. Lefferts, G. Grefenstette, S.H. Handerson, W.R. Hersch and A.A. Archbold (1993), CLARIT TREC design, experiments and results. TREC-1 proceedings, pp. 251-286.
8. J.L. Fagan (1988), *Experiments in automatic phrase indexing for document retrieval: a comparison of syntactic and non-syntactic methods*, PhD Thesis, Cornell University.
9. A. Gelbukh, G. Sidorov, S.-Y. Han, E. Hernández-Rubio (2004), Automatic Syntactic Analysis for Detection of Word Combinations. In: A. Gelbukh (Ed.) Computational Linguistics and Intelligent Text Processing (CICLing-2004). Springer LNCS 2945 p. 243-247.
10. G. Grefenstette (1996), Light parsing as finite state filtering. Workshop on Extended finite state models of language, Budapest, ECAI'96.
11. C.H.A. Koster, Affix Grammars for Natural Languages. In: H. Alblas and B. Melichar (Eds.), *Attribute Grammars, applications and systems*. SLNCS 545, Heidelberg, 1991, p. 469-484.
12. C.H.A. Koster and M. Seutter (2002), Taming Wild Phrases, Proceedings 25th European Conference on IR Research (ECIR 2003), Springer LNCS 2633, pp 161-176.
13. C.H.A. Koster and E. Verbruggen, The AGFL Grammar Work Lab, Proceedings of the FREENIX/Usenix conference 2002, pp 13-18.
14. M. Krier and F. Zaccà (2002), Automatic Categorisation Applications at the European Patent Office, World Patent Information 24, pp. 187-196, Elsevier Science Ltd.
15. D. D. Lewis (1992), *Representation and Learning in Information Retrieval*. PhD thesis, Department of Computer Science; Univ. of Massachusetts; Amherst, MA 01003.
16. D. Lin (1995), A dependency-based method for evaluating broad-coverage parsers. *Proceedings IJCAI-95*, pp. 1420-1425.
17. M. Marcus, B. Santorini and M. Marcinkiewicz (1994), Building a Large Annotated Corpus of English: The Penn Treebank. In: *Computational Linguistics*, 19(2):313-330, 1994.
18. K. Sparck Jones (1998), Information retrieval: how far will *really* simple methods take you? In: Proceedings TWTL 14, Twente University, the Netherlands, pp. 71-78.
19. K. Sparck Jones (1999), The role of NLP in Text Retrieval. In: [22] pp. 1-24.
20. A.F. Smeaton (1997), Using NLP and NLP resources for Information Retrieval Tasks. In: T. Strzalkowski (Ed.), *Natural Language Information Retrieval*, Kluwer Academic Publishers.
21. T. Strzalkowski (1995), Natural Language Information Retrieval, *Information Processing and Management*, 31 (3), pp. 397-417.
22. T. Strzalkowski, editor (1999), *Natural Language Information Retrieval*, Kluwer Academic Publishers, ISBN 0-7923-5685-3.
23. T. Winograd (1983), *Language as a Cognitive Process: Volume I: Syntax*, Reading MA, Addison-Wesley, 650 pp.