Use of Dependency Tree Structures for the Microcontext Extraction

Martin HOLUB
Department of Software Engineering
MFF UK, Malostranské nám. 25
CZ-118 00 Praha, Czech Republic
holub@ksi.ms.mff.cuni.cz

Alena BÖHMOVÁ
Institute of Formal and Applied Linguistics
MFF UK, Malostranské nám. 25
CZ-118 00 Praha, Czech Republic
bohmova@ufal.ms.mff.cuni.cz

Abstract

In several recent years, natural language processing (NLP) has brought some very interesting and promising outcomes. In the field of information retrieval (IR), however, these significant advances have not been applied in an optimal way yet.

Author argues that traditional IR methods, i.e. methods based on dealing with individual terms without considering their relations, can be overcome using NLP procedures. The reason for this expectation is the fact that NLP methods are able to detect the relations among terms in sentences and that the information obtained can be stored and used for searching. Features of word senses and the significance of word contexts are analysed and possibility of searching based on word senses instead of mere words is examined.

The core part of the paper focuses on analysing Czech sentences and extracting the context relations among words from them. In order to make use of lemmatisation and morphological and syntactic tagging of Czech texts, author proposes a method for construction of dependency word microcontexts fully automatically extracted from texts, and several ways how to exploit the microcontexts for the sake of increasing retrieval performance.

1 Introduction

Empirical methods in natural language processing (NLP) employ learning techniques to automatically extract linguistic knowledge from natural language corpora; for an overview of this field see (Brill and Mooney 1997). This paper wants to show their usefulness in the field of information retrieval (IR). As the effects and the contribution of this discipline to IR has not been well examined and evaluated yet, various uses of NLP techniques in IR are only marginally mentioned in well known monographs published in last ten years, e.g. (Salton 1989), (Frakes and Baeza-Yates 1992), (Korfhage 1997).

A textual IR system stores a collection of documents and special data structures for effective searching. A textual document is a sequence of terms. When analysing the content of a document, terms are the basic processed units — usually they are words of natural language. When retrieving, the IR system returns documents presumed to be of interest to the user in response to a query. The user's query is a formal statement of user's information need. The documents that are interesting for the user (relative to the put query) are relevant; the others are non-relevant. The effectiveness of IR systems is usually measured in terms of *precision*, the percentage of retrieved documents that are relevant, and *recall*, the percentage of relevant documents that are retrieved.

The starting point of our consideration on IR was a critique of word-based retrieval techniques. Traditional IR systems treat the query as a pattern of words to be matched by documents. Unfortunately, the effectiveness of these word-matching systems is mostly poor

because the system retrieves only the documents that contain words that occur also in the query. However, in fact, the user does *not* look for the words used in the query. The user desires the *sense* of the words and wants to retrieve the documents containing words having the same sense. In contrast to the word-based approach, a sense-based IR system treats the query as a pattern of the required sense. In order to match this sense by the sense of words in documents, the senses of ambiguous words must be determined. Therefore a good word sense disambiguation is necessary in a sense-based IR system.

Ambiguity and synonymity of words is a property of natural language causing a very serious problem in IR. Both ambiguous words and synonyms depress the effectiveness of word-matching systems. The direct effect of polysemy on word-matching systems is to decrease precision (e.g., queries about financial banks retrieve documents about rivers). Synonymity decreases recall. If one sense is expressed by different synonyms in different documents, the word-matching system will retrieve all the documents only if all the synonyms are given in the query. Unfortunately, polysemy has another negative effect: polysemy also prevents the effective use of thesauri. Consequently, thesauri cannot be directly used to eliminate the problem with synonyms.

In our opinion, if a retrieval system is not able to identify homonyms and synonyms and to discriminate their senses, ambiguity and synonymity will remain one of the main factors causing 1) low recall, 2) low precision, and 3) the known and inevitable fact that recall and precision are inversely related. There are some evidences that lexical context analysis could be a good way how to eliminate or at least decrease these difficulties — see below.

How to take the step from words towards senses? Since an application of word contexts is the only possibility to estimate the sense of words, the way of dealing with word contexts is a central problem in sense-based retrieval. Knowing word contexts we can determine the *measure of collocating*, i.e. the extent to which a pair of words collocates. The knowledge of collocations can be used in IR for several purposes: making up contextual representations of words, resolving word ambiguity, estimating semantic word similarity, tuning the user's query in interaction with the user and quantifying the significance of words for retrieval according to entropy of their contexts.

Section 2 expresses our motivation: the investigation of word contexts helps us to develop an efficient IR system. Next section is devoted to analysing Czech texts and suggests a construction of dependency microcontext structures making use of the tree structure automatically created in the process of Prague Dependency Treebank annotation. Further part focuses on applications of contextual knowledge in IR and refers to the project working on an experimental IR textual database. Finally we summarise the results of this study.

2 Significance of word contexts

Word senses are *not* something given a priori. Humans create word senses in the process of thinking and using language. Thinking forms language and language influences thinking. It is impossible to separate them. Word senses are products of their interaction. In our opinion, the effort to represent word senses as fixed elements in a textual information system is a methodological mistake.

Many researchers consider the sense of a word as an average of its linguistic uses. Then, the investigation of sense distinctions is based on the knowledge of contexts in which a word appears in a text corpus. *Sense representations* are computed as groups of similar contexts. For instance, Schütze (1998) creates sense clusters from a corpus rather than relying on a pre-established sense list. He makes up the clusters as the sets of contextually similar occurrences of an ambiguous word. These clusters are then interpreted as senses.

According to how wide vicinity of the target word we include into the context we can speak about the *local* context and the *topical* context. The local or *"micro"* context is generally considered to be some small window of words surrounding a word occurrence in a text, from a few words of context to the entire sentence in which the target word appears. The topical context includes substantive words that co-occur

with a given word, usually within a window of several sentences. In contrast with the topical context, the microcontext may include information on word order, distance, grammatical inflections and syntactic structure.

In one study, Miller and Charles (1991) found evidence that human subjects determine the semantic similarity of words from the similarity of the contexts they are used in. They summarised this result in the so-called *strong contextual hypothesis*:

Two words are semantically similar to the extent that their contextual representations are similar.

The *contextual representation* of a word has been defined as a characterisation of the linguistic context in which a word appears. Leacock, Towell and Voorhees (1996) demonstrated that contextual representations consisting of both local and topical components are effective for resolving word senses and can be automatically extracted from sample texts. No doubt information from both microcontext and topical context contributes to sense selection, but the relative roles and importance of information from different contexts, and their interrelations, are not well understood yet.

Not only computers but even humans learn, realise, get to know and understand the meanings of words from the contexts in which they meet them. The investigation of word contexts is the most important, essential, unique and indispensable means of understanding the sense of words and texts.

3 Analysing Czech texts

Linguistic analysis of an input Czech text consists of a sequence of procedures depicted in Figure 1. The input is a Czech sentence and the results of the analysis are the two target structures: the *dependency microcontext structure* (DMCS) which we use for the microcontext extraction and the *tectogrammatical tree structure* (TGTS) which represents the underlying syntactic structure of the sentence. As the main intention of this paper is to describe the DMCS, building of the TGTS is distinguished by dashed line in Figure 1; we mention it here only for completeness and for comparison with the DMCS.

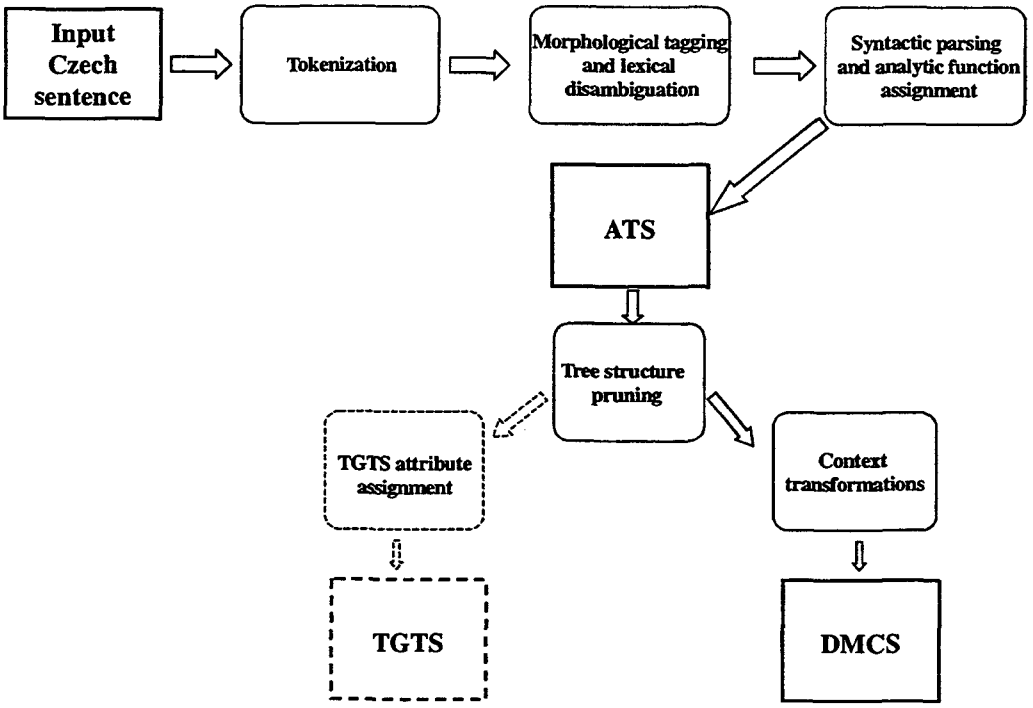


Figure 1: The sequence of procedures in the analysis of a Czech sentence.

Key algorithms used in the process of the analysis are based on empirical methods and on previous statistical processing of training data, i.e. natural language corpora providing statistically significant sample of correct decisions. Consequently, the ability of these procedures to provide a correct output has a stochastic character. These procedures were developed during the past years in the process of the Czech National Corpus and Prague Dependency Treebank creation. For a detailed descriptions see Hajič (1998), Hladká (2000) and Collins, Hajič, Ramshaw, Tillmann (1999).

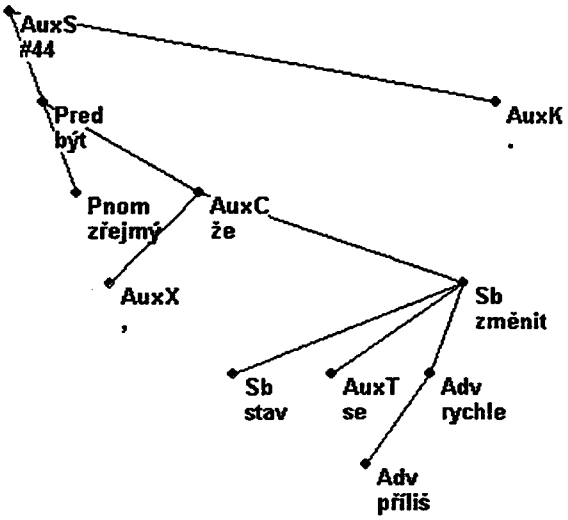
As shown in Figure 1, the first procedure is tokenization. The output of tokenization is the text divided into lexical atoms or tokens, i.e. words, numbers, punctuation marks and special graphical symbols. At the same time the boundaries of sentences and paragraphs are determined.

The following procedure, i.e. morphological tagging and lexical disambiguation, works in two stages. The first is the morphological analysis, which assigns each word its lemma, i.e. its basic word form, and its morphological tag. Since we often meet lexical ambiguity (i.e. it is not possible to determine the lemma and the tag uniquely without the knowledge of the word context), the morphological analyser often provides several alternatives. In the second stage, the result of the analysis is further used as an input for the lexical disambiguation assigning a given word form its unique lemma and morphological tag.

The next procedures work with syntactic tree structures. This process is described in the following subsection.

3.1 Syntactical analysis

The first step of the syntactic tagging consists in the building of the *analytic tree structure* (ATS) representing the surface syntactic dependency relations in the sentence. We use the statistical Collins's parser to create the structure of the tree and then a statistical procedure to assign words their syntactic functions. Two examples of the ATS are given in figures 2 and 3.

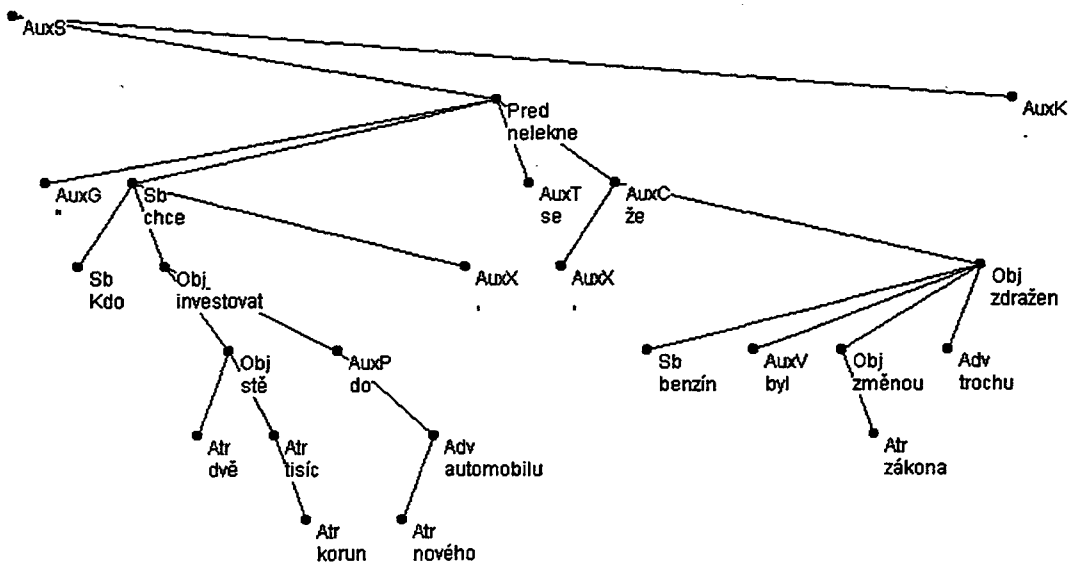


"Bylo zřejmé, že stav se příliš rychle nezmění."
(Lit.: It-was clear that the-state too fast will-not-change. E: It was clear, that the state will not change too fast.)

Figure 2: An example of an ATS.

The automatically created ATS is a labelled oriented acyclic graph with a single root (dependency tree). In the ATS every word form and punctuation mark is explicitly represented as a node of the tree. Each node of the tree is annotated by a set of attribute-value pairs. One of the attributes is the *analytic function* that expresses the syntactic function of the word. The number of nodes in the graph is equal to the number of word form tokens in the sentence plus that of punctuation signs and a symbol for the sentence as such (the root of the tree). The graph edges represent surface syntactic relations within the sentence as defined in Bémová et al (1997).

The created ATS is further transformed either to the TGTS or to the DMCS. In the Prague Dependency Treebank annotation, the transduction of the ATS to the TGTS is performed (see Böhmová and Hajičová 1999). For the sake of the construction of word contexts, we use the lemmas of word forms, their part of speech, their analytic function and we adapted the algorithms aiming towards the TGTS to build a similar structure, DMCS. Since, in comparison with the ATS, in both the TGTS and the DMCS only autosemantic words have nodes of their own, the first stage of this transformation (i.e. the pruning of the tree structure) is common.



"Kdo chce investovat dvě stě tisíc korun do nového automobilu, nelekne se, že benzín byl změnou zákona trochu zdražen."
(Lit.: Who wants to-invest two hundred thousand crowns in new car, he-does-not-get-frightened that petrol was by-change of-law a-little made-more-expensive. E: Those who want to invest two hundred thousand crowns in a new car, do not get frightened that petrol was made a little more expensive by the change of law.)

Figure 3: An example of an ATS.

3.2 From ATS towards DMCS

The transduction of the ATS to the DMCS consists of the four procedures:

1. Pruning of the tree structure, i.e. elimination of the auxiliary nodes and joining the complex word forms into one node.
2. Transformation of the structures of coordinations and appositions.
3. Transformation of the nominal predicates.
4. Transformation of the complements.

The first step of the transformation of the ATS to the respective DMCS is deletion of the auxiliary nodes. By the auxiliary nodes we understand nodes for prepositions, subordinate conjunctions, rhematizers (including negation) and punctuation. In case the deleted node is not a leaf of the tree, we reorganise the tree. For the IR purposes the auxiliary verbs do not carry any sense, so the analytical verb forms are treated as one single node with the lemma of the main verb. The purpose of the next three procedures is to obtain the context relations among words from the sentence, so we call them context transformations.

The constructions of coordination and apposition are represented by a special node (usually the node of the coordinating conjunction or other expression) that is the governor of the coordinated subtrees and their common complementation in the ATS. The heads of the coordinated subtrees are marked by a special feature. In case of coordinated attributes, the transformation algorithm deletes the special node, which means that a separate microcontext (X, Atr, Y) is extracted for each member of coordination. The same procedure is used for adverbials, objects and subjects. If two clauses occur coordinated, the special node remains in the structure, as the clauses are handled separately.

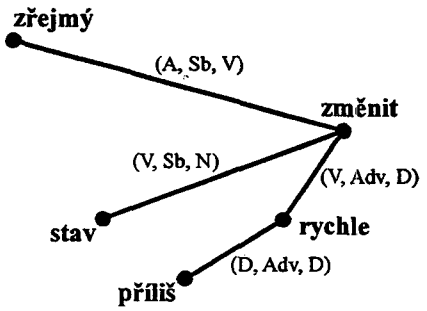


Figure 4: The DMCS of the sentence from Figure 2.

Probably the main difference from the syntactic analysis is the way we are dealing with the nominal predicate. We consider the nominal predicate to act as a normal predicate, though not expressed by a verb. This way of understanding a predicate is very close to predicate logic, where the sentence "The grass is green" is considered to express a formula such as "green(grass)".

In the ATS the complement (word syntactically depending both on the verb and the noun) is placed as a daughter node of the noun and marked by the analytical function of Atv. In the DMCS this node is copied and its analytical function is changed to Attr for the occurrence of the daughter of the noun and Adv for the new token of the daughter of the governing verb.

As we cannot go into details here, we illustrate the DMCS by two examples given in figures 4 and 5. The nodes of the trees represent semantically significant words. The edges of the graphs are labelled by so called dependency types (see below).

3.3 Extraction of microcontexts from the DMCS

There are 10 parts of speech in Czech and 18 types of analytic function in ATSs. However, we will consider only four parts of speech, namely nouns (N), adjectives (A), verbs (V) and adverbs (D), and four types of analytic function, namely subject (Sb), object (Obj), adverbial (Adv) and attribute (Attr), because only these are significant for the purpose of retrieval.

The construction of the dependency microcontext is based on the identification of *significant dependency relationships* (SDRs) in the sentence. An SDR consists of two words and a *dependency type*. An SDR is a triple [w_1, DT, w_2], where w_1 is a head word (lexical unit), DT is a dependency type and w_2 is a depending word (lexical unit). A dependency type is a triple (P_1, AF, P_2), where P_1 is the part of speech of the head word, AF is an analytic function and P_2 is the part of speech of the depending word.

For example, (A, Adv, D) is a dependency type expressing the relationship between words in expression "very large" where "very" is a depending adverb and "large" is a head adjective. [large, (A, Adv, D), very] is an example of an SDR.

Considering 4 significant parts of speech and 4 analytic functions, we have 64 (= 4×4×4) possible distinct dependency types. In Czech, however, only 28 of them really occur. Thus, we have 28 distinct dependency types shown in Table 1. Table 2 summarises the number of dependency types for each part of speech. The dependency types marked by an asterisk are not the usual syntactic relations in Czech, they were added on account of the transformation of the nominal predicate.

The number of SDRs extracted from one sentence is always only a little smaller than the number of significant, autosemantic words in the sentence, because almost all these words are depending on another word and make an SDR with it.

Now we define the *dependency word microcontext* (DMC). A DMC of a given word w is a list of its *microcontext elements* (MCEs). An MCE is a pair consisting of a word and a dependency type. If a word w occurs in a sentence and forms an SDR with another word w_1, i.e. if there is an SDR [w, DT, w_1] or [w_1, DT', w], then w_1 and the dependency type DT or DT', respectively, constitute a microcontext element [DT, w_1] or [w_1, DT'], respectively, of the word w. The first case implies that w is a head word in the SDR and in the second case the word w is a dependant.

Thus, each SDR [w_1, DT, w_2] in a text produces two MCEs: [w_1, DT] is an element of the context of w_2 and [DT, w_2] is an element of the context of w_1.

In the following Table 3 we exemplify the microcontexts extracted from the sentences used in the examples above.

Dependency types		
(N, Atr, N)	(V, Sb, N)	(V, Obj, N)
(N, Atr, A)	(V, Sb, V)	(V, Obj, V)
(N, Atr, V)	(V, Sb, A)	(A, Obj, A)
(N, Adv, N)*	(N, Sb, N)*	(D, Obj, N)
(N, Adv, V)*	(N, Sb, A)*	(A, Adv, A)
(N, Adv, D)*	(N, Sb, V)*	(A, Adv, D)
(V, Adv, N)	(A, Sb, N)*	(A, Adv, N)*
(V, Adv, V)	(A, Sb, A)*	(A, Adv, V)*
(V, Adv, D)	(A, Sb, V)*	(D, Adv, D)
		(D, Adv, N)

Table 1: Dependency types.

	Number of dependency types	
	as governing	as depending
N	9	10
A	8	6
V	8	8
D	3	4

Table 2: Number of dependency types for each part of speech.

4 Applications

4.1 Contextual knowledge in IR

As we have already mentioned, the knowledge of word contexts can be used for resolving word ambiguity. Word sense disambiguation is a central problem in NLP. Its task is to assign sense labels to occurrences of an ambiguous word. Researchers dealing with WSD methods often inspect also the way it affects retrieval performance if used in a retrieval model. Krovetz and Croft (1992) demonstrated that WSD can improve text retrieval performance. Later, Schütze and Pedersen (1995) found a noticeable improvement in precision using sense-based retrieval and word sense discrimination. Towell and Voorhees (1998) showed that, given accurate WSD, the lexical relations encoded in lexicons such as WordNet can be exploited to improve the effectiveness of IR systems.

Schütze (1998) introduced an interesting method: word sense discrimination. This technique is easier than full disambiguation since it only determines which occurrences of a given word have the same meaning and not *what* the meaning actually is. Moreover, while other disambiguation algorithms employ various sources of information, this method dispenses of an outside source of knowledge for defining senses. For many problems in information access, it is sufficient to solve the discrimination problem only. Schütze and Pedersen (1995) measured document-query similarity based on word senses rather on words and achieved a considerable improvement in ranking relevant documents. No references to externally defined senses are necessary for measurement of similarity.

4.2 Using microcontexts

In this subsection we give several ideas how to employ the microcontexts for improving the retrieval performance. Their significance and the extent of their usefulness is to be verified experimentally. For more details refer to Holub (2000).

In the literature, we can meet different definitions of collocation (cf. Ide and Véronis, 1998). Following Yarowsky (1993), who explicitly addresses the use of collocations in the WSD work, we adopt his definition, adapted to our purpose: A collocation is a co-occurrence of two words in a defined relation. Dependency microcontexts and collocations can be treated as mutually equivalent concepts in the sense that collocations can be derived from the knowledge of microcontexts and vice versa. In order to separate significant collocations from word pairs which occurred merely by a coincidence, we can compute the *measure of collocating* of a word and an MCE as the mutual information of the probability of their occurrence.

We also use the knowledge of collocations for computing the context similarity measure of two words. Assuming the "strong contextual hypothesis", the context similarity of words implies their semantic similarity, too. Consequently, we can estimate the semantic similarity of words.

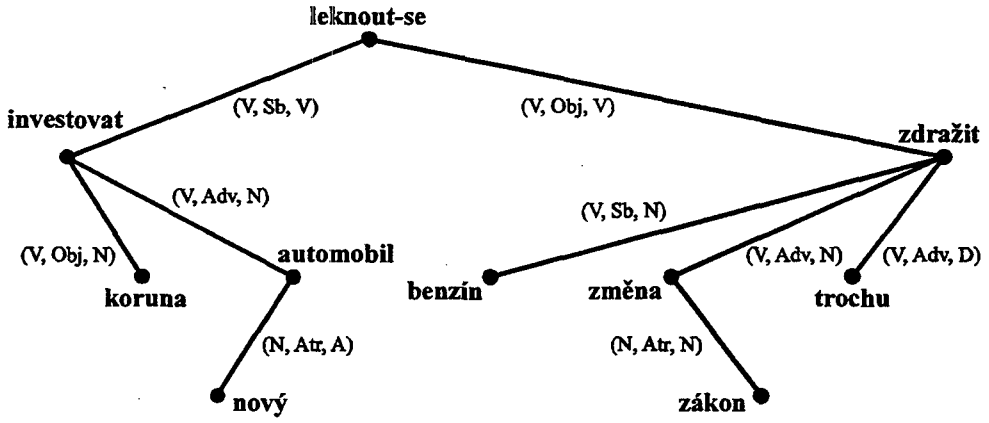


Figure 5: The DMCS of the sentence from Figure 3.

Word	Extracted MCEs	SDR used for derivation
zřejmý	[(A, Sb, V), změnit]	[zřejmý, (A, Sb, V), změnit]
změnit	[(V, Sb, N), stav] [(V, Adv, D), rychle] [zřejmý, (A, Sb, V)]	[změnit, (V, Sb, N), stav] [změnit, (V, Adv, D), rychle] [zřejmý, (A, Sb, V), změnit]
stav	[změnit, (V, Sb, N)]	[změnit, (V, Sb, N), stav]
rychle	[změnit , (V, Adv, D)]	[změnit , (V, Adv, D), rychle]
příliš	[rychle, (D, Adv, D)]	[rychle, (D, Adv, D), příliš]
leknout-se	[(V, Sb, V), investovat] [(V, Sb, V), zdražit]	[leknout-se , (V, Sb, V), investovat] [leknout-se , (V, Sb, V), zdražit]
investovat	[leknout-se , (V, Sb, V)] [(V, Obj, N), koruna] [(V, Adv, N), automobil]	[leknout-se , (V, Sb, V), investovat] [investovat, (V, Obj, N), koruna] [investovat, (V, Adv, N), automobil]
koruna	[investovat, (V, Obj, N)]	[investovat, (V, Obj, N), koruna]
automobil	[investovat, (V, Adv, N)] [(N, Atr, A), nový]	[investovat, (V, Adv, N), automobil] [automobil, (N, Atr, A), nový]
nový	[automobil, (N, Atr, A)]	[automobil, (N, Atr, A), nový]
zdražit	[leknout-se , (V, Sb, V)] [(V, Sb, N), benzín] [(V, Adv, N), změna] [(V, Adv, D), trochu]	[leknout-se , (V, Sb, V), zdražit] [zdražit , (V, Sb, N), benzín] [zdražit , (V, Adv, N), změna] [zdražit , (V, Adv, D), trochu]
benzín	[zdražit , (V, Sb, N)]	[zdražit , (V, Sb, N), benzín]
změna	[zdražit , (V, Adv, N)] [(N, Atr, N), zákon]	[zdražit , (V, Adv, N), změna] [změna , (N, Atr, N), zákon]
zákon	[změna , (N, Atr, N)]	[změna , (N, Atr, N), zákon]
trochu	[zdražit, (V, Adv, D)]	[zdražit, (V, Adv, D), trochu]

Table 3: Dependency microcontexts extracted from the two example sentences (from figures 2 and 3).

Another application of microcontexts consists in determining the context entropy of the words. Based on the context entropy we can distinguish vague and specific words and give them different weights for retrieval.

In order to improve retrieval performance by a modification of the query, two methods can be employed. The first is *query expansion* replacing words in the query with a set of words of the same meaning. It should ensure a higher recall. The second is *query refinement*, i.e. specifying the senses of query terms more precisely to avoid ambiguity of the query.

Asking a query, the user can be offered collocations of the terms used in the query. Then the user can decrease the vagueness of the (ambiguous) query terms by the choice of collocations that are characteristic for the sense required. It seems to be a good way of refining a query. The user can be also offered a list of words identified by the system as similar to query terms. Then the user can modify the query or even compose an alternative expression for the same query sense. This is a way to decrease or eliminate the negative influence of synonyms in relevant documents.

4.3 Experimental databases

In order to test the methods mentioned above we are developing two experimental databases. The first is the database of dependency microcontexts extracted from a large text corpus. We should obtain a lot of useful statistical data from it.

The second experimental database is a textual system MATES (MAster of TExt Sources). The main purpose of MATES is to serve as a textual database for experiments with various information retrieval methods.

MATES is constructed universally, not only for certain given retrieval algorithms, and it is adapted for the work with Czech language. Using MATES, it is possible to store both the originals of the input documents and their linguistically pre-processed versions. MATES supports grouping of documents into collections. For each collection an index is built and additional data structures are created that enable storing all the additional information about each term, each document and about their relations. This additional data can be used by the retrieval module.

In the near future, the MATES system should enable us to test the methods proposed here and evaluate their contribution to IR as well.

5 Conclusion

In the presented study, it is pointed out that ambiguity of language as well as synonymy are the serious obstacles preventing retrieval based on sense of the user's query. We describe an approach employing the lexical contexts to overcome or at least to reduce these difficulties. In order to recapitulate the results of this study and to make them more clear, we can sum up the essential and most important ideas into the following principles:

1. As to retrieval performance, word-based IR systems can be superseded by sense-based ones using effective techniques that are able to identify and compare meanings or senses of words. The structure of the IR system should contain the word context information retrieved from texts.
2. The closest core of the word context cannot be extracted based on word order. Therefore knowledge of the syntactic relations, which does carry this information, should be used.
3. The dependency tree containing all the surface dependency relations (ATS) contains information not relevant for the contexts extraction (with respect to IR needs), therefore we reduce this structure and we gather a structure containing only the semantically significant words and 4 main types of syntactic dependencies.
4. We present an algorithm for construction of the DMCS meeting the previously mentioned requirements. the DMCS allows for extraction of word microcontexts. The accuracy of this process depends on the quality of the used syntactic parser.
5. The statistical knowledge of lexical contexts can help especially to determine the importance of lexical units for retrieval and to tune the user's query in interaction with the

user using the knowledge of collocations and word similarity. Thus, the database of the retrieved microcontexts can be used for improving the performance of sense-based IR systems.

6. Uncertainty and vagueness in the text retrieval cannot be eliminated entirely since they are caused primarily by the character of the human thinking necessarily determining also the character of natural language.

Our long-term goal is to design an efficient IR system using the best methods of natural language analysis. The presented analyses as well as building the experimental textual database MATES are likely to be significant steps towards that goal.

Acknowledgements

This study has been supported by MŠMT (the FRVŠ grant no 1909).

References

Bémová, A.; Buráňová, E.; Hajič, J.; Kárník, J.; Pajas, P.; Panevová, J.; Štěpánek, J.; Urešová, Z. (1997) *Anotace na analytické rovině - příručka pro anotátory*, Technical Report #4, LJD UFAL MFF UK, Prague, Czech Republic. (in Czech)

Brill, E.; Mooney, R. J. 1997. *An Overview of Empirical Natural Language Processing*. In: AI Magazine, Vol. 18, No. 4.

Böhmová, A.; Hajičová, E. (1999) *How much of the underlying syntactic structure can be tagged automatically?* In: Proceedings of the ATALA Treebanks Workshop, Paris.

Collins, M.; Hajič, J.; Ramshaw, L.; Tillmann, Ch. (1999) *A Statistical Parser for Czech*. 37th Annual Meeting of the ACL, Proceedings of the conference, pp. 505-512.

Frakes, W. B.; Baeza-Yates, R. 1992. *Information Retrieval. Data structures and Algorithms*. 504 pp. Prentice Hall, Englewood Cliffs, New Jersey.

Hajič, J. (1998) *Building a syntactically annotated corpus: The Prague Dependency Treebank*. In: Issues of Valency and Meaning. Studies in Honour of Jarmila Panevová (ed. by E. Hajičová) (pp. 106-132). Prague: Karolinum.

Hladká, B. (2000) *Morphological Tagging of Czech Language*. PhD Thesis. MFF UK Prague.

Holub, M. (2000) *Use of Dependency Microcontexts in Information Retrieval*. Accepted for publication at Sofsem 2000 conference.

Ide and Véronis (1998)

Korfhage, R. 1997. *Information Storage and Retrieval*. 349 pp. John Wiley & Sons.

Krovetz, R.; Croft, W. B. (1992) *Lexical ambiguity and information retrieval*. In: ACM Transactions on Information Systems, 10(2), 1992, pp. 115-141.

Leacock, C.; Towell, G.; Voorhees, E. M. (1996) *Toward building contextual representations of word senses using statistical models*. In: B. Boguraev and J. Pustejovsky (editors), Corpus Processing for Lexical Acquisitions. pp. 97-113, MIT Press.

Lin, D. (1998) *Extracting Collocations from Text Corpora*. In: Computerm '98. Proceedings of the First Workshop on Computational Terminology. Montreal.

Lyons, J. (1977) *Semantics*. Cambridge University Press.

Miller, G. A.; Charles, W. G. 1991. *Contextual correlates of semantic similarity*. In: Language and cognitive processes, 6(1).

Salton, G. 1989. *Automatic Text Processing. The Transformation, Analysis, and Retrieval of Information by Computer*. 530 pp. Addison-Wesley.

Schütze, H.; Pedersen, J. O. (1995) *Information Retrieval Based on Word Senses*. In: Proceedings of the Fourth Annual Symposium on Document Analysis and Information retrieval, pp. 161-175, Las Vegas, NV.

Schütze, H. (1998) *Automatic Word Sense Discrimination*. In: Computational Linguistics, March 1998, Vol. 24, Number 1, pp. 97-123.

Towell G.; Voorhees, E. M. (1998) *Disambiguating Highly Ambiguous Words*.

In: Computational Linguistics, March 1998, Vol. 24, Number 1, pp. 125-145.

Yarowsky, D. 1993. *One sense per collocation.* In: Proceedings of ARPA Human Language Technology Workshop, pp. 266-271, Princeton, NJ.