Bull. Nov. Comp. Center, Comp. Science, 31 (2010), 91–109

Methods of syntactic analysis and comparison of constructions of a natural language oriented to use in search systems

A.A. Perfiliev, F.A. Murzin, T.V. Shmanina

Abstract. This work is dedicated to an actual problem of efficient information search in the Internet. The work is based on the algorithms of sentences comparison taking into account the schemes of syntactic analysis generated by Link Grammar Parser software. The main idea is that syntactic diagrams give us a primitive structure of a text, which allows us to select phrases in a text which have a syntactic structure similar to that given in a request. According to these ideas, the Information Retrieval System (IRS) iNetSearch was developed. Our study showed that it is often sufficient to remain on the syntactic level and obtain rather good search results. The final part of the article represents the results of testing for the methods implemented within iNetSearch.

Key words: Information Retrieval System, Link Grammar Parser, information search, semantic tree, relevance

1. Introduction

Under conditions of rapid growth of volumes of information resources, there is a necessity of quality improvement of information search [1]. It forces the developers of search systems to improve the algorithms of search and document ranking so that they be capable to consider the inquiry semantics.

Many researchers tend to the necessity of carrying out deep semantic analysis in order to make some semantic images of texts on the basis of which it is possible to carry out fine ranking of documents [2]. This approach, undoubtedly, is the most reasonable; however, it requires careful and long-term work on creation of suitable tools for automatic text processing. In particular, the detailed description of various fields of knowledge can be required. Therefore, search of partial solutions, one of which is presented in this work, is also expedient.

The main goal is to construct algorithms which can deduce an adequate estimation of the text relevance by getting into its structure. It is important that the given estimation would be deduced on the basis of the context of search inquiry, and would not be limited only by keywords, their similarity or frequency.

The method described in this work allows us to compare the natural language constructions and in some cases to identify even the paraphrased

variants of sentences on the basis of the analysis of their syntactic structures. Thus we can compare a search inquiry with a text in order to find out its relevance to this search inquiry. The method is based on processing and using the diagrams of links created by Link Grammar Parser software.

2. Metasearch system iNetSearch

Within the frameworks of the implemented project, the search robot iNetSearch has been developed which automates information search in a network. Its interface is simplified as much as possible. The user task is reduced to entering an inquiry into the program and waiting for the search finish and gathering of information from the Internet. On completion, it will suggest to look through the search results.

The features of the Information Retrieval System (IRS)

1. The system is installed on the user part and requires the Internet connection.
2. It uses the results of inquiries to existing search systems (for example, the search service nigma.ru was used for testing, because this system forwards an inquiry to other search systems, thereby increasing the possible area of search). The implemented system corrects the search results and specifies them.

The meta-information is not sufficient – semantic connectivity of text is important. The system looks through the content of the Internet pages received from a standard search service (for example, from nigma.ru) as a basis for analysis. If a source does not contain a text corresponding to certain criteria, it is rejected.

The process of loading the Internet pages assumes the following actions.

1. Replenishment of a user inquiry by means of dictionaries of synonyms, hyperonyms (hyponyms are also possible), sending this inquiry to a search system, analysis of the received hypertext, and replenishment of the list of hyperlinks.
2. Loading the content of Internet pages from the list.
3. Viewing the hypertext, search and gathering of hyperlinks.
4. Gathering the information satisfying the user inquiry.

IRS can work in several modes:

- purposeful loading of the list of entered Internet addresses, search for information corresponding to the inquiry;

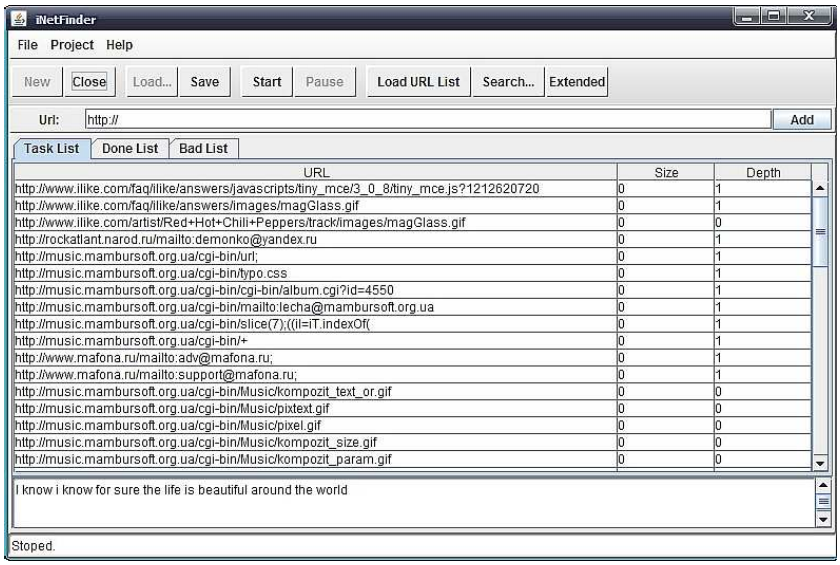


Figure 1. The main window of the iNetSearch system

- sending the inquiry to a search system, receiving and looking through the list of Internet addresses, search for information corresponding to the inquiry;
- recursive browsing of a catalogue, files viewing, search for information corresponding to the inquiry;
- purposeful loading of a file of the specified type from a site.

3. Search kernel of the iNetSearch system

A text inquiry arrives into the system from a user. At the following stage, keywords and some terms are picked out from the inquiry. Replenishment of the inquiry with synonyms and hyponyms (words of more specific meaning) gives a wider range of search for words. If the insufficient number of documents was found, the search inquiry repeats taking into consideration the hyperonyms (words of a more general sense; for example, a dog — an animal), which considerably expands the search area.

The search base of iNetSearch contains the text content of Internet-pages, which are taken from the download manager built into the system. Further, the text samples are passed to the system of preliminary filters, where a preliminary estimation of the text relevance is done. The presence of the corresponding keywords shows possible relevance of the considered samples. Preliminary filters reduce the parser run-time, which considerably accelerates the work.

The input sentences are translated into syntactic diagrams. The compiler

carries out lemmatization of words adding some meta-information and adds syntactic links between them attributing types to these links.

The syntactic parser allows us to consider attributing dependences between subordinate sentences. Thus we obtain rather essential information about a sentence. The syntactic parser generates the diagrams of syntactic analysis used in the system. They reflect the syntactic interrelation between words.

4. Link Grammar Parser

Some words about the parser used in the system. Link Grammar Parser (or Link) is a syntactic analyzer of the English language developed in 1990th at the Carnegie Mellon University, USA. It is based on the usage of grammatical links from a non-classical theory of English syntax [4].

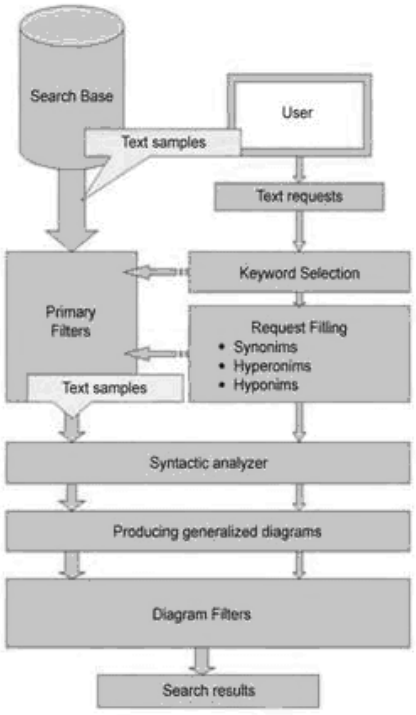


Figure 2. The scheme of a search kernel of the system iNetSearch

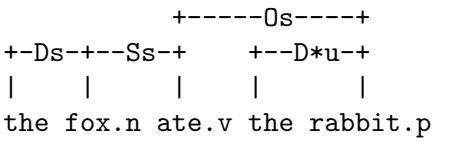
Having received a sentence, Link attributes it with a syntactic structure which consists of a set of marked links connecting the pairs of words. The mark of each link corresponds to some case of correct usage of the given pair of words in the sentence.

For example, the mark S corresponds to a link between a subject and a predicate, Ot – between an object and a predicate, etc. Besides, the mark can have a compound bottom index, which is necessary for checking the grammatical concordance and the word compatibility control. In addition, the system attributes the words of a sentence with the values of their basic classes. For example, nouns receive the signature ".n", verbs – ".v", etc.

Link is implemented in the C language for Unix and Windows, it has an open code and is distributed under the license compatible with GNU GPL. The parser dictionary includes about 60000 dictionary forms. It covers a huge part of syntactic constructions, including numerous rare expressions and idioms.

The parser work is stable; it can skip a part of a sentence which it cannot understand and define some structure for the rest part of a sentence. It is capable to process an unknown lexicon, and do reasonable assumptions about the syntactic category of unknown words from the context and writing. It has data about various names, numerical expressions, and various punctuation marks. Inside the parser, the methods of dynamic programming are used for comparison of links between words. For today, this software tool is one of the most promising for text processing [3, 4].

An example of analysis of the sentence "the fox ate the rabbit" is given below:



The diagrams obtained are, as a matter of fact, analogues to the so-called submission trees for sentences. In the submission trees, it is possible to put a question from a main word to a secondary word. Thus, words build up a treelike structure.

5. Algorithms of comparison

5.1. Basic algorithm

So we have an analysis tree. Further, generalization of such trees is made. At this stage, normalization of word forms is carried out. Some transformations of sentences can be made, for example, an inverse word order is replaced by a direct order. Complex forms of verbs are "cut off" to simple forms. A verb is transformed into one normalized form in a present simple tense. Difficult combinations of pretexts are unified. As a result, we have the tree skeleton where speech constructions are removed. These trees are compared with the diagram of a user inquiry.

In the beginning, filtration of diagrams is carrying out as follows. Before comparing, the words are passed to a simple filter to define a word form – it would be incorrect to consider a verb and a noun as identical. Comparison of words is simple: conformity of two words is checked using a set of rules. If all rules are checked and conformity is not revealed, words are considered as being distant in their sense. The set of rules represents the conditions that allow us to consider words akin. Such rules are: direct equality of words, partial coincidence (when we do not take into consideration the endings of words), synonymy of words, the presence of the hyponym-hyperonym

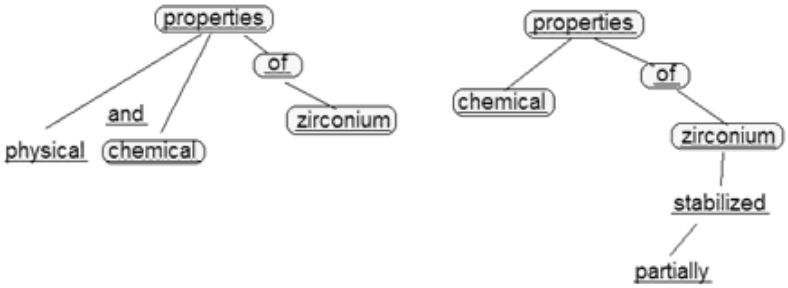


Figure 3. Example of imposing of two trees

relation, words with transmutations, and other possible relations between words [5].

The algorithm of expression comparison works similar to a finite automaton working on nodes of a tree. The finite automaton is constructed on the basis of a word-combination. If the text contains word-combinations that satisfy certain conditions, then the automatic machine transfers its heads to the next states. If at least one phrase transferred the automaton into its final state, this means that it is suitable and relevant to the inquiry. The number of automaton's heads depends on the set of rules to be checked. The use of automata essentially accelerates expression processing, especially in combination with a fast syntactic parser. Moreover, the majority of sentences are filtered out at the initial stage.

Thus, the resulted degree of estimations allows us to enter a certain measure of sentences closeness. It takes into account connection between words and search for word-combinations.

Sentences that passed the last filter are considered as relevant and are output to the user. When the work is complete, the system iNetSearch forms the summary of the relevant text found.

5.2. Additional features of the system

Note that in our system it is possible to form some specific inquiries. For example, we can find out all possible adjectives appearing together with some chemical element and thus to find out its chemical and physical properties.

Typical methods are as follows:

1) Selection of word-combinations. There is a word-combination: a noun and some adjectives. We search for word-combinations with the same noun and a partial set of adjectives.

1. Subject-predicate. There is a subject and a predicate or, simply, a verb and a noun. We build all possible verb forms. If there are similar constructions in the text under consideration, then we select them.

2. Transformation of abbreviations, idioms, search for synonyms. A required word-combination is supplemented with all possible reductions, abbreviations and synonyms.

5.3. Indistinct search for words and correction of errors

The mode of an indistinct search allows us to find documents which contain words similar in writing to the words of an inquiry. For example, words with misprints, some colloquial expressions, reductions, transliterations, and so on. We can also consider correction of words written in similar symbols from other languages and special symbols that are usually used by hackers for words masking.

6. Algorithms for comparison of the paraphrased sentences

6.1. Mathematical model

To compare natural language constructions and find the paraphrased variants of sentences, it was necessary to do some theoretical research and develop the corresponding methods based on the syntactic structure analysis.

Let L be a set of words of a natural language presented in dictionaries and documents.

The function $x' = Norm(x)x$, $x' \in L$, is defined on L, where x' is the normal form of x. For example, for an arbitrary noun, the result of the function is this noun's singular form in the Nominative case.

Besides, there are the series of one-place, two-place and three-place predicates and mappings defined on L. The validity of each predicate in the model is established by means of the corresponding dictionaries and algorithms. Each mapping in the model corresponds to one of the predicates and is defined according to its semantics.

6.2. Basic meta-words

Let POS be the set of parts of speech, and $NF \subset L$ be the set of all words of the language which are in the normal form. The two-place predicate $PartOfSpeech(\hat{P}, x')$ is a set on the Cartesian product $POS \times NF$. $PartOfSpeech(\hat{P}, x')$ is true if and only if $x' \in NF$ and $\hat{P}$ is a part of speech of x'.

Let $\bar{x} = \{x_1, ..., x_n\}$, where $\forall i\ x_i \in L$, be an arbitrary sentence over L. $\varphi : \bar{x} \to POS \times NF$ is a mapping, such that $\varphi(x) = \left\langle \hat{P}, x' \right\rangle$, where $x \in \bar{x}$, $x' = Norm(x)$, and the predicate $PartOfSpeech(\hat{P}, x')$ is true. This map is univalent assuming that it is always possible to determine the part of speech of the given word x, for example, from a context.

A pair $\left\langle \hat{P}, x' \right\rangle$ (further it is denoted as $\hat{P}[x']$) is called the basic meta-word corresponding to the word x.

Besides, let us define the following predicates on L which show that the given word $x \in L$ is a member of one of the auxiliary verb groups:

1) $Vaux_{11}(x)$, where $x \in \{will,' ll, may, might, should, must, can, could, would,' d, shall\}$;
2) $Vaux_{12}(x)$, where $x \in \{won't, shouldn't, mustn't, can't, couldn't, wouldn't\}$;
3) $Vaux_2(x)$, where $x \in \{isn't, aren't, wasn't, weren't\}$;
4) $Vaux_3(x)$, where $x \in \{hasn't, haven't, hadn't\}$;
5) $Vaux_4(x)$, where $x \in \{don't, doesn't, didn't\}$.

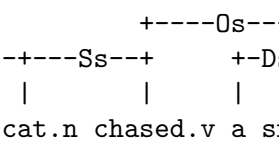
It is obvious that the property

$$\forall x \left(Vaux_i(x) \to PartOfSpeech(Verb, x') \right),$$

where $x' = Norm(x)$ and $Verb$ is a part of speech "verb", is fulfilled. Therefore, words from the introduced groups are represented by meta-words $\langle Verb, Vaux_i \rangle$, where $Vaux_i$ is considered to be a word from L.

6.3. The predicates associated with Link Grammar Parser

Let us put two-place predicates in correspondence with the Link Grammar Parser connectors. If Q is a name of a Link Grammar Parser connector, and both x_1 and x_2 are words from the sentence $\bar{x}$, then the predicate $Q(x_1, x_2)$ is true on $\bar{x}$ if and only if there is a connection with the label Q between x_1 and x_2 in the connection diagram of the given sentence. For example, in the sentence "The cat chased a snake" with the following connection diagram:



the predicates $Ds(the, cat)$, $Ss(cat, chased)$, $Ds(a, snake)$, $Os(chased, snake)$ are true. And there are no other predicates of this kind that are true on the given sentence.

6.4. Derivative meta-words and their construction formulas

Suppose that a set of basic meta-words

$$\varphi(\bar{x}) = \{\varphi(x_1), \ldots, \varphi(x_n)\} = \{\hat{P}_1[x'_1], \ldots, \hat{P}_n[x'_n]\}$$

corresponds to the sentence $\bar{x} = \{x_1, ...x_n\}$, and $P(\bar{x})$ is the set of subsets of words from $\bar{x}$, $P(\varphi(\bar{x}))$ is the set of subsets of meta-words from $\varphi(\bar{x})$, and $MN = \{PredAct,\ PredActNo,\ PredPas,\ PredPasNo,\ InfAct,\ InfActNo,\ InfPas,\ InfPasNo,\ ...\}$ is the set of type identifiers of composite words and sentence parts.

Composite words and composite sentence parts (further, composite units of a sentence) are such units of a sentence which are expressed by several words, but are indivisible, i.e. they cannot be segmented without modification or loss of the sense of these parts. For example, infinitives and gerunds, nominal and verbal predicates are considered to be composite units of a sentence.

Let us define a two-place predicate $SentenceMember(Name, U)$ on the Cartesian product $MN \times P(\bar{x})$. It is true if and only if $U \in P(\bar{x})$ is a composite unit of a sentence, $Name \in MN$ is a type identifier of U.

Thus the set of identifiers MN is defined in such a manner that

$$\forall U \left(\begin{array}{l} SentenceMember(Name, U) \rightarrow \\ (\neg \exists Name' \in MN\,((Name \neq Name')\ \ \&\ \\ SentenceMember(Name', U))) \end{array} \right).$$

Let us define a mapping $\psi : P(\varphi(\bar{x})) \rightarrow MN \times (NF^+)$, such that

$$\psi(\varphi(U)) = \left\langle Name, \tilde{U} \right\rangle \text{ for any } U = \{u_1, ...u_k\} \in P(\bar{x}),$$

where the predicate $SentenceMember(Name, U)$ is true and $\tilde{U} = u'_{i1} \ldots u'_{ir}$ is a concatenation of normal forms of the words included in U, $\{u'_{i1}, ..., u'_{ir}\} \subset \{u'_1, ..., u'_k\}$ (if auxiliary verbs or link-verbs were included in the structure U, they are filtered away).

The pair $\left\langle Name, \tilde{U} \right\rangle$ (or $Name[\tilde{U}]$) is called the derivative meta-word corresponding to U.

Let us describe the principle of construction of a projection of ψ-mapping onto the set $\{PredAct\} \times NF$, where the identifier $PredAct$ corresponds to a predicate expressed by a verb in active voice and positive form.

Let the sentence $\bar{x}$ be in correspondence with its connection diagram. Then the validity of the predicate $SentenceMember(PredAct, U)$ on some $U \subset \bar{x}$ is equivalent to the validity of the following formulas on U:

$$(\exists x \in U)(\exists x' \in NF)[x' = Norm(x)\ \&\ [PartOfSpeech(Verb, x')\ \&$$
$$(\exists y \in L)\,(S(y,x)) \vee (\exists y \in U)[PartOfSpeech(Verb, x')\ \&\ Vaux_{11}(y)\ \&$$
$$I(y,x)\ \&\ \neg N(y,"not") \vee PartOfSpeech(PartAct, x)\ \&$$
$$Norm(y) = "be"\ \&\ Pg(y,x)\ \&\ \neg N(y,"not") \vee$$
$$PartOfSpeech(PartAct, x)\ \&\ Vaux_{11}(y)\ \&\ I(y,"be")\ \&\ Pg("be", x)\ \&$$
$$\neg N(y,"not") \vee PartOfSpeech(PartPas, x)\ \&\ Norm(y) = "have"\ \&$$
$$PP(y,x)\ \&\ \neg N(y,"not") \vee PartOfSpeech(PartPas, x)\ \&\ Vaux_{11}(y)\ \&$$
$$I(y,"have")\ \&\ PP("have", x)\ \&\ \neg N(y,"not") \vee$$
$$PartOfSpeech(PartAct, x)\ \&\ Norm(y) = "have"\ \&\ PP(y,"be")\ \&$$
$$Pg("be", x)\ \&\ \neg N(y,"not") \vee PartOfSpeech(PartAct, x)\ \&\ Vaux_{11}(y)\ \&$$
$$I(y,"have")\ \&\ PP("have","be")\ \&\ Pg("be", x)\ \&$$
$$\&\ \neg N(y,"not")]]]$$

Here several connectors of Link Grammar Parser are involved:

I – connects a verb and an infinitive;

PP – connects the form of the word "have" and past tense participle;

Pg – connects the form of the word "be" and present participle.

It is possible to find similar formulas for every predicate $SentenceMember(Name,\ \cdot\ )$, where $Name \in MN$.

Let us consider that $Q(x,y) \to Q\left(\hat{P}_x[x'], \hat{P}_y[y']\right)$ takes place for any connector Q of Link Grammar Parser, where $x, y \in \bar{x},\ \varphi(x) = \hat{P}_x[x'],\ \varphi(y) = \hat{P}_y[y']$. Therefore, it is possible to rewrite each formula corresponding to the predicate $SentenceMember(Name,\ \cdot\ )$ in terms of meta-words. The obtained formulas define the order of construction of the corresponding derivative meta-words $Name[\tilde{U}]$ from the basic meta-words, that is a projection of ψ-mapping to the set $\{Name\} \times NF^+,\ Name \in MN$. The set of all constructed projections gives the required mapping ψ.

For example, the formula for construction of a derivative meta-word $PredAct[X]$ (here Xrepresents the notional verb in its normal form) looks like:

$$V[X]\ \&\ \left(\begin{array}{l}(\exists N[Y])\,(S(N[Y], V[X])) \vee (\exists PRON[Y])\,(S(PRON[Y], V[X])) \vee \\ (\exists GER\_ACT[Y])\,(S(GER\_ACT[Y], V[X]))\end{array}\right) \vee$$
$$I(V["Vaux_{11}"], V[X])\ \&\ \neg N(V["Vaux_{11}"], PRTC["not"]) \vee$$
$$Pg(V["be"], PART\_ACT[X])\ \&\ \neg N(V["be"], PRTC["not"]) \vee$$
$$I(V["Vaux_{11}"], V["be"])\ \&\ Pg(V["be"], PART\_ACT[X])\ \&$$
$$\&\ \neg N(V["Vaux_{11}"], PRTC["not"]) \vee$$
$$PP(V["have"], PART\_PAS[X])\ \&\ \neg N(V["have"], PRTC["not"]) \vee$$
$$I(V["Vaux_{11}"], V["have"])\ \&\ PP(V["have"], PART\_PAS[X])\ \&$$
$$\&\ \neg N(V["Vaux_{11}"], PRTC["not"]) \vee$$
$$PP(V["have"], V["be"])\ \&\ Pg(V["be"], PART\_ACT[X])\ \&$$
$$\&\ \neg N(V["have"], PRTC["not"]) \vee$$
$$I(V["Vaux_{11}"], V["have"])\ \&\ PP(V["have"], V["be"])\ \&$$
$$\&\ Pg(V["be"], PART\_ACT[X])\ \&\ \neg N(V["Vaux_{11}"], PRTC["not"]).$$

6.5. Semantics-syntactical relation predicates and meta-connections

Assume that a Link Grammar Parser connection diagram and a set of meta-words $MW = \left\{\hat{P}_{i1}[x'_{i1}], ..., \hat{P}_{ik}[x'_{ik}]\right\}$, consisting of all derivative meta-words built on $\bar{x}$ and all basic ones not used in construction of derivatives, correspond to the sentence $\bar{x} = \{x_1, ..., x_n\}$.

Further the syntactic submission relations between words or composite units of a sentence are considered (this units can be considered as words without loss of generality). In each word pair connected by the syntactic submission relation, one of them is principal and the second one is dependent. The presence of the syntactic submission relation between these words is determined by the capability to state a question from the principal word to the dependent.

Syntactic submission relations are considered only between meaningful words of a sentence, that is, between the words that are not particles, prepositions, or auxiliary verbs.

Let SR be the set of type identifiers of syntactic submission relations. Each identifier from this set characterizes the type of the syntactic relation between two meaningful words (for example, the relation "predicate-object") and also fixes the set of features of this syntactic relation (for example, "a predicate in active voice and positive form with a direct object"), which as a whole allows us to establish the type of the semantics-syntactical relation between two given words.

The three-local predicates of syntactic subordination relations are defined on the set of words from $\bar{x}$:

$SyntacticRelation(RelationName, w_1, w_2)$, which is true if and only if w_1 and w_2 are a pair of meaningful words of the sentence connected by the syntactic subordination relation of the type $RelationName \in SR$, and w_1 is a principal word in this pair and w_2 is subordinate.

Let us further define the mapping$\chi : MW \times MW \to SR \times NF \times NF$, such that $\chi\left(\hat{P}_1[w'_1], \hat{P}_2[w'_2]\right) = \langle RelationName, w'_1, w'_2\rangle$, where

$$\varphi(w_i) = \hat{P}_i[w'_i],\ w_i \in \bar{x},\ \ i = 1, 2,$$

and the predicate $SyntacticRelation(RelationName, w_1, w_2)$ is true.

The ordered triple $\langle RelationName, w'_1, w'_2\rangle$ is called meta-connection, where the identifier $RelationName$ is the meta-connection name, and w'_1 and w'_2 are accordingly the principal and subordinate words of the meta-connection. The notation $RelationName[w'_1, w'_2]$ is also used for a meta-connection representation along with $\langle RelationName, w'_1, w'_2\rangle$.

Thus, each meta-connection contains a pair of meaningful words between which the syntactic submission relation is determined. Information on the

recognized syntactic relation type is included in a meta-connection title, along with some additional features, on the basis of which the conclusion about semantic-syntactical relation between two given words can be made. For example, the meta-connection $PRED\_ACT\_DIR\_OBJ$ $[get, award]$ will be put in correspondence with the selected word combination in the sentence "Ann **got** an **award**". This meta-connection indicates that the word "award" is a direct object for the predicate expressed by the verb "get", being in active voice and positive form, hence "award" is a direct object of the operation "get". This information will be necessary for construction of the semantic graph of a sentence.

A mapping χ is constructed similarly to the mapping ψ. It sets the construction order for derivative meta-words. Namely, for each $SyntacticRelation(RelationName,\ \cdot\ ,\ \cdot\ )$ predicate, an equivalent formula is written in terms of predicates that correspond to Link Grammar Parser connectors and $PartOfSpeech(\ \cdot\ ,\ \cdot\ )$, $SentenceMember(\ \cdot\ ,\ \cdot\ )$ and $SyntacticRelation(\ \cdot\ ,\ \cdot\ ,\ \cdot\ )$ predicates. Then each formula is rewritten in terms of meta-words and meta-connections. The resulting set of formulas defines the order in which the corresponding meta-connections are constructed from meta-words and Link Grammar Parser connectors.

6.6. The semantic graph of a sentence

Let us define $Syn(x, y)$, a predicate on L which is true if and y belong to the same part of speech and are synonyms.

Let $\overline{x} = \{x_1, x_2, ..., x_n\}$ be a sentence over L, and $S \subset \bar{x}$ be the set of all meaningful words and compound sentence units from $\overline{x}$ which, as earlier, are considered as words.

The semantic graph G of the sentence $\overline{x}$ is a marked digraph the nodes of which are marked by the sets of synonymous words, and arcs are marked by the sets of semantic-syntactical relations:

$G = \langle V, E, s, r \rangle$, where

V is the set of nodes of the graph G, $|V| = |S|$;

E is the set of arcs of the graph G, $E \subseteq V \times V$;

$s : V \to P(L)$, where $P(L)$ is the set of subsets of L;

$r : E \to P(M)$, where M is the set of marks of arcs.

Thereby the function s should satisfy the following condition:
$(\forall x \in S)(\exists! v \in V)\left((Norm(x) \in s(v)) \ \&\ s(v) = \{y \in L | Syn(x, y)\}\right)$, i.e. there will be a unique node of the semantic graph corresponding to each meaningful word x of the sentence $\overline{x}$. This node will be marked by the set of synonyms of the word x.

Thus, the mark of each arc of the semantic graph should represent the semantic-syntactical relation between the sentence words, mapped into nodes incident to the given arc.

For example, in the semantic graph of the sentence "Cats were sitting in the box", there will be an arc $sit \xrightarrow{iplaceCIR} box$ corresponding to the relation between an operation and an adverbial modifier of a place.

7. A method of natural language construction comparison

7.1. Construction of the semantic graph of a sentence

Let a Link Grammar Parser connection diagram be constructed for a sentence $\bar{x} = \{x_1, ..., x_n\}$. The semantic graph is constructed by means of a consecutive application of mappings ϕ, ψ and χ to the connection diagram of the sentence $\bar{x}$. Because references to other meta-connections could exist in the formulas of meta-connection construction, it is necessary to set the order of syntactic relation recognition for the mapping χ. In particular, it is possible to find the connection between the sentence parts of the highest order in the free tree, obtained as a result of application of the mapping ψ, and to consider one of meta-words incident to the found connection as a root. In this case it is necessary to start meta-connection construction from leaves of the resulting rooted tree. This method is based on the assumption that the child nodes in the tree correspond to the sentence parts subordinate to the parent node or lie on paths to such nodes. The formulas of meta-connections are constructed in such a manner that they can refer only to those meta-connections that correspond to subtrees with their roots being meta-words involved in the relation under consideration.

As a result of application of these mappings, the list of meta-connections containing all information on recognized semantic-syntactical relations for the sentence is determined. Thus, each meta-connection in the set of rules of meta-connection construction corresponds to one or several marks of the semantic graph arcs.

Construction of the semantic graph on the basis of a meta-connection list is made as follows. In each meta-connection of the kind

$$MetaLinkName[w_1, w_2],$$

w_1 is a principal word and w_2 is subordinate, so it is necessary to build an arc in the semantic graph from the node marked by w_1 to the node marked by w_2 and to mark it with the set of marks corresponding to the title of $MetaLinkName$ meta-connection. Additionally, for some meta-connections in the dictionary, it could be specified that it is necessary to build an inverse arc from w_2 to w_1 with some mark. Therefore, the semantic graph is generally not a tree.

After all meta-connections are mapped into arcs of the semantic graph, each its node is marked by a set of synonyms of a word corresponding to it. Thus, the semantic graph construction is complete.

7.2. Comparison of semantic graphs of two sentences

Let us assume that two sentences $\bar{x}_1$ and $\bar{x}_2$ are given and it is necessary to compare the second sentence with the first. The sentence$\bar{x}_1$ is called the query, and a sentence$\bar{x}_2$ is the applicant.

Let $G_1 = \langle V_1, E_1, s_1, r_1 \rangle$ and $G_2 = \langle V_2, E_2, s_2, r_2 \rangle$ be the semantic graphs of $\bar{x}_1$ and $\bar{x}_2$, accordingly. Each arc of the semantic graph of the query is put into correspondence with some arc in the semantic graph of the applicant by a certain rule. For this purpose, the map $F : G_1 \to G_2$ is defined with the following properties:

1) $domF \subseteq E_1$;
2) $rangeF \subseteq E_2$ and it is maximum (in terms of weight);
3) F is injective on its domain;
4) for any connectivity component $K = \langle V_1^K, E_1^K \rangle$ of the graph G_1, it holds that $F(E_1^K \cap domF)$ is the set of arcs also belonging to the same connectivity component $S = \langle V_2^S, E_2^S \rangle$ of the graph G_2;
5) for any connectivity component $S = \langle V_2^S, E_2^S \rangle$ of the graph G_2, it holds that $F^{-1}(E_2^S \cap rangeF)$ is the set of arcs belonging to the same connectivity component $K = \langle V_1^K, E_1^K \rangle$ of the graph G_1.
6) $(\forall e \in domF)\,(r_1(e) \cap r_2(F(e)) \neq \emptyset)$;
7) $(\forall v \in V_1)\,(\forall w \in V_1)\,(\langle v, w \rangle \in domF \to$
$((s_1(v) \cap s_2(F(v)) \neq \emptyset)\ \ \&\ \ (s_1(w) \cap s_2(F(w)) \neq \emptyset)))$.

Thus, F puts in correspondence two arcs of G_1 and G_2 in case that they have at least one common mark and are incident to nodes marked with synonymous words. So, the beginning and the end of the arc in the first graph "are synonymous" to the beginning and the end of the arc in the second graph, accordingly. The arcs of the graph G_2 belonging to $rangeF$ are called coincident, and the arcs belonging to $E_2 \backslash rangeF$ are noncoincident.

The function F with the above properties could be not unique. In particular, non-uniqueness of F could be caused by the fact that G_1 or G_2 may contain isomorphic subgraphs.

To construct the mapping F of the kind here described, it is possible to use well-known algorithms of isomorphic graph enclosure search.

7.3. Evaluation of the coincidence coefficient for two sentences

To estimate the coincidence coefficient for an applicant-sentence and a query-sentence, it is reasonable to take into account the number of both coincident and noncoincident arcs in the applicant's semantic graph. The number of coincident arcs is considered to be the main coincidence factor, but the

number of noncoincident ones is an auxiliary factor used for correcting the estimation of the coincidence coefficient that could be useful for ranking the set of applicants which have gained an equal estimation at calculation of the number of coincident arcs .

In particular, the following hypothesis is used: the more the number of noncoincident arcs in the semantic graph of the applicant, the more the probability that the segment of the applicant, which is similar to some part of the query, has little significance for the applicant or gains another meaning in it. Therefore, if two applicants have equal numbers of equivalent matches with the query, then the sentence of the smaller length is considered to be the most similar.

Besides, the arcs of semantic graphs with different marks have different weight. The weight depends on importance of the semantic-syntactical relation corresponding to a mark.

The formula below evaluates the coincidence coefficient for two sentences, it is appropriate for ranking the applicants and satisfies the above-stated principles:

$$y = \frac{\sum\limits_{i=1}^{N} p_i - \left(\frac{\sum\limits_{i=1}^{M} q_i}{\sum\limits_{i=1}^{\tilde{M}} t_i} \right)}{\sum\limits_{i=1}^{K} r_i}, \text{ where}$$

y is the coincidence coefficient for the applicant and query;

K and N are the number of arcs in the semantic graph of the query and in the applicant subgraph consisting of coincident arcs;

$\tilde{M}$ is the total number of arcs in the semantic graph of the applicant;

M is the number of noncoincident arcs in the semantic graph of the applicant, $M = \tilde{M} - \sum\limits_{i=1}^{N} N_i$.

r_i and t_i are the weights of arcs of the semantic graphs of the query and the applicant, respectively;

p_i is the weight of a coincident arc in the semantic graph of the applicant;

q_i is the weight of a noncoincident arc in the semantic graph of the applicant.

Thus, the more is the number of coincident arcs in the semantic graph of the applicant and the higher are their weights, the higher is the estimate it should get.

Besides, the formula contains the correcting component $-\frac{\left(\sum\limits_{i=1}^{M} q_i\right)}{\left(\sum\limits_{i=1}^{\tilde{M}} t_i\right)}$. Its absolute value is not greater than 1 and serves for ranking the applicants having the same number of equivalent coincidences with the query. Thus,

the lighter are the noncoincident arcs, the less decreasing is the applicant's estimate.

The magnitude $\sum_{i=1}^{K} r_i$ normalizes the coincidence coefficient for two sentences. Thus, $y = 1$ only if the semantic graphs of the query and applicant are completely identical.

7.4. Restrictions

The method offered above is applicable only to sentences which can be analyzed by Link Grammar Parser. Besides, the method is based on the assumption that, as an input, it takes the connection diagram that properly reflects all relations between concepts. Otherwise, the resulting semantic graph will reflect relations between concepts incorrectly.

This method cannot compare paraphrases in case when the compared sentences contain formally different concept systems, or concepts are connected with each other by different semantic-syntactical relations, for example, as it is in the sentences "The fox attacked the rabbit" and "The rabbit fell a victim to the attack of the fox". It takes place because the method does not take into account the semantics of words and operates only with the syntactical categories.

At last, the system of rules for semantic graph construction has been developed in such a way that identical graphs could correspond only to those sentences that set identical semantic-syntactical relations between concepts. In that scope this method is correct. However, if we consider semantics of a sentence in a broad sense, [6] the assertion formulated above cannot be fulfilled for many reasons: the method ignores intonations and accents in a sentence, which can change the meaning of a phrase; semantically different relations could be identified in case they cannot be distinguished by grammatical indications, and so on.

8. The results of iNetSearch testing.

To demonstrate efficiency of **iNetSearch**, experiments have been made using this system. Ten simple inquiries from the field of inorganic chemistry have been generated. For each inquiry, the lists of addresses with their description, usually returned to the user by a search system, have been loaded. On the basis of these short snippets, the resource estimation has been made. For comparison with another search system (namely, with the system nigma.ru, since it can readdress inquiries to other systems), the statistics of inquiries relative to ten sentences of inquiries has been made. The system left relevant references, rejecting irrelevant by its estimation. As a result of testing, on the average, the system allocated 5-15 qualitative relevant references out of 100 references received from nigma.ru, accepted about

Table 1. The results of testing of the basic algorithm of the iNetSearch system

Request	Total number of links from Internet search system	Number of relevant links collected by system	Number of relevant links skipped by system	Number of irrelevant links collected by system
the burning rate of rocket fuels	99	15	8	1
using of liquid nitrogen	85	29	2	0
physical and chemical properties of zirconium	96	8	2	9
raw material for pharmaceutical product	121	26	7	9
using of zirconium in medicine	97	9	1	1
harmful influence of strontium on a man	102	6	0	0
molecular structure of products of alcohol disintegration	85	20	1	12
methods of glycerin production	89	3	2	0
physical properties of oxides	95	17	4	8
classification of separation techniques	107	10	0	1

5 incorrect references as relevant and rejected others as irrelevant, which corresponds to reality. This demonstrates that the system could make filtration at a good level. The results of testing are shown below.

Further, two methods for natural language constructions have been compared – the basic, used in the initial version of the iNetSearch system, and a new one, which takes into account the sentences rephrasing. The original method is based on comparison of links diagrams for an inquiry and a phrase from the document under estimation, and comparison uses some generalizations and simplifications for taking into account the possibilities of paraphrasing.

The inquiries, paraphrases of which had to be found, were made on various subjects. The sources of the inquiries are as follows:

1) a collection of scientific papers on more than 20 subjects;
2) a collection of educational texts.

To estimate the quality of search, the following characteristics have been chosen:

1) $Precision = \frac{|Relevant \cap Retrieved|}{|Retrieved|}$;
2) $Recall = \frac{|Relevant \cap Retrieved|}{|Relevant|}$;
3) $Fall - out = \frac{|NotRelevant \cap Retrieved|}{|NotRelevant|}$.

Here we have:

Relevant – a set of documents from a collection relevant to the inquiry;

NotRelevant – a set of documents irrelevant to the inquiry;

Retrieved – a set of documents approved by the iNetSearch system.

As a collection of documents, all documents received by iNetSearch from search systems were considered.

The table containing the results of testing is represented below, namely, the average values of precision, completeness and losses for each inquiry.

	Precision, %	Recall, %	Fall-out, %
Basic method of iNetSearch	0,520	0,875	0,576
Comparison of semantic trees	0,551	0,893	0,504

Thus on the average, the search system approves less irrelevant and more relevant documents.

9. Conclusion

The main purpose of this work was to develop the methods for comparison of natural language constructions and for identification, among others, of the paraphrased variants of sentences by analyzing their syntactic structure.

In the process of solving these problems, we have found the methods that allow us to represent semantic-syntactical relations between semantic units of a sentence, to construct this representation on the basis of diagrams of Link Grammar Parser, and also to calculate the degree of coincidence of natural language constructions. Besides, these methods have been implemented and integrated into the metasearch system iNetSearch. Testing was also performed, showing their applicability to information search problems.

As a result, we see high efficiency of the approach here presented. On the other hand, the method that uses rephrasing allowed us to improve the results of the iNetSearch system, but testing showed that this improvement is insignificant in comparison with the basic algorithm. It is also possible to make a conclusion that further development of this method will not lead to substantial improvement of the obtained results. One of the reasons is that the possibilities of Link Grammar Parser at the current stage of work are almost completely exhausted. And, in spite of the fact that Link Grammar Parser possesses a number of advantages (high speed, partial coverage of semantics, many examples of its successful application in the systems of Internet texts filtration), it makes us to stay at the level of syntax with partial semantics coverage. Therefore, if we want to have essential advancement, it is necessary to move to a higher level [7, 8], to knowledge engineering.

References

[1] Text REtrieval Conference (TREC). – Available at: http://trec.nist.gov/

[2] Salton G. Automatic Information Organization and Retrieval. – McGraw-Hill, 1968. – 514 p.

[3] Batura T.V., Murzin F.A. The Machine-Oriented Logic Methods for Representation of Semantics of a Text in a Natural Language. – Novosibirsk: Publishing Company of NGTU, 2008. – 248 p. (in Russian)

[4] Temperley D., Sleator D., Lafferty J. Link Grammar Documentation. – 1998. – Available at: http://www.link.cs.cmu.edu/link/dict/index.html

[5] Grinberg D., Lafferty J., Sleator D. A robust parsing algorithm for link grammars. – Pittsburgh, 1995. – (Tech. Rep. / Carnegie Mellon Univ. Computer Science; CMU-CS-95-125).

[6] Betty Schrampfer Azar. Understanding and Using English Grammar, 3rd ed. – NY: Pearson Education, 2002. – 567 .

[7] Nirenburg S., Raskin V. Ontological Semantics. – Cambridge, MA: MIT Press, 2004. – 420 p.

[8] Thompson C. Acquiring word-meaning mappings for natural language interfaces // J. of Artificial Intelligence Res. – 2003. – Vol. 18. – P. 1–44.