Semantic Paraphrasing for Information Retrieval and Extraction*

Juri D. Apresjan[1], Igor M. Boguslavsky[1,2], Leonid L. Iomdin[1], Leonid L. Cinman[1], and Svetlana P. Timoshenko[1]

[1] Institute for Information Transmission Problems (Kharkevich Institute), RAS, Bolshoj Karetnyj per. 19, Moscow, 127994, Russia
{apr,bogus,iomdin,cinman,timoshenko}@iitp.ru
[2] Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Boadilla del Monte, Madrid, Spain
igor@opera.dia.fi.upm.es

Abstract. The paper is devoted to the development of a system of synonymous and quasi-synonymous paraphrasing and its practical applications, first of all in the domain of search engine optimization and information extraction. This system is part of the ETAP-3 multifunctional NLP environment created by the Laboratory of Computational Linguistics of the Kharkevich Institute for Information Transmission Problems. Combinatorial dictionaries of Russian, English and some other languages and a rule-driven parser constitute the core of ETAP-3 while a variety of generating modules are used in a number of applications. The paraphrase generator, based on the apparatus of lexical functions, is one such module. We describe the general layout of the paraphrase generator and discuss an experiment that demonstrates its potential as a tool for search optimization.

Keywords: paraphrase generator, information retrieval, information extraction, lexical functions.

1 Introduction

Insufficient precision (even despite high recall) of search is known to be one of the major drawbacks of modern search engines. In many cases, documents relevant for the query are found by the engine but they are drowned in the ocean of irrelevant documents offered by the search engine together with the relevant ones. To give an example, if Google is asked for the "population of Tatarstan" it will yield over 100,000 results, which for the most part have no relation whatsoever to the request (which is obviously triggered by the wish to know how many people live in Tatarstan). In particular, in the majority of documents found the two words occur independently of each other and do not constitute the notion of interest to the requesters.

* This study was supported in part by the Russian Foundation of Basic Research with a grant No. 08-06-00344, for which the authors are grateful.

T. Andreasen et al. (Eds.): FQAS 2009, LNAI 5822, pp. 512–523, 2009.

One can resort to the precise query option and require that the two words appear in the text in exactly the same form as in the query. In this case the engine will overlook those documents in which the same meaning is expressed somewhat differently, even if the difference is very slight: "population of Tataria" (a synonymous name for this autonomous republic of Russia) or "Tatarstan's population".

We have hypothesized that the search is likely to be more accurate and precise if it is based on meanings rather than on words. In most search tasks what we need is not the documents that contain words listed in the query but the documents that render the sense which is of interest to us, irrespective of the words that convey it. As a matter of fact, most, if not all, search engine users implicitly resort to this idea: if upon sending a query expression one does not obtain a satisfactory result from the system, one tries to reformulate the query with other words.

Synonymous paraphrasing of utterances is a natural way of upgrading from word-based to meaning-based search. The notion of meaning is extremely hard to formalize. One of the more constructive definitions of meaning has it that **meaning is the invariant of synonymous paraphrasing**, i.e. it is the common property shared by all texts which are viewed by natural language speakers as denoting the same thing. Hence, if we succeed in creating a generator of paraphrasing, it will be an important stage in "harnessing" the meaning.

In addition to information retrieval, another possible application of synonymous paraphrasing is information extraction. The systems of extracting thematically restricted information from large bulks of natural language texts available now have a major disadvantage: the creation of the system itself and its tuning to another subject domain is a costly enterprise that requires large amounts of knowledge engineering. For this reason, numerous attempts have been made to automate this work using machine learning techniques (see e.g. [1], [2] and many other studies in the field). Of great interest is the direction of research that is aimed at full exclusion of the processes of tuning and retuning of the system to particular subject domains [3], [4], [5]. This is achieved with the help of the approach that, from the start, the system is not targeted to any particular subject. In response to a user query which specifies a topic for information extraction the system automatically generates patterns, or text fragments, with variables relevant to this topic. The next step is to specify clusters of paraphrases, or expressions that are close in meaning to each other. These clusters are needed to ensure the system's robustness with regard to the variation of meaning expression. The performance of the whole system largely depends on how successful one is in both types of operations: pattern generation and synonymous pattern clusterization.

The method of paraphrase generation used by S. Sekine in [5] is based on machine learning. Automatic generation of paraphrases heavily relies on the availability of various texts that describe the same real situation or event. One of the rare sources of such texts is the multitude of news sites and pages for the same time span, which are likely to describe the same events or incidents.

To give an example, on June 22, 2009 many news agencies reported on the latest developments in Iran where protests against fraud at the presidential election had been crushed by the police. Among the reports, the following three described the same course of events using different lexical units and grammatical means:

-A security crackdown appears to have quelled street protests organized in the wake of Iran's disputed presidential election.
-Iranian riot police and militiamen appear to have halted protests in the capital, Tehran, after days of clashes over the country's disputed election.
-The Basij, a plainclothes militia under the command of the Revolutionary Guard, have been used to quell street protests that erupted after the election result was announced.

A good syntactic parser backed by an ontology-like resource having lists of synonyms, hyponyms and hypernyms may help to form certain paraphrases: *police – militia*, *country's election – presidential election*, and may be a few others less trivial. However, this method of paraphrase extraction has an important drawback, too: texts so close in meaning that they can be viewed as paraphrases of each other are not a massive or representative source; besides, the precision of this method is not too high.

The paraphrase generator proposed in this paper which is intended to be used in both information retrieval and extraction is based on different principles. It is a system of rules heavily relying on a rich dictionary of lexical functions created by expert lexicographers (Section 2). Synonymous paraphrasing is, for the most part, of a universal linguistic character hardly depending on the content of the query to be served. Accordingly, the same paraphrase generator can be used to process queries belonging to different subject domains and does not require retuning when the query's subject domain is changed. The apparatus of lexical functions allows achieving a systematic and broad coverage of synonymic means of the language. We can therefore expect that the method of lexicographic presentation of synonymy will be more adequate that the method of machine learning.

We have made the first steps in this direction. On an experimental scale, we have created a system of synonymous paraphrasing (primarily, for Russian, although work has been started recently to extrapolate the results of this work to English). The generator of synonymous paraphrasing we are developing (Section 3) is able to vary the form of an utterance while its sense is preserved. This can be exemplified by a host of paraphrases which could be good matches in response to the query about the height of Mount Elborus (irrespective of the wording of the query itself): *Mount Elborus is 5642 metres high <tall>*; *Mount Elborus stands 5642 meters high <tall>*, *Mount Elborus stands at 5642 meters high <tall>*, *Mount Elborus rises 5642 meters*, *Mount Elborus measures 5642 meters in height*, *The height of Mount Elborus is 5642 meters*, *The height of Mount Elborus equals to 5642 meters*, *The height of Mount Elborus amounts to 5642 meters,* etc. The evaluation of this experiment is given in Section 4.

2 Lexical Functions and Paraphrases

As mentioned above, the paraphrase generator is a system of rules heavily relying on a rich dictionary of lexical functions (LFs) created by expert lexicographers. The apparatus of lexical functions allows achieving a systematic and broad coverage of synonymic means of the language.

The notion of LF was first proposed by the author of the Meaning ⇔ Text linguistic theory, Igor Mel'čuk, in 1970s and has been extensively studied and developed by the Moscow Linguistic School and, in particular, by the laboratory of Computational

Linguistics of the Kharkevich Institute of Information Transmission Problems. For more information, see [6], [7], [8], [9], [10]. The laboratory has developed a number of NLP applications using LFs, including machine translation, where LFs are used to resolve lexical and syntactic ambiguity, to achieve idiomatic translation of collocations and to improve syntactic parsing [11], [12].

Another interesting benefit of lexical functions can be obtained in the notoriously difficult task of anaphora resolution, in particular, of finding antecedents of pronouns. At present, this task is mainly solved in NLP by statistical methods, whose precision is not too high. The precision may be increased by using the information on lexical functions and semantic features stated in argument frames of predicate words. Our working hypothesis is that, of several grammatically acceptable candidates for antecedents, the priority should be given to the one that better meets the semantic and lexical functional requirements of the context in which the pronoun appears. To give an example, sentence *The convention was signed by the United States, Honduras, El Salvador, Dominican Republic, Haiti, Argentina, Venezuela, Uruguay, Paraguay, Mexico, Panama, Bolivia, Guatemala, Brazil, Ecuador, Nicaragua, Colombia, Chile, Peru and Cuba but **it** was later annulled* has as many as 22 nouns that, theoretically, may be antecedents for the pronoun *it*. However, there is only one correct antecedent, *convention*, despite the fact that this word occurs furthest from the pronoun. This conclusion could be made due to the fact that the pronoun *it* occupies the object position of the lexical functional verb *annul*, and the argument of this LF ($LiquFact_0$) can only be the word *convention*.

The generator of paraphrases is another application based on LFs.

A prototypical LF is a triple of elements {R, X, Y}, where R is a certain sense or a certain general semantic relation obtaining between the argument lexeme X (the keyword) and some other lexeme Y which is the value of R with regard to X (by a lexeme we mean either a word in one of its lexical meanings or some other lexical unit, such as a set expression). Y is often represented by a set of synonymous lexemes Y_1, Y_2, …., Y_n, all of them being the values of the given LF R with regard to X; e.g.,

MAGN (*desire*) = *strong / keen / intense / fervent / ardent / overwhelming*,

where MAGN is a LF for which the underlying semantic relation is 'high degree'.

Two major types of lexical functions are distinguished – paradigmatic LFs (substitutes) and syntagmatic LFs (collocates).

Substitute LFs are those which replace the keyword in the given utterance without substantially changing its meaning or changing it in a strictly predictable way. Examples are synonyms, antonyms, and converse terms. A special subclass of substitute LFs is represented by various types of derivatives of X (nomina actionis, as in *to encourage – encouragement*, typical agents, as in *to build – builder* or *to judge – judge*, typical patients, as in *to nominate – nominee*, *to teach – student* and the like). All such substitute LFs play an important role in paraphrasing sentences within our generator. For example: *She bought a computer for 500 dollars from a retail dealer – A retail dealer sold her a computer for 500 dollars – She paid 500 dollars to the retail dealer for a computer – The retail dealer got 500 dollars from her for a computer.*

Collocate LFs are those which appear in an utterance alongside the keyword. Typically, such LFs either dominate the keyword syntactically or are dominated by it, even though more elaborate syntactic configurations between the keyword and an LF

value are not infrequent. Typical examples of collocate LFs are adjectival LFs, such as the already mentioned MAGN, or support verbs of the OPER / FUNC / LABOR family. LFs of the latter type play a leading role in the paraphrasing system, providing paraphrases like

(1) *John respects his teachers* – (2) *John has respect for his teachers* – (3) *John's teachers enjoy his respect* – (4) *John treats his teachers with respect.*

In this series of synonymous sentences, we have the verb *respect 1* in (1), the noun *respect 2* which is the value of substitute LF S_0 (nomen actionis) for the verb *respect 1* in (2), (3) and (4), the value of LF $Oper_1$ *have* for the noun *respect 2* in (2), the value of LF $Oper_2$ *enjoy* for the noun *respect 2* in (3), and the value of LF $Labor_{12}$ *treat* for the noun *respect 2* in (4).

In a very simplified form, rules of paraphrasing can be represented as

$X \Leftrightarrow Oper_1 + S_0(X)$, where the subject of X is inherited by $Oper_1(X)$ and the first object of X becomes the first object of $S_0(X)$;

$X \Leftrightarrow Oper_2 + S_0(X)$, where the subject of X becomes the first object of $S_0(X)$ and the first object of X becomes the subject of $Oper_2(X)$;

$X \Leftrightarrow Labor_{12} + S_0(X)$, where the subject and the first object of X are inherited by $Labor_{12}(X)$, and $S_0(X)$ appears as the second object of $Labor_{12}(X)$.

Using these rules, the paraphrase generator produces equivalences like *The United Nations ordered Iraq to write a report on chemical weapons – the United Nations gave Iraq an order to write <submit, prepare, make, produce,… > a report on chemical weapons – Iraq was ordered by the United Nations to write <submit, prepare, make, produce,… > a report on chemical weapons – Iraq received an order from the United Nations to write <submit, prepare, make, produce,… > a report on chemical weapons* (in this case, two elements of the sentence are subject to paraphrasing: *order* and *report*, while the paraphrasing diversity is increased because the LF $Oper_1$ for the argument *report* has several values).

Here are some other paraphrasing rules used by the generator, accompanied by examples.

$X \Leftrightarrow Copul + S_1(X)$
He taught me at school – He was my teacher at school.

$X \Leftrightarrow Func_0 + S_0(X)$
They are arguing heatedly – A heated argument between them is on.

$X \Leftrightarrow Func_1 + S_0(X)$
He is afraid – Fear possesses him.

$IncepOper_1 + S_0(X) \Leftrightarrow IncepOper_2 + S_0(X)$
He conceived a dislike for her – She caused his dislike.

$FinOper_1 + S_0(X) \Leftrightarrow FinOper_2 + S_0(X)$
England lost control of this territory – This territory went out of England's control.

$LiquOper_1 + S_0(X) \Leftrightarrow LiquOper_2 + S_0(X)$
The government deprived the monopolies of control over the prices – The government took the prices out of the monopolies' control.

$LiquOper_1 + S_0(X) \Leftrightarrow LiquFunc_1 + S_0(X)$
We freed him of this burden – We lifted this burden from him.

$X \Leftrightarrow IncepOper_1 + S_{res}(X) \Leftrightarrow IncepFunc_1 + S_{res}(X)$.
He learned physics – He acquired the knowledge of physics.

$X \Leftrightarrow CausOper_1 + S_{res}(X)$ etc
He taught me physics – He gave me the knowledge of physics.

$LiquOper_1 + S_{init}(X) \Leftrightarrow LiquFunc_1 + S_{init}(X)$ etc.
A sudden bell woke him up – A sudden bell interrupted his sleep.

$CausFact_0\text{-}M + X$ / $CausFact_1\text{-}M + X$ / $CausReal_1\text{-}M + X \approx IncepFact_0\text{-}M + X$ / $IncepReal_1\text{-}M + X$ etc.
They sent him on leave for a few days – He went on leave for a few days.

$LiquFact_0\text{-}M + X$ / $LiquFact_1\text{-}M + X$ / $LiquReal_1\text{-}M + X \approx FinFact_0\text{-}M + X$ / $FinReal_1\text{-}M + X$ etc.
He was deprived of his last chance to win in this event – He lost his last chance to win in this event.

$Anti_1Fact_0\text{-}M(X) + X = negFact_0\text{-}M(X) + X$ etc.
The plans of pacifying the aggressor failed – The plans of pacifying the aggressor did not succeed; The hypothesis of the pulsing Universe was disproved – The hypothesis of the pulsing Universe was not confirmed.

$Anti_1Real_1\text{-}M(X) + X \Leftrightarrow negReal_1\text{-}M(X) + X$ etc.
The board of directors declined the compromise – The board of directors did not accept the compromise, The champion let slip his advantage – the champion did not use the advantage.

$Anti_1Real_2\text{-}M(X) + X = negReal_2\text{-}M(X) + X$ etc.
He swallowed up the insult – He did not avenge the insult, The whole group failed the examination – The whole group did not pass the examination.

$Anti_1Real_3\text{-}M(X) + X \Leftrightarrow negReal_3\text{-}M(X) + X$ etc.
The lecturer ignored the questions of the audience – The lecturer did not answer the questions of the audience, He neglected my advice and smarted for it – He did not follow my advice and smarted for it, Any soldier who violates the order is subject to court martial – Any soldier who does not obey the order is subject to court martial.

A paraphrasing system of this kind requires a good lexicographic source from which the appropriate LF values of words could be extracted. Such a source is provided by the combinatorial dictionary: two such dictionaries, for English and Russian, are available at the Laboratory of Computational Linguistics as part of the multipurpose linguistic processor, ETAP-3. The principal tools of establishing semantic links between words in the combinatorial dictionary, in addition to LFs, are semantic features that refer the word to a specific semantic class, and argument frames, or government patterns, which establish semantic links between predicates and its

arguments. In all, the two combinatorial dictionaries make use of over 120 LFs, each of which corresponds to a specific semantic relation of a universal nature, about 60 partially hierarchized semantic features, and tens of thousands argument frames, individually created for all classes of predicate words – verbs, nouns, adjectives, adverbs, and prepositions.

3 The Search Query Paraphrasing Module

Though the paraphrasing system outlined in Section 2 was primarily built as a practical implementation of the Meaning ⇔ Text linguistic theory, it proved promising in an experiment staged to check whether such a system can be used to increase the precision of information retrieval. To adapt the paraphrase generator to the needs of search engine optimization, we have slightly modified it so to obtain a new module, called Search Query Paraphrasing Module, which only works with structures that contain quantitative information. It does not generate a new sentence from the source sentence; instead, it produces a set of incomplete sentences which lack only the numerical data from a noun phrase with the parametric keyword as a master. Quantitative information is represented by different measure scales or properties which need to be measured. The names of these scales form a lexical class which we call parametric words: *height, capacity, volume; duration, age; power, mass, pressure; rate* (as in *birth* or *death rate*); *price, value; entropy, level, coefficient, index,* etc. This class constituted the experimental material, for the search query paraphrasing module.

In Russian, the rules of lexical semantics that describe the behavior of parametric words are extremely strict. This strictness is a specific feature of the language: even though numerical properties are universal predicates, their prototypical representations are nouns rather than verbs [13]. Such verbs as *stoit'* ('to cost'), *vesit'* ('to weigh'), *dlit'sja* ('to last'), *vmeščat'* ('to hold') are few. There is no verb which could be used to attribute a height to an object – we cannot naturally translate a simple English sentence *The Pisa tower rises 56 meters* into Russian retaining its syntactic structure. A routine way would be to use a support verb with the parametric word*: Pizanskaja bašnja dostigajet v vysotu 56 metrov* 'The Pisa tower reaches 56 meters in height'.

The structure of this sentence can be represented in terms of LFs: *dostigat'* 'reach' = $Labor_{12}$(*vysota* 'height'). Since, as stated above, there is no Russian verbal correlate of *vysota,* one cannot paraphrase this sentence on the basis of the rule X ⇔ $Labor_{12}$ + S_0(X). Instead, one can use it to deduce new equations.

Specifically, from three equations (X ⇔ $Oper_1$ + S_0(X), X ⇔ $Func_2$ + S_0(X), X ⇔ $Labor_{12}$ + S_0(X)) we get the following rules, in which three lexical functions are used:

$Oper_1$ + X ⇔ $Func_2$ + X
$Oper_1$ + X ⇔ $Labor_{12}$ + X
$Func_2$ + X ⇔ $Labor_{12}$ + X

With these rules, the module can perform the following transformations:

-*imet' glubinu* 'reach a depth'⇔ *glubina sostavljajet* 'a depth reaches';
-*imet' glubinu* 'reach a depth' ⇔ *imet' v glubinu* 'reach to the depth';
-*glubina sostavl'ajet* 'a depth reaches' ⇔ *imet' v glubinu* 'reach to the depth'.

If a lexical function has several values, the paraphrasing module generates all possible variants. To give an example, for a noun phrase like *glubina Marianskoj vpadiny* 'the depth of the Mariana trench', the module produces a host of paraphrases:

-*glubina Marianskoj vpadiny ravna* 'The depth of the Mariana trench is equal to';
-*glubina Marianskoj vpadiny sostavljaet* 'The depth of the Mariana trench amounts to';
-*glubina Marianskoj vpadiny dostigajet* 'The depth of the Mariana trench reaches';
-*glubina Marianskoj vpadiny ravnjaetsja* 'The depth of the Mariana trench equals';
-*Marianskaja vpadina imeet v glubinu* 'The Mariana trench has in depth';
-*Marianskaja vpadina dostigajet v glubinu* 'The Mariana trench attains in depth';
-*Marianskaja vpadina imeet glubinu* 'The Mariana trench has the depth';
-*Marianskaja vpadina dostigajet glubiny;* 'The Mariana trench reaches the depth'.

We can also use this module to get the whole set of paraphrases from any of the above paraphrases.

This example shows that the same verb can serve as value for different lexical functions, but it is not necessarily so. The theory predicts that words belonging to one semantic class (as the parametric words do) will have similar values of lexical functions. But it is also true that real language systems have many exceptions, especially in the domain of the lexicon. So, the verbs representing values of lexical functions can vary from one parametric word to another and form a unique co-occurrence area of the word.

For example, Russian word *moščnost'* 'power' has only $Oper_1$ and does not have $Labor_{12}$, but the set of $Oper_1$ values is richer than the same set for other words. Beside common trivial values *imet'* 'to have' and *dostigat'* 'to reach' we find a new verb *razvivat'* 'to develop', which co-occurs only with two parametric words, *moščnost'* 'power' and *skorost'* 'speed'. Interestingly, this co-occurrence rule can be literally translated into English.

As stated above, the difference between the standard paraphrasing system and the search query paraphrasing module lies in the completeness of both input and output structures. There are two more rules in the search query paraphrasing module that, first, expand the noun phrase to a whole sentence by adding a trivial verbal value of $Func_2$ and a temporary formal object, and, second, delete the formal object. Once the complete sentence is built, the module works in a standard way. The chain of transformations looks like this:

(a) *glubina Marianskoj vpadiny* 'the depth of Mariana trench'
(b) *glubina Marianskoj vpadiny ravnjajetsja čemu-to*
'the depth of Mariana trench equals something'
(c) *Marianskaja vpadina* [$Oper_1$] *glubinu čto-to* 'Mariana trench [$Oper_1$] the depth of something'
(d) *glubina Marianskoj vpadiny* [$Func_2$] *čemu-to* 'the depth of Mariana trench [$Func_2$] something'
(e) *Marianskaja vpadina* [$Labor_{12}$] *čto-to v glubinu* 'Mariana trench [$Labor_{12}$] something in depth'

The algorithm begins to delete the "empty" complement only after it has generated all possible paraphrases. It seems excessively complicated when we look at the linear

structure, and a question suggests itself: could we simply add a verb without adding and removing any "empty" complements? The answer is: no, we could not. We must take into consideration that such transformations occur at the deep syntactic level where language information is represented as a tree. When we produce sentence (c) from sentence (b) we do not actually replace one word in the word chain with another word, but we transform the marked branches of the tree, and it is important not to miss any features like case or gender. For example, we must know that in (c) the verb $Oper_1$ is bound to the word *glubina* 'depth' by the predicative relation to choose the correct gender if we want to transpose the sentence in the past tense. We must keep in mind that dependencies are very important when we deal with a syntactically rich language like Russian.

For the sake of completeness, we should also discuss one possibility we already mentioned. Some queries need to be transposed in the past or future tense, for example, *vodoizmeščenie "Titanika"* 'the displacement of Titanic', *vmestimost' kosmičeskogo korablja Orion* 'carrying capacity of the spacecraft Orion'. There is one more difficulty of the same kind in Russian - the opposition of two grammatical aspects. We can treat this problem in two ways. First, we can expand the algorithm and make it generate variants with all possible sets of grammatical features. It will significantly increase the number of paraphrases and could become a serious disadvantage if we try to apply the paraphrase generator in the real search engine. Another way to treat this problem is to select the verbs or at least morphological features when preparing the queries. In this case the system will keep all the features. As a matter of fact, this is the way our system functions now.

We already mentioned that all transformations of a query occur at the deep syntactic level. This level is deep enough to provide a translation from one language to another. It follows that deep syntactic structures may be considered, with some natural reservations, as an invariant for different languages. So we tried to get English paraphrases from the same deep syntactic structures using lexical functions. For the input *vysota Pizanskoj bašni* 'the height of the Pisa tower', the module generated the following sets of paraphrases:

The height of the Pisa tower equals
The height of the Pisa tower reaches
The height of the Pisa tower amounts to
The height of the Pisa tower attains.

4 Evaluation

We made a list of 100 short Russian queries (a parametric word and a subject of property, such as *glubina Marianskoj vpadiny* 'the depth of Mariana trench'). The relevance of these queries was confirmed by the data on query statistics provided by the Russian search engine Yandex [14]. The search query paraphrasing module generated a set of paraphrases for each query. During the experiment, these paraphrases were offered one by one to Yandex.

Since we were interested to find out if a possibility exists to improve search precision, we disregarded the time of query execution and the time of query processing by the system ETAP-3. The same concerns also the date when the experiment was

carried out and the load of the search engine servers. Since the paraphrasing module yields the holistic structures, not going beyond the sentence limits and without lacunas and omissions, we chose the form of a precise query for testing. The goal of the experiment was to learn how significant the increase of question-answering precision could be. We used the following estimation protocol.

A result was recognized as relevant when numerical information appeared in the snippet proposed by the search engine. We did not verify if the answer was factually correct or not.

The first result containing numerical information received the MMR (mean reciprocal rank) grade. If we consider that several paraphrases generated from one noun phrase work for one query and their results present a kind of entity, it appears reasonable to calculate the MRR for one set of paraphrases, and only then calculate the MRR for all paraphrases. If, for example, two paraphrases returned the answer in the first snippet, and other six did not find anything at all, then we consider only two paraphrases and take MRR for this query to be equal to 1, because the imaginary question-answering system will have a result to show. In such a case the MRR of search for queries in Russian is 0.683. For comparison, the MRR of search for the queries fed to the search engine without paraphrasing is 0.503.

Is this MRR difference promising for our paraphrasing module? It should be admitted that the data are rather difficult to interpret: the paraphrases do not return any answer in a greater number of cases than in the rough queries. This is the price we have to pay for the query precision.

We believe that if we convert a paraphrased query into a less demanding form, which requires the appearance of all words and their correct sequence, but allow the paraphrase to be split by one “foreign” word in between, the MRR will be noticeably higher.

Nevertheless, our data indicate that the precision of search using linguistically enriched queries is very high. Among the queries with paraphrases yielding the answer there was only one case where not a single paraphrase brought any answer in the first snippet. In all the remaining cases at least one paraphrase gave an answer in the first snippet.

A similar experiment has been carried out with the English data. The English input was parsed and paraphrased, and the paraphrases were sent to the Google search engine. However, this experiment was not as successful as the Russian one. The paraphrases obtained could not improve the search – in many cases the engine did not return any answer. It can be accounted for by the specific feature of English, in particular by the well-known fact that in English properties are usually expressed with adjectives meaning a high degree of a quality (*30 feet high, 6 feet tall, 25 years old*). It is more natural for English speakers to attribute qualities with structures like this: *The Cupola is 55 meters high and 16 meters wide.* To process English queries properly, we needed some extra paraphrasing rules not bound to lexical functions. These rules are under development now. We can quite easily transform the noun phrase *the height of the Statue of Liberty* into the incomplete sentence *The Statue of Liberty is ... high* using the same principle that we described for Russian paraphrasing, but other parametrical words require special treatment.

Our search query paraphrasing module is a closed-domain query-answering system. As compared with other query-answering systems, it has both advantages and drawbacks.

Advantages:

1) Though we use sophisticated linguistic data the algorithm is quite simple because it functions in a rich and well integrated multifunctional NLP environment.
2) The whole World Wide Web can be used as a source of data. Our algorithm needs no special linguistic markup of document collections.
3) We need no knowledge bases except our own, i.e. the combinatorial dictionaries, and they also may be incomplete – the algorithm will process the query even if it cannot recognize some of the words, for example the word *Cotopaxi* in the query *the height of the volcano Cotopaxi.*

Drawbacks:

1) The application domain is limited to certain types of data.
2) The efficacy of the module is strongly dependent on document collections, like the efficacy of all query-answering systems.

Considering the fact that the development of the module claimed minimal effort we could conclude that the positive result of evaluation experiment is encouraging.

5 Conclusion

Even though empirical methods are widely used in modern information processing, they cannot solve all the problems alone. The optimal decision would be to unite the advantages of empirical and heuristic methods into an integrated NLP system. In particular, in the context of information extraction, the patterns used are specific for every subject domain and they should probably be produced by machine learning methods, whilst paraphrase clusters in patterns should be rather built with uniform rules of paraphrasing in terms of lexical functions, so it is more beneficial to obtain them through heuristics.

We believe that the creation of a full-scale lexicographic resource populated with values of lexical functions will be a useful step in the direction of semantics-oriented natural language processing in a variety of applications.

References

1. Yangarber, R.: Acquisition of domain knowledge. In: Pazienza, M.T. (ed.) SCIE 2003. LNCS (LNAI), vol. 2700, pp. 1–28. Springer, Heidelberg (2003)
2. Lin, W., Yangarber, R., Grishman, R.: Bootstrapped Learning of Semantic Classes from Positive and Negative Examples. In: Proceedings of the 20th International Conference on Machine Learning: ICML 2003 Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, Washington, D.C (2003)
3. Shinyama, Y., Sekine, S.: Paraphrase Acquisition for Information Extraction. In: The Second International Workshop on Paraphrasing: Paraphrase Acquisition and Applications (IWP2003), Sapporo, Japan (2003)

4. Sekine, S.: Automatic Paraphrase Discovery based on Context and Keywords between NE Pairs. In: Proceedings of the International Workshop on Paraphrase 2005, Jeju Island, Korea (2005)
5. Sekine, S.: On-Demand Information Extraction. In: ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, July 17-21 (2006)
6. Mel' čuk, I.A., Žolkovskij, A.K.: Tolkovo-kombinatornyj slovar' sovremennogo russkogo jazyka. In: Opyt semantiko-sintaksičeskogo opisanija russkoj leksiki, Wiener Slawistischer Almanach, Wien (1984)
7. Apresjan, J.D.: Izbrannye trudy. Leksičeskaja semantika. Sinonimičeskie sredstva jazyka. Jazyki slavjanskix kul' tur, Moscow (1995)
8. Mel' čuk, I.: Lexical Functions: A Tool for the Description of Lexical Relations in a Lexicon. In: Wanner, L. (ed.) Lexical Functions in Lexicography and Natural Language Processing, Amsterdam, Philadelphia, pp. 37–102 (1996)
9. Mel' čuk, I.: The Meaning-Text Approach to the Study of Natural Language and Linguistic Functional Models. In: Embleton, S. (ed.) LACUS Forum, vol. 24, pp. 3–20. LACUS, Chapel Hill (1998)
10. Mel' čuk, I.A.: Opyt lingvističeskix modelej "Smysl <=> Tekst". Semantika, sintaksis. Shkola Jazyki russkoj kul'tury, Moscow (1999)
11. Apresjan, J.D., Cinman, L.L.: Formal' naja model' perifrazirovanija predloženij dlja sistem pererabotki tekstov na estestvennyx jazykax. In: Russkij jazyk v naučnom osveščenii, vol. 4, pp. 102–146 (2002)
12. Apresjan, J.D., Boguslavsky, I.M., Iomdin, L.L., Cinman, L.L.: Lexical Functions in Actual NLP Applications. In: Wanner, L. (ed.) Selected Lexical and Grammatical Issues in the Meaning-Text Theory. In Honour of Igor Mel' čuk, pp. 199–230. Benjamins Academic Publishers, Amsterdam (2007)
13. Apresjan, J.D.: Osnovanija sistemnoj leksikografii. In: Jazykovaja kartina mira i sistemnaja leksikografija. Škola Jazyki russkoj kul'tury, Moscow (2006)
14. Search query statistics of Yandex, http://wordstat.yandex.ru/