INFORMATION RETRIEVAL SYSTEM FOR NEWS ARTICLES IN RUSSIAN

Alexandr Zharikov, Konstantin Kristalovsky, Vasiliy Pivovarov
Interfax
e-mail: alexander.zharikov@interfax.ru,
konstantin.kristalovskiy@interfax.ru, vasiliy.pivovarov@interfax.ru

Abstract

We present a description of the natural language processing system developed for information retrieval project scan.interfax.ru. The system should process news articles in Russian and retrieve as much information as possible about persons, organizations or other text objects mentioned there. The conceptual system idea is to process and "understand" documents without using any time dependent named object databases. The system should retrieve and interpret person names, organizations, geography and some other text structures. And only on the second stage — to make identification of retrieved named objects via database. The resulting information is used in Scan project to allow complex-query search of news articles, to fill a named object database in automatic manner and to provide information for analytical services. We discuss functionality of the system, main approach ideas used and challenges to be resolved in future work. The article mostly covers entity extraction procedure and fact extraction mechanics. The problems of geography entity extraction are discussed particularly.

Keywords: *computational linguistics, natural language processing, information retrieval.*

1. INTRODUCTION

We present the natural language processing system aimed to provide information necessary for a search engine with complex query language, like "find all news with «Organization Name» and «Person Name» mentioned", and for analytical services based on mentioning count analysis for almost arbitrary text objects types. The content of documents to be processed is predetermined by news articles written in Russian. The system is applied in project www.Scan.Interfax.ru.

2. RELATED WORK

The conceptual system idea is to extract as much information as possible from document without using any time dependent named object databases (fig. 1). The ideal system behavior we are trying to reach should be similar to a human, who reads a news article written in his mother language, but with a subject he is absolutely unfamiliar with. He does not know geography of mentioned regions, he has never heard about people mentioned in the article, but his language skills allow him to understand almost everything. And, if interested, after reading he will query Google to know more about people or companies mentioned. So the system does: only after document "understanding" it will query a highly

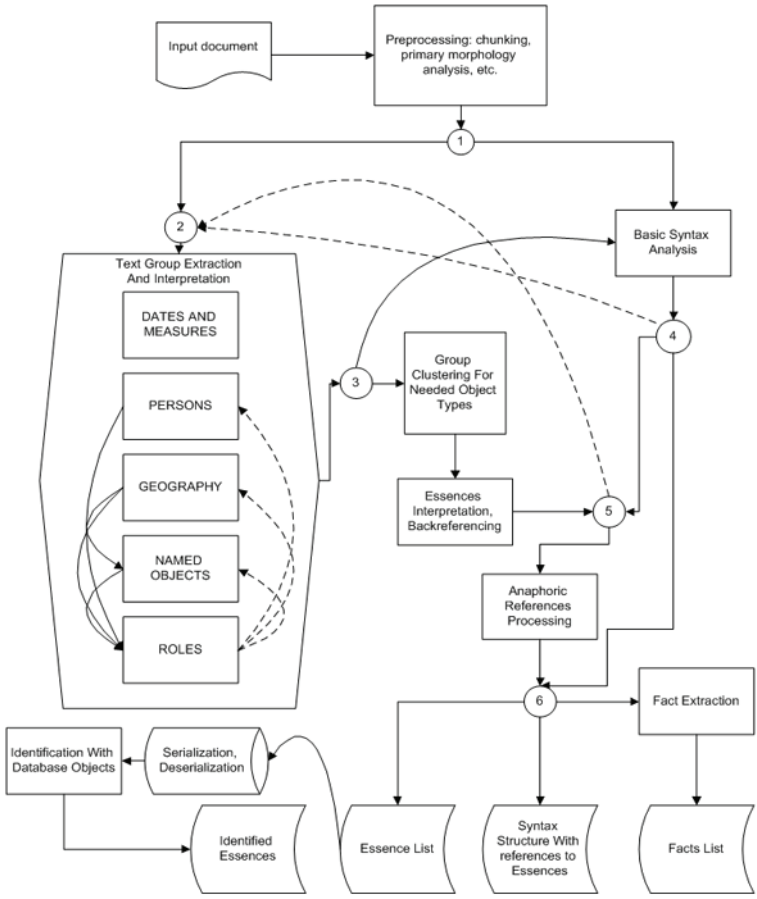

Figure 1. Document processing stages. Dotted lines refer to backreferencing

temporal-dependent database to identify named objects. And if the unresolved object can be treated as interesting by the system, it should be added into the database in automatic way (though human verification is not unneeded yet). Such "interesting" objects are good qualified persons (with full name and a role given) or fully named companies or such.

2.1. Preprocessing

Preprocessing includes sentence chunking, tokenization, basic morphology marking (influenced by AOT morphology tables [1]) and querying

semantic dictionaries, which include geography dictionary (OKATO database with some manual extensions), person name forms dictionary and the "homemade" qualifier dictionary (see below). On that stage we also use some statistical algorithms for entity extraction. The system has a long history and on its first stages entity extraction was mostly statistical, based on Markov Chains approach. But the quality was not sufficient, and now we use statistical approaches only to make first estimates for positioning person name objects in text. The usage of deterministic algorithms allowed us to improve precision/recall rates from approximately 75/75% up to 98/98%.

2.2. Named Entity Extraction

The extremely important role in our entity extraction approach is played by qualifiers semantic dictionary. Qualifiers, like "правительство", "глава", "президент", are markers of a particular text object categories. Our methods are based mostly on expanding from qualifiers to text objects using their dictionary properties.

Fig. 2. presents our results in a format based on a well known MUC-6 with some inner changes and shortening for better readability. Each essence corresponds to single object: in ideal case all mentions of an object in a text should refer to its single essence. Syntax structure is represented by a list of predicates with subjects and target objects.

2.3. Geography entity extraction

Our entity extraction procedure, as it was mentioned, is mostly based on "type qualifiers". The logic of extraction for objects like organization is quite complex but rather distinct. Most of organization names can be extracted using "qualifier chains". For example organization named «бухгалтерия московского филиала завода "Красные Копыта"» contains three qualifiers and can be divided into three chains (combinations of these chains give birth to three essences). It is widely known approach. The similar logic can be applied to a various non-organization objects, but our attempt to use it in geography entity extraction leaded to surprising results. The logic of geography chain extraction turned to be much more obscure. We present several interesting examples below.

Consider two geography names (capitalization can be arbitrary):

Name1=«Донецкий район Украины» and Name2=«Индийский город Дели».

These names are syntactically similar and, furthermore, have quite similar morphology, but they contain different number of essences: Name1 contains Name3=«Донецкий район» and Name4=«Украина», while

```
<ENAMEX T=»DATE» norm=»01.10.2010-31.10.2010»>В октябре
прошлого года</ENAMEX> премьер-министр <ENAMEX T=»GEO»
E=»0»>РФ </ENAMEX> <ENAMEX T=»PERSON» E=»1»>В. Путин</
ENAMEX> подписал ряд указов.… <ENAMEX T=»PERSON»
E=»1»>Глава </ENAMEX> <ENAMEX T=»ORG» E=»2»>российского
правительства </ENAMEX> подписал Указ «О Помиловании».
… «Этот указ позволит развиваться…» - заявил <ENAMEX
T=»PERSON» E=»1»>он</ENAMEX>.
<ESSENCE E=»0» T=»GEO» NAME=»Российская Федерация»></
ESSENCE>
<ESSENCE E=»1» T=»PERSON» name=»В. Путин»
role=»премьер-министр, GEO =0»; role =»Глава, ORG=2»
<ESSENCE E=»2» T=»ORG» T=»правительство»
name=»российское правительство»> GEO=»0»</ESSENCE>
<VERB predicate=»подписал» subject=»премьер-министр РФ
В. Путин (essences: 1)» object=»ряд указов» />
<VERB predicate =»подписал» subject=»Глава российского
правительства (essences: 1)» object=»Указ «О
Помиловании»» />
<VERB predicate =»позволит развиваться» subject=»Этот
указ» />
<VERB predicate =»заявил» subject=»он (essences: 1)»/>
```

Figure 2. Xml output of document processor (essence list and basic syntax structure). “ENAMEX” tag refer to entity (text object group), ENAMEX attribute “T” denotes entity type, “E” — essence number.

Name2 can only be shortened to itself («Дели»). This property of extraction is defined by qualifiers, and is specific geography feature.

Suppose geo dictionary does not contain Name2 and Name3 (it is typical situation due to our dictionary includes only major world objects outside Russia). For Name2 human intuitively extracts «Индия» as holonym (father) essence – the system can extract it too. For Name3 the essence «Донецк» can be derived using the same logic, but for now it turns to be meronym (child) essence (from a geo location point of view). Let’s suppose it is also a qualifier property. But there exist Name5=«Тверской район Новгорода», which has equal structure with one of Name1. The only difference is that Name5 refer to a city and Name1 to a country, but it is not clear from context and underivable without dictionary. And our logic of extraction name «Тверь» from Name5 leads to crucial failure.

After examples mentioned, it seems that a full geo dictionary can be very useful. We have made it for Russian geography, but a lot of new problems aroused. Now we have hundreds of cities and 10 times more villages with

names similar to commonly used words, like «Северный», «Центральный», «Лесной» and regions («район» qualifier) with the same names. The brute force document marking via geo dictionary will mark about a quarter of it. Nevertheless, the geographic dictionary is the only one object dictionary we still use in document processing stage. In our "global" ideology it corresponds to basic geography knowledge of a news-reader person.

We can add to this logical maze several external cases like: 1) correct capitalization is not always provided; 2) phrase polysemy: for example, fragment «в районе Дели» can mean "in the region called Delhi" as well as "somewhere near Delhi" or "in one of districts of Delhi"; furthermore «Дели» without capitalization is a commonly used verb; 3) unknown geo names are often followed by person names «мэр города Ху Лин Сяо» and the correct separation is not always evident.

After all, we can see that perfect geo entity extraction is not possible without very complex and detailed logic, backreferencing from syntax analyzer and from interpretation results of other type entities (from node 5 in Fig. 1). The backreferencing is also necessary for all other entity types as long as they are syntactically connected to each other: the organization entities can include geography and role entities include both of them («глава администрации президента РФ»).

We have also made attempt to create procedures, which can extract father geography terms from demonyms (like «москвич») and "geo adjectives" (like «московский»). We found, that in case of geo adjectives that procedure can provide reliable results only if the father term can be verified by geo dictionary. The results of demonym processing are more reliable, though unexpected interpretations also take place (for example, extraction of «Вена» and «Куба» from fragments «снять венок» or «поднять кубок»).

2.4. Entity clustering

After entities in text are revealed and interpreted we make clustering to form a list of essences. These are geography essences, organizations and persons. The clustering stage is especially interesting for the last ones. First of all it is because of short person naming, which is typical natural language feature.

Figure 3 presents typical situation, which is not resolvable without strong limitations of generality (like "«Янукович» can never be patronym") or without much deeper context interpretation (person naming in form "Name Patronym" is possible mostly in a specific context like direct speech). The clustering procedure for persons is also not standard. The metrics for person objects is non-Euclidian and clustering procedure

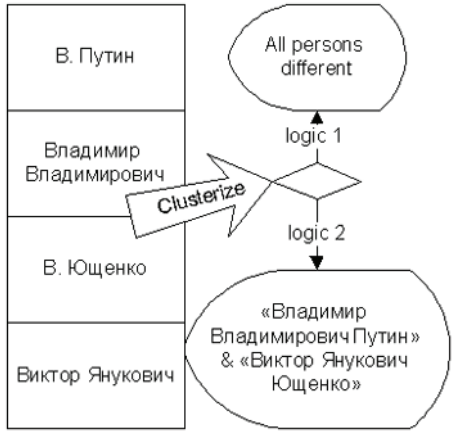

Figure 3. Person cluster challenge. Both possible logic schemas produce incorrect result

should include conflict-resolving algorithms. In fig. 4 it is shown how stable links between the first three essences can become unstable if the forth one is added.

2.5. Anaphoric references identification

The anaphoric references processing (identification) approaches are of two main types. The first one is based on reliable information presented in qualifiers and geo dictionary. The simplified logical scheme for identification of «Russian government leader» with «prime-minister of Russia V. Putin» is presented in fig. 5. The second type of anaphoric links is situation, when fully reliable conclusion is not derived from semantics. The example of that type is pronoun identification (see fig.2) or situation presented in fig. 5 with geography markers excluded.

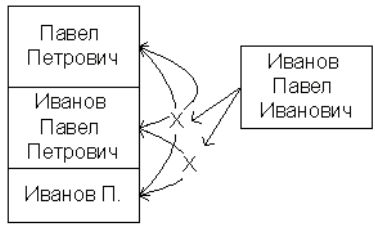

Figure 4. Discrete non-Euclidian nature of person objects. X-arrows refer to unstable links

Figure 5. Reliable logic scheme for anaphoric link identification

2.6. Syntax analysis and application

Our syntax revealing algorithms are empiric and mostly deterministic, and they are on the first stages of development. In cases of ambiguous interpretation we use a priori information on news content specifics. For example, fragment «в районе Южного Полюса нашей планеты» has unresolvable ambiguity (non evident variant is «нашéй»), but specific content allows us to ignore vocatives in non-speech phrases. Correct syntax structure allows us to boost named and anaphoric entities extraction. We are also planning to use it for fact extraction.

3. FACT EXTRACTION

It's well known that the rules for extraction of variety of interesting objects in a text (e.g., noun phrases, date groups etc.) could be effectively represented in the form of regular expressions over the annotated text [2]. Such the regular expressions could check the agreement between the members of text structure (e.g., to verify coincidence in the cases, numbers and genders etc.)

The similar technique could be in many situations applied to the more complex text objects. It's enough for that to supply the annotated text with the appropriate meta objects and to append necessary additional attributes to the affected low-level text items. Such the meta objects could correspond to the named entities etc.

The annotated text generated in such a manner becomes the substantially equivalent to the "semantic network" (in terminology of [4]), but remains the flat structure of the ordinary text. Hence, its processing could be achieved without usage of the algorithms of tree navigation, which verify certain conditions in the tree nodes. The practically important difference between the cases is that the rules of tree navigation should be verified sequentially whilst for the rules represented in the form of large

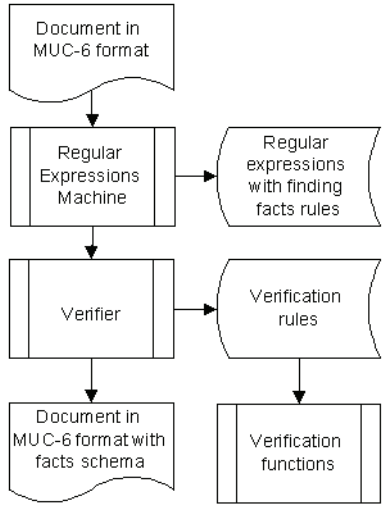

Figure 6. Fact Extraction scheme

regular expression with the plenty of alternatives there is well-known formal way to obtain all the results in the single pass. So, the latter representation is preferable though isn't always possible.

In fact, the processing rules in our system are structured in the form resembling the decision tree method (fig. 6). The rule given could have some child rules, which processing is performed in the case of success of the parent node. These child rules could have some arguments depending on the result of the match of the parent rule. The results generated by the rules are ordered by priority. The rules from the different subtrees could exchange their results through the document model. The result of serialization of the document model is the annotated text itself. To hide from the user the internal details of annotated text generation, the regular rules are written in the specialized language, which allows to compose rules with the stress to the semantics of the action and various arithmetic and bitwise conditions to be checked. The rules are now composed manually.

While stepping over the tree levels, optional additional actions could be performed: the conditions which could hardly be represented by the regular rules could be checked and the processing output could be generated (fig. 7). Such the actions address the document model in the simple functional style.

So, all the regular rules with the same path prefix in the tree could

```
<s FACT=»0»><ENAMEX TYPE=»ORG» E=»0»>Парламент
Армении </ENAMEX> принял в понедельник в первом чтении
изменения и дополнения в закон «Об адвокатуре»…</s>
<ESSENCE E=»0» T=»ORG» TYPE=»Парламент» NAME=»Парламент
Армении»>
<FACT FACT=»0» NAME=»Законотворчество» SUBJ=»0»>
```

Figure 7. Xml output with a fact extracted

be processed simultaneously. It should be noted, though, that the majority of the modern regular expression (RE) engines don't allow to gain the benefits resulting from this possibility. Their implementation is similar to the well-known Perl engine and doesn't support the simultaneous pass through the multiple alternatives. Their parsing speed is inverse proportional to the number of alternates; moreover, in the standard (Perl) mode the machine could systematically lose the long right alternatives. The alternative (POSIX) mode is 1–2 order slower. So, the specialized machine which implements partially compiled DFA with the delayed build is used. The special precautions are taken to preserve the possibility of using of the tagged-groups similar syntactic constructions. The main disadvantage of such machines — rather high memory consumption [3] — is in this case compensated by the much higher performance (with the smoothed dependence on the number of rules) and by the possibility to easily obtain all the matches including overlapped ones. The latter allows to implement processing of the uniform lists in particular.

4. PROBLEMS AND CHALLENGES

1. Deep syntax analysis and reliable anaphoric links processing for all text object types.
2. Person name extraction, interpretation and clustering.
3. More accurate entity identification via database, using supervised learning, automatic database filling.
4. Automatic qualifier dictionary making.
5. Refining homonymy resolution algorithms (especially for geography terms).
6. Fact extraction based on syntax sentence structure.
7. Ontology interpretation of text objects and facts retrieved.

5.ACKNOWLEDGMENTS

Special thanks to Interfax IT-center team.

REFERENCES

1. **Сокирко А.В.** Синтаксический анализ // AOT:: Технологии :: Синтаксический анализ: http://www.aot.ru/docs/synan.html (2005).
2. **Ножов И.М.** Морфологическая и синтаксическая обработка текста (модели и программы). Дисс. на соискание ученой степени канд. техн. Наук, 2003.
3. **Ахо, Сети, Ульман,** "Компиляторы: Принципы, Технологии, Инструменты", Москва, «Вильямс», 138
4. RCO Fact Extractor SDK (документация), http://rco.ru/product.asp?ob_no=5047&part=docs