

Complex Data Mining Algorithm, Based on Logical Deduction Rules

Denis V. Krayushkin
Sytech, Russia
kraus@sytech.ru

Elena B. Kozerenko
Institute for
Informatics Problems
of the Russian
Academy of Sciences
kozerenko@mail.ru

Abstract

This article considers a method of data mining from text documents by using logical deduction rules. This algorithm is used in information processing of heterogeneous sources. The method allows to mark out a set of data objects from a flow of semistructured full-text information, presented in natural languages, links between data flows, and to form a repository of factual information. This algorithm includes procedures of grapheme, morphological, syntax and logical-semantic analysis. The method has a software support and practical application in automated information systems and is used in accomplishing tasks of analytical data treatment.

Introduction

Nowadays the systems of processing semistructured documentary information are considered to be an essential instrument of analytical departments in many organizations. A lot of algorithms are elaborated for data processing. They allow revealing different patterns in flows of documentary information. In spite of the fact that the general principles of data processing in natural language are common for different systems, the concrete methods, used in them, have their specific features.

This article observes a technology of processing documentary information, which is used in “ARION” system. It allows picking out data objects, their characteristics and information about them in documentary texts automatically. The particularized method of forming rules for data processing which is used in this system allows forming rules for text processing in different languages (such as Russian, English, German, French and Spanish). Linguists with proper qualification skills are required to solve this problem. These skills include knowledge of language structures and peculiarities of different languages. Developers are also needed to transform knowledge, got from linguists, into rules of logical deduction.

Technology of data processing in natural language

Processing of documentary data is realized by using grapheme, morphological, syntax and logical-semantic analysis; moreover, the initial texts are also indexed and remain in data warehouse.

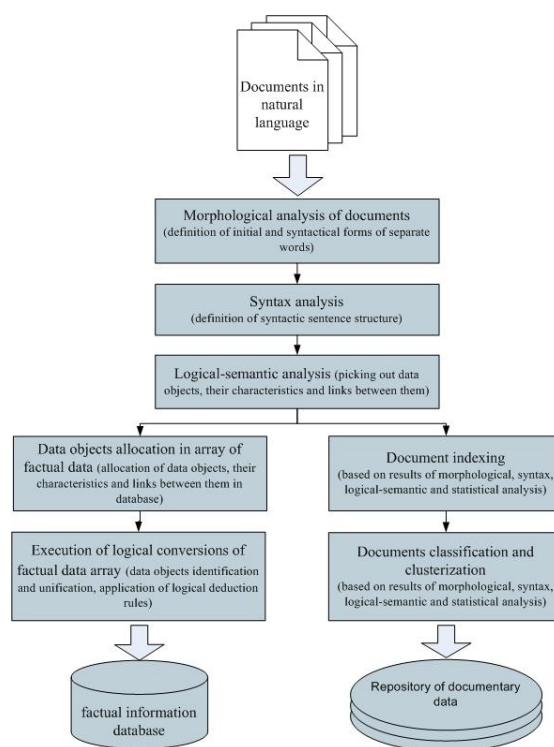


Fig. 1. Technology of data processing in natural language.

Grapheme analysis

The phase of grapheme analysis (fragmentation into lexemes without morphology) includes text fragmentation to separate recoveries of special types: a `word_in_quotes`, a `word`, a `punctuation_symbol`, a `numerical_block`, an `alphanumeric_block` and so on. The following characteristics are marked out for every fragment: position, length and priority. Moreover, such characteristics as `Up`, `Lw`, `UpLw`, representing the register they are written in, are marked out for these examples. As result of this work is system of separate lexemes with number in text and without division of word. If a word has several variants of parsing, then several fragments of system correspond to it with one number.

As an example illustrating data processing we will take the following text (a news report of 24.03.2006 which was taken from BBC website <http://news.bbc.co.uk>):

Belarus riot police halt protests

03/24/2006

Riot police in the Belarussian capital, Minsk, have broken up a five-day demonstration against the re-election of President Alexander Lukashenko.

Fig. 2. Natural language text.

The English variant for grapheme analysis of this text is presented in the following way:

WORD (0, 1, 1, BELARUS, UpLw, 706)
 WORD (1, 1, 1, RIOT, Lw, 714)
 WORD (2, 1, 1, POLICE, Lw, 722)
 WORD (3, 1, 1, HALT, Lw, 730)
 WORD (4, 1, 1, PROTESTS, Lw, 738)
 SPEC_SYMBOL (5, 0, 1, NEW_LINE, 746)
 NUM_BLOC (5, 1, 1, 03/24/2006, 753)
 SPEC_SYMBOL (6, 0, 1, NEW_LINE, 760)
 WORD (6, 1, 1, RIOT, UpLw, 768)
 WORD (7, 1, 1, POLICE, Lw, 776)
 WORD (8, 1, 1, IN, Lw, 784)
 WORD (9, 1, 1, THE, Lw, 792)
 WORD (10, 1, 1, BELARUSSIAN, UpLw, 800)
 WORD (11, 1, 1, CAPITAL, Lw, 808)
 WORD (12, 1, 1, MINSK, UpLw, 816)
 WORD (13, 1, 1, HAVE, Lw, 824)
 WORD (14, 1, 1, BROKEN, Lw, 832)
 WORD (15, 1, 1, UP, Lw, 840)
 WORD (16, 1, 1, A, Lw, 848)

WORD (17, 1, 1, FIVE, Lw, 856)
 WORD (18, 1, 1, DAY, Lw, 864)
 WORD (19, 1, 1, DEMONSTRATION, Lw, 872)
 WORD (20, 1, 1, AGAINST, Lw, 880)
 WORD (21, 1, 1, THE, Lw, 898)
 WORD (22, 1, 1, RE-ELECTION, Lw, 910)
 WORD (23, 1, 1, OF, Lw, 918)
 WORD (24, 1, 1, PRESIDENT, UpLw, 924)
 WORD (25, 1, 1, ALEXANDER, UpLw, 933)
 WORD (26, 1, 1, LUKASHENKO, UpLw, 945)
 PUNC_SYMBOL (27, 1, 1, ., 987)

Fig. 3. Result of the grapheme analysis(eng).

Morphological analysis.

During the phase of morphological analysis all the words from the text are being analyzed. A number in the text, an initial form, part of speech, form in which the word was met in the text and morphological characters are written then for every word of the text.

As a result of morphological analysis is a system, consisting of separate lexemes with their own number in the text and without division of abbreviation. If a word has several variants of analysis, then it is corresponded to several fragments of the system with one number.

START_FORM_MORF (0, 1, 1, BELARUS, гж, 768)
 START_FORM_MORF (0, 1, 1, BELARUSSIAN, ra, 758)
 START_FORM_MORF (0, 1, 1, POLICE, ra, 798)
 VERB (1, 1, 1, HALT, 2780)

Fig. 4. Result of the morphological analysis, not complete (eng).

Syntax analysis.

During syntax analysis of the text sentences are processed in series. Sentences consist of order of words, punctuation marks, such as dash, inverted commas, brackets, colons, commas (other punctuation marks are deleted during morphological analysis), and also special sequences of symbols like numbers, a set of Latin letters etc. Later this phase is used in determination of the fact and colors of connections between marked objects.

Logical-semantic analysis.

On basis of syntax analysis the final structures of texts are transformed into a semantic network (Fig. 6.), in which nodes are presented by a great number of frequent terms- words and set expressions. These network nodes are connected associatively between each other with different strength, which depends on frequency of cooccurrence of concepts in the sentences of the text. This semantic network can be used as a model of data domain in processing new unknown documents.

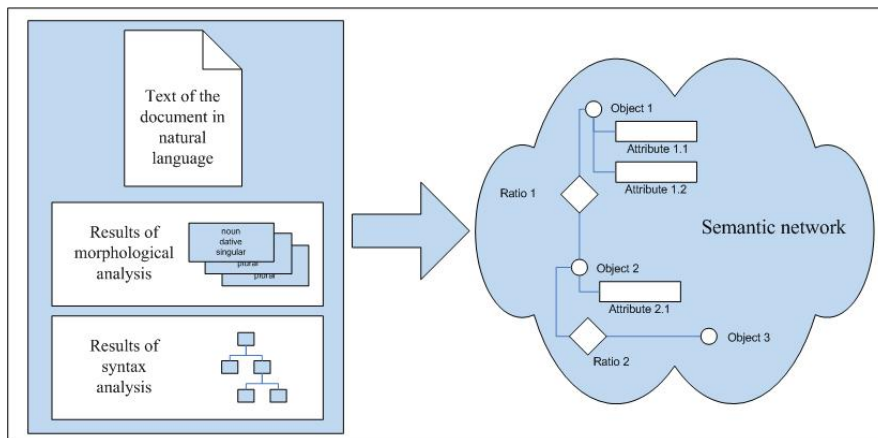


Fig. 5. A general scheme of logical-semantic analysis

Processing procedure consists of the following phases:

- picking out lexical definitions;
- analysis of domain objects;
- creating connections between marked objects.

Picking out lexical definitions

For picking out lexical definitions from texts preformed dictionaries are usually used. Structure of these dictionaries is a set of couples “goal – possible decoding”, where goal is an abbreviation or a concrete instance of any object. It is possible to have several sources for an object dictionary and several decodings for abbreviations. General dictionary structure usually looks like:

- for abbreviation dictionaries:

```
<ITEM>
  <SRC>abbreviation</SRC>
  <DEST>decoding_1</DEST>
  <DEST>decoding_2</DEST>
</ITEM>
```
- for object dictionaries:

```
<ITEM>
  <SRC>object_name_1</SRC>
  <SRC> object_name_2</SRC>
  <DEST>decoding</DEST>
</ITEM>
```

An example of using such an object dictionary we can see in Fig. 4. Minsk was identified here like “city” as dictionary contains the following structure:

```
<ITEM>
  <SRC>ADLER</SRC>
  <SRC>... </SRC>
  <SRC>MINSK</SRC>
  <SRC>... </SRC>
  <SRC>YARTSEVO</SRC>
  <DEST>CITY</DEST>
</ITEM>
```

As a result of processing the text, which was taken like an example, the following lexical definitions were picked out:

```
CITY (12, 1, 1, MINSK, 32019)
STATE_STRUCTURE (24, 1, 1, PRESIDENT, 3946)
INO_NAME (25, 1, 1, ALEXANDER, 32271)
```

Fig.6. The results of picking out lexical definitions (eng).

Marking out domain objects

While parsing domain objects, addresses, phone numbers, names, organizations, dates etc., are marked out – the final set of objects is determined directly in the rules of logical deduction. While analyzing objects proper rules are progressed in series (proper functions are called). Every function marks out necessary lexemes by using its own set of rules and after that it creates a new object on the base of these lexemes, saving necessary data in the object. All the processed lexemes are marked in a special way to exclude a probability of re-processing.

This set of rules is called “a first leveled set of rules”, their general structure is presented in following:

```
<RULE priority = "n">
<!--inquiry division -->
  <QUERY>OBJECT_NAME_1 (list_of_object_parameters)</QUERY>
  .....
<!-- blocks ORQUERY -->
  .....
  <IGNORE> OBJECT_NAME_N(list_of_object_parameters) </IGNORE>
  .....
  <QUERY> OBJECT_NAME_N (list_of_object_parameters) </QUERY>
<!--functions division -->
  <FUNCTION> function_name_1 (function_parameters) </FUNCTION>
  .....
  <FUNCTION> function_name_N (function_parameters) </FUNCTION>
<!--action division -->
```

```

<CREATE> OBJECT_NAME _1 (list_of_object_parameters) </CREATE>
<INHERIT> inquiry_number </INHERIT>
<CREATE> ATTRIBUTE (attribute name_1_1, meaning) </CREATE>
.....
<CREATE> ATTRIBUTE (attribute name_1_N, meaning) </CREATE>
.....
<CREATE> OBJECT_NAME _N (list_of_object_parameters) </CREATE>
<CREATE> ATTRIBUTE (attribute name _N_1, meaning) </CREATE>
.....
<CREATE> ATTRIBUTE (attribute name _N_N, meaning) </CREATE>
<DESTROY> INITIAL_FORM ($x,$y) </DESTROY>

```

</RULE>

Here:

- a list of parameters – meanings of parameters, listed with commas; every parameter can be changed with a variable name (such as \$ - symbol, after which a Latin name comes, or an anonymous variable, marked with _ - symbol);
- a scope of variable quantity – is one rule;
- a parameter “priority” points at priority of the rule. If several rules worked in a set of lexemes and the used lexemes are intersected, then only one object will be chosen during filtration, the one which was got by the rule with the highest priority.

While the rule is in progress, first comes the revise of object models, pointed between tags QUERY with models available. A transfer to the next part of rule happens only in case of revealing a combination of the corresponding models, following inseparably in a mentioned sequence.

Further call a functions, pointed by tags FUNCTION, follows. Every function returns Boolean number. Transition happens only in case when the result is TRUE, but it is possible that function changes meanings of included variable quantities.

After that creation of final objects, described with tags CREATE, and creation of attributes which correspond them (by using an operation <CREATE>ATTRIBUTE (parameter_name, parameter_meaning) <CREATE>), or attribute inheritance of existing objects with a tag INHERIT, happens.

It is also possible to destroy objects according to masks, pointed in tags DESTROY. This function is created to have an opportunity of cleaning up temporary and system objects in information processing to reduce noise level and probability if getting false objects and attributes because of wrong order of using rules. When the first – leveled rules are done, all the temporary and system objects are automatically deleted.

The following example of using the first – leveled rules shows a process of forming an object:

- this rule marks out a set like “name, last name”:


```

<RULE>
  <QUERY>WORD($I,"UpLw")</QUERY>
  <QUERY> WORD ($F,"UpLw")</QUERY>
  <FUNCTION>MakeFI($F,$I, $RF,$RI, $g)</FUNCTION>
  <CREATE>PERSON($RF,$RI,"", $g)</CREATE>
</RULE>

```

The rule, used in this example, marked out a set “Alexander Lukashenko”;

- this rule, based on marked sets “last name, name, middle name” forms an object “natural person”:


```

<RULE priority="3">
  <QUERY>MAN($a,$b,$c,$d)</QUERY>
  <FUNCTION>concat_space($all,$a,$b)</FUNCTION>
  <FUNCTION>concat_space($all,$all,$c)</FUNCTION>
  <CREATE> NATURAL_PERSON ($all)</CREATE>
  <CREATE>ATTRIBUTE(LSAT_NAME,$a)</CREATE>
  <CREATE> ATTRIBUTE (NAME,$b)</CREATE>
  <CREATE> ATTRIBUTE (MIDDLE_NAME,$c)</CREATE>
  <INHERIT>0</INHERIT>
</RULE>

```

Using this rule in the example an object NATURAL_PERSON- “Alexander Lukashenko” was marked out.

As a result of applying first-leveled rules in the text, used like an example, the following objects were marked out in addition to objects, which were marked out by using dictionaries:

```

CITY (12, 1, 1, MINSK, 32019)
STATE_STRUCTURE (24, 1, 1, PRESIDENT, 3946)
INO_NAME (25, 1, 1, ALEXANDER, 32271)

DATE (5, 1, 1, 03/24/2006, 7399)
ATTR (5, 1, 1, YEAR, 2006, 7399)
ATTR (5, 1, 1, DAY, 24, 7399)
ATTR (5, 1, 1, MONTH, 03, 7399)
PERSON (24, 2, 4, LUKASHENKO ALEXANDER, 7904)
ATTR (24, 2, 4, NAME, ALEXANDER, 7904)
ATTR (24, 2, 4, SURNAME, LUKASHENKO, 7904)

```

Fig.7. The result of applying first-level rules (eng).

Creation of links between marked objects

The most simple method is to link all the objects, placed in the same sentence by the principle of belonging to one affair or fact, described in the sentence.

The more complicated is a procedure of extraction the links between objects, based on context. For example if one sentence contains an object “natural person” and “address” and there is a verb “lives” between them, a link “place_of_living” will be marked out.

Rules of extraction the links between objects (second-level rules) are described the same as rules of the first level, but there are some differences:

- sequence of elements, forming the rule, is not important; they can follow in any order;
- every rule is given a priority coefficient for a created link;
- presence of gaps in a great amount decreases priority of the result;
- words from sentences used in every example are written in it in the form of bit mask;
- rules are adapted separately to every sentence in the text.

The overview of the second-level rule is:

```

<RULE2 scope="sentence | paragraph| <all>" direction="up | down| <all>" multiobj1 multiobj2 >
<COEF> coefficient </COEF>
<!--inquiry block -->
    <QUERY>OBJECT_NAME_1 (list_of_object_parametres)</QUERY>
    .....
    <QUERY> OBJECT_NAME _N(list_of_object_parametres)</QUERY>
<!--function division -->
    <FUNCTION>function_name_1(function_parametres)</FUNCTION>
    .....
    <FUNCTION> function_name _N(function_parametres)</FUNCTION>
<!-- action division -->
    <CREATE> LINK_NAME_1(OBJECT_1, OBJECT_2,VERB)</CREATE>
    .....
    <CREATE> LINK_NAME _N(OBJECT_1, OBJECT_2,VERB)</CREATE>
</RULE2>

```

The example of using the first-level rules shows how the links between objects are realized.

This rule forms a link between a person and a state structure:

```

<RULE2 scope="sentence" direction="all">
    <COEF>10</COEF>
    <QUERY> STATE_STRUCTURE ($A)</QUERY>
    <ORQUERY>
        <QUERY>VERB(TO_BE_ELECTED)</QUERY>
        <QUERY>VERB(TO_BE_REELECTED)</QUERY>
        <QUERY> VERB (TO_APPOINT)</QUERY>
        <QUERY> VERB (TO_EXECUTE)</QUERY>
        <QUERY> VERB (TO_RULE)</QUERY>
    </ORQUERY>
    <QUERY>NATURAL_PERSON($M)</QUERY>
    <CREATE>LINK_ NATURAL_PERSON _STATE_STRUCTURE($M, $A, $orqueryA)
</CREATE>
</RULE2>

```

As a result of using the second-level rules in text, which was taken like an example, the following link was marked out – “state authority “president” (id: 2964) is associated with a natural person Alexander Lukashenko (id: 3904)”:

LINK_PERSON_STATE_STR (00000012221111100, 1, 2946, 3904, 2822, 12301)

Fig. 8. Results of using second-level rules. (eng).

Adapting the results of processing documentary information in the procedures of knowledge extraction

Procedure of processing information in natural language, which is described above, is usually the first phase of documentary information processing. The general result of initial processing in the observed system is an array of constrained factual data, presented as a semantic network.

For marked out of data objects from different sources, an identifying procedure is held out. It helps to reveal similar data objects which were got from different sources. When identifying objects, two basic types of links are chosen: links of similarity and links of coincidence, but at the same time a possibility of automatic junction of concurrent objects is provided. Links of similarity are usually processed by an analyst (analyst uses his expertise to find out if the data objects are concurrent and, if it is necessary, he makes their junction by himself) before starting procedure of knowledge extraction.

A possibility of correlating data objects that are newly placed in a factual basis to the ones that are already available is considered to be an important peculiarity of identifying process. It allows solving problems of monitoring standard situations.

When identification and junction of concurrent data objects are done, a formed array of factual data is ready for knowledge to be extracted from it.

This procedure supposes using different procedures of processing factual data:

- searching procedures:
 - attributive search;
 - indistinct search;
 - a full-text search of similar documents;
- analytic procedure:
 - context-dependent analysis;
 - situational analysis;
 - searching for communication series;
 - procedure of modeling;
 - simulation;
 - prediction of situational development in long time.

These procedures of data processing are usually used in solving applications for supporting assumption, monitoring situation and analytic researches and so on.

References

- [1] A.V. Gubin , D.V. Krayushkin , V.V. Kuzmin. «Selection of Technology for Creating Knowledge Management System» // Digest of IIP RAS, 2004.
- [2] D.V. Krayushkin «Documentary Information Preliminary Processing Technology Analysis» // Digest of IIP RAS, 2005.
- [3] D.V. Krayushkin «Enterprise and Territory Security Threats Detection and Analysis Using Monitoring and Forecasting Centers System»// First Eurasian Forum for Informational Security «INFOFORUM-Almaty», 2005.
- [4] D.V. Krayushkin, A.A. Kaschenko «Data Mining from Publicly Available Information Sources»// Xth International Scientific and Practical Conference «Complex Information Protection», 2006.
- [5] Igor Kuznetsov. «Semantic Representations». Moscow: Science, 1978. 294 p. (in Russian).
- [6] Igor Kuznetsov. «Methods of report processing which reveal the characteristics of figurants and incidents. International workshop»// "Dialogue'98": Computational Linguistic and its applications. Vol2. Kazan, 1998. P. 961-700.
- [7] I.Kuznetsov, M.Charnine. «Semantic-Oriented System For Factual Search With the Interface in Russian and English» // Systems and Facilities of Informatics. Moscow: Science, 1995, V 7.
- [8] Igor Kuznetsov, Andrey Matskevich. «System for Extracting Semantic Information from Natural Language Text» // "Dialogue'02": Computational Linguistic and its applications. Vol2. Moscow: Science, 2002.