Methodology for Building Extraction Templates for Russian Language in Knowledge-Based IE Systems.

Valery Solovyev, Vladimir Ivanov, Rinat Gareev, Sergey Serebryakov, Natalia Vassilieva

HP Laboratories
HPL-2012-211

Keyword(s):
event extraction; dictionaries; rules; patterns; meaning-text model; Chomsky grammars

Abstract:
In this technical report we describe methodology for building information extraction (IE) rules. Rules are usually developed by experts and are widely used in knowledge-based IE systems. They consist of two parts: the left-hand side (LHS) of a rule is a template that matches a certain syntactico-semantic structure (SSS) and the right-hand side is an action that is executed when LHS template is matched against a particular text fragment. In the report we describe the process of building a more complex LHS part (template). This methodology was used for developing the information extraction system that extracts business events from news articles written in Russian language.
Methodology for Building Extraction Templates for Russian Language in Knowledge-Based IE Systems

Valery Solovyev
Kazan Federal University
Kazan, Russia
maki.solovyev@mail.ru

Vladimir Ivanov
Kazan Federal University
Kazan, Russia
nomemm@gmail.com

Rinat Gareev
Kazan Federal University
Kazan, Russia
aldvinaldv@gmail.com

Sergey Serebryakov
Hewlett-Packard Laboratories
Saint-Petersburg, Russia
sergey.serebryakov@hp.com

Natalia Vassilieva
Hewlett-Packard Laboratories
Saint-Petersburg, Russia
nvassilieva@hp.com

ABSTRACT
In this paper we describe methodology for building information extraction (IE) rules. Rules are usually developed by experts and are widely used in knowledge-based IE systems. They consist of two parts: the left-hand side (LHS) of a rule is a template that matches a certain syntactico-semantic structure (SSS) and the right-hand side is an action that is executed when LHS template is matched against a particular text fragment. In the paper we describe the process of building a more complex LHS part (further in the paper we will refer to LHS as template). This methodology was used for developing the information extraction system that extracts business events from news articles written in Russian language.

Categories and Subject Descriptors
H.3 [Information Storage and Retrieval]: Content Analysis and Indexing; H.4 [Information Systems Applications]: Miscellaneous

General Terms
Theory

Keywords
IE, rules, templates, free-context grammars, meaning-text

1. INTRODUCTION
Rules developed by experts are widely used in information extraction knowledge-based (IE) systems. The rules are productions consisting of two parts. The left-hand side (LHS) of a rule is a template that matches a certain syntactico-semantic structure (SSS) and the right-hand side is an action that is executed when LHS template is matched against a particular text fragment. A typical action is to annotate a matched text fragment with a new SSS. Once the text is annotated in such a way, knowledge is extracted and stored in a particular format. Rule-based approach is described in a number of works [5, 2, 11, 9]. To our knowledge, there are no papers providing the set of rules developed for a particular domain or describing a methodology used to develop the rules themselves. In this paper we describe our methodology for building extraction rules. In particular, we describe the process of building a more complex LHS part (further in the paper we will refer to LHS as template). This methodology was used for developing the information extraction system that extracts business events from news articles written in Russian language.

2. TEMPLATES
Event extraction (EE) problem is recognized to be a high-level problem in IE. Before EE, it is necessary to preprocess the entire document – split the text into sentences and tokens, extract named entities such as Persons, Companies etc. These named entities are considered to be arguments of events. For instance, event of type Company Announcement would have two arguments: company (named entity of type Company) and date-time (named entity of type Temporal Expression).

Dictionary-based approach is another way to extract information from free texts. It is common to use dictionaries of named entities besides EE rules. In our system we use dictionaries of company names and given and family names of people. Additional dictionaries such as dictionaries of countries and positions in companies are also utilized.

We introduce the notion of Person Identifier for more precise people identification. Dictionary of person identifiers contains such words as businessman, entrepreneur, oligarch etc. We denote the dictionaries of given and family names as Name and Surame. A simple example of the rule is:

\[ \text{Name Surname} \rightarrow \text{Person} \]

The rule 1 says that if a text fragment annotated as a person's given name is followed by a text fragment annotated as a person's family name, then these two text fragments can be united into SSS mentioning person. TextMARKER [6] engine is used for representing rules and annotating documents. The same rule in TextMARKER notation looks as follows:

\[ \text{Name Surname} \rightarrow \text{MARK(} \text{Person, 1, 2) } \]

MARK here is a marking operator that creates new annotations, digits 1 and 2 show template elements in a certain order that will be included into Person annotation. Further on, we will be representing rules in the format (1), without paying attention to TextMARKER notation details.
It is hazardous to rely only on intuition because it is easy to miss something important. Only a very experienced developer can afford it. For the beginners we recommend to use other information sources. It is quite natural to be guided by examples from real texts. Looking through the examples from the texts one can write rules for every example. However, a developer will hardly be able to read millions of news articles. Some grammatically valid constructions are likely not to be found in a small-size corpora and thus a number of extraction templates would be missing in the final rule set resulting in low recall of IE system.

Chomsky grammars like other formal models are intended for constructing all grammatically valid sentences. That is why they can be used for generating a complete set of templates.

As an example we provide a set of rules for extracting named entity of type "Person". These rules are presented in the form of context-free Chomsky grammars (bottom-up parsing [1]). In this example we use the dictionaries of Companies, Positions, SurnamePrefix dictionary which includes such words as Mc, van, de, ter etc. and the Initials dictionary that contains all letters with dots (initials). Symbol W stands for any word, CW denotes any word beginning with a capital letter. The tokens "and" and "," are also used in the examples below.

\[
\begin{align*}
\text{Surname} & \rightarrow \text{Person} \\
\text{SurnamePrefix SurnamePrefix} & \rightarrow \text{SurnamePrefix} \\
\text{SurnamePrefix Surname} & \rightarrow \text{Surname} \\
\text{Initials Initials} & \rightarrow \text{Initials} \\
\text{Initials Surname} & \rightarrow \text{Person} \\
\text{Surname Surname} & \rightarrow \text{Person} \\
\text{Name Name} & \rightarrow \text{Person} \\
\text{Surname Surname} & \rightarrow \text{Surname} \\
\text{Name Person} & \rightarrow \text{Person} \\
\text{Person Name} & \rightarrow \text{Person} \\
\text{Person Indicator Person} & \rightarrow \text{Person} \\
\text{Person Indicator} & \rightarrow \text{Person} \\
\text{CW CW} & \rightarrow \text{CW} \\
\text{Position Company} & \rightarrow \text{Person} \\
\text{Position W Company} & \rightarrow \text{Person} \\
\text{Person Person} & \rightarrow \text{Person} \\
\text{Person}, " \text{Person} & \rightarrow \text{Person} \\
\text{Person} "\text{and}" \text{Person} & \rightarrow \text{Person} \\
\text{Person} "\text{and}" \text{W} "\text{and}" & \rightarrow \text{Person} \\
\text{Person} W W & \rightarrow \text{W} \\
\text{Position} "\text{and}" \text{Position} & \rightarrow \text{Position} \\
\end{align*}
\]

Now, we provide the example of bottom-up parsing. The text fragment "The Chairperson of "GasProm" Board of Directors Viktor Zubkov" is turned into "Position Company GivenName FamilyName" after dictionary-based lookup and then consequently into "Person FamilyName" and "Person" after applying the above mentioned rules.

It is important to mention that rules W W → W and CW CW → CW allow to transform long constructions into smaller ones of the same semantic type. Transformation of the rules into the TextMARKER notation is not a difficult task and can be done automatically. One should notice that the above mentioned set of rules is a theoretical model and the peculiarities of the TextMARKER must be taken into account while transforming rules into its notation.

Strict bottom-up parsing is not the only acceptable method. We have implemented a more complicated algorithm with the altered order of different phases of analysis. The general idea of the algorithm is the following. At the beginning all the words with capital letters are marked as Possible-Name (first words in the sentences are processed in a special way using dictionary of stop words). Then the rules which determine the concrete type of these words are applied (these rules determine if particular PossibleName is Person or Company with help of PersonIndicator and CompanyIndicator semantic classes). Afterwards the templates of events are applied, and only then, if the ambiguity of names identification still remains, the dictionaries of companies, given and family names are used.

4. REPHRASING

The use of the I.A. Melchuk's model "Meaning ↔ Text" is effective at the sentence level for Russian language [8]. This model includes the description of transformations which retain the grammatical correctness (rephrasing). In this model the set of such lexical functions (LFs) as Syn, Conv, Der, Gener is defined (see definition [8]). LFs are functions that describe lexical relations between lexical units (LU) - lexemes and/or phrasems used to construct sentences. We provide only a short description of these LFs, for complete definitions we refer to paper [7].

- **Syn(L)** [synonym] returns synonyms of L: Syn(car) = {auto, automobile, roadster}

- **Conv(L)** [conversive] returns conversion for L: for two text fragments 'workshop follows conference' and 'conference precedes workshop' Conv(follows) = {precedes}.

- **Der(L)** [derivative] returns any deep syntactic derivative for L. This function breaks into four subfunctions:
  - S₀(L) [substantival] returns a noun that is congruent to L: S₀(analyze) = {analysis}.
  - A₀(L) [adjectival] returns adjective that is congruent to L: A₀(city) = {urban}.
  - V₀(L) [verb] returns verb that is congruent to L: V₀(analyze) = {analyze}.

3. SOURCES OF TEMPLATES

While building templates, the developer should take into account all the peculiarities of the grammar of natural language, i.e. it is necessary to build a formal model of a fragment of grammar that is adequate to the task. For this purpose the developer may use the following information sources: a) his own intuition; b) examples from the texts; c) language model based on Chomsky grammars; d) other formal language models (for example, I.A. Melchuk's "Meaning ↔ Text" model for the Russian language [8, 5]).

These rules are presented in the form of context-free Chomsky grammars (bottom-up parsing [1]).
The use of transformations of such kinds is described in a number of works [11] but these transformations have not been used systematically.

Let us consider the following example of using this approach for obtaining new templates for Mergers and Acquisitions events. Event of this type has two mandatory arguments (actants, slots) (1) acquirer (company buying another company) and (2) acquiree (company that is bought by other company). We use AcquisitionIndicator dictionary of words that clearly indicate the potential presence of the events under consideration in free texts. We will be referring to AcquisitionIndicator as $C_0$ and to events arguments as $X$ and $Y$ for acquirer and acquiree respectively. The semantic type of arguments $X$ and $Y$ is Company.

- Input parameters:
  - key word $C_0$, which denotes an event, i.e. an indicator of the event ("buy") from the AcquisitionIndicator dictionary;
  - actants (arguments, slots) of an event $(X, Y)$;

- Output parameters:
  - tuples for forming templates in the form: $X, C_0, Y$, which in the case of selecting certain classes for $X, C_0, Y$ turn into "Company AcquisitionIndicator Company";

Description of the method:

STEP 1
To create tuples combining $C_0, X, Y$. One can use permutation operator and removing of actants operator. The permutation operator selects all possible arrangements of $C_0, X, Y$ according to the meaning of $C_0$ and roles of $X, Y$ (e.g. while tuple $(C_0, X, Y)$ considered as possible in Russian, tuple $(X, Y, C_0)$ rather uncommon or even wrong if $C_0$ is a verb). The removing of actants operator deletes one or more actants from the tuple. This operator is usefull when the source tuple contains additional (non-obligatory) actants.

STEP 2
To generate the set of synonyms for event indicator $T_{syn} = \{Sym(C_0)\}$ (for instance, such as buy, acquire and all their variations). If $T_{syn}$ is empty, go to STEP 3, otherwise set $C_0 \leftarrow t_{syn}$ for every element $t_{syn}$ of the set and go to STEP 1. After bypassing all the elements from the set go to STEP 3.

STEP 3
To generate the set of conversions for event indicator $T_{conv} = \{Conv(C_0)\}$ (for instance, such as buy ↔ sell). If $T_{conv}$ is empty, go to STEP 4, otherwise set $C_0 \leftarrow t_{conv}$ for every element $t_{conv}$ from the set and go to step STEP 1. After bypassing all the elements of the set go to STEP 4.

STEP 4
To generate the set of derivates $T_{der} = \{Der(C_0)\}$. Form additional actants $F_i$, defined by the meaning of the situation, denoted as $t_{der}$, for every element $t_{der}$ of the set, $C_0 \leftarrow t_{der}$ and execute STEP 1. After bypassing all the elements of the set go to STEP 5. These additional actants $F_i$ are intended to fill new semantic valencies which $t_{der}$ may have.

STEP 5 (optional)
To generate the set $T_{syn} = \{Genr(C_0)\} \cup \{Genr(T_{syn})\} \cup \{Genr(T_{conv})\} \cup \{Genr(T_{der})\}$. For every element $t$ of the $T_{syn}$ generate additional actants $F_j$ (just like on STEP 4), set $C_0 \leftarrow t$ and go to STEP 1. After bypassing all the elements of the set finish execution.

This method allows us to obtain, for example, the construction "the owner sold the company $B$ to company $A$" from the construction "company $A$ bought company $B$". For instance, for another event of type Management Position Change using the method automatically for construction "Temporal-Expression 'Person A' fired 'Person D' from 'Position B' of 'Company C" we get 44 variants, that would be difficult to enumerate manually.

Context-free grammars in Chomsky normal form and rephrasing method can be used both for initial creation of the template set and for its expanding in the active learning approach in machine learning of templates [10]. Method of rephrasing was used earlier in Question Answering (QA) [3, 4].

5. CONCLUSION
Our experience in developing a system for extracting knowledge from free texts written in Russian language shows that the most effective method is a combined one which includes both studying the real examples of entities (relations) to be extracted in texts and applying the theoretical methods such as Chomsky grammars and I.A.Melchuk’s model "Meaning ↔ Text". Making use of theoretical methods is aimed at providing the complete set of templates. It also allows us to represent the set of templates in a more elegant way. Further research will be devoted to experimental evaluation of the system, developed on the basis of the described methodology.

6. REFERENCES


