УДК 004.912

ГЕНЕРАЦИЯ СЛОВАРЯ МОДЕЛЕЙ УПРАВЛЕНИЯ ДЛЯ ЗАДАЧИ ИЗВЛЕЧЕНИЯ СОБЫТИЙ

Ф.Николаев, В.Иванов

GENERATING A DICTIONARY OF SUBCATEGORIZATION FRAMES FOR EVENT EXTRACTION

F.Nikolaev, V.Ivanov

Казанский федеральный университет, f12v@yandex.ru

Модели управления являются важным понятием в ряде задач обработки текста на естественном языке. В частности, специальным образом составленный словарь моделей управления глаголов-индикаторов событий может быть применен для задачи извлечения событий и аргументов из текстов. Особенно актуален этот способ для языков со свободным порядком расположения слов, таких как русский, так как традиционный способ извлечения на основе линейных шаблонов является в этом случае сложно применимым. В статье предлагается способ полуавтоматического формирования такого словаря с использованием корпуса Google Books Ngram и последующего его дополнения экспертом с помощью разработанного авторами веб-приложения. На основе составленного словаря разработан алгоритм извлечения, показаны примеры его применения к новостным текстам.

Keywords: извлечение событий, модели управления, корпус n-грамм Google Books

Subcategorization frames are important in a number of natural language processing tasks. In particular, a specially constructed dictionary of subcategorization frames of words that indicates some events (usually verbs) can be used for the task of extracting the events and their arguments. This is especially useful for languages with free words order like Russian for which the traditional approach based on linear templates is not quite easy to use. This article proposes a method for semi-automatic construction of such dictionary using Google Books Ngram corpus and following expert assignment with help of a specially developed web application. On the basis of the constructed dictionary an algorithm for event extraction was designed. In this article we show some results of its work.

Ключевые слова: event extraction, subcategorization frames, Google Books Ngram corpus

1. Introduction and Motivation

Event extraction is an important task in extracting information from unstructured texts. This task attracted a number of researchers in the last decade. An event extraction system aims at capturing certain parts of a text (e.g. event type, participants and attributes). One of the central concepts in event extraction is an indicator word (usually a separate verb) denoting a type of event [1]. On one hand, the indicator word indicates presence of an event in a sentence. On the other hand, the indicator is considered as a main part in knowledge-based (KB) approach to event extraction.

According to this approach, rules (or patterns) and dictionaries are used. These patterns may be generated automatically [2] or defined manually [3]. However, in languages with free word order (e.g. Russian) a developer of those patterns should also take into account all possible arrangements of words in a sentence. In this case it is more natural to define pattern parts as independent pairs: "event-participant" which will be automatically mapped to "predicate-argument" pairs that denote subordination in a parse tree of a sentence at hand. Thus, a complete subordination dictionary becomes a crucial element of a knowledge-based event extraction system. A well-known limitation of recent works in this area is insufficient dictionary size that prevents using such dictionaries in a computer system.

In 2013 Klyshinsky et al. [4] generated such dictionary for Russian verbs using a set of web corpora; all corpora together contain about 10-11 billion tokens. The authors proposed a method for automatic generation of dictionary for verbs and prepositions. Klyshinsky et al. reported that the dictionary size was about 25-30 thousand verbs. Their method deals only with lexical information, i.e. extraction of verb(-preposition)-noun dependencies was done with six simple finite automata, and no parsing step was performed. Treebanks of the Russian language also have insufficient corpus size for automatic generation of a complete (for most Russian verbs) subordination dictionary. The main difference with previous works is that ambiguous part of text was not processed at all. Resulted set was filtered to exclude case ambiguity, infrequent words and ngrams which are not allowed in Russian grammar. The dictionary was evaluated on

Table 1

Event types and their brief description

Event type	Description
CHARGE-INDICT	A CHARGE-INDICT Event occurs whenever a person or organization is accused of a crime by a state.
CONTACT	A CONTACT Event occurs when two or more people interact with one another and directly engage in discussion.
DECLARE-BANKRUPTCY	A DECLARE-BANKRUPTCY Event will occur whenever an organization officially requests legal protection from debt collection due to an extremely negative balance sheet.
DIE	A DIE Event occurs whenever the life of a person ends. DIE Events can be accidental, intentional or self-inflicted.
ELECT	An ELECT Event occurs whenever a candidate wins an election designed to determine the Person argument of a START-POSITION Event.
END-ORG	An END-ORG Event occurs whenever an organization ceases to exist (in other words 'goes out of business').
END-POSITION	An END-POSITION Event occurs whenever a person stops working for (or changes offices within) an organization.
FINE	A FINE Event takes place whenever a state actor issues a financial punishment to a person or organization, typically as a result of court proceedings.
INJURE	An INJURE Event occurs whenever a person experiences physical harm. INJURE Events can be accidental, intentional or self-inflicted.
MERGE-ORG	A MERGE-ORG Event occurs whenever two or more organizations come together to form a new organization.
NOMINATE	A NOMINATE Event occurs whenever a person is proposed for a new position, through official channels.
START-ORG	A START-ORG Event occurs whenever a new organization is created.
START-POSITION	A START-POSITION Event occurs whenever a person begins working for (or changes offices within) an organization.
SUE	A SUE Event occurs whenever a court proceeding has been initiated for the purposes of determining the liability of a person or organization accused of committing a crime or neglecting a commitment.
TRANSFER-MONEY	TRANSFER-MONEY Events refer to the giving, receiving, borrowing, or lending money when it is not in the context of purchasing something.
TRANSFER-OWNERSHIP	TRANSFER-OWNERSHIP Events refer to the buying, selling, loaning, borrowing, giving, or receiving artifacts or organizations.
TRANSPORT	A TRANSPORT Event occurs whenever an artifact or a person is moved from one place to another.
TRIAL-HEARING	A TRIAL-HEARING Event occurs whenever a court proceeding has been initiated for the purposes of determining the guilt or innocence of a person or organization accused of committing a crime.

a corpora of Russian fiction texts and news site texts, and it showed good results.

In this paper we present an alternative method for generating a subordination dictionary using Google Books Ngram Corpus (contains of 67 billion tokens). The main motivation behind this work is to facilitate an event extraction system for Russian which is focused on event types described in ACE [1]. Here we consider the case when an indicator is the main verb (or predicate) which acts as a syntactic head for all participants of a corresponding event (participants of the event act as syntactical arguments of the predicate). For now, we extract only the subset of all ACE event types. Event types we work with are shown in Table 1. We start with a discussion of the method for generating a subcategorization dictionary. Then we give a brief overview of the user interface that can be used for both pattern definition and dictionary correction.

2. A subordination dictionary

The main idea of the work is based on using the Google Books Ngram Corpus (GBNC) that was enriched with morphological information and filtered with certain rules. The study was carried out for Russian, but this method is applicable to other languages for which the Google Books Ngram Corpus and morphological dictionary are available.

2.1. Google Books Ngram Corpus

The Google Books Ngram Corpus describes how often words and phrases (i.e., ngrams) were used over a period of five centuries, in eight languages; it reflects 6% of all books ever published. Russian subset of GBNC contains 67,137,666,353 tokens extracted from 591,310 volumes [6], mostly over the past three centuries. The most part of books was drawn from university libraries. Each book was scanned with custom equipment and the text was digitized by means of OCR. Only ngrams that appear over 40 times across the corpus are included into a dataset. The data is available for download and can also be viewed through the interactive Google Books Ngram Viewer*. Latest version of GBNC introduced syntactic annotations: words were tagged with their part-of-speech, and head-modifier relationships were recorded. We extensively exploited these relationships in our work.

2.2. Corpus preprocessing

The original GBNC data set contains statistics on occurrences of n-grams (n = 1…5) as well as frequencies of binary dependencies between words. These binary dependencies represent syntactic links between words from the Google Books texts. An accuracy of unlabeled attachment for Russian dependency parser reported in [6] is 86.2%. As GBNC stores all statistics on a year-by-year basis, each datafile contains tab-separated data in the following format: *ngram, year, match_count, volume_count.*

We have preprocessed the original data set in a special way. First, for each 2-gram dependency (the same step for each 3-gram) we have collected all its occurrences on the whole data set and summated all "match_count" values since 1900. Aggregated data set consists of pairs (n-gram, count) for n=2, 3. This step also joined n-grams typed in different cases (lower and upper) into a single (lower case) n-gram.

The next step was to assign each word in the data set a POS-tag and morphological features. For this purpose we used a morphological dictionary provided by OpenCorpora [5] to generate a POS-tag and morphological features for 1-grams only. Thus, we got an enriched dataset that has the following format:

n1, match_count, pos, lemma, gram,

where *n1* is a word from the GBNC 1-gram dataset; *pos*, *lemma* and *gram* stand for the POS-tag, lemmatized word form and vector of grammatical features respectively. Ambiguous words have led to several records in the enriched dataset. For instance,

n1, match_count, pos, lemma_id, gramA,

n1, match_count, pos, lemma_id, gramB,

where ambiguous word *n1* has two sets of grammatical features: *gramA* and *gramB*. In all such cases we omit these conflicting rows from the dataset because taking these records into account adds a lot of noise.

2.3. Dictionary of verbal government construction

Let us briefly describe a technique we use for generating a dictionary of direct subject government. To this end, we capture all pairs (head, dep) with a POS-tag of the head part equal to 'VERB' and having a certain grammatical case for the dependent part (dep), say 'gent' for Genitive. Finally, we group all these pairs by "lemma_id" (in order to regard different forms of the same verb) and count the number of records and summate match_count values. Basically, we run the following SQL-query against the preprocessed dataset:

```
CREATE TABLE direct_verbal_control as
   SELECT
     dep_bigrams.lemma_id,
     dep_bigrams.n1,
   SUM(CASE
     WHEN dep_bigrams.gram LIKE '%nomn%'
     THEN dep_bigrams.count
     ELSE 0 END) AS nomn,
   ...
   SUM(CASE
     WHEN dep_bigrams.gram LIKE '%loct%'
     THEN dep_bigrams.count
     ELSE 0 END) AS loct,
   FROM dep_bigrams
   WHERE dep_bigrams.pos='VERB'
   GROUP BY dep_bigrams.lemma_id;
```

In this example we have six aggregation (sum) functions (one for each grammatical case, e.g. 'loct' for the Locative). Each aggregation function in the query calculates total amount of dependency links between verbs given a lemma_id and arbitrary word forms in a certain grammatical case. We apply the same technique when generating subcategorization frames of a preposition from a 3-gram dataset. Corresponding SQL-queries for prepositions differ only in the WHEN-condition and GROUP BY operator that include additional restriction on the second word in a 3-gram.

3. User interface for pattern and dictionary construction

For managing our dictionary we developed a user interface shown in Fig. 1. Existing verbs can be edited and new verbs can be added. In a simple tabular interface the user can set preposition and grammatical case of the argument. The interface also allows defining non-linear

* URL: https://books.google.com/ngrams

extraction patterns by selecting the type of an event argument. The type of event can be chosen from a drop-down in the top bar. The panel below shows argument types for the event type. For all 18 types of events we defined about a hundred indicators (verbs) and about 180 patterns for argument extraction.

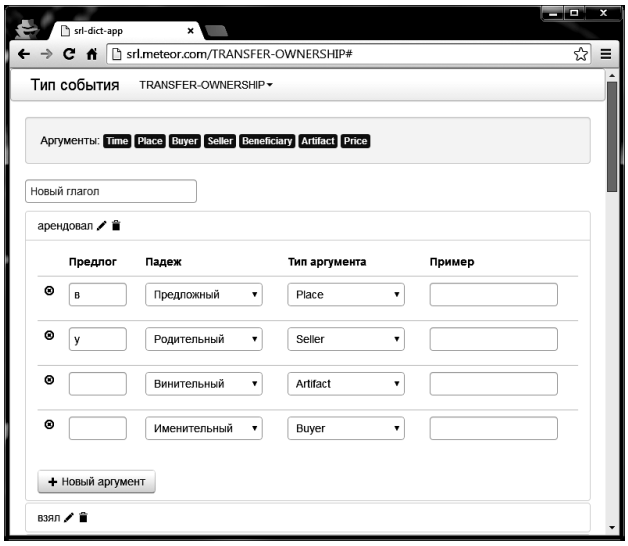

Fig.1. A simple user interface for event definition

4. Results and future work

We have run two types of queries described in previous section against the whole Google Books Ngram dataset. We have got about 24 thousand rows (one row per verb) from the dataset of dependency pairs and about 51.5 thousand rows from the dataset of 3-grams (a verb + preposition + case per row). Some samples from the resulted dictionary are provided in Table 2 and Table 3. The interesting result is that many verbs can subordinate words in almost any grammatical case. This result differs significantly from the ones presented in [4]. We cannot consider this as an error of our calculation or the parsing method but rather as an effect of variations in sense of the verb. It might be useful to compare our dictionary to the one generated from a web corpus [4].

Next, on the basis of constructed dictionary we developed an algorithm for events and arguments extraction. For this purpose we use our CRF-based implementation of Russian PoS-tagger (its accuracy is about 93%). For dependency parsing we use Minimum-Spanning Tree Parser [7] trained on the SynTagRus corpus**. The whole program was designed as Apache UIMA*** pipeline, so for the event extraction task we developed UIMA annotator that uses the results of previous annotators (PoS-tags and dependencies) and finds events and their arguments by matching templates from our dictionary to them.

In order to measure the effectiveness of our extractor, we constructed some test corpus containing 10 news articles and run our algorithm on it. Next, we manually counted true positive, false negative and false positive counts for events and arguments (we considered only arguments of correctly recognized events). Resulted

Table 2

A part of generated dictionary for few frequent Russian verbs

Verb	Main case	Genitive	Dative	Accusative	Ablative or Instrumental
Сказать	Dat.	0.183	0.573	0.057	0.133
Дать	Dat.	0.194	0.511	0.252	0.025
Говорить	Dat.	0.192	0.434	0.070	0.166
Писать	Dat.	0.207	0.389	0.174	0.123
Указать	Dat.	0.216	0.377	0.338	0.056
Изменить	Acc.	0.131	0.338	0.352	0.115
Объяснить	Ablt. or Instr.	0.093	0.292	0.113	0.489
Читать	Acc.	0.196	0.198	0.449	0.102

Table 3

Government of prepositions for the verb "купить" (to buy)

Verb	Prep.	Main case	Genitive	Dative	Accusative	Ablt. or Instr.	Locative
купить	для	Gent.	1.0	0.0	0.0	0.0	0.0
купить	из	Gent.	1.0	0.0	0.0	0.0	0.0
купить	без	Gent.	1.0	0.0	0.0	0.0	0.0
купить	до	Gent.	1.0	0.0	0.0	0.0	0.0
купить	с	Gent.	0.595	0.0	0.0	0.405	0.0
купить	в	Loc.	0.0	0.011	0.068	0.0	0.921
купить	за	Ablt. or Instr.	0.0	0.0	0.393	0.607	0.0
купить	к	Dat.	0.0	1.0	0.0	0.0	0.0
купить	на	Loc.	0.0	0.049	0.138	0.005	0.808
купить	по	Dat.	0.0	1.0	0.0	0.0	0.0
купить	под	Ablt. or Instr.	0.0	0.0	0.0	1.0	0.0
купить	со	Ablt. or Instr.	0.0	0.0	0.0	1.0	0.0

** URL: http://testsynt.soiza.com
*** URL: http://uima.apache.org

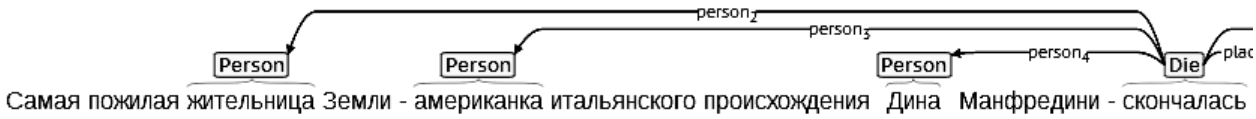

Fig.2. Example of a composite argument extraction

F_1-score was 0.57 for events and 0.62 for arguments which we consider to be an acceptable result at this stage of work. One of the interesting results we discovered was the ability of our algorithm to discover composite entities composed of multiple words (see Fig. 2).

As causes of low F_1-score we can name a few. First, the current dictionary is not full and needs more expert work, especially with regard to a number of event indicators. Second, we do not consider composite event indicators. We can do this by using the method proposed in [8]. Many errors were produced due to ambiguity and lack of semantic context interpretation.

1. Doddington G. R., Mitchell A., Przybocki M. A., Ramshaw L. A., Strassel S., Weischedel R. M.. The automatic content extraction (ACE) program - tasks, data, and evaluation. Proc. of the 4th Int. Conf. on Language Resources and Evaluation. Lisbon, 2004, pp. 837-840.
2. Dzendzik D., Serebryakov S. Semiautomatic generation of linear event extraction patterns for free texts. Proc. of the 9th Spring Researchers Colloquium on Databases and Information Systems (SYRCoDIS 2013). Kazan, Russia, 2013, vol. 1031, pp. 5-9.
3. Solovyev V. et al. Methodology for Building Extraction Templates for Russian Language in Knowledge-Based IE Systems. Technical Report HPL-2012-211, HP Laboratories.
4. Klyshinskii E. S., Kochetkova N. A. Metod avtomaticheskoi generatsii modeli upravleniia glagolov russkogo iazyka [Method for automatic generation of Russian verbs control model]. Trinadtsataia natsional'naia konferentsiia po iskusstvennomu intellektu s mezhdunarodnym uchastiem KII-2012 [Proc. of the 13th Nat. Conf. on Artificial Intelligence with international participation (CAI 2012)]. Belgorod, Russia, 2012, vol. 2, pp. 227-235.
5. Granovsky D. V. Protopopova E. V. Stepanova M. E. Surikov A. V. Bocharov V. V., Alexeeva S. V. Crowdsourcing morphological annotation. Proc. of the Annual Int. Conf. Dialogue 2013 on Computational Linguistics and Intellectual Technologies. Bekasovo, Russia, 2013, iss. 12, vol. 1, p. 109.
6. Lin Y., Michel J. B., Aiden E. L., Orwant J., Brockman W., Petrov S. Syntactic annotations for the google books ngram corpus. Proc. of the ACL 2012 System Demonstrations, ACL '12. Stroudsburg, PA, USA, 2012, vol. 2, pp. 169-174.
7. McDonald R., Pereira F., Ribarov K., Hajiv J. Non-projective Dependency Parsing using Spanning Tree Algorithms. Proceedings of HLT/EMNLP, 2005.
8. Solovyev V., Ivanov V. Composite Event Indicator Processing in Event Extraction for Non-configurational Language. MICAI 2013, Part 1, LNAI 8265, pp. 329-341. Springer-Verlag Berlin Heidelberg, 2013.