Named Numeric Characteristics Extraction from Text Data in Russian

Ivan Uskov[1][0000−0002−7734−6531], Alexander Yarkeev[1][0000−0001−9682−7253], and Evgenij Tsopa[1][0000−0002−7473−3368]

Faculty of Software Engineering and Computer Systems, ITMO University, Saint Petersburg, Russia
https://en.itmo.ru/en/
{ivan.uskov,alexander.yarkeev,evgenij.tsopa}@cs.ifmo.ru

Abstract. The article is focused on the problem of named numeric characteristics extraction from the text data in Russian. This problem have a significant impact to the wide variety of natural language processing tasks such as marketing analysis, statistic tools and data aggregation solutions. There are a lot of existing approaches such as scrappers and parsers, various natural language processing tools (Stanford CoreNLP, spaCy, Natural language toolkit, Apache OpenNLP) and Tomita parser from Yandex that are considered in the article. The structure of the numerical data in Russian has its own specific that have an impact to the algorithms of converting numbers from text form to their value that is also covered in the article. As a result of the research, the new method for extracting numerical data from texts in a natural language was proposed and software module for the proposed method proof was developed. The proposed solution uses semantic networks and semantic frames to determine the boundaries and extract the numerical data from the text. The developed software module was tested on a variety of data sets extracted from the different sources such as Avito and Yandex.Market. The results of the testing shows the effectiveness of the proposed method in comparision with existing solutions.

Keywords: Semantic networks · Semantic frames · Natural language processing.

1 Introduction

The data processed by computers is heterogeneous so there are a lot of different approaches to process it. A large amount of data is structured using various databases, files in certain formats, etc. The structure of this data is pre-calculated for processing by certain programs that allows to simplify the creation of software products.

Nevertheless, huge amounts of data are loosely structured so there is a special class of tasks focused on data preprocessing for structurization to solve this problem. The canonical example of such data is website content. The structure of web pages is optimized for data displaying and isn't intended for data extraction and processing. Partial solution of this problem is based on the principle that every web page has some structure, so it's possible to use patterns to extract the necessary data from it. CSS classes, semantic markup, XPath's can be used as some kind of pointers to data elements. Software products for data processing from unstructured sources is very complicated, because they need a special parser programs to extract the necessary data from sources. These parsers are not so complicated, but programmer should write parser for every data source and these parsers require permanent support because the web site structure tends to permanent changes.

The data examples described above have some structure that simplifies its processing. But large amounts of data are stored in an unstructured form: texts, internal documents of companies (reports, plans, etc.), news articles, and so on. Processing of these data requires special approaches, and these approaches creation task is still actual.

One of the most important automatic text processing tasks is named numeric characteristics extraction. This task solution can be useful in many areas: marketing (competitors' offers analysis), statistics (numerical data extraction from various natural language sources), data aggregators (for example, a site that collects information about the characteristics and prices of goods from different online stores) and many others.

A numerical characteristic is an object characteristic that can be expressed as numbers and units of its measurement. In a natural text, these characteristics can have one of these forms:

- The object the characteristic relates to.
- Name of the characteristic (can be implicit).
- Numerical value of the characteristic.
- Measurement units of the characteristic.

A typical structure for numerical characteristics representation is shown on the figures 1 and 2.

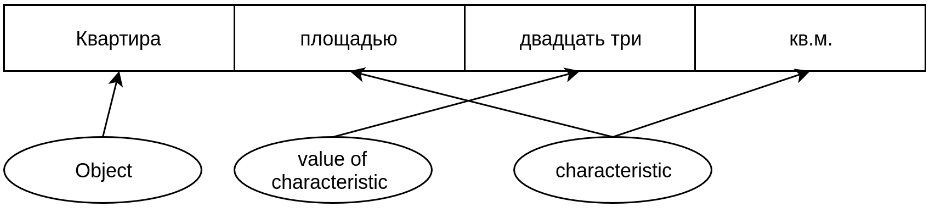


Fig. 1. Numerical characteristics representation structure example

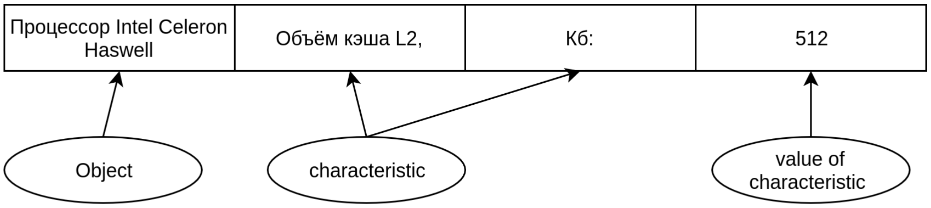


Fig. 2. Yet another example of the numerical characteristics structure representation

2 Targets and goals

The goal of the work is to develop a solution for numerical data extracting from natural language. Natural language processing is a long-studied problem, and there are a lot methods and algorithms developed in this area. Therefore, the first task is to observe existing approaches applicable to achieve the goal. If there is a suitable solution, it must be adapted. Otherwise, suitable approach must be developed.

3 Existing solutions review

These approaches and tools were investigated:

1. Scrappers and parsers.
2. Various natural language processing tools (Stanford CoreNLP [1], spaCy, Natural language toolkit [2], Apache OpenNLP).
3. Tomita parser.

Each of these solutions has a number of disadvantages make it inapplicable for the problem solve:

1. Scrappers and parsers are based on structure or metadata (for example, HTML page layout), so they are unsuitable for natural language analysis.
2. Natural language processing tools possess a search mechanism for named entities, but there is no any numerical data-like entity implemented in these tools.
3. Tomita parser doesn't contain any mechanism for hierarchical concepts combination. Because of this it's impossible to use it to formulate a request "all areas of residential premises" that will automatically include "all areas of apartments", "all areas of houses", "all areas of estates", etc.

Since no suitable solution was found, it became necessary to develop the new one. The semantic networks and semantic frames were taken as its basis [3] [4] [5].

4 General provisions of the algorithm

To analyze the text, semantic frames based on the Fillmore frame semantics were used [6] [7]. This approach is based on the concept that a specific fact is described by a set of lexical and semantic units in the text. Frames also allow to extract the data (text) enclosed between these units [8]. An example of such frame is shown on the figure 3. As you can see, the presence of certain semantic and lexical units is optional.

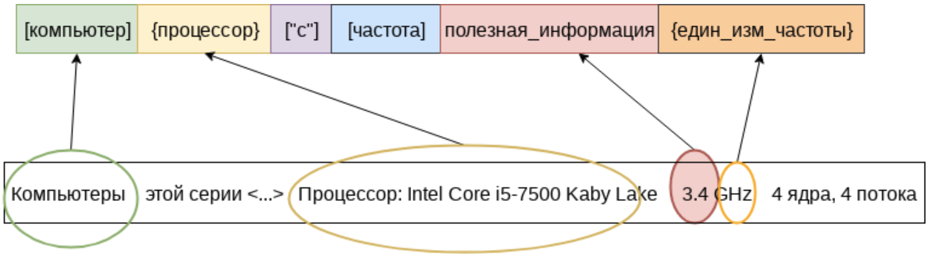


Fig. 3. An example of a frame and its mapping to text

The semantic frame works with tokenized text and consists of the following units:

1. Lexical unit (lex_unit) - describes a single string value.
2. The semantic unit (semantic_unit), describing the value represented by the node of the semantic network. Comparison of the token for coincidence is made with all word forms of this node. Two subtypes of units are distinguished depending on the target set of word forms:
   - Single - the target set of word forms is taken from the specified network node.
   - Hyponymic - the target set is taken from the specified node and all its hyponyms [9].
3. Information unit (payload_unit). The purpose of this unit is not to check for coincidence, but to remember all tokens. This is used to extract text data enclosed between a pair of semantic units (lexical or semantic). Because of this it's possible to extract data from the text depending on its semantic structure.

Thus, semantic frames allow us to localize the position of numerical data in a natural language text. So the task is reduced to:

1. Fill the semantic network with data on various units of measurement.
2. Compile the semantic frames for the numeric data representation options in the specified text.
3. Select the search algorithm and normalize the numbers in a localized area (translate number from text representation form into its value).

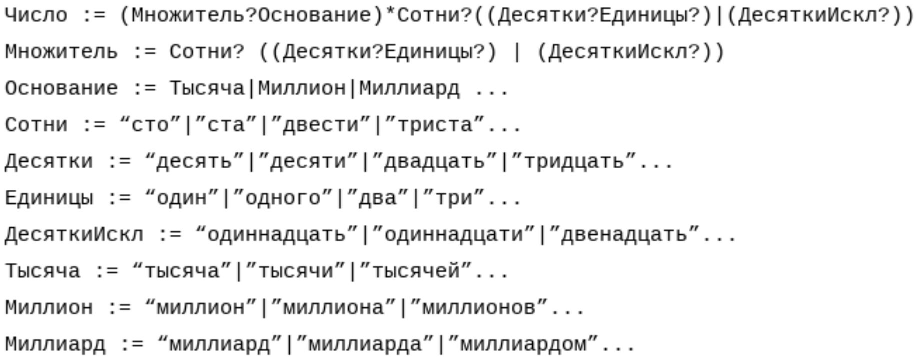
5 Implementation

Examples in figures 1 and 2 could be described by two semantic frames, that contains the object related by the characteristic and the measurement unit of this characteristic.

Hyponymic semantic units should be used to describe an object and a measurement unit, and information units should be used to extract the suitable text. As a result of the frame work, a piece of text with numerical data will be extracted by information unit. This text can contain some redundant text, so it should be filtered out.

The tool or algorithm for recognizing and converting numbers from a text format should work with the Russian language. The final solution must work with a natural text in Russian. There is no any existing solution distributed under a free license that is meet for this requirement.

The algorithm for Russian numbers recognition and conversion is based on the fact that compound numerals (expressing numbers) in the Russian language have a strictly defined structure shown on the figure 4. The final algorithm is based on this structure.



```
Число := (Множитель?Основание)*Сотни?((Десятки?Единицы?)|(ДесяткиИскл?))
Множитель := Сотни? ((Десятки?Единицы?) | (ДесяткиИскл?))
Основание := Тысяча|Миллион|Миллиард ...
Сотни := “сто”|”ста”|”двести”|”триста”...
Десятки := “десять”|”десяти”|”двадцать”|”тридцать”...
Единицы := “один”|”одного”|”два”|”три”...
ДесяткиИскл := “одиннадцать”|”одиннадцати”|”двенадцать”...
Тысяча := “тысяча”|”тысячи”|”тысячей”...
Миллион := “миллион”|”миллиона”|”миллионов”...
Миллиард := “миллиард”|”миллиарда”|”миллиардом”...
```

Fig. 4. The structure of numbers in Russian

6 Results

Suggested approach was tested on datasets from one of the largest online retail platform in Russia Yandex.Market and classified advertisements portal Avito. Both of these resources contains a sets of good test samples of objects with numeric characteristics. The difference between these two resources is that Yandex.Market contains poorly-structured data while data in Avito is completely non-structured.

Test datasets were imported from the resources using two different ways. Yandex.Market data was imported using the scrapper module extracting the product characteristics block from the page. Avito data was imported using parser extracting the advertisement source. There were a couple of subject areas-related datasets selected for the testing purposes:

1. Computer equipment
   - Processors
   - RAM
   - Power supplies
2. Apartments
3. Video cameras
4. Coffeemakers

Characteristics of computer parts were taken for "computer equipment" subject area. These characteristics contain a processor, a random-access memory and a video card.

The following numeric object characteristics were chosen for recognition in text data:

Table 1. Objects and characteristics.

Object	Characteristic	Measure unit
Processor	Frequency	Hz
	MOSFET scaling	nm
	Cache size	MB
RAM	Frequency	Hz
	Memory size	GB
Power supply	Power	W
Apartment	Area	M^2
	Cost	Rub
Video camera	Image sensor	MP
Coffeemaker	Volume	L
	Pressure	Ba
	Weight	kg

There were separate semantic networks developed for the each subject area domain and its characteristics. Each semantic network was built by the linguist expert based on different sources. There was the global semantic network based on Wiktionary [10] and RuThes [11] translingual data used as the main semantic data source [12] [13]. Usage of this network allowed to solve these two important problems:

- Eliminates the need to set all word forms manually for each node of the semantic network.
- Allows to use nodes and relations already existing in the global network and modify it for the purposes of the specific task.

Separate subsets of semantic frames were built for each subject area domain. The resulting semantic network and frames were used as an input data for the software module implements the developed algorithm.

The main characteristics of the test samples used are number of test samples for each domain and number of values for each characteristic. These characteristics are shown in the table 2.

Table 2. Test data.

Object	Number of samples contains objects	Characteristic	Number of samples contains characteristic
Processor	200	Frequency	200
		MOSFET scaling	165
		Cache size	200
RAM	200	Frequency	180
		Memory size	195
Power supply	80	Power	65
Apartment	150	Area	145
		Cost	90
Video camera	80	Image sensor	70
Coffeemaker	40	Volume	25
		Pressure	20
		Weight	15

There were precision, recall and F1-score metrics values taken as testing results. The average values of these metrics values are shown in the table 3.

Table 3. Testing results.

Object	Precision	Recall	F_1
Processor	0.705	0.691	0.693
RAM	0.840	0.875	0.856
Power supply	1	0.677	0.807
Apartment	0.734	0.607	0.654
Video camera	0.942	0.929	0.935
Coffeemaker	0.728	0.730	0.723

Testing results proved the high efficiency of the numeric characteristics extraction using the proposed approach. The average value for the F1-score was 78% for the used data sets.

Nonetheless, there are some improvements in the algorithm and software module that are needed to widely use it in practical tasks. At the moment there are some limitations in the software module abilities to recognize the float numbers. Also there are some troubles with relatively rarely used untypical numbers formats (e.g. word "thousand" is implied but not used in text). This is the area of further development of the system. Also it should be noted that the effectiveness of the numbers extraction highly depends on the quality characteristics of the used semantic network and frame set.

References

1. Manning C. et al. The Stanford CoreNLP natural language processing toolkit //Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. – 2014. – Pg. 55-60.
2. Bird S., Loper E. NLTK: the natural language toolkit //Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. – Association for Computational Linguistics. – 2004. – Pg. 31.
3. Bessmertny I., 2010. Knowledge visualization based on semantic networks //Programming and Computer Software. – 2010.
4. Francopoulo G. (ed.). LMF Lexical Markup Framework. – 2013.
5. Eckle-Kohler J. et al. lemonUby – A large, interlinked, syntactically-rich lexical resource for ontologies //Semantic Web. – 2015. – Vol. 6. – №. 4. – Pg. 371-378.
6. Fillmore C. J. Frame semantics and the nature of language //Annals of the New York Academy of Sciences. – 1976. – No. 1. – Pg. 20-32.
7. Fillmore C. J., Baker C. F. Frame semantics for text understanding //Proceedings of WordNet and Other Lexical Resources Workshop. – 2001.
8. Barsalou L. W. Frames, concepts, and conceptual fields //In Frames, fields, and contrasts, ed. Adrienne Lehrer and Eva Feder Kittay. – 1992. – Pg. 21–74.
9. Stern D. Making Search More Meaningful: Action Values, Linked Data, and Semantic Relationships. – 2015.
10. Krizhanovsky A., Smirnov A. An approach to automated construction of a general-purpose lexical ontology based on Wiktionary //Journal of Computer and Systems Sciences International. – 2013. – Pg. 215-225.
11. Loukachevitch N., Dobrov B. RuThes linguistic ontology vs. Russian wordnets //Proceedings of Global WordNet Conference GWC-2014. – 2014.
12. Klimenkov S. et al. Reconstruction of Implied Semantic Relations in Russian Wiktionary //Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (KDIR). – 2016. – Pg. 74-80.
13. Osika V. et al. Method of Reconstruction of Semantic Relations using Translingual Information //Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. – 2017. – Vol. 2. – Pg. 239-245.