UDK: 811.163.41’322.2
004.9:811.111’374

TRANSDUCERS FOR ANNOTATING WEATHER INFORMATION IN METEOROLOGICAL TEXTS IN SERBIAN

Vesna Pajić, svesna@agrif.bg.ac.rs, University of Belgrade, Faculty of Agriculture, Departmant for Agricultural Engineering, Nemanjina 6, 11080, Zemun, Belgrade, Republic of Serbia
Staša Vujičić Stanković, stasa@matf.bg.ac.rs, University of Belgrade, Faculty of Mathematics, Studentski trg 16, 11000, Belgrade, Republic of Serbia
Miloš Pajić, paja@agrif.bg.ac.rs, University of Belgrade, Faculty of Agriculture, Departmant for Agricultural Engineering, Nemanjina 6, 11080, Zemun, Belgrade, Republic of Serbia

Abstract
We present a process of extracting information on meteorological phenomena from texts in Serbian. We used finite state automata and transducers for both text processing and information extraction, through software specialized for linguistic text processing. Information extraction was done by annotating text segments. The extraction rules were described with transducers (finite state transducers and recursive transition networks). Some details of used transducers are presented in this paper, aiming to demonstrate the application of different electronic resources for Serbian, especially the electronic morphological dictionary. Transducers are very efficient tools for language processing. In the case of processing Serbian, it is very important to create different resources and corpora which could allow linguistic research. Therefore, we plan to form a collection of transducers and make it publicly available for different kinds of research in the computational linguistics domain.

Keywords: Information Extraction, Serbian, Natural Language Processing, Finite State Transducers, Recursive Transition Networks

1. Introduction

Information extraction is a subfield of artificial intelligence that studies and develops techniques for discovering and extracting relevant information from large documents. Today, there are huge collections of text documents containing different types of human knowledge in the form of various textual descriptions. Such information is difficult to find and use. Developing intelligent tools and methods that would enable access to information from the text is of great importance for effective management of human knowledge. It is information extraction that tries to explore different methods and features to access textual information more efficiently. The ultimate goal of this scientific discipline is the representation of extracted information in a structured form, suitable for further computer processing and analysis.

Information extraction area was originally developed in the series of conferences called *Message Understanding Conference* (MUC), organized by *Defense Advanced Research Projects Agency of the USA* (DARPA) (Grishman and Sundhaim, 1996) in the late 80's and early 90's. During these conferences, the main goal was to extract information that describes events, specifically the terrorist attacks in Latin America. Thus, the majority of research in this area generally referred to named entity extraction from texts, such as proper names, place names, etc. (Friburger and Maurel, 2004; Gucul-Milojevic, 2010, Maynard et al., 2003; Sekine and Ranchhod, 2009), especially in the beginning. Today, information extraction methods are being used in other areas as well, for extracting different types of information (Burns et al., 2008, Feng et al., 2007; Goh et al., 2006, Jim et al., 2004; Korbel et al., 2005; MacDonald and Beike, 2010; Tamura and D'haesleer, 2008).

Among others, texts on the topic of weather conditions in a narrower or wider locality have been studied over the years in this and other related disciplines (Slocum, 1985; Kononenko et al., 1999; Kononenko et al., 2000; Matetic and Brkic, 2007; Labsky et al., 2007). These kinds of texts are interesting because of their properties on one hand, and the possibility of different usage of data, on the other hand. The obtained data are often used for other information systems, such as automatic translation from one language to another, data visualization, to merge data from multiple sources, and similar.

As a part of this research, we will present a process of data extraction on weather conditions from texts in Serbian, which can be used for different purposes (for example, to automatically create a lexicon or for automatic annotation of text).

2. Textual corpus of meteorological descriptions

Articles on weather conditions were collected during the years 2010, 2011 and 2012, from several electronic sources (*Republic Hydrometeorological Service of Serbia*[1], *Meteos*[2] agency, daily newspaper *Politika*[3], *B92*[4], *SMedia*[5] and web portal *Krstarica*[6]). We downloaded 13,705 text descriptions with a total of 45,862 sentences. All descriptions were stored in a relational database, from which a single text file was created, containing textual descriptions for analyzing and processing.

Most downloaded articles contained weather forecasts for one or more days, but there were a few texts with observed data. Since the articles were downloaded from several sources, their structure was quite heterogeneous. However, there were certain regularities pertinent to all of these texts, no matter what source they came from.

Here is an example of several descriptions of weather conditions:

1. Oblačno i hladnije, povremeno sa kišom koja će krajem dana preći u susnežicu i sneg. Vetar slab i umeren severni i severozapadni. Jutarnja temperatura oko 3 °C, najviša dnevna oko 6 °C, tokom noći u padu. (*Cloudy and cold, with occasional rain that will change to sleet and snow at the end of the day. Wind weak to moderate, north and northwest. Morning temperature around 3 °C, the highest daily around 6 °C, at*

1 http://www.hidmet.gov.rs
2 http://www.meteos.rs
3 http://www.politika.rs
4 http://www.b92.net
5 http://www.smedia.rs
6 http://www.krstarica.com

night in the fall.)

2. U Srbiji danas pretežno sunčano, posle podne u brdsko-planinskim predelima umereno oblačno. Vetar slab, severni. Maksimalna temperatura od 16 do 23 °C. (*Today in Serbia mostly sunny, in the afternoon in the mountainous regions moderately cloudy. Weak north wind. Maximum temperature from 16 to 23 °C.*)

3. Narednih dana pretežno sunčano, temperatura oko 20 °C. (*In the next few days mostly sunny, temperature around 20 °C.*)

The descriptions of weather conditions are very specific and easily recognizable. The limited set of words of a natural language (in this case Serbian, but similar applies to other natural languages) that is used to describe the weather phenomenon can be seen as a sublanguage of natural language, along with its characteristics:

- Limited vocabulary – the same words are used for describing a phenomenon in most of the descriptions, with no synonyms or different phrases. Thus, it is common to use words *promenljivo* (variable) or *nestabilno* (unstable) for unstable weather conditions, instead of some other synonyms (*nestalno, varijabilno)*.
- Disregarding grammar rules and the syntax of a natural language – phrases and statements in the weather descriptions usually do not include auxiliary verbs, and often do not have a predicate (*Vetar slab, jugoistočni*. – *The wind is weak, southeast.*) or articles.
- Structure of the text – it is not possible to clearly separate statements on the basis of the punctuation; a sentence often contains several statements, or a few sentences (statements) merged into one with commas (*U većem delu promenljivo oblačno, mestimično kratkotrajna kiša, pljuskovi i grmljavina, a u oblasti Sredozemlja i Crnog mora pretežno sunčano i toplo*. – *In most parts variably cloudy, locally depleting rain, severe rain and thunder, in the area of the Mediterranean and Black Sea mostly sunny and warm*.)

On the one hand, the existence and usage of such a sublanguage facilitate the text processing, simply because many of the syntax rules are simplified in relation to natural language. On the other hand, the disregard of natural language syntax rules makes already existing electronic resources and grammars, developed and available for some natural language, useless and inapplicable.

3. Techniques and tools used for detecting and annotating information in the text

The goal of the extraction process was to annotate pieces of information that are contained in a text description. The three types of information were of interest: location, time, and weather phenomena. All pieces of information found in texts were annotated and structured. The data were not transformed into any other data formats (for example, a relational database), since this procedure is trivial, and it depends on further research needs. The rules for annotating information were given by finite state transducers.

3.1 Finite state transducers in natural language processing

Finite state transducers are finite state abstract machines that define a relationship between two sets of strings in the sense that they are able to transform one string into another. Formally, the *finite state transducer* (FST) is defined as a 6-tuple $\tau = (\Sigma_1, \Sigma_2, Q, i, F, \Delta)$, where:

- Σ_1 and Σ_2 are an input and output alphabet
- Q is a finite set of states
- $i \in Q$ is an initial state
- $F \subset Q$ is a set of final states
- $\Delta \subset Q \times \Sigma_1 \times \Sigma_2 \times Q$ is a transition relation, whose elements are called *arcs*.

Finite state transducers have been used in almost all areas of computer science for many years, and have a special role in computational linguistics. Their use is justified from the standpoint of linguistics, as well as computer science. From the linguistics point of view, finite state transducers are adequate as a means to describe a relevant local phenomenon of a language, as

well as to model a part of a natural language (its phonology, morphology, syntax, etc.). Some examples of adequate representation of different linguistic phenomena with finite state machines are given in (Gross and Perrin, 1987). From the standpoint of computer science, the use of finite state machines is motivated by their time and space efficiency. Time efficiency is achieved by using deterministic finite state machines. The output of this class of machines depends mostly on the size of input data, so they are considered to be optimal (Jurafsky and Martin, 2000; Vitas, 2006). The space efficiency is achieved by the minimization of deterministic machines.

The transducers' main characteristic, which distinguishes them from the rest of finite state machines, is their ability to produce an output. It is this feature that determines how finite state transducers are used in natural language processing. Also, the finite state transducers can be represented by graphs, which makes them very comfortable to use. Finite state transducers are being used in computational linguistics for morphological parsing, describing the spelling rules, word formation, describing inflectional rules, etc. A detailed overview of the theoretical and practical use of finite state transducers in natural language processing can be found in (Casacuberta et al., 2005; Friburger and Maurel, 2004; Hobbs et al., 1997; Jurafsky and Martin, 2000; Kornai, 1999; Krstev, 2008; Pajic, 2010; Pajic, 2011; Pajic et al., 2011; Pajic et al., 2011b; Roche, 1999; Roche and Schabes, 1997; Vitas, 2006).

Finite state transducers could be complex and complicated to create and modify, which in practice leads to significant problems. For example, if someone tries to describe the syntax of a natural language using a finite state transducer, the corresponding graph would be huge and immense. Finding a specific piece of information in it, such as the part that describes the syntax of noun phrases, would be practically impossible. Therefore, in practice, instead of one large graph, we use a collection of smaller subgraphs. This approach has its theoretical background in the theory of *recursive transition networks* (RTN). Recursive transition network is an extension of context-free grammars (Sastre and Forcada, 2007; Sastre, 2009; Vitas, 2006). States in graphs that represent a recursive transition network are usually labeled arbitrarily, only for identification purposes. On the other hand, arcs of the recursive network are labeled in accordance with the grammar symbols or subgraphs, which are called when the graph passes that arc.

In this and similar types of research, it is not of interest whether a single graph or collection of graphs and subgraphs is being used. The important thing is that at the point where the graph reaches its final state, some transformation of an input string is done (translation, insertion of a string, string replacements, etc.). Therefore, we will use the term *transducer* in the rest of the paper, for all abstract machines that are used for transduction, referring to a finite state transducer or a recursive transition network.

There are several software tools and systems designed for linguistic research and natural language processing, which are directly based on transducers (Olivier et al., 2006; Paumier, 2011; Silberztein, 1993). We have chosen the Unitex software system (Paumier, 2011) for text processing and application of extraction rules (application of transducers) in this research.

3.2 *Unitex* as a tool for text processing and application of transducers

Unitex (Paumier, 2011) is a collection of programs developed for the analysis of texts written in natural language using linguistic resources and tools. Resources consist of morphological electronic dictionaries and grammars specially developed for particular languages. This system is open-source. It was designed to be portable, i.e. it can run on different operating systems such as Windows, Linux, MacOS and others. Programs within *Unitex* are written in programming languages *C*/*C*++, while the graphical user interface is written in the *Java* programming language. *Unitex* is designed to support a variety of natural languages.

Morphological electronic dictionaries in *Unitex* are in the DELA format (Silberztein, 1993). The dictionaries were constructed by appropriate teams of linguists for each particular language e.g., for English (Chrobot et al., 1999; Klars-

feld and Hammany-McCarthy, 1991; Monceaux, 1995; Savary, 2000), for French (Courtois, 1996; Silberztein and Courtois, 1990; Labelle, 1995)). Morphological electronic dictionary for Serbian (Vitas and Krstev, 2005; Vitas et al., 2003; Krstev, 2008) contains Serbian words, along with proper nouns, classified as simple or compound words. It was created by researchers at the Human Language Technology Group at the Faculty of Mathematics, University of Belgrade. It included a total of 128,327 lemmas of simple words and 4,484,431 inflected forms of simple words, as well as 9,598 compound words' lemmas and 186,114 appropriate compound words' forms (as reported in June 2012). There are over 35,000 lemmas of proper names (geopolitical terms, the Serbian people's names, foreign names of people and encyclopedic knowledge).

Each dictionary entry represents a word form, its lemma, and the codes that indicate different grammatical categories (parts of speech, gender, number, etc.), as well as various codes that indicate derivational, syntactic, semantic, or other characteristics of the lemma. The following is a lexical entry example:

Evropi,Evropa.N+NProp+Top:fs3q:fs7q

The string *Evropi* denotes the form of the word, *Evropa* is the corresponding lemma, *N* indicates that it is a noun, *NProp* that it is a proper noun, and *Top* denotes a toponym. Codes *fs3q* and *fs7q* denote properties of inflectional word forms more closely (*f* – feminine, *s* – singular, *3* – the third case, *q* – inanimate). A list of all grammatical categories that are used in morphological electronic dictionary of Serbian, together with a list of codes are given in (Krstev, 2008). For annotating proper nouns, a set of tags is based on the (Grass et al., 2002).

The existence of this type of information in dictionaries allows usage of the lexical masks within Unitex to refer to items from the morphological electronic dictionary. For example, the lexical mask <*N*+*NProp*+*Top*> corresponds to all inflectional word forms that are marked with these codes, i.e. all nouns (*N*), more precisely (*NProp*), marked as toponyms (*Top*). These masks are used in the extraction process related to weather conditions, to determine the location of the weather phenomenon occurrence, mentioned in the text. Without the dictionaries, effective extraction of this type of data would not be possible.

3.3 *Unitex* graphical user interface and creation of graphs

Unitex has a well-developed, easy-to-use and intuitive graphical user interface designed for creating graphs. Each graph consists of an initial state (marked by an arrow symbol), a final state (marked by a square symbol) and an arbitrary number of boxes which correspond to transitions of automata or transducers. More details about the *Unitex* system can be found in (Paumier, 2011).

The functionalities of graphs are achieved by entering different labels and symbols in different boxes. Thus, for example, Figure 1 shows the following sequence:

ja+ti+on+ona+ono+mi+vi+oni+one+ona

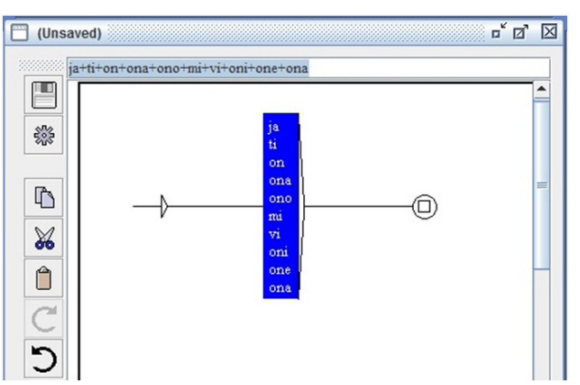

Figure 1. Creating a box that contains more than one state transition word.

The left mouse button's click on the initial state of the graph and then on the box creates the transition between the state and the box. Graph

created in this way, presented in Figure 1, recognizes personal pronouns' nominative in Serbian. This graph corresponds to the regular expression *ja|ti|on|ona|ono|mi|vi|oni|one|ona*.

By entering different values in the box, different transitions and thus different types of graphs are obtained. The following types of labels could be entered into the box:

- Lexical masks – are marked with symbols < and >. It can relate to a dictionary reference or can contain special characters. For example, lexical mask *<pevati.V>* coresponds to all forms of the verb *pevati* (*sing*). Some of the special symbols are *<TOKEN>* (any token – the basic unit of text, usually a word or a digit,), *<MOT>* (words consisting only of letters), *<MAJ>* (words written in capital letters), *<NB>* (uninterrupted sequences of digits) and others.

- Morphological filters – denoted by the symbols << and >>, within which a regular expression for describing a class of tokens is given. The POSIX syntax is used for describing the regular expression (Laurikari, 2009). Some possible filters are *<<izam$>>* (all words that end with the *izam* string) *<<^a>>* (words beginning with *a* character, *<<a.*s>>* (words that contain a character *a*, followed by any sequence of characters and ending with a character *s*).

- Subgraph calls – represented with the name of the graph that is being called, preceded with a character *:* (for example, the input sequence *alpha+:beta+gamma+:E:\greek\delta.grf* in the box recognizes strings *alpha* and *gamma*, as well as the strings identified by subgraphs *beta* and *delta*; it is expected that the subgraph *beta* is in the same directory as the main graph, while the *delta* subgraph is specified with an appropriate absolute path on the disk). It is possible to specify the full path to the graph, to use a relative path or to use the so-called graphs' repositories (special directories that contain collections of graphs, defined at the level of *Unitex*, see (Paumier, 2011))

Transducer's outputs – assignment of an output to a box creates a graph that corresponds to a transducer. In order to define the output, the character / is used. All characters to the right of that character represent the transducer's output. For example, the graph in Figure 2 is the result of attributing sequence *jedan+dva+tri/broj* (*one+two+three/number*) to the box.

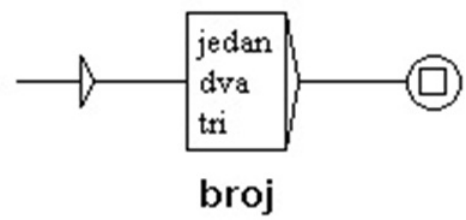

Figure 2. Transducer that recognizes the words *jedan*, *dva* or *tri* (*one*, *two* or *three*) and generates the word *broj* (*number*) as an output.

- Variable – using special boxes, it is possible to mark the beginning and the end of a variable that is generated by passing through a graph. The variable can later be used as an output of the transducer. Variable gets value based on the recognized sequence, i.e. one of its parts, which is defined with the beginning (box contains mark *$var1(*) and the end (box contains mark *$var1)*). Figure 3 shows a graph that recognizes a date format *januar 2012* (*January 2012*) and transfers it to the format *2012. godina, mesec januar* (*2012., the month of January*). For the months' name recognition, the subgraph *mesec* (*month*) is used.

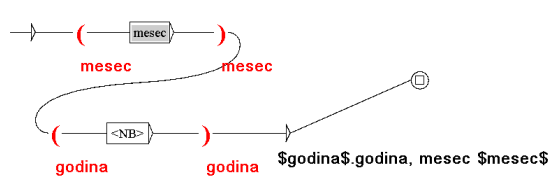

Figure 3. Transducer example with two variables, mesec (month) and godina (year).

All transducers within *Unitex* could be ap-

plied to the text in two modes, the *Merge* mode and the *Replace* mode. In the *Merge* mode, the text that represents the output of the transducer is inserted into the original text, to the left of the identified sequences. In the *Replace* mode, recognized sequences are replaced with the sequences produced by the transducer. For the purposes of this study all transducers are applied in the *Merge* mode, in order to perform annotation of the text.

4. Semantic classes used for structuring information

Information contained in the textual descriptions of weather conditions, which was of interest in this research are grouped into semantic classes of different levels. The semantic classes are associated with the extracted text fragments, while some of them contained an additional classification. For the purpose of this research, we used the class hierarchy shown in Table 1. This hierarchy is created based on the classification represented in (Kononenko et al., 2000), but the classes are modified to correspond the texts that are processed in this research.

In the process of information extraction that will be described in this paper, the goal is to identify segments of text which contain defined characteristics. Annotation of the recognized text segments and assignment of the proper semantic class to it were done by inserting special tags directly in the text.

The semantic classes represented in the column *Features* in Table 1 were used as the annotations' names. Their organization into a higher-level classes and co-reference resolution will be the subject of our further research. The annotations have the following syntax:

<feature> text segment </feature>

Table 1. Class hierarchy used to structure information extracted from the text

Element	Feature	Value examples
Padavine (*Precipitation*)	TipPadavina (*PrecipitationType*)	*kiša, sneg ... (rain, snow...)*
	ObimPadavina (*Precipitation-Amount*)	*slaba, jaka, ... (weak, strong...)*
Oblačnost (*Cloudiness*)	PrisustvoOblaka (*CloudPresence*)	*sunčano, oblačno (sunny, cloudy)*
	ObimOblačnosti (*CloudAmount*)	*promenljivo, potpuno... (variable, fully...)*
Vetar (*Wind*)	PravacVetra (*WindDirection*)	*jugoistočni, severni ... (southeast, north ...)*
	JačinaVetra (*WindAmount*)	*jak, slab... (strong, weak...)*
	BrzinaVetra (*WindSpeed*)	*16 m/s*
Temperatura (*Temperature*)	Temperatura (*Temperature*)	*12 stepeni, 12 C, dva stepena, ispod nule ... (12 degrees, 12 C, two degrees, below zero…)*
	KatTemperature (*Temperature-Category*)	*najviša, jutarnja ... (maximum, morning…)*
	OpisTemperature (*Temperature-Description*)	*hladno, toplije, porast ... (cold, warmer, rising…)*
Pojava (*Phenomenon*)	TipPojave (*PhenomenonType*)	*magla, oluja ... (fog, storm...)*
Teritorija (*Territory*)	ImeTeritorije (*TerritoryName*)	*Srbija, Evropa, Beograd ...*
	DeoTeritorije (*TerritoryPart*)	*severoistok, južni delovi (northeast, southern parts)*
Lokalitet (*Locality*)	Lokalitet (*Locality*)	*na planinama, lokalno... (in the mountains, localy)*
Dan (*Day*)	Datum (*Date*)	*15. januar (January 15th)*
	ImeDana (*DayName*)	*ponedeljak, utorak .. (Monday, Tuesday...)*
	DeoDana (*DayPart*)	*ujutru, posle podne (in the morning, in the afternoon)*
Period (*Period*)	Period (*Period*)	*sledeće nedelje, tokom februara (next week, during February…)*

Hence, the example sentence *U većem delu zemlje promenljivo oblačno, mestimično slaba kiša, pljuskovi i grmljavina.* (*In most parts of the country variable cloudiness, with areas of light rain, showers, and thunder.*), should be annotated as follows:

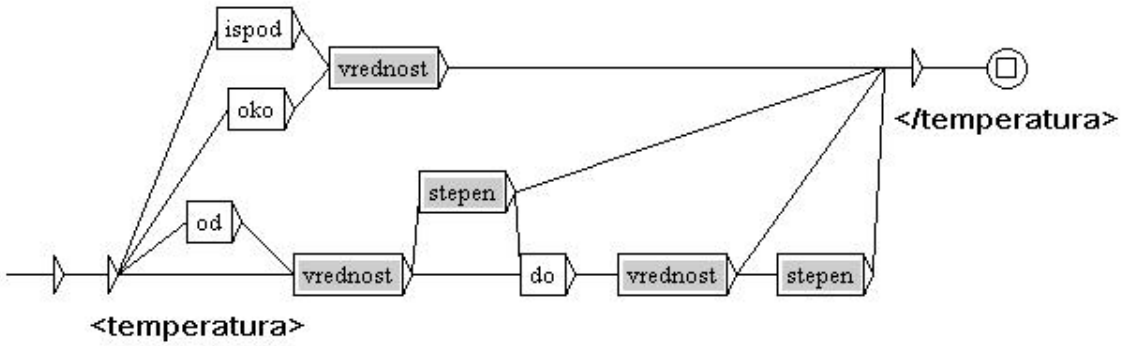

Figure 4. The main transducer temperatura.grf within the recursive transition network for extracting information about the temperature.

*<lokalitet>**U većem delu**</lokalitet>*
*<obimOblacnosti>**promenljivo**</obimOblacnosti>*
*<prisustvoOblaka>**oblačno**</prisustvoOblaka>,*
*<lokalitet>**mestimično**</lokalitet>*
*<obimPadavina>**kratkotrajna**</obimPadavina>*
*<tipPadavina>**kiša** </tipPadavina>,*
*<tipPojave>**pljuskovi**</tipPojave> **i***
*<tipPojave>**grmljavina**</tipPojave>.*

(*<Locality>**In most parts of the country** </Locality>*
*<CloudAmount>**variable**</CloudAmount>*
*<CloudPresence>**cloudiness**</CloudPresence>,*
*<Locality>**with areas**</Locality> of*
*<PrecipitationAmount>**light**</PrecipitationAmount>*
*<PrecipitationType>**rain**</PrecipitationType>,*
*<PhenomenonType>**showers**</PhenomenonType> and*
*<PhenomenonType>**thunder**</PhenomenonType>.*)

5. Transducers - extraction rules

The extraction rules for annotating text segments with the features from the Table 1 were created. The rules are presented as graphs corresponding to finite transducers or recursive transition networks, where in most cases one transducer is corresponding to one class of features. In some cases, when it is more efficient, a transducer is used to extract two features, such as, for example, the graph presented in Figure 7, which is described later in Section 5.2. All transducers are designed and implemented through the software system *Unitex*, and the text structuring was done through annotating of the text segments. If necessary, designed transducers can easily be modified in order to do annotation in some other way, depending on the problem.

The application of the transducer was performed sequentially, one by one. For the most of the created transducers the order of their implementation was not crucial, although it is possible to improve the efficiency of the extraction process by successive application of transducers (the cascade of transducers, where the transducer uses the results of thepreviously applied ones (Friburger and Maurel, 2004)). This section will present some of the used transducers, while Section 5.3 describes the case where the order of the transducer application is important.

5.1 Transducers for extracting information about the temperature

Temperature data have been presented in the texts in two ways:

- Quantitative (*12 stepeni – 12 degrees, 12°C, 12 C, dva stepena – two degrees, ispod nule – below zero, minus 5 ...*) and

- Qualitative (*hladno – cold, hladnije – colder, toplo – warm, toplije – warmer, pad temperature – the temperature drop, temperatura u porastu – the temperature rising ...*).

For each way of representing temperature, a special extraction rule has been created. Figure 4 shows the main transducer (*temperatura.grf*)

in the recursive transition network for extracting information related to the temperature presented quantitatively.

Subgraph calls are marked with gray colour. Subgraph *vrednost.grf* recognizes different expressions for the specific value (number of degrees) of the temperature. This subgraph is shown in Figure 5.

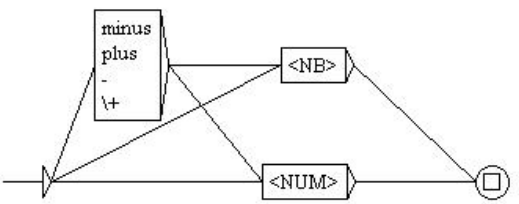

Figure 5. Subgraph vrednost.grf that recognizes numeric values written as numbers or text.

The lexical mask <NB> recognizes successive digits. The lexical mask <NUM> recognizes all the words from the dictionary that are marked with a code NUM (jedan, dva, tri – one, two, three...). Thus, this subgraph recognizes, among others, the following expressions: 10, minus dva (minus two), +5 or jedanaest (eleven).

It should be noted that the graph represented in Figure 5 recognizes the token C as a number, given that the morphological electronic dictionary contains an entry that refers to the Roman numeral C:

C,.NUM+Roman

Since Roman numerals in the morphological electronic dictionary are marked with Roman code, it would be possible to use the mask <NUM~Roman>, for recognizing all the words marked with a code NUM, which are not marked with Roman code. The operator ~ is used in Unitex as a negation of some grammatical code. On the other hand, when the graph vrednost.grf is used in the graph temperatura.grf, and since that in that case the appearance of numbers is requested in a particular context, the Roman numerals do not appear in the results, so the graph vrednost.grf is used in the form presented in Figure 5.

The main transducer temperatura.grf (Figure 4) contains a subgraph call stepen.grf. This graph is intended to recognize expressions that describe the degrees on the Celsius scale, as the common unit of temperature measure, in the texts in Serbian. Subgraph stepen.grf is shown in Figure 6.

The lexical mask which refers to a dictionary word (<*stepen*>) recognizes any form of the word *stepen – degree* (*stepena*, *stepeni*, *stepenima* etc.).

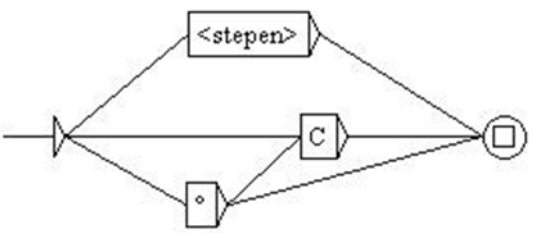

Figure 6. Subgraph stepen.grf that recognizes phrases for marking degrees on the Celsius scale.

Graph *temperatura.grf* recognizes the following phrases:

oko +8 °C (*approximately +8 ° C*)
- 1C
- 30 °C
- 4 stepena (*- 4 degrees*)
-1 C do 1 C (*from -1 C to 1 C*)
-1 do +3 stepena (*-1 to +3 degrees*)
-12 do -8 (*-12 to -8*)
od 11 do 15 stepeni (from 11 to 15 degrees)
11 stepeni (*11 degrees*)
od 15 do 18 (*from 15 to 18*)
od 15 do 19°C (*from 15 to 19°C*)
od pet do devet stepeni (*from five to nine degrees*)
od sedam do 10 stepeni (*from seven to 10 degrees*)
oko +2 (*approximately +2*)
oko četiri (*approximately four*)
oko minus 12 (*approximately minus 12*)
oko plus tri (*approximately plus three*)
ispod 0 (*below 0*)
ispod deset (*below ten*)

Graph *temperatura.grf* takes into account that in many cases weather forecast descriptions do not contain the word *stepen* (*degree*) or any other label for the measure of temperature, but it is implied. Of course, this is not true in general, but in the corpus of meteorological texts analyzed in this study, especially in sentences that specify several values of temperature, there are often situations where the first value specifies the unit of measure, but not the next (for example, *jutarnja temperatura oko 4 stepena, maksimalna dnevna oko 15. – morning temperatures around 4 degrees, the maximum daily around 15*.). This fact complicates the extraction process, because it is necessary to compromise between the situation where part of the information remains unrecognized (if it requires explicit reference to unit measures in order for information to be extracted) or to recognize expressions that do not represent the temperature (for example, *vidljivost je oko 200 m – visibility is about 200 m*).

Initially, the transducer in Figure 4 (temperatura.grf) was created to recognize segments such as preko 30 (over 30). However, the increase in recall that has been achieved was small compared to the decrease of the precision. In fact, there were only three results beginning with the word preko (over), and referring to the temperature, but a very large number (over 200) of the results that were incorrectly recognized and mostly were related to the amount of snow cover (preko 1 m – over 1m) and wind speed (preko 20 m/s – over 20 m/s). Consequently, a branch of the graph that starts with the word preko (over) has been removed. On the other hand, the branch that starts with the word oko (approximately) was retained, because it returned 9,564 results, of which only 16 incorrectly recognized (oko 17 m/s – approximately 17 m/s, oko 30 l/m^2“ approximately 30 l/m^2 etc.).

This kind of adjustments of the transducers is possible based on the analysis of the texts to be processed. Also, it is possible that such transducers would not be effective in the same way when applied to some other texts or to some other corpus.

5.2. Transducers for extracting information related to the wind

The transducer for extracting the features *Pravac Vetra (Wind Direction)* and *Jacina Vetra (Wind Amount)* considers the context in which these information appear. Through the analysis of texts about the weather conditions, it was determined that these pieces of information are almost always located in one sentence, very close to each other. The most common form of statements that contained this information was:

Vetar slab, jugozapadni.
(*The wind is weak, southwest.*)
Vetar severni, slab.
(*North wind, weak.*)
Jak severozapadni vetar...
(*A strong northwest wind...*)
Severni, jak vetar ...
(*North, strong wind ...*).

The words of the statements were in different morphological forms. The processing of words such as severni (northern), jugozapadni (southwestern), and the like, was of particular interest. It was important to recognize them as a direction of the wind and not as a cardinal direction that could be related to the location of the observed phenomena (as in ... u istočnim krajevima kiša… – ... in the eastern parts the rain...). Therefore, this graph was created to require the existence of the word vetar (wind) close to the recognized segment of text. Figure 7 shows the transducer for extracting information about the strength and direction of the wind, in the case when the word vetar (wind) is at the beginning of the segment that carries information. Transducer for the case when the word vetar (wind) is placed at the end is very similar and will be omitted here.

This transducer recognizes, among others, the

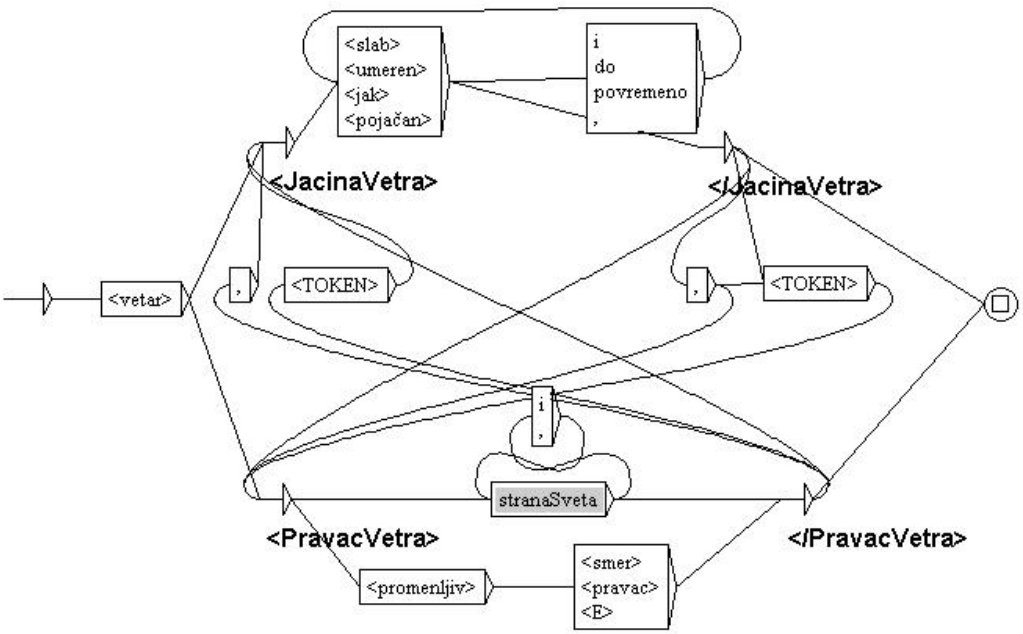

Figure 7. Transducer for extracting information related to the wind amount and direction.

following expressions:

Vetar jak zapadni.
(*The wind is weak, west*)
Vetar slab i umeren, jugoistočni.
(*Winds weak to moderate, southeasterly*.)
Vetar umeren i povremeno jak ...
(*Wind moderate and occasionally strong...*)
Vetar jak, uglavnom severni i severoistočni
(*Strong wind, mainly northern and northeastern*)
vetar jugozapadni, jak
(*southwesterly wind, strong*)

After application of the transducer shown in Figure 7, the above expressions have acquired the following forms, with extracted parts of the segments shown bolded:

*Vetar <JacinaVetra>**jak**</JacinaVetra>*
*<PravacVetra>**zapadni**</PravacVetra>*
*Vetar <JacinaVetra>**slab i umeren**</JacinaVetra>,*
*<PravacVetra>**jugoistočni**</PravacVetra>*
*Vetar <JacinaVetra>**umeren i povremeno jak***
*</JacinaVetra> Vetar <JacinaVetra>**jak**</JacinaVetra>,*
*uglavnom <PravacVetra>**severni i severoistočni***
*</PravacVetra> vetar <PravacVetra>**jugozapadni***
*</PravacVetra>, <JacinaVetra>**jak**</JacinaVetra>*

5.3 Transducers for extracting information related to the location

Information about the location where the weather phenomenon occurred or where it is expected to occur was represented in a number of different ways in the text. The location information is divided into three semantic classes (three features):

- *ImeTeritorije – TerritoryName* (*na Balkanskom poluostrvu – at Balkan peninsula*, *u Beogradu – in Belgrade*, *Srbija – Serbia* etc.)

- *DeoTeritorije – TerritoryPart* (*na istoku – in the east*, *u severozapadnim krajevima – in the northwest region* etc.)

- *Lokaltet – Locality* (*u kotlinama – in the valleys*, *na planinama – in the mountains* etc.)

Feature *ImeTeritorije* (*TerritoryName*) represents a toponym, and its application requires dictionaries that contain information about the

toponyms (Gucul-Milojevic, 2010). The main graph that is used to extract toponyms is shown in Figure 8.

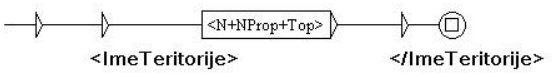

Figure 8. Transducer for extracting information related to the territory name.

Figure 9 shows the graph for the extraction of information related to the part of the territory. The following bolded parts were extracted:

u centralnim delovima *kontinenta*
(***in the central parts*** *of the continent*)
na severozapadu *kontinenta*
(***in the northwest*** *of the continent*)
u južnim i jugoistočnim delovima *Evrope*
(***in the southern and southeastern parts*** *of Europe*)
na severu *Evrope*
(***in northern*** *Europe*)
u istočnim krajevima *Srbije*
(***in the eastern parts*** *of Serbia*)
iznad ***većeg dela*** *poluostrva*
(***over most part*** of the peninsula)

Graph *deoTeritorije.grf* requires for the words representing cardinal directions to be followed by some of the words such as *deo* (*part of*), *kraj* (*region*), *kontinent* (*continent*), *država* (*state*), *zemlja* (*country*), *grad* (*city*) or a toponym, in order to be recognized as a part of the territory (*u južnim i jugoistočnim krajevima – in the south and southeastern regions*), because otherwise the recognized segment probably refers to the direction of the wind (*vetar slab, južni i jugoistočni – the wind is weak, south and southeast*). In order to achieve good effectiveness of the graph, it is essential that the graph is applied before the application of the graph for recognizing the *ImeTeritorije* (*TerritoryName*) feature. Otherwise, the annotations would be inserted in front of the toponyms. Of course, it is possible to change the order of application of the two transducers, but in that case the rules of extraction should be adjusted. This is one example where the order of the graphs application is important, i.e. when the order of graphs application affects the design of the graphs.

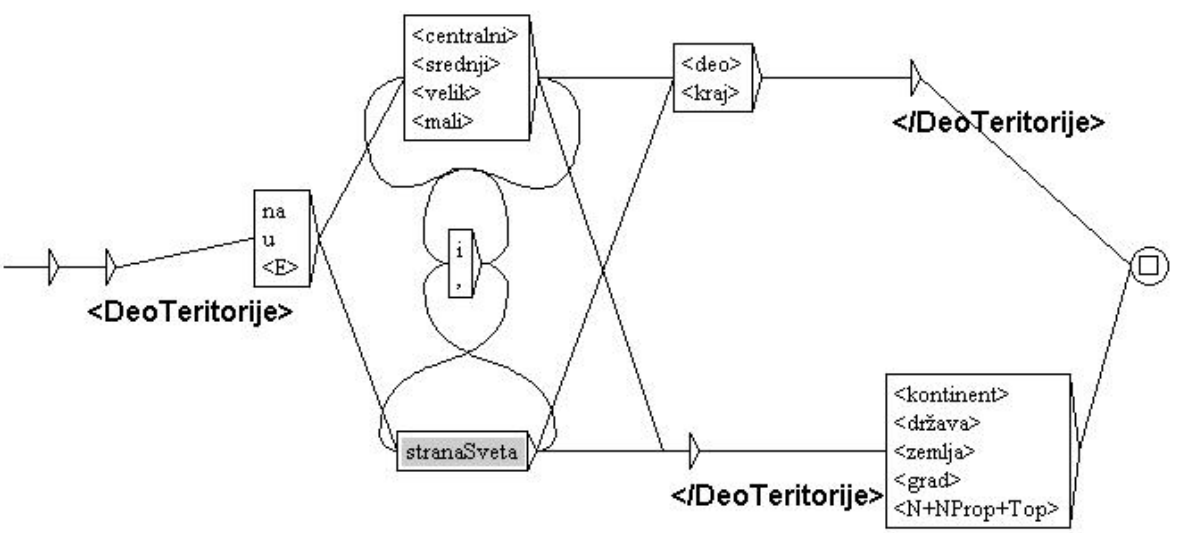

Figure 9. Transducer deoTeritorije.grf for extracting information related to the parts of the territory

6. Conclusion

Although finite state transducers and other abstract machines, such as recursive transition networks, have been used for several decades in natural language processing, they are still the most attractive for research because of their ability to achieve great precision. The transducers described in this paper are only part of a wider process of extracting information. However, they are extremely important, for several reasons.

Firstly, although there are major efforts of the Human Language Technology Group from the Faculty of Mathematics to improve researches related to Serbian, the results have remained relatively modest compared to other world languages (Vitas et al., 2012). Therefore, the creation of new electronic resources for Serbian is of great importance, which is recognized by the META – Multilingual Europe Technology Alliance group[7]. Transducers created in this research should be observed as a part of those resources, taking into account that they are suitable for use in a variety of natural language processes, and not just in one for which they were created for. For example, transducers designed in this study with minor modifications can be used for automatic translation into another language.

Secondly, the data extracted using transducers are more precise than the data acquired using other methods from the information extraction field. This means that there is very little incorrectly extracted information, so these data sets can be used for further processing with high reliability. Additionally, given that they are created by humans, and not machines, their precision can be adjusted by the analysis of the results and modifying of the transducers, which is shown in the example of the transducer for extracting information related to temperature (Section 5.1).

Our further research will enhance information extraction from the previously described meteorological corpora, as well as from other texts in Serbian. Collection of the transducers that will arise in these studies could be re-used for different purposes. Therefore, it is planned to create this collection as a special structure that can eventually one day be made public and available to other researchers in this area.

Acknowledgments

This research was conducted through the project 178006, «Serbian and its resources: theory, description and applications», financed by the Serbian Ministry of Science.

References

Brkic, M., and Matetic, M. 2007. Modeling Natural Language Dialogue for Croatian Weather Forecast System. In *Proceedings of the 18th International Conference on Information and Intelligent Systems*, *Varaždin, Croatia*, 391-396

Burns, G., Feng, D., and Hovy, E. 2008. Intelligent Approaches to Mining the Primary Research Literature: Techniques, Systems, and Examples, Computational Intelligence in Medical Informatics. In *Studies in Computational Intelligence*, 17-50, Berlin, Heidelberg: Springer.

Casacuberta, F., Vidal, E. and Picó, D. 2005. Inference of finite-state transducers from regular languages*, Pattern Recognition,* 38(9): 1431-1443.

Chrobot, A., Courtois, B., Hammani-Mc Carthy, M., Gross, M. and Zellagui. K. 1999. Dictionnaire electronique DELAC anglais: noms composés. In *Technical Report 59*, LADL, Université Paris 7.

Courtois, B. 1996. Formes ambiguës de la langue française. *Lingvisticæ Investigationes,* 20(1): 167-202.

Courtois, B. and Silberztein, M. 1990. *Les dictionnaires électroniques du français*, Larousse, Langue française, vol. 87.

Feng, D., Burns, G. and Hovy, E. 2007. Extracting Data Records from Unstructured Biomedical Full Text. In *Proceedings of the EMNLP conference*, Prague, Czech Republic.

Friburger, N. and Maurel, D. 2004. Finite-state transducer cascades to extract named entities in

texts, *Theoretical Computer Science* 313: 93-104.

Goh, C. S., Gianoulis, T. A. Liu, Y. Li, J. Paccanaro, A. Lussier, Y. A. and Gerstein, M. 2006. Integration of curated databases to identify genotype-phenotype associations. *BMC Genomics*, 7: 257-257.

Grishman, R. and Sundheim, B. 1996. Message Understanding Conference –6: A Brief History, In *Proceedings of COLING'96, Copenhagen, Denmark*, 466-471.

Grass, T., Maurel, D. and Piton, O. 2002. Description of a multilingual database of proper names. In *PorTAL, Volume 2389 of Lecture Notes in Computer Science*, eds. Elisabete Ranchod and Nuno J. Mamede, 137–140. Berlin: Springer.

Gross, M. and Perrin, D. 1987. Electronic Dictionaries and Automata in Computational Linguistics. In *Proceedings of LITP Spring School on Theoretical Computer Science, Saint-Pierre d'Oleron, France, May* 25-29.

Gucul-Milojević, S. 2010. Proper names in information extraction, *INFOtheka* 11(1): 47-58.

Hobbs, J. R., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M. and Tyson, M. 1997. FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In *Finite-State Language Processing*, eds. Roche E. and Y. Schabes, 383-406, Cambridge, MA: The MIT Press.

Jim, K. Parmar, K. Singh, M. and Tavazoie, S. 2004. A cross-genomic approach for systematic mapping of phenotypic traits to genes. *Genome Resources*, 14(1): 109-115.

Jurafsky, D. and Martin J. H. 2000. *Speech and language processing*, Prentice-Hall Inc.

Klarsfled, G. and Hammani-Mc Carthy, M. 1991. Dictionnaire électronique du ladl pour les mots simples de l'anglais (DELASa). *Technical report,* LADL, Université Paris 7.

Kononenko, I., Popov, I. and Zagorulko, Yu. 1999. Approach to Understanding Weather Forecast Telegrams with Agent-Based Technique. In *A. Ershov Third International Conference «Perspectives of System Informatics»*, 295-298.

Kononenko, I., Kononenko, S., Popov, I. and Zagorulko, Yu. 2000. Information extraction from non-segmented text (on the material of weather forecast telegrams). *RIAO* 2000:1069-1088.

Korbel, J. Doerks, T. Jensen, L. J. Perez-Iratxeta, C. Kaczanowski, S. Hooper, S. D. Andrade and M. A. Bork, P. 2005. Systematic association of genes to phenotypes by genome and literature mining. *PLoS Biol*. 3: 134-134.

Kornai, A. 1999. *Extended finite state models of language*, Cambridge University Press.

Krstev, C. 2008. *Processing of Serbian: Automata, texts and electronic dictionaries*. Belgrade: University of Belgrade, Faculty of Philology.

Krstev, C. and Vitas, D. 2005. Corpus and Lexicon - Mutual Incompletness. In *Proceedings of the Corpus Linguistics Conference*, *Birmingham*.

Krstev, C., Vitas, D., Obradović, I. and Utvić, M. 2011. E-Dictionaries and Finite-State Automata for the Recognition of Named Entities. In *Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing*, *Blois (France), July 12-15, 2011*, 48–56

Labelle, J. 1995. Le traitement automatique des variantes linguistiques en français: l'exemple des concrets, *Lingvisticæ Investigationes*, 19(1): 137-152

Labsky, M., Nekvasil, M., and Svatek, V. 2007. Towards web information extraction using extraction ontologies and (indirectly) domain ontologies. In *K-CAP '07 Proceedings of the 4th international conference on Knowledge capture*, *ACM New York, NY, USA*.

Laurikari, V. 2009. TRE library 0.7.6, http://lautikari.net/tre.

MacDonald, N. J. and Beiko, R. G. 2010. Efficient learning of microbial genotype-phenotype association rules. *Bioinformatics,* 26: 1834-1840.

Maynard, D., Bontcheva, K. and Cunningham, H. 2003. Towards a semantic extraction of Named Entities, In *Recent Advances in Natural language Processing*, *Bulgaria*.

Monceaux, A. 1995. Le dictionnaire des mots simples anglais: mots nouveaux et variantes orthographiques, *Technical Report 15*, IGM, Université de Marne-la-Vallée, France.

Olivier, B., Constant, M. and Laporte, E. 2006. Outilex, plate-forme logicielle de traitement de textes ecrits. In *Proceedings of TALN'06*. Leuven, Belgium: UCL Presses universitaires de Louvain.

Pajić, V. 2011. Putting Encyclopaedia Knowledge into Structural Form: Finite State Transducers Approach. *Journal of Integrative Bioinformatics, Informationsmanagement in der Biotechnologie e.V.*, Germany, 8(2): 164, ISSN 1613-4516.

Pajić, V., Pavlović-Lažetic, G. and Pajić, M. 2011a Information Extraction from Semi-structured Resources: A Two-Phase Finite State Transducers Approach. In *Implementation and Application of Automata: Proceedings of 16th International Conference CIAA*, *Lecture Notes in Computer Science*, 282-289, Berlin, Heidelberg: Springer, ISBN 978-3-64-222255-9.

Pajić, V., Pavlović-Lažetic, G., Beljanski, M., Brandt, B. and Pajić, M. 2011b Towards a Database for Genotype-Phenotype Association Research: Mining Data from Encyclopedia. *International Journal of Data Mining and Bioinformatics*, Inderscience publishers, **ISSN (Online): 1748-5681, ISSN (Print): 1748-5673,** http://www.inderscience.com/browse/index.php?journalID=189&action=coming.

Pajic, V. 2010. *Konačni transduktori u nadgledanju veba*, Magistarska teza. Faculty of Mathematics, University of Belgrade, Serbia.

Paumier, S. 2011. *Unitex 2.1 User Manual*. Université Paris-Est Marne-la-Vallée. http://www-igm.univ-mlv.fr/˜unitex/UnitexManual2.1.pdf.

Roche, E. 1999. Finite state transducers: parsing free and frozen sentences. In *Extended finite state models of language,* 108.-120, Cambridge University Press.

Roche, E. and Schabes, Y. 1997. *Finite-state language Processing*, The MIT Press.

Sastre, J. M. and Forcada, M. 2007. Efficient parsing using recursive transition networks with output, In *3rd Language & Technology Conference* (LTC'07). 5-7 October 2007, ed. Zygmunt Vetulani, 280-284.

Sastre, J. M. 2009. Efficient Parsing Using Filtered-Popping Recursive Transition Networks. *Lecture Notes in Computer Science* 5642: 241-244.

Savary, A. 2000. *Recensement et description des mots composés - méthodes et applications*. Thèse de doctorat. Université de Marne-la-Vallée, France.

Sekine, S. and Ranchhod, E. 2009. *Named entities: Recognition, classification and use*. Amsterdam: John Benjamins Publishing Company.

Silberztein, M. D. 1993. Dictionnaires ´electroniques et analyse automatique de textes. *Le systeme INTEX*. Paris: Masson.

Slocum J. 1985. A Survey of Machine Translation: its History, Current Status, and Future Prospects. *Computational Linguistics* 11(1): 1-17.

Tamura, M. and D'haeseleer, P. 2008. Microbial genotype-phenotype mapping by class association rule mining. *Bioinformatics* 24: 1523-1529.

Vitas, D. 2006. *Prevodioci i interpretori: Uvod u teoriju i metode kompilacije programskih jezika*. Belgrade: Faculty of Mathematics.

Vitas, D., Krstev, C., Obradović, I., Popović, Lj. and Pavlović-Lažetić, G. 2003. Processing Serbian Written Texts: An Overview of Resources and Basic Tools. In *Workshop on Balkan Language Resources and Tools*, *Thessaloniki, Greece,* 97-104.

Vitas, D., Popović, Lj., Krstev, C., Obradović, I., Pavlovic-Lažetic, G. and Stanojevic, M. 2012. *Српски језик у дигиталном добу - The Serbian Language in the Digital Age*. In META-NET White Paper Series, eds. Georg Rehm and Hans Uszkoreit, Berlin, Heidelberg: Springer, ISBN 978-3-642-30754-6.