Language Identification:
The Case of Serbian

Anđelka Zečević and Staša Vujičić Stanković

University of Belgrade, Faculty of Mathematics, Studentski trg 16, Belgrade, Serbia,
{andjelkaz,stasa}@matf.bg.ac.rs

Abstract. Serbian and other national standard languages that are used instead of common standard Serbo-Croatian have a phonologically based orthography. The characteristics of this orthography are that Serbian can be written in two alphabets (Latin and Cyrillic) and in two dialects (Ekavian and Ijekavian) which is directly reproduced in a written language. Consequently, Serbian is hard to identify because there are languages that are very similar (sharing alphabets and dialects). Therefore the problems typical for closely related languages are strongly presented in Serbian. The existing top-level tools do not give results comparable to the other classes of languages, so it is necessary to locate the problem and use the cumulative linguistic knowledge to overcome it.
This paper summarizes the first results towards that goal. We have chosen several top-level language identification tools and tested theirs sensibility for both the alphabets and both the dialects. For the testing purpose we have created corpora encompassing the newspaper articles, the literary works written by Serbian authors and the translations of many widely-circulated novels. The obtained results indicate that not all the tools support Latin and Cyrillic scripts and confirm that the language identification of documents written in Ijekavian variant is much more error prone in comparison to documents written in Ekavian variant.

Keywords: Language Identification, Serbian

1 Introduction

A language identification is a problem of identifying the language a document is written in. This task can be considered solved just to a certain extent as for different languages and different document types an accuracy of identification varies significantly. Having in mind an importance of a digital visibility of a language and extensive development of natural language processing applications, we investigated the case of Serbian. It might be challenging as it can be written in two alphabets. Apart from these, a historical background makes the language closely related to the languages that are used instead of common standard Serbo-Croatian. Therefore, all our intentions were addressed to the choice of top-level tools and thoughtful evaluation.

The paper is organized as follows. Section 2 covers the characteristics of the Serbian language and lists important properties that should be taken into account. In Section 3 an overview of the language identification approaches is given from the early days up to now. Our experiment regarding the successful identification of Serbian with a special remark to closely related languages is explained in Section 4. Section 5 summarizes the obtained results, while Section 6 concludes and presents some future ambitions.

2 Characteristics of the Serbian Language

The Serbian language as a representative of the South Slavic languages has rich morphology and as such represents a challenge for natural language processing tasks. Additionally, it is less-resourced language with a modest number of speakers, which is why it is important to be maintained through the development of various tools and language resources, as well as, to be properly identified deploying the existing ones.

Standard Serbian is the standard national language of Serbs and the official language in the Republic of Serbia. It was formed on the basis of Ekavian and Ijekavian Neo-Štokavian South Slavic dialects. As discussed in [15], Vuk Stefanović Karadžić (1787-1864), the major reformer of the Serbian language, has been reformed and modernized Serbian written language, and has been made the standardization of the Serbian Cyrillic. Vuk Stefanović Karadžić has been propagated the principle “Write as you speak and read as it is written”, and to the present time the Serbian language has phonologically based orthography (with a few exceptions).

In the 20^{th} century, in the common state of Yugoslavia, this language was officially encompassed by Serbo-Croatian, a name that implied a linguistic unity with Croats (and later with other nations whose languages were based on Neo-Štokavian dialects). In the last decade of the 20th century in Serbia the name Serbo-Croatian has been replaced in general usage by the name Serbian[1] [17].

There are three dialects that are used in the countries of the world in which the Serbian language is spoken – Ekavian (prevalent in Serbia), and Ijekavian and Ikavian (common in some parts of the northern Serbia, Bosnia and Herzegovina, Croatia, and Montenegro). The dialect difference origins from the ‘*jat*’, old Slavic vowel, which has been conflated into the vowel ‘*e*’ in the case of Ekavian, diphthongs ‘*ije*’ and ‘*je*’ in the case of Ijekavian dialects, and vowel ‘*i*’ in the case of Ikavian. In the Table 1 are presented Ekavijan, Ijekavian, and Ikavian forms for the word *cvet* (eng. ‘*flower*’) in nominative singular and nominative plural.

As discussed in [17], it could be made use of the facts that languages of Štokavian provenance share the Ijekavian dialect in their standard forms, and therefore have a lot in common. For instance, the tools and resources that are developed for one such language, could be used with minor modifications or without any modifications in the process of natural language processing of its

[1] In 2006, the Constitution of the Republic of Serbia prescribes: “Serbian language and Cyrillic script shall be in official use in the Republic of Serbia”.

Table 1: Ekavijan, Ijekavian and Ikavian forms for the word *cvet* (eng. '*flower*').

	Ekavian dialect	Ijekavian dialect	Ikavian dialect
nominative singular	cv**e**t (*long e*)	cv**ij**et	cv**i**t
nominative plural	cv**e**tovi (*short e*)	cv**j**etovi	cv**i**tovi

closely related language. However, from the language identification point of view, these facts represent the difficulty.

Although the Cyrillic alphabet is in the mandatory use in Serbia in communication with the state authorities, both Cyrillic and Latin alphabets are equally used in the Serbian written texts. The possibility to combine both alphabets in the same text is also legitimate. In Table 2 list of 30 Serbian Cyrillic and Latin graphemes are presented, with digraphs in the Latin alphabet marked in grey color. Nevertheless, the transliteration is not unified. For example, the toponym *New York* could be represented in the Latin alphabet as *Njujork* (or even *NJujork*), but in the Cyrillic alphabet it could be represented as *Њујорк* or on rare occasions *Нујорк*.

Table 2: Serbian Cyrillic and Latin graphemes. Digraphs in the Latin alphabet are marked in grey color.

	Serbian letters														
Cyrillic	А	Б	В	Г	Д	Ђ	Е	Ж	З	И	Ј	К	Л	Љ	М
	а	б	в	г	д	ђ	е	ж	з	и	ј	к	л	љ	м
Latin	A	B	V	G	D	Đ	E	Ž	Z	I	J	K	L	Lj	M
	a	b	v	g	d	đ	e	ž	z	i	j	k	l	lj	m
Cyrillic	Н	Њ	О	П	Р	С	Т	Ћ	У	Ф	Х	Ц	Ч	Џ	Ш
	н	њ	о	п	р	с	т	ћ	у	ф	х	ц	ч	џ	ш
Latin	N	Nj	O	P	R	S	T	Ć	U	F	H	C	Č	Dž	Š
	n	nj	o	p	r	s	t	ć	u	f	h	c	č	dž	š

Having in mind all above discussed Serbian language-specific aspects, and in particular alphabetic and dialectic variants, we could deduce that four different combinations for modelling must be taken into account in the process of identifying the language.

3 Language Identification

Many traditional as well as modern language applications we encounter in everyday lives rely on a successful language identification. Thus, for example, for the automatic text preprocessing such as tokenization, morphological analysis or parsing, a language of the given text should be known in advance. Search engines such as Google, Yahoo! or Bing are expected to recognize the language of a user's query and to generate the list of relevant pages in the same language. Tremendous efforts are put into an improvement of the automatic machine translation where the source language is necessary to identify in order to apply appropriate knowledge and translation techniques. Equally challenging task is set to web crawlers, as they also need to extract the fragments of crawled texts and incorporate them into specific web corpora or redirect them to a further analysis. All these examples indicate that tools for automatic language identification ought to work well with both short and long text forms as well as multilingual documents. They need to be able to adapt to a large number of written languages, styles and genres that cover all domains of an application.

Theoretical approaches to a language identification are numerous and quite diverse. However, the great majority of them uses statistical language models and results derived from a machine learning field as this task can be seen as a classification problem: assign a given text to the one of the pre-existing language classes. More about the historical overview of the language identification problem is presented below.

The first language identification attempts were based on unique character combinations extracted from languages' samples [4]. Thus, for example, "*eux*" combination was specific to French, "*ery*" to English, and "*lj*" to Serbo-Croatian. Obviously, not all the documents written in these languages should contain the listed combinations. Even more, these combinations might appear as a part of citations or loanwords, so they cannot be considered as reliable. Almost in parallel the lists of language common words were examined [8]. Since there is a tendency of common words grouping around a less common word, these approaches were applied successfully only when a document in question was long enough to contain many common words.

The usage of n-grams brought many novelties in the language identification field. Since n-grams of different types and different lengths are able to capture various languages' elements such as prefixes, suffixes, stop words, lemmas or stems, punctuation and many more, they were a reasonable choice. Apart from these, they do not require any text preprocessing and adapt quite easily to various error types that can be found in informal or OCRed texts.

Cavnar and Trenkle [3] described one of the first language identification algorithms based on character n-grams. The algorithm compares the profiles of the pre-existing language categories to the profile of a given document. The profiles consist of most frequent n-grams of size from 1 to 5 characters sorted in descending order by the number of appearances. For the comparison is used so called "out-of-place" measure that calculates the total difference of the n-grams' positions in the profiles. The assignment is done intuitively according to the lowest value of the “out-of-place” measure, that is, to the language category with the greatest number of n-grams overlapping. The algorithm was tested on the collection of 3,713 messages from the Usenet newsgroups in the eight languages (English, Dutch, French, German, Italian, Polish, Portuguese, and Spanish) and the precision of 99.8% was reported. The algorithm implementation by Gertjan van Noord [14] under the name TexCat was the first publicly available implementation. It was implemented in Perl and covered 74 language/encoding combinations. Nowadays, many implementations are available in a variety of programming languages and environments (for example, R [7]).

In the same year as Cavnar and Trankle, Dunning [6] proposed the algorithm based on language models and probabilistic reasoning. Markov models of the same order were created for each language category and in further steps these models were used for the likelihood estimation according to the Bayesian decision rule. The algorithm was tested on the collection of English and Spanish text segments and it was reported that when a text size was greater than 100 bytes and the size of each collection was greater than 50K bytes, there was 90% confidence that the precision would be greater than 99%. For the reported collection the best results were obtained for the bigrams and trigrams models.

Nowadays, the task of language identification faces a large group of languages as the Web has become a host for both social and professional activities. It is not unusual to have a single Web page comprising many of these languages. One of the papers addressing the complex task of language identification of multilingual documents is presented in the paper [18]. It refreshes a kind of dictionary model looking for the words that appear in one of the nine European languages (French, English, Italian, Spanish, Slovakian, Czech, Slovenian, and Polish). Here, dictionaries are defined as a set of (word, relevance) mappings learned from Wikipedia dumps. The algorithm is tested on a collection of 1.000 multilingual documents. The reported accuracy of 98,34% is at the level of powerful monolingual identification tools.

Many short text formats such as search engine queries, e-Bay messages or Tweets need to be identified. In the paper [2] the impacts of different conditions such as the length of the document and the amount of training data are tested on three popular corpora (EUROGOV, TCL and Wikipedia) and what is reported confirms the language identification task is much harder for shorter documents and smaller training sets.

For the number of languages in the phase of identification the choice has to be made among closely related languages (for instance, Danish, Swedish and Norwegian or Spanish and Catalan), language varieties (for instance, European and Brazilian Portuguese) or language dialects. The paper [10] presents the three-phase model for identification of closely related languages. The algorithm addresses differentiation of Croatian from Slovenian and Serbian. In the first phase the documents written in any of these three languages are singled out by the rule of 100 most frequent words and the rule of special character elimination[2]. In the next phase character based second-order Markov model is developed aiming to distinguish languages among themselves. In order to improve the distinction between Croatian and Serbian, which is found very difficult, in the final phase the lists of forbidden words are introduced. Those are the lists of words appearing in one language but not in the other. The model is trained and tested on the news collection and the achieved accuracy of 0.9918 is the best ever reported for this group of languages.

The identification of language varieties is discussed in [19] for the case of European and Brazilian Portuguese. As the differences between these two varieties can be described at orthographic, lexical and syntactic level, the algorithm analyses three groups of features: character n-grams (n varying from 2 to 6), word unigrams and word bigrams. The language models are calculated by using the Laplace probability distribution and evaluated on the journalistic corpora containing texts from the both varieties classified according to their length in tokens. The achieved accuracies differ slightly for used features but are still at the level of expected: 99.8% for 4-grams, 99.6% for word unigams and 91.2% for word bigrams.

As can be observed, all listed approaches either general or specific, require the advanced linguistic knowledge and complex statistics. In some cases, the task of identification is accomplished with high accuracy, but there are still some cases waiting for improvements and further adaptations.

4 The Experiment

Considering the modest number of native speakers of the Serbian language, its characteristics and an importance of digital visibility of a language, we wanted to test if modern language identification tools are able to cope with Serbian and recognize the texts written in it. We have chosen `langid.py`, Google's `CLD` module as well as the tool described in [16] (in further text denoted as `Tiedemann&Ljubešić`). The motives are numerous: the `langid.py` is a very popular, pre-trained and easy to use language identification tool covering a wide range of languages; the `CLD` module is integrated in Google Chrome web browser and available through several Google's services; the `Tiedemann&Ljubešić` tool analyses the case of closely related languages with the focus on languages that are used instead of common standard Serbo-Croatian. As reported, all these tools recognize Serbian successfully.

4.1 Used Language Identification Tools

In the next paragraphs is given a brief description of the used language identification tools.

[2] On the Wikipedia page available at `http://en.wikipedia.org/wiki/Wikipedia:Language_recognition_chart` there are some elementary, a bit rough, language recognition charts)

The `langid.py` is a top-level language identification tool developed by Lui and Baldwin [12]. It is based on the multinomial Naive Bayes classifier which operates on the set of features (byte level unigrams, bigrams and trigrams) selected so that their information gain represents the characteristics of the language rather than the characteristics of the training domain [11]. The training corpus encompasses governmental documents, news-wire, online encyclopedia, software documentation and the Internet crawl extracted from the following sources: JRC-Acquis, ClueWeb 09, Wikipedia, Reuters RCV2, and Debian i18n. In the case of Serbian, the training collection includes XML wiki dumps for the period July-August 2010 as well as the set of manually translated content strings for a number of Debian software packages. The number of so far covered languages is 97. This tool expects a string or a redirected file as an input and generates an ISO 639-1 language code followed by a probability estimation. It also offers an useful option for narrowing the set of possible languages for a given document.

The `CLD` (`Compact Language Detection`) [5] is a library embedded in a Google's Chromium browser able to detect one of 83 different languages of Unicode UTF-8 texts, either plain text or HTML/XML. The algorithm is probabilistic in nature and uses Naive Bayes classifier. For the languages apart from those having unique scripts and those counting over 80.000 characters, sequences of 4 letters (quadgrams) are used. The training corpus is manually constructed from chosen web pages for each language, and then augmented by careful automated scraping of over 100M additional web pages. The obtained quadgrams are stored into tables which can be accessed in a very short time. The main quadgrams lookup table consists of 256K entries covering about 50 languages. In our testing we have used a Python implementation of the algorithm available by courtesy of Michael McCandless [9]. This tool outputs a language name, a reliability parameter with boolean value calculated by comparing the first and the second best language, number of analyzed bytes per document, and optionally, a list of three top languages among which the result is calculated.

The `Tiedemann&Ljubešić` classifier addresses the problem of closely related languages. It is in the main multinomial Naive Bayes classifier trained over a parallel collection. The usage of the parallel training set resulted in outperforming the state-of-the-art tools significantly since the data parallelism provided focus on subtle differences among languages. The results published in the introductory paper concern very closely related languages Croatian, Bosnian and Serbian. The classifier is trained over a parallel news collection from Southeast Europe titled SETimes ([13]) which is available in the eight languages (Albanian, Bosnian, Croatian, English, Macedonian, Serbian, Russian, and Ukrainian). Approximately 2.7 million words per language are found in training sets. The evaluation data contains of 200 documents per language varying from 70 thousand words for Croatian to 113 thousand words for Serbian. The reported overall accuracy is 95.7% which is significantly higher in comparison to other reported results. For the same learning and evaluation sets, one additional algorithm is tested ([1]). That is the algorithm based on the lists of words that appear quite frequently in one language but never in the other languages. The words from these lists are usually called "blacklisted words" and produced by comparing the words' frequencies among languages. The lists for Croatian, Bosnian and Serbian are applied in cascade fashion for each pair of languages resulting in the overall accuracy of 97%. We have used the first classifier that for the given input generates two language codes and probability estimation.

4.2 Test Corpus

For the testing purpose, we have created a corpus which consists of documents in both Ekavian and Ijekavian variant (Table 3). Since Serbian can be written in Cyrillic or Latin script, all the documents are transliterated into Latin script.

Table 3: The structure of the corpus.

	Size (in number of words)	Size (in MB)
Ekavian part	2.078,172	13.2
Ijekavian part	528,749	3.2

The Ekavian part of the corpus includes the articles from the daily newspaper *Politika*[3] for the years 2007 and 2010, the literary works written by the Serbian authors and the translations of many widely-circulated novels. The list of all used materials is reported in Appendix 1.

The Ijekavian part of the corpus includes the articles from the daily newspaper *Glas Srpske*[4] for the period January-July 2013, some columns taken from *Deutsche Welle* website[5] and representative works written in the Ijekavian dialect. Appendix 2 depicts all the details.

4.3 Processing Part

Due to the nature of the used tools and comparability with other reported results, we have split both the Ekavian and Ijekavian parts of the corpora into lines on average 400 words long and randomly singled out 100 lines from each. For the testing purpose of the `langid.py` tool each line is saved as a separate file because a redirection mode is used. The input for `CLD` and `Tiedemann&Ljubešić` tools is a single document containing all concatenated lines: the first 100 lines are from the Ekavina part and the rest from the Ijekavian part.

[3] *Politika* newspaper: `http://www.politika.rs`.

[4] *Glas Srpske* newspaper: `http://www.glassrpske.rs`.

[5] *Deutsche Welle*: `http://www.dw.de`.

5 Results

The obtained results are summarized in Table 4.

Table 4: The results.

Tool		Serbian	Bosnian	Croatian	Other
`langid.py`	Ekavian	100	0	0	0
	Ijekavian	0	0	100	0
`CLD`	Ekavian	25	0	75	0
	Ijekavian	0	0	100	0
`Tiedemann&Ljubešić`	Ekavian	98	2	0	0
	Ijekavian	1	98	1	0

All the tested lines are classified as written is Serbian, Croatian or Bosnian, which proves that there is a good external distinction among this group of languages and other languages groups. However, on the intrinsic level the results are quite discouraging. The lines written in Serbian Ekavian variant are recognized to some extent with overall accuracy of 74.3%. That is far beyond all reported results for all the tools. The identification of Serbian Ijekavian variant is much more difficult task even for the tool developed with an idea of closely related languages in mind resulting in the accuracy close to 0%.

The `langid.py` tool recognizes Serbian Ekavian variant with 100% accuracy and Serbian Ijekavian variant with 0% accuracy. We notice that this tool recognizes only the texts written in official Cyrillic alphabet which might cause the misclassification of all tested Ijekavian lines, usually written in Latin script, as Croatian.

Google's `CLD` obviously favours Croatian in both Ekavian and Ijekavian cases. In all the iterations the algorithm's confidence parameter is set on the true value which means it is quite sure about the final outcome. The average number of analysed bytes ranges from 80 to 100. After the inspection of the wrong results referring to Ekavian tests, we found that in 25 iterations the second proposed language was Slovenian, in 8 iterations it was Serbian, and in 5 iterations it was Slovak. In all the reminder iterations the algorithm was completely sure about Croatian. In the case of Ijekavian tests, in 16 iterations the second proposed language was Slovenian, in 3 iterations it was Slovak and in just 14 iterations it was Serbian. There was one iteration for each of the languages: Spanish, Italian, and Indonesian.

The `Tiedemann&Ljubešić` tool is very accurate in classifying Ekavian tests, but almost all (98 out of 100) Ijekavian tests are recognized as written in Bosnian. In 83 of these 98 iterations, the second proposed language was Croatian, and only in 15 of them it was Serbian. Our explanation for this phenomenon is the absence of Ijekavian documents from the training collections. If we expect it to be representative, it should contain both Ekavian and Ijekavian translations. We address the same comment to the epilogue of the introductory paper ([10]) with the list of the strongest discriminators among the observed languages which in the case of Bosnian contains many regular Serbian words in Ijekavian pronunciation (for instance, *izvještajima*, *posjetioci*, *djelimično*).

6 Conclusions and Further Work

The obtained results indicate that Serbian as a language cannot be identified with a high accuracy in the context of the national standard languages that are now used instead of common standard Serbo-Croatian. Moreover, it cannot be identified close to the level expected from some modern language applications. The accent on both the alphabets and both the variants need to be stressed more sharply to the groups of interest. In the time of writing this paper, the Internet Assigned Number Authority accepted and published in its language sub-tag registry codes for all four variant-alphabet combinations[6]. We hope that this will influence the awareness of language communities and the adaptability of future identification tools. Our next research steps will focus on discrimination lists among languages from the region in more broad and domain independent way as well as the named entity tracking as they are the witnesses of spelling differences among languages.

Acknowledgments. This research was conducted through the project 178006, financed by the Serbian Ministry of Science.

References

1. A Blacklist Classifier for Language Discrimination. Available at https://bitbucket.org/tiedemann/blacklist-classifier/wiki/Home.
2. Timothy Baldwin and Marco Lui. Language identification: The long and the short of the matter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 229–237. Association for Computational Linguistics, 2010.
3. William B Cavnar, John M Trenkle, et al. N-gram-based Text Categorization. *Ann Arbor MI*, 48113(2):161–175, 1994.
4. Gavin Churcher. Distinctive character sequences. *Personal communication*, 1994.
5. CLD (Compact Language Detection). Available at https://code.google.com/p/chromium-compact-language-detector/.
6. Ted Dunning. *Statistical identification of language*. Computing Research Laboratory, New Mexico State University, 1994.

[6] Special thanks to Goran Rakić for initiating this procedure http://blog.goranrakic.com/.

7. Ingo Feinerer, Christian Buchta, Wilhelm Geiger, Johannes Rauch, Patrick Mair, and Kurt Hornik. The textcat package for n-gram based text categorization in R. *Journal of Statistical Software*, 52(6):1–17, 2013.

8. Stephen Johnson. *Solving the problem of language recognition.* Technical report, School of Computer Studies, University of Leeds, 1993.

9. Language detection with Google's Compact Language Detector. Available at `http://blog.mikemccandless.com/2011/10/language-detection-with-googles-compact.html`.

10. Nikola Ljubesic, Nives Mikelic, and Damir Boras. Language identification: How to distinguish similar languages? In *Information Technology Interfaces, 2007. ITI 2007. 29th International Conference on*, pages 541–546. IEEE, 2007.

11. Marco Lui and Timothy Baldwin. Cross-domain feature selection for language identification. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 553–561, 2011.

12. Marco Lui and Timothy Baldwin. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30. Association for Computational Linguistics, 2012.

13. SETimes. Available at `http://www.setimes.com`.

14. Software by Gertjan van Noord. Available at `http://odur.let.rug.nl/vannoord/software.html`.

15. Živojin Stanojčić and Ljubomir Popović. *Grammar of the Serbian Language.* Institute for textbook publishing and teaching aids, Belgrade, 2011.

16. Jörg Tiedemann and Nikola Ljubesic. Efficient Discrimination Between Closely Related Languages. In *COLING*, pages 2619–2634, 2012.

17. Duško Vitas, Ljubomir Popović, Cvetana Krstev, Mladen Stanojević, and Ivan Obradović. *Languages in the European Information Society – Serbian.* META-NET White Paper Series. Georg Rehm and Hans Uszkoreit (Series Editors). Springer, 2011.

18. Radim Řehůřek and Milan Kolkus. Language identification on the web: Extending the dictionary method. In *Computational Linguistics and Intelligent Text Processing*, pages 357–368. Springer, 2009.

19. Marcos Zampieri, Binyam Gebrekidan Gebre, and Holland Nijmegen. Automatic identification of language varieties: The case of Portuguese. In *Proceedings of KONVENS*, pages 233–237, 2012.

Appendix 1 – The Structure of the Ekavian Part of the Corpus

Da Vinci Code by Dan Brown
1984 by George Orwell
Around the World in Eighty Days by Jules Verne
The Little Prince by Antoine de Saint Exupéry
The Diary of Anne Frank
The Hobbit by J. R. R. Tolkien
The Lord of the Rings by J. R. R. Tolkien
Solaris by Stanisław Lem
Winnie-the-Pooh by A. A. Milne
Bridget Jones's diary by Helen Fielding
For and Against Vuk by Meša Selimović
articles from the *Politika* newspaper

Appendix 2 – The Structure of the Ijekavian Part of the Corpus

Springs of Ivan Galeb by Vladan Desnica
Selected works of Petar Kočić
two novels by Branko Ćopić[a]
Dove Hole by Jovan Radulović
Rebel and Rebel Janko by Simo Matavulj
Spiders and Searching the bread by Ivo Ćipiko
The Dervish and Death by Meša Selimović
articles from Glas Srpske newspaper
column written by Nenad Veličković[b]

[a] Titles in Serbian are: *Magareće godine* and *Glava u klancu, noge na vrancu.*

[b] Column written by Nenad Veličković – `http://www.dw.de/skljocam-i-zvocam/a-4461937`.