Improved Methods of Word Acquisition in developing *Hascheck* Spell Checker Web Service System

Jakov Pavlek, University of Zagreb
(prof. Šandor Dembitz, University of Zagreb)
Šandor Dembitz, University of Zagreb
Marko Matasić, University of Zagreb

Abstract

Public service *Hascheck* (*Croatian Academic Spell CHECKer*) is a free Web service on the global level with continually growing base of its users and with rapidly increasing service volume.

In this paper we discuss methods used for processing and learning new, previously unknown words to the *Hascheck* system.

Interface for manual word acquisition has been developed using *Google Web Search* engine from appropriate given domains as a part of the improvement of the *Hascheck* service. In this matter already existing systematized knowledge resources, specifically *Wikipedia* and *Croatian Spell Checker for MS Word*, have been intensively used.

Program modules for automatic retrieval and classification of word types based on information about domain, language, and way of spelling have been developed.

As a result, some 135000 of new word types have been processed and classified into adequate classes using the developed software.

We also evaluate earlier methods used in the same process and compare them to the new ones regarding their accuracy, efficiency and the time they take to process words.

Combining new methods the processing of word types, that is, supervised learning in the *Hascheck* system, has been accelerated and the time of decision-making process has been significantly reduced.

Key words: *spell checker*, *word acquisition*, *web service*, *Google Search*, *Wikipedia*.

1. Introduction

The public service *Hascheck* (*Croatian Academic Spell CHECKer*) has confirmed itself as a very useful service on the global level with users in 47 countries. During its lifetime *Hascheck* has processed over 76 million tokens and this number is constantly increasing. Daily it provides 500 services on average, where the corpus of 130000 tokens is processed. The total number of users exceeds 20000 people, and more than 100 individuals use the service daily.

There are several main advantages of the *Hascheck* spell checker over most of the other spell checking systems: its average text coverage has passed 95 % long time ago (as could be seen from [2]), and it is learning new words on daily basis; it is available 24 hours a day, 365 days a year; it is accessible all over the world wherever there is internet access; finally, it is free.

The main disadvantage of *Hascheck* is that it's only a word based spell checker, thus it still doesn't provide context sensitive spell checking nor grammar checking. However, that is planned for further development as well as its integration with other applications. (First of them is *Hascheck Gadget*, a program for using *Hascheck* on desktop in *MS Vista* sidebar made recently by Domagoj Pavlešić [3].)

More details on *Hascheck* spellchecking service can be found in [2] and [4].

As *Hascheck*'s user base is growing, so is the amount of text for processing. In order to improve *Hascheck*'s service and to support it's learning process, we have reexamined the methods used during the word type processing.

2. Methods and methodology

As the main part of the project we had set ourselves a task to process 135000 word types previously unknown to *Hascheck* in order to increase its word coverage by another inch. The word types were divided into 7 lists (*L1* to *L7*), each of them consisting of approximately 20000 words (6 times 20000 words and single 15000 word list).

As *Hascheck* has gone to saturation concerning the learning of common word types, most of the new word types entering *Hascheck*'s dictionary belong to the class of proper names, especially the foreign proper names. They enter to Croatian language in their original form if coming from Latin alphabetical origin (phonetically transcribed otherwise). When coming in inflected form, Croatian suffixes are added. Foreign common words, which enter the dictionary as a class of special language entities, are very frequent and they usually appear in text in italics standing alone or with other foreign words in context. The word types, which are a combination of those two, are also frequent. This means that foreign common word types appear with Croatian inflectional suffixes. These should be avoided in Croatian and such words should be replaced with equivalent Croatian common word, unless these

words are commonly accepted over time and most usually phonetically adapted to Croatian. Little less frequent are common words in inflected form which haven't been recorded earlier by the *Hascheck* system. Then come the compounds, abbreviations, dialectal words, slang words, archaisms and neologisms, which are classified either as words or non-words (errors), depending from case to case. The rest of word types are typos and they are classified as errors.

There is another point, which needs to be mentioned. *Hascheck* tends to be rather discriminative to those words from languages similar to Croatian, which are very similar to some Croatian words. The same is valid for some words coming from rather distant languages strayed to the close vicinity of particular Croatian words. This problem could be possibly resolved if *Hascheck* were a context sensitive spell checker.

The time frame to complete the task mentioned earlier was fixed to 60 days and the choice of methods was constrained only to not using the *Hascheck* system alone as these word types were unknown to it.

There were several approaches to the task, but eventually 3 of them have emerged:

- using the MS Word's spell checker for Croatian;
- creating the interface for faster manual word acquisition using *Google search*;
- creating automatized system for automatic retrieval and word type classification using *Google search* and *Wikipedia*.

2.1. MS Word Spell Checker

Word's spell checker for Croatian has been developed by Silić et al. [1] and it was later bought by *Microsoft* and incorporated to the *Word*. It is based on morphological modeling so it is rather robust. However, it is somewhat closed, meaning that its dictionary database is relatively small, so its word coverage is not as high as *Hascheck*'s.

2.2. Interface for faster manual word acquisition

A quite significant improvement has been done in reduction of unnecessary copy pasting and writing and rewriting words and their tags. The main result of this approach has been in creating the interface for fast manual word acquisition over *Google search* on any chosen web domain.

2.3. *Google Search* and *Wikipedia*

Google search, the most popular web-searching engine, ensures the coverage of the web space. Its spelling suggestion (*Did you mean*) is a very powerful spell checking help. However, focused on the global scale and statistically oriented, it still misses language specific features for a highly inflected language such as Croatian. On the other hand, *Wikipedia*, the most popular and most probably the largest, free, multilingual and open content encyclopedia project, ensures the coverage of wide area of topics in different languages which are very likely (frequently) checked and edited from multiple users. Recent *Wikipedia* reports from 2007 [6] state that there was 7735252 of total articles in all *Wikipedias*, and 1858154 of that number in English, 518945 in French, 604538 in German, 315927 in Italian, 248592 in Spanish, 268876 in Portuguese, 396936 in Polish, 185174 in Russian, 70876 in Czech, 63586 in Hungarian. When looking at the languages of countries with comparable size of population to Croatia, the languages with the most articles were Finnish (120963 articles), Norwegian (117132 in Bokmål, 23831 in Nynorsk), Slovak (71804) and Danish (64988). There were (only) 32219 articles in Croatian, so there is still a lot of room for improvement.

Since then these numbers have become significantly higher. Just for instance, there has been 2,554,497 articles on English *Wikipedia* and 47108 articles in Croatian (compared to 94114 articles in Danish) *Wikipedia* in the moment of writing of this article (middle of September 2008) [7]. In that sense, *Wikipedia* and *Hascheck* share some qualitative aspects: they are both public, free and always and everywhere available services, their popularity is increasing as well as their knowledge and quality of service. In time *Hascheck* as well would become even more open to collaborative contents creation in the sense of word acquisition and information editing and correction, just like *Wikipedia* is.

Some argue that *Wikipedia* is not very reliable. We have accepted that and are ready to act with caution. However, we argue that *Wikipedia* is still more reliable than average web, at least concerning the spelling. And as Wilkinson and Huberman have shown in [8], "*Wikipedia article quality continues to increase, on average, as the number of collaborators and the number of edits increases.*"

2.3.1 Wictionary

Wictionary is another interesting project associated to *Wikipedia*. It is designed as the lexical companion to *Wikipedia*. It is in development and growing. However, we haven't been testing it since its size is (still) much smaller than the size of *Wikipedia* and it's context is not so rich, although its potential provides perfect surrounding and a base for expansion from spell checker dictionary to the fully operable dictionary created by internet community enthusiasts. The most recent data from *Wictionary* was the amount of 897570 entries with English definitions from 272 languages and total of 7,991 entries from Croatian in it. Croatian version of *Wictionary* is called *Wječnik*, is active since the end of 2004 and currently has 2859 entries.

3. Word type processing

Although each of the three approaches described either has limited coverage, or speed or accuracy, as a result of combining them together, an improved methodology has been developed.

3.1. Processing by Word

Word types are filtered through a spell checker in *MS Word*. The problem here is in capacity of the word spell checking processor confronted to a list of word types over a large number of pages. If formatted differently to fewer pages, it could probably be faster. However, the surprising problem was merely deleting all the word types, which the spell checker hadn't recognized. And doing that, it could take from 3-5 hours to process a single 15000-20000 word token list with approximately 2500-3500 recognized words. Thus, we have dismissed the idea to use *Word*'s spell checkers for other languages, which are less represented in the list of given words.

For instance, to process 500 words of German origin in the list of 15000 word types takes 3 hours, which gives the speed of 166 words per hour and that is pretty close to manual spell checking speed. If words could (only) be grouped by their language origin, then it could be worth the trouble. This emphasizes the importance of the language origin information. We'll come to that later when discussing more on this approach. After we've processed and tagged all the words using *Word*'s spellchecker and reduced the list, we continue to the next two approaches.

3.2. Automatic Google Search

The idea is to preprocess the list by retrieving all the necessary information on each word type from the Web. More precisely, using the *Google search* engine to collect and save the information for each word type on the suggested word alternative (if any), the category of the word according to *Hascheck*'s automatic word tagging based on degree of the word peculiarity, its total web frequency and internal *Hascheck*'s frequency, the data on language distribution and the most common ways of spelling over the most significant *Google search* page results for a given word.

ALT: -	Word: julkinog	class: -ll-	fr: 2	com: P	Freq: 25	Jezici: 4	[3: hr 1: au	Title: 0 []	Contxt: 1 [Julkinog: 3]
ALT: -	Word: jušadi	class: -ll-	fr: 2	com: => junadi?	Freq: 25	Jezici: 1	[3: hr]	Title: 0 []	Contxt: 1 [jušadi: 1]
ALT: kadhem	Word: kadhema	class: -ll-	fr: 2	com: => badema? k	Freq: 50	Jezici: 7	[2: hr 2: kv	Title: 0 []	Contxt: 1 [Kadhema: 7]
ALT: -	Word: kahofer	class: -ll-	fr: 2	com: => Kalofer?	Freq: 32	Jezici: 3	[7: at 1: fr	Title: 1 [Ka	Contxt: 1 [Kahofer: 1]
ALT: katara	Word: kajtara	class: -ll-	fr: 2	com: => kantara? ka	Freq: 48	Jezici: 6	[4: cz 2: hu	Title: 2 [Ka	Contxt: 2 [Kajtara: 4 kajtara: 1]
ALT: -	Word: kakidudis	class: -ll-	fr: 2	com: => kakiouzis?	Freq: 35	Jezici: 1	[4: hr]	Title: 0 []	Contxt: 2 [kakidudis: 1 Kakidudis: 1]
ALT: kaskida	Word: kaksida	class: -ll-	fr: 1	com: => kasida?	Freq: 30	Jezici: 4	[4: hr 2: cc	Title: 0 []	Contxt: 2 [kaksida: 3 KAKSIDA: 1]
ALT: -	Word: kalikeamicin	class: -ll-	fr: 1	com: -	Freq: 20	Jezici: 1	[4: hr]	Title: 0 []	Contxt: 1 [kalikeamicin: 2]
ALT: kamauf	Word: kamaufa	class: -ll-	fr: 2	com: => kamala? ka	Freq: 50	Jezici: 2	[7: hr 1: sk	Title: 0 []	Contxt: 2 [Kamaufa: 4 kamaufa: 1]
ALT: -	Word: kaneharu	class: -ll-	fr: 1	com: => kamenaru?	Freq: 45	Jezici: 2	[7: jp 1: au	Title: 1 [Ka	Contxt: 2 [Kaneharu: 4 kaneharu: 2]
ALT: -	Word: kangrgino	class: -ll-	fr: 2	com: -	Freq: 19	Jezici: 2	[6: hr 4: yu	Title: 0 []	Contxt: 1 [Kangrgino: 5]
ALT: -	Word: kanna	class: -ll-	fr: 1	com: => kanan? kar	Freq: 35	Jezici: 5	[3: jp 2: de	Title: 1 [Ka	Contxt: 3 [Kanna: 5 KANNA: 1 kanna: 1]
ALT: -	Word: kanzianu	class: -ll-	fr: 2	com: => kandiranu?	Freq: 34	Jezici: 5	[4: si 2: hr	Title: 0 []	Contxt: 2 [Kanzianu: 8 KANZIANU: 1]
ALT: kardumovic	Word: kardumovu	class: -ll-	fr: 2	com: P	Freq: 25	Jezici: 1	[6: hr]	Title: 0 []	Contxt: 2 [Kardumovu: 3 kardumovu: 2]
ALT: -	Word: kauflandove	class: -ll-	fr: 1	com: P	Freq: 30	Jezici: 3	[7: hr 2: cz	Title: 0 []	Contxt: 1 [Kauflandove: 8]
ALT: kaukenas	Word: kaukenasa	class: -ll-	fr: 2	com: P	Freq: 35	Jezici: 4	[4: hr 3: pl	Title: 0 []	Contxt: 2 [Kaukenasa: 7 kaukenasa: 1]
ALT: kavkaski	Word: kavkavskih	class: -ll-	fr: 1	com: => kavkaskih?	Freq: 30	Jezici: 2	[6: hr 1: ru	Title: 0 []	Contxt: 2 [kavkavskih: 4 Kavkavskih: 2]
ALT: kavkaski	Word: kavkavskim	class: -ll-	fr: 1	com: => kavkaskim?	Freq: 21	Jezici: 2	[4: hr 2: yu	Title: 0 []	Contxt: 1 [kavkavskim: 1]
ALT: -	Word: kavrevišvili	class: -ll-	fr: 2	com: -	Freq: 20	Jezici: 1	[3: hr]	Title: 0 []	Contxt: 1 [Kavrevišvili: 1]

Fig. 1. Information retrieved using *Google Search* (sample).

Collecting this information can be rather slow since *Google* allows mostly manual search. It is possible to distribute the web queries to multiple computers in order to achieve greater throughput. The alternative is to use *Google API*, which is restricted to 1000 queries per day with separate queries for word frequency and for word suggestion. Except that, there are no more *Google API* keys for new users. Although, the keys, which have been already assigned, are free for individual use. Besides, *Google* might be open toward academic projects with such usage of *Google search* and could possibly allow them less restrictive use.

At this stage, the main domain which is chosen is the *wikipedia.org* domain while it contains the language information and we presume that spelling could be more reliable then on the Web in general, since it is a public project and presumably more people are active on the same article and they probably correct each other's materials when/if making mistakes. The final result of such queries could be seen in Fig. 1.

3.3. Automatic Classification

With all the necessary information collected and saved locally, it is relatively easy to use it in order to make a decision tree system for automatic classification of word types. Rules used in this process somewhat depend on the general orthography rules for the specified language. Beside the information on the possible language origin of the word type, the decision tree based automatic classifier is using the information on the most common ways of spelling, on frequency and the

spelling suggestion information.

During this process, words are grouped to the different files according to the information on language origin of the words.

Since the classifier is far from being error free, these files should be later processed and rechecked manually using the interface for faster manual word acquisition and classification.

3.4. Manual Classification interface

The main advantage of such an interface is that it saves time and work since there is no more copy pasting or typing of the words. Mouse clicking is almost all what is necessary to classify the word correctly. There is a click to check the context, 2 clicks to select the class of the word, possibly one more to capitalize the first letter and one click to confirm the decision. So, no more than five mouse clicks is needed to classify a word type.

After classifying the word types found by *Word*'s spell checker and those found on *wikipedia.org* domain, it is needed to cover the rest of the word types.

This demands minor changes concerning the domain search while retrieving information from the Web using the *Google Search* engine. In order to reach the words originating from smaller languages such as Croatian, it was found useful to exclude most of the *general Top Level Domai*ns (*gTLD*), in order to extract country domains and through that, the relevant language information. The rest of the process is same as it was with *Wikipedia* domain.

Method used before the improvement consisted of opening the file with a wordlist, then opening the internet browser on *Google* webpage and copying words one by one to the *Google search* engine and checking the context and consulting the dictionary, then tagging the word and moving to the next one. That method was simpler and more direct. However, it was slower. A well-trained person could process between 100 and 150 word types per hour, depending on the sample and the category of the word types. As the service has become more popular, the need to speed up the word classification process or to distribute it to more individuals becomes stronger.

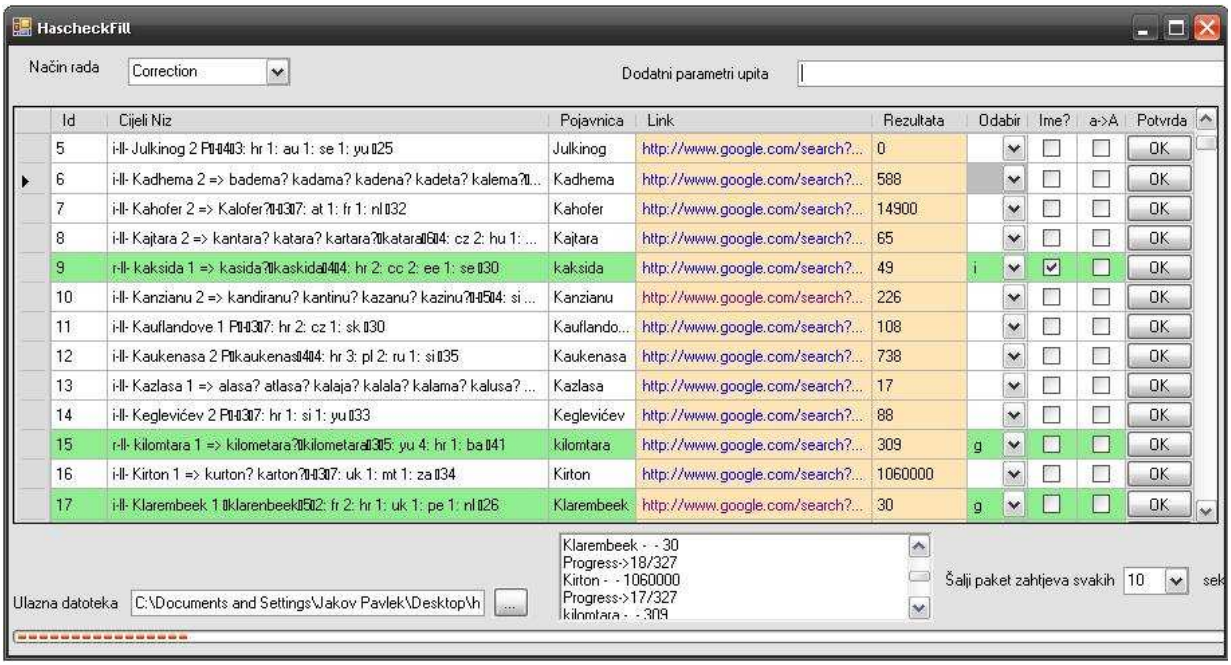


Fig. 2. Interface for manual word acquisition and classification.

Improved method provides significant advantage of processing speed. Using the interface for manual word acquisition and classification raises the speed of processing to 200 to 250 word types per hour in average. This is improved further by automatic word type classification where much work is saved by not having to classify word types, which are correctly automatically pre-classified. However, it is not known in advance which word types are correctly pre-classified and which are not, so every word has to be rechecked manually using the interface. Thus, some word types need reclassification or some spelling changes (letter capitalization or lowercasing) so the speed stabilizes at approximately 250 word types per hour. Of course, combined with *Word*'s spell checker this speed increases to approximately 330 word types per hour in average, depending on word type category.

The processing speed could be much improved only with near 100 % accurate automatic classification and somehow improved with more precise automatic classification, which could be achieved by using several independent methods of automatic classification.

4. Results

The coverage of *Word* Spell checker even in this stage of *Hascheck*'s development is still considerably high and is around 17 % (Tab. 1.). This makes it a very useful tool (even) for learning purposes of *Hascheck*. If *Hascheck* had morphology model implemented in the way *Word* Spell Checker has it, this could be significant to *Hascheck*'s text coverage.

Tab. 1.

Word coverage samples L5 and L7 for *Word Spell Checker*

Word Spell Checker	Sample *L5*	Sample *L7*
Processed	3534	2506
Total	20474	15121
%	17.3 %	16.6 %

4.1. Coverage samples

The coverage of *Wikipedia* is between 20 and 30 %, which was expectedly higher than the one of *Word Spell Checker*, but still not so high as it was expected (Tab. 2.).

Tab. 2.

Word coverage samples L5 and L7 for *Wikipedia*

Wikipedia	Sample L5	Sample L7
Processed	4486	4223
Total	20474	15121
%	21.9 %	27.9 %

However, applying the method to the rest of Web proved to be useful. The coverage is high enough (Tab. 3.), but decision-making is less secure and it demands careful manual check-up so the time saving is reduced and further improvement of the method would be welcome.

Tab. 3.

Word coverage: samples L5 and L7 for WWW *Google*

WWW_Google	Sample L5	Sample L7
Processed	12454	8392
Total	20474	15121
%	60.8 %	55.5 %

4.2. Accuracy of the method

By comparing the accuracy of the automatic method on several samples of word lists (Tab. 4.) it is possible to assess the method used in this work.

Tab. 4.

Word accuracy for different methods

Word Spell Checker	99%
Wikipedia	77%
WWW	66%

It shows that the accuracy of the method is slightly better on Wikipedia domain than in www national domains. This supports the thesis that quality of articles on Wikipedia is higher than on the average Web.

Fig. 3. shows the accuracy of the developed method on the WWW sample of words and it is equal to similarity of the files.

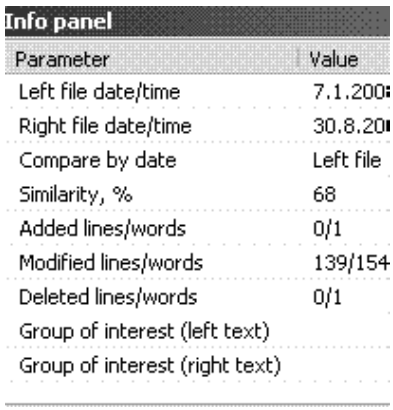


Fig. 3. File comparison for automatically classified and manually corrected sample file in order to assess the accuracy for the WWW method (68% in this sample). The file samples were evaluated using the *Compare Suite* comparison tool [9].

4.3. Total speed and time

When taking into consideration the average speeds for different methods in Tab. 5.;

Tab. 5.

Average speed for different methods

Word Spell Checker	850 words/hr
Wikipedia	250 words/hr
WWW	250 words/hr
Manual	100-150 words/hr

the total estimated time to process the whole list of 135000 words would approximately be 475 hrs as can be seen from Tab. 6.:

Tab. 6.

Estimated times to process the entire list of 135000 word types

Word Spell Checker	27 hrs
Wikipedia	151 hrs
WWW	297 hrs
Total	**475 hrs**
Manual Equivalent	900-1350 hr

Real time spent for processing and for programming differs from the estimate. In total, it's shorter, because not all of the words processed have been manually checked yet. (Tab. 7.) However, it should be longer when we add the time spent for preprocessing and programming.

Tab. 7.

Total time spent for processing and programming

L1	28 hrs
L2, L4	53 hrs (2 lists)
L3	80 hrs
L5	54
L6	97
L7	54 hrs
Total	**366 hrs**
Total average speed	**335 words/hr**

5. Conclusion

Combining new methods the processing of word types, that is, supervised learning in the Hascheck system has been accelerated and the time of decision have significantly reduced making process.

The automatization of decision-making process is very promising and should be further continually researched.

Wikipedia has turned out as a good searching domain for word acquisition. Its quality is improving and, as its volume is increasing, it should become even more appropriate. The same could be expected from *Wictionary* project in the near future.

6. Further Research & Development

Further work should include improving the developed method by:

- fixing some minor bugs in program;
- adjusting close language discrimination and overall error ration;
- enhancing the automatized preprocessing by increasing the speed of word retrieval from the web using the Google API and by distributing the program modules further;
- introducing multiple search for possible flections and using the frequency differences as help in decision making classification process.

Further research should be directed to enhancement of the automatic decision making process by replacing the decision tree method with machine learning algorithms with the same information parameters and comparing these methods.

Acknowledgments

We thank Kristina Mrvelj, Srđan Nanut, Ante Machiedo, Domagoj Mijić and Igor Bajo, the students who have also participated in the project and who have much contributed to it bringing in their enthusiasm, patience, creativity and persistency, spending many hours in word acquisition and classification.

Bibliography

[1] Batnožić Slaven, Ranilović Branko, Silić Josip: *Croatian Computerised Orthography* (in Croatian), Matica Hrvatska, Zagreb, 1996.

[2] Dembitz Šandor, Knežević, Petar, Sokele Mladen: *Hascheck - The Croatian Academic Spelling Checker*, Applications and Innovations in Expert Systems IV / Milne, Robert; Macintosh, Ann; Bremer, Max (ed.), London, Berlin, Heidelberg, New York, Barcelona, Hong Kong, etc: Springer, 1999. pp. 184-197.

[3] Domagoj Pavlešić: *Hascheck Gadget – Checking Croatian Ortography from Windows Vista Side Bar* (in Croatian), http://domagoj.bloger.hr/post/hascheck-gadget--provjera-hrvatskog-pravopisa-iz-bocne-trake-windows-viste/749306.aspx, 2008.

[4] *Hascheck - The Croatian Academic Spelling Checker Web Service* (in Croatian), http://www.hachek.tel.fer.hr, 1995.

[5] *Wikipedia - Wikipedia, the free encyclopedia,* http://en.wikipedia.org/wiki/Wikipedia, 2008.

[6] *Wikipedia:Multilingual statistics (panorama)*, http://en.wikipedia.org/wiki/Wikipedia:Multilingual_statistics_(panorama), 2008.

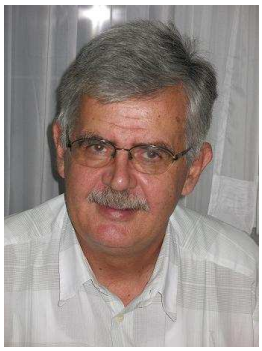
[7] Wikipedia – Statistics, http://en.wikipedia.org/wiki/Special:Statistics, 2008.

[8] Wilkinson Dennis, Huberman Bernardo: *Assessing the value of cooperation in Wikipedia*, First Monday, vol. 12, no 4., 2007.

[9] *Compare Suite - compare word, excel, html, pdf, binary files,* http://www.comparesuite.com, AKS-Labs, 2008.

Authors:

Pavlek, Jakov, dipl. ing.
University of Zagreb
Faculty of Electrical Engineering and Computing
Unska 3
10000 Zagreb
Croatia (Hrvatska)
tel: +38516129763
GSM: +385915751117
email: *jakov.pavlek@fer.hr*



Dembitz, Šandor, PhD.
University of Zagreb
Faculty of Electrical Engineering and Computing
Unska 3
10000 Zagreb
Croatia (Hrvatska)
tel: +38516129760
email: *sandor.dembitz@fer.hr*

Matasić, Marko
(University of Zagreb
Faculty of Electrical Engineering and Computing)
Kralji 19
10360 Sesvete
tel. +385958024480
email: *mmatasic@gmail.com*