УДК 811.163.41'255:004
https://doi.org/10.18485/primling.2016.17.22

Stevan Ostrogonac
Faculty of Technical Sciences, University of Novi Sad, Serbia

AUTOMATIC DETECTION AND CORRECTION OF SEMANTIC ERRORS IN TEXTS IN SERBIAN

Summary: Spell checking tools have been developed for many languages, but for most of them (including Serbian) such applications are based on simple dictionary lookup and can, therefore, handle only so-called non-word errors. This research is focused on developing advanced spell checking software for Serbian. Semantic errors are the most difficult ones to handle and this research focuses mostly on solving this problem by analyzing parallel output probabilities of word-based and morphologic class-based statistical language models. An algorithm and a prototype of a system are presented along with the results of the evaluation of the prototype.

Keywords: semantic errors, spell checking, N-gram, language model, Serbian, morphology.

1. Introduction

The Information Age has introduced numerous changes into everyday life. Written communication in forms of instant messaging, *e*-mail, social networking or transfer of textual files has become a common way of exchanging information. Spelling error handling is a very time consuming and an exhausting task. Fortunately, for many languages there are applications for spelling error correction, which solve this problem to a certain extent. Their quality depends on the language itself, available language resources and techniques used to handle spelling errors.

There are generally three types of spelling errors that need to be addressed by a spell checking software. The first type includes so-called non-word errors. For the English language, these errors can be typographic, cognitive and phonetic, depending on the reason of their occurrence (Liang, 2005). For Serbian, this division is not applicable, since almost all of the words are written as they are spoken (phonetically). The detection of non-word errors is fairly simple if an adequate vocabulary exists. Many lookup techniques were developed long ago in order to efficiently check if a word exists in a vocabulary (De Schryver, Prinsloo, 2004). A system attempting to correct an error finds a set of candidate words that could replace the misspelled chunk of text and offers the choices to the user. If the goal is to automatically correct an error, only the best candidate is searched for. The means of finding candidate sets are numerous and usually based on phonetic similarity between the misspelled text and words from

the vocabulary, as is the case with minimal edit distance (MED) algorithm (Navarro & Gonzalo, 2001). Furthermore, a rule-based approach can improve the speed and/or accuracy of the process of finding a replacement for misspelled text (Mozgovoy, 2011). Fine examples of spelling errors that could be treated by creating a set of rules include swapping the positions of adjacent letters within words or simply pressing a key adjacent to the intended one on the keyboard. Some general rules are language-independent, for example – it is far less likely for the positions of adjacent letters to be swapped at the beginning or at the end of a word then in the middle. Furthermore, many errors are related to the positions of keys on the keyboard, which means this could be used to estimate probabilities of errors for which not enough data exist within available textual corpora. In addition to non-word errors, there are grammatical and semantic errors. While the non-word errors can be detected by dictionary lookup, grammatical and semantic errors require context analysis (Verberne, 2002). Context analysis requires a morphologic dictionary and a large textual corpus for training language models. Correcting these types of errors is more complicated as well, since the candidates for replacement of misspelled words do not necessarily need to be phonetically very similar. In case of grammatical errors, this depends on the rules for inflections, which vary significantly for different languages. Semantic errors, however, represent the biggest challenge, because the meaning of the words and phrases need to be learned by a spell checking system somehow. It should be noted here that the aforementioned error categories are defined for practical purposes, mostly driven by the techniques used for their detection and correction. In reality, the nature of spelling errors is more complicated. For example, grammatical errors can be considered a subcategory of semantic errors. Furthermore, semantic errors may be categorized by the causes of their appearance – some of them were simple typographic errors which resulted in valid words instead of non-word errors, while some are related to the writer's intent.

The means of finding semantic errors and finding suitable candidate words for error correction are the focus of this research and will be discussed in detail. Section 2 describes language resources for Serbian, which are needed for constructing a spell checking software that is capable of detecting and correcting all of the previously mentioned types of errors. Collecting language resources is an expensive and a very time-consuming task. For some languages, great amounts of data have been collected (Vosse, 1992). However, for many languages, there is very little data, or the resources are not adequately exploited (Liang, 2005). For Serbian, a respectable amount of language resources has been collected and they are being used within text-to-speech (TTS) synthesis (Sečujski & Delić, 2007) and automatic speech recognition (ASR) (Janev et al, 2010) systems. These resources represent the basis for constructing advanced spell checking software as well. For advanced spell checking (handling grammatical and semantic errors), statistical language models are commonly used (Verberne, 2002), usually in combination with hand-written rules or some other techniques. Language models for Serbian are described within Section 3. Section 4 gives a detailed description of the proposed architecture for an advanced spell checker for Serbian. Semantic errors are discussed further within Section 5. Experiments along with the initial results of prototype testing are given in Section 6 and, finally, in Section 7, conclusions are drawn and future research is discussed.

2. Language Resources

Serbian is a highly inflective language, which implies that greater amounts of data (in comparison to e.g. English) are needed in order to obtain statistically relevant information regarding grammar and semantics. The resources collected so far, which are relevant to spell checking, include a morphologic dictionary and textual corpora for training language models.

The morphologic dictionary for Serbian contains around four million inflected word forms (Sečujski, 2002). An entry consists of the part-of-speech, lemma, inflected form, accentuation (within square parentheses) and values of relevant morphologic categories (such as case, number, gender, person etc.). This efficient representation combined with a fast search algorithm makes this dictionary a good basis for different practical applications.

Textual corpora have been collected from many sources and the textual content has been categorized by literary style. Four categories significant to language model training exist: administrative, scientific, literature and journalistic. Currently, the entire textual corpus contains roughly 20 million words (tokens), out of which 14.4 million tokens within journalistic corpus, 3.8 million tokens comprising literature corpus, 1.2 million tokens of scientific corpus and 0.6 million of tokens of administrative corpus. Over 350,000 different word forms (types) are present in the entire textual corpus, not counting punctuation and special characters, which are also important for training language models. Details about the textual corpora for Serbian can be found in (Ostrogonac et al, Sep. 2012), but it should be noted that the corpora are being updated continuously. Journalistic corpus proved to be the most adequate for training general-purpose language models used in applications such as spell checking.

By using morphologic dictionary and software for morphologic sentence analysis, developed within previous research (Delić et al, 2013), corpora for training class *n*-gram language models were created. The corpora consist of sentences in which the words are replaced with corresponding morphologic class IDs. The classes were defined based on morphologic information contained within the dictionary and they were given names that indicate this information. A total of 1124 classes were defined, not including punctuation marks, each of which can be considered to represent a class of its own in the context of spell checking. An example of a sentence from the class-based training corpus is given below.

> pred_d b_osn_jedan_d_sr_j prid_poz_d_sr_j i_sr_nv_aps_d_j pred_d i_zr_nv_aps_d_j i_mr_v_top_etn_g_j .
> (Derived from original sentence in Serbian: "*U jednome malom mestu u blizini Kopenhagena.*" (In a small town near Copenhagen.))

The class names in this form are convenient for manual review of the word-to-class conversion process, but they can be converted into another type of data in order to speed up spell checking.

3.Language Models

A statistical language model (LM) is a probability distribution over all possible sequences of words, given a vocabulary. It could be thought of as a black box for which

the input is a sequence of words and the output is a number that represents the probability of the input sequence. A statistical language model consists of *N*-grams, which are basically sequences of words of length *N* (or less), their corresponding probabilities (more specifically an *N*-gram probability is the probability of the last word of the sequence, given the previous *N*−1 words) and back-off coefficients, which will be explained in the following text. The probability of each sequence of words *w* can be estimated by using the *N*-gram probabilities, more precisely by using the chain rule:

$$P(w) = \sum_{i=1}^{N} P(w_i | w_1 \dots w_{i-1})$$

At the beginning of a sequence for which the probability estimation is needed, lower order *N*-gram probabilities (unigram, bigram…*N*–1-gram) are used, naturally. Lower order *N*-gram probability is also used when an *N*-gram has not been seen in training corpora and therefore there is no direct probability estimate for it within the language model. Since lower order *N*-gram probability may not be an adequate replacement for the probability of the intended *N*-gram, this back-off procedure should be adjusted. For this adjustment, previously mentioned back-off coefficients are used. Details on the mathematical basis for word sequence probability estimation, along with other details related to language models (such as methods for smoothing the probability distribution in order to avoid assigning zero probability to unseen *N*-grams) can be found in (Manning & Schütze, 1999). In (Mikolov, 2012), recurrent neural network language models (RNNLMs) are presented. The RNNLMs were not used in this research because of the computationally expensive processes they introduce. However, as the technology progresses and computers become more powerful, RNNLMs are likely to eventually replace *N*-gram LMs.

The models used within this research were trained using only the journalistic portion of the textual corpora. The training was done by using the SRILM toolkit (Stolcke, 2002). Trigram models were used for tests, since it was shown within a previous research (Ostrogonac et al, Nov. 2012) that the analysis of longer contexts does not increase the accuracy of spelling error detection because current training corpus is insufficient for accurate estimation of 4-gram of 5-gram probabilities. However, class-based models can represent longer contexts well, even with the current training corpus, but for the application architecture, which will be described within the next section, parallel use of word-based and class-based LMs is necessary, and they must be LMs of the same *N*-gram order.

Parallel outputs of word-based and class-based language models are used for detecting grammatical and semantic errors in texts. However, probabilities for correctly as well as for incorrectly spelled sentences can vary significantly and are not sufficient for detecting spelling errors. Normalization by sentence length may help a little but this would still not assure high accuracy of error detection. Fortunately, alongside with the probabilities, there is also information on when the back-off to lower order *N*-grams occurs, and to which degree the back-off is being applied. The SRILM toolkit provides this information and, when combined with the output probability, this makes a good basis for grammatical and semantic error detection. This will be further discussed in Section 5, after the proposed architecture for advanced spell checking application for Serbian is introduced in the following section.

4. Proposed Architecture for an Advanced Spell Checking Application for Serbian

In previous research (Bojanić, 2012), a spell checker for Serbian (*anSpellChecker*) was developed in order to detect the non-word errors in texts. The spell checker was based on the dictionary that was mentioned in Section 2. Microsoft has also developed and included a spell checker for Serbian into Microsoft Word, but it is, like *anSpellChecker*, currently able to detect only non-word errors by applying a simple dictionary lookup. The first step towards creating an advanced spell checker for Serbian was described in (Ostrogonac et al, 2015). The schematic representation of the prototype described in that research will be given here as well, in order to provide introduction to the discussion on semantic error handling, which will be the topic of the following section.

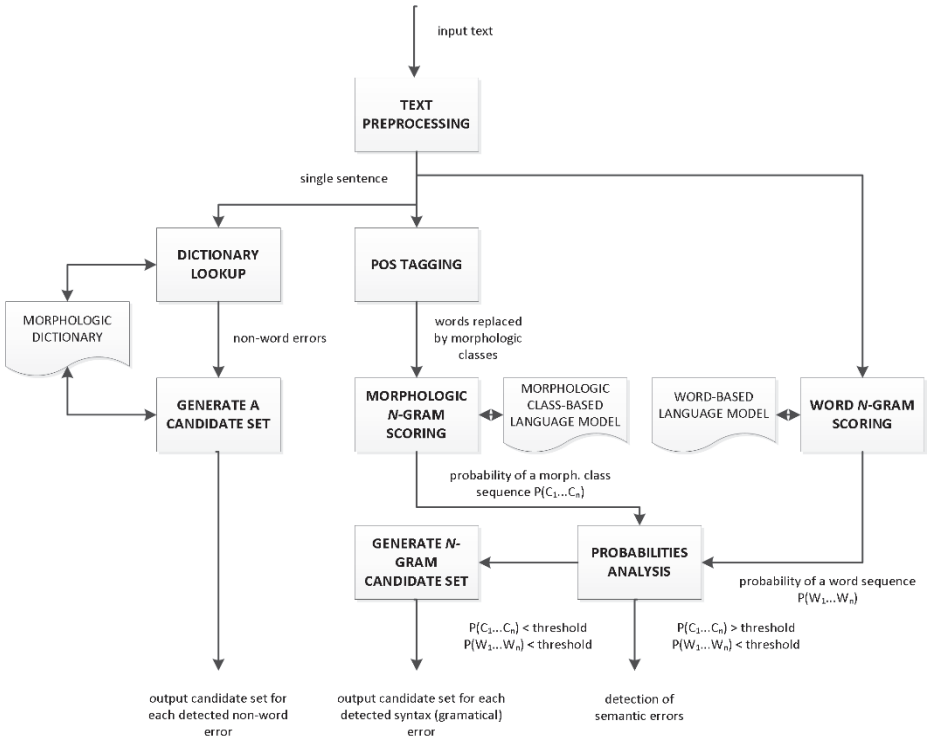

Fig. 1. Advanced spell checker for Serbian – proposed architecture

As can be seen in Fig. 1, the advanced spell checker treats three types of spelling errors. Non-word errors are detected by dictionary lookup, and the candidates for error correction are chosen by Minimal Edit Distance (MED) criterion. The other two types of errors require language models trained on a large textual corpus. For training the morphologic class-based LM, the corpus is created by part-of-speech (POS) tagging of the original textual corpora and replacing words with their corresponding morphologic class IDs.

The detection of these errors comprises calculating sentence probabilities by using both the word-based and the class-based LM (naturally, in order to obtain class-based LM output, the sentence must first be analyzed, POS tagged and converted to a sequence of morphologic class IDs) and then analyzing the probabilities (final values as well as their change after each step in the chain rule) to determine if an error occurred. For simplicity some details are omitted from Fig. 1, e.g. the probability normalization by sentence length, the fact that information on back-off occurrence is used along with the probabilities and so on.

Within initial research, the goal was to test how well the language models distinguish correctly spelled sentences from incorrectly spelled ones, when the errors are grammatical or semantic (rather than out-of-dictionary). The results were very promising – the correct sentences were given higher probabilities than the sentences derived from the original ones by replacing a single word with a different one (whether the replacement resulted in a grammatical or a semantic error, the probability decrease was detected), but the details on how the output probabilities should be used to detect errors (how the thresholds should be determined), as well as the details on how the candidate words for error correction were to be found, were not discussed at that point, since the test did not include evaluation on that level. Furthermore, the prototype was a collection of the components which would have to be manually handled in order to obtain results of such an experiment. For this research, the prototype was made functional in the sense that all the information related to error detection and correction is automatically presented to the user, and if the user sets a few parameters, he can evaluate the accuracy of the system for those parameters. However, these parameters should be fine-tuned and the means to do so will be discussed within the next section. The focus of this research is aimed at determining if the fine tuning is possible at all at this point and if not – what the impediments are.

5.Semantic Error Detection and Correction

Semantic errors, as mentioned before, represent the most difficult category of spelling errors to handle. Grammatical errors are a subcategory of semantic errors, which are usually treated separately because the techniques for detection and correction of these errors are somewhat simpler than those developed for semantic errors in general. While semantic errors occur less frequently than non-word errors, they can cause more inconveniences, since they may not be apparent to the person reading the text and may cause a wrong interpretation of the content. Unfortunately, some semantic errors are simply not possible to detect even by a human, except by the author of the text. Such errors are, however, very rare. For the rest of semantic errors, context analysis usually gives good results. For example, for the following two sentences:

Ja volim da jedem grožđe i jabuke. (I like to eat grapes and apples.)

Ja volim da jedem gvožđe i jabuke. (I like to eat iron and apples.)

The difference is hard to detect for a human, since the error within the second sentence is only one letter, and at first glance the sentences are visually the same. However, context analysis should clearly show that *gvožđe* (iron) is probably an incorrectly spelled word. It should be noted here that with semantic errors the information presented to the user of a spell-checking application should always be in the form of a warning,

as opposed to the situation with the non-word errors, which can be detected with much greater certainty. This difference can be presented in a variety of ways, for example by color-coding the error markers, as is the case with spell checking within Microsoft Word.

For a sentence **W** (and the corresponding sequence of morphologic classes **C**), context analysis for semantic error detection involves tracking the calculation of probabilities P(**W**) and P(**C**) given by word-based and morphologic class-based language models, respectively. The tracking is performed step-by-step while the chain rule is applied. This process is shown in Fig. 2 for the sentence "*Ja volim da jedem gvožđe i jabuke.*" (I like to eat iron and apples).

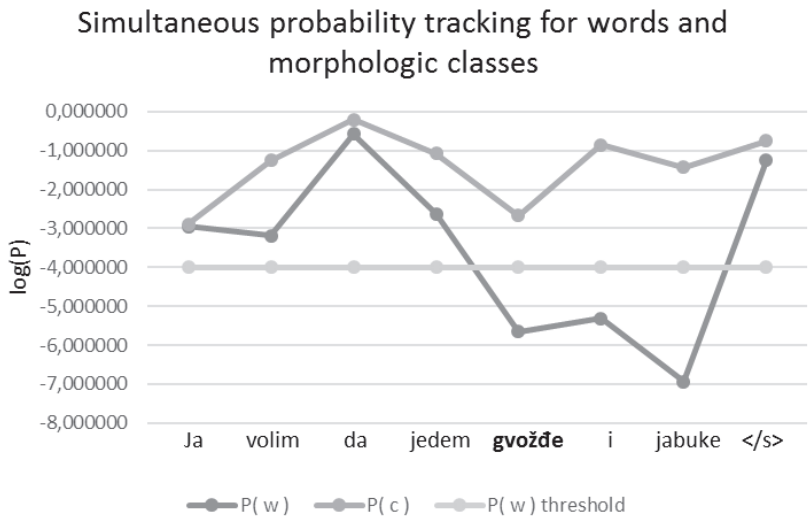

Fig. 2. Word-based and morphologic class-based LM probability estimates in case a semantic error exists in text

Logarithmic values of P(**w**) and P(**c**) are used, where **w** and **c** represent current *N*-grams that are being evaluated. It should be noted that P(**w**) and P(**c**) are used to detect errors by comparing them with threshold values, but P(**W**) and P(**C**) (probabilities of entire sentences) need to be calculated as well, as they can be useful in determining threshold values (e.g. if a sentence as a whole has a relatively low probability, the thresholds should be accordingly low as well). If a semantic error exists within a sentence, when the chain rule includes the incorrectly spelled word, the probability given by the word-based LM will most likely decrease significantly in comparison to the previous value, and the low values will continue in the following steps, the number of which depends on the order of the language model (as well as on the number of remaining words in the sentence). At the same time, morphologic class-based LM should indicate no such decrease (unless the error falls into the category of grammatical errors, in which case special techniques are to be applied for detection of the exact location of the error as well as for finding the candidate set of words for error correction). As can be seen, the location of the semantic error is fairly straightforward to determine, as is often the case. However, in some cases, when back-off procedure is applied, the estimated probability of an *N*-gram may be significantly higher (or lower) than the "real" probability. In these cases, the location of an error cannot be determined solely by

analyzing probability values in the neighboring steps. Fortunately, by using information on whether back-off has been applied and to which degree, this problem can often be resolved. Namely, if the number of consecutive steps in which the probability P(**w**) is low (while P(**c**) does not show the same) is smaller than the order of LMs, the back-off procedure probably caused incorrect probability estimation. The other reason would be that the incorrect word is frequent in a context defined by the particular *N*-gram (even though in a longer context it does not make sense), but in that case the probability decrease should still be observable in the neighboring steps of calculation. In any case, by analyzing the neighboring steps, the exact location of an error can be determined with fair certainty. For example, if trigram LMs are used and the *N*-gram probability is very low in two consecutive steps (and these are not the last two steps of calculation for the sentence), the erroneous word is most likely located either at the first step with probability decrease or at the previous step. A sketch of this situation is shown in Fig. 3 (Situation 1), along with another frequent situation.

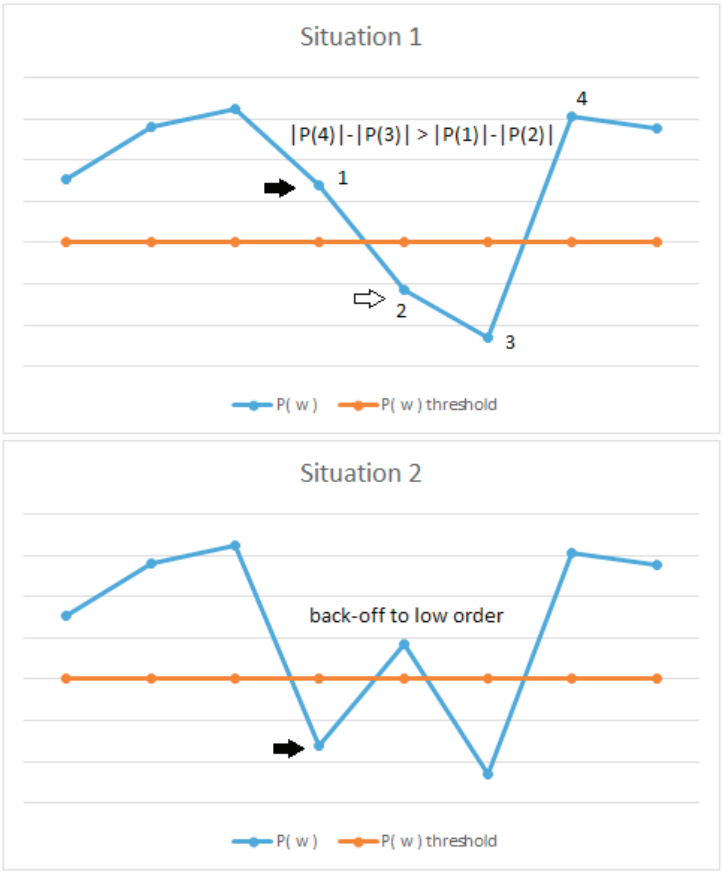

Fig. 3. Sketches of some of the typical situations when detecting semantic errors based on word sequence probabilities (it is implied that class sequence probabilities do not show significant decrease)

The decision on the location can, in the first case, be reached by simply comparing the absolute values of differences between probabilities at the steps 1 and 2 (in Fig. 3) and 3 and 4. Data analysis confirmed that if the absolute value of the decrease of probability between steps 1 and 2 is higher than the absolute value of the increase of probability between steps 3 and 4, the error had most likely occurred at step 1. Otherwise, the location is probably at step 2. Situation 2, which is illustrated in Fig. 3 can be resolved by taking into account the degree to which the back-off procedure has been applied at the point in between two steps for which the probabilities are below threshold, since back-off is likely to cause inaccurate probability estimation. Furthermore, if the decrease of probability has been observed in only one step of calculation, it is likely that it was simply a rare *N*-gram, rather than a semantic error (unless it is the last step of calculation, in which case the degree of back-off could be considered as an additional indicator). As mentioned before, the probability of the entire sentence can help determine the threshold for error detection. If three steps show probability decrease, the first of those steps is likely to be the one indicating a semantic error, as was shown in Fig. 2. If more than three steps indicate probability decrease, either adjacent semantic errors occurred, or several rare *N*-grams happened to be located near each other. In these cases, it is best to warn the user that there might be one or more errors in the sentence, but candidate sets should not be searched for, since that would be computationally expensive and probably wouldn't give adequate results.

It is obvious that different situations can be expected and, for each of them, the algorithm for error detection should be defined and certain parameters (thresholds) should be adjusted. There are several ways to determine the parameters. The best approach includes obtaining a textual corpus which contains semantic errors in each sentence. Such a corpus can be constructed by replacing a word (or multiple words, depending on the situations being simulated) within a sentence with another word that belongs to the same POS. The pairs of original sentences and sentences with semantic errors can then be used to analyze probability changes and learn appropriate values of parameters for error detection. If semantic information is previously extracted (which can be done in a variety of ways, but an approach for word clustering based on semantics for Serbian has been described in (Ostrogonac et al, Nov. 2015)), then along with parameters for error detection, additional information for error correction can be learned. Namely, the value of probability decrease would not only indicate a semantic error, but it would also indicate the degree of change in semantics, meaning that the group of initial candidates for error correction could be reduced by using semantic similarity (difference) information. This would make the search for error correction candidate set faster.

As it is for all the spelling error types, semantic error correction means finding the most appropriate replacements for a misspelled word. Automatic error correction would mean finding one, most adequate, replacement and performing the replacement automatically, but this is not a good practice when it comes to semantic errors, since the frequency of false positives is higher in comparison to the case of detecting non-word errors. Therefore, it is a much better approach to find a certain number of candidate words for error correction and leave it to the user to choose one of them (or even a word

that is not among the offered set). The described approach for Serbian suggests that the candidate words should be sought within the morphologic class to which the original (misspelled) word belongs. Furthermore, the initial candidate list can be reduced, if necessary, by using previously extracted semantic information, as mentioned before. After that, a shortlist of candidates can be found by applying MED algorithm. The MED search can be improved by implementing advanced scoring, which would incorporate information on the position of keys on the keyboard, as well as the likelihood of adjacent letters being swapped. After the shortlist of candidates is created, the candidates should, naturally, be organized in the descending probability order before being presented to the user. This is done by replacing the original word with each of the candidates and recalculating sentence probabilities. The word that causes the highest sentence probability increase is the most likely candidate for error correction, and so on.

However, there are errors that require special treatment. Some of those errors are related to the incorrect insertion of spaces between words. There are also issues related to nonstandard use of punctuation marks, which is a frequent problem (that can significantly influence semantics), especially in colloquial writing. These and many other types of problems are best addressed by incorporating hand-written rules.

6. Initial Experimental Results

The initial evaluation of the prototype of the advanced spell checker for Serbian included the analysis of word-based and class-based LMs outputs to a hundred sentences, each containing one semantic error. Punctuation marks were disregarded at this point as well as the information related to capitalization. Even though the evaluation experiment was not extensive enough to determine general parameter values related to semantic error detection, it was adequate for gaining an insight into the potential of the described approach. The general conclusion was that the available information, combined with probability analysis on a larger corpus containing sentences with artificially inserted semantic errors, can result in a fine-tuned spell checker which would be able to detect semantic errors with great accuracy. Furthermore, it is clear that the textual corpus for training word-based LMs needs to be significantly larger, since false positives are frequent at the moment, and back-off is also frequent even at phrases that can be considered to be common in everyday use. Since the results of the experiment are practically data which need to be analyzed, only a few examples will be shown and discussed here. In Fig. 4, LM output data are given for the sentences, the left side containing the word-based LM output, and the right side containing the morphologic class based LM output. The first steps of calculation that include semantic errors are given in bold letters.

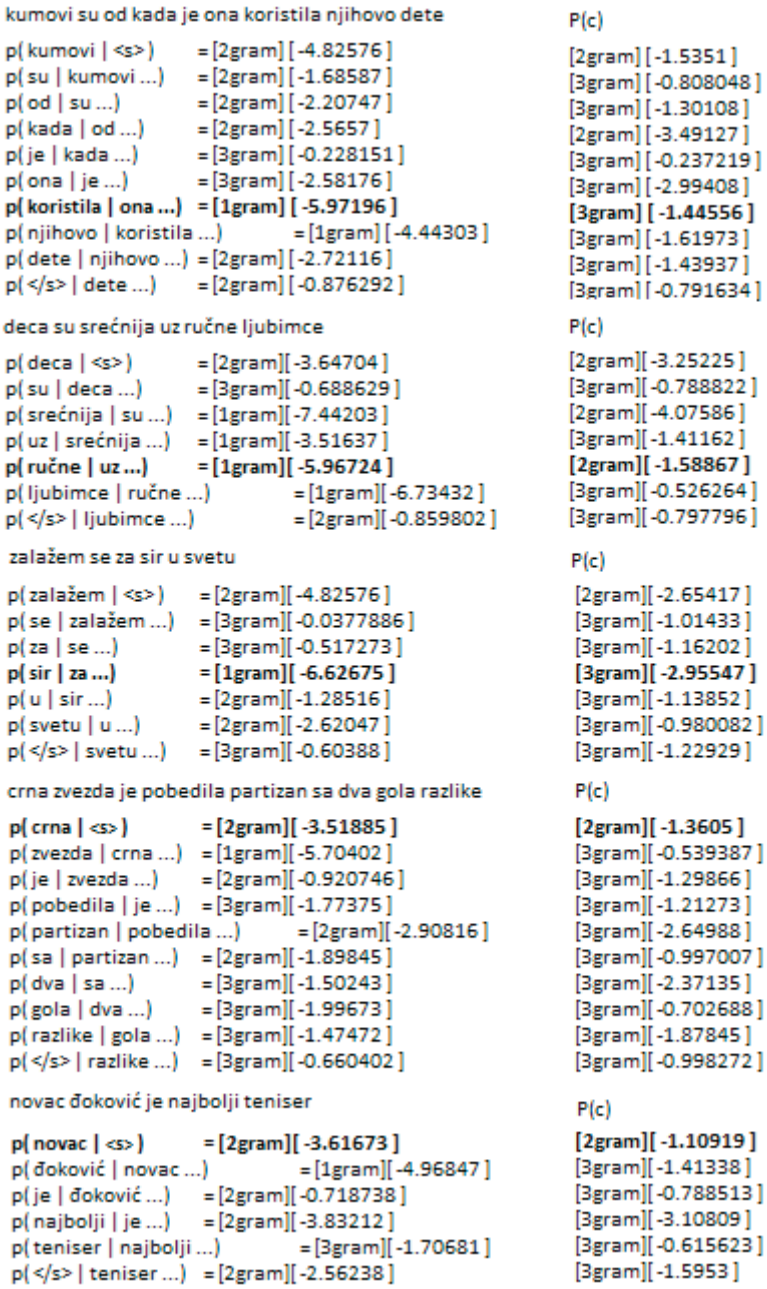

Fig. 4. Experimental results

In the first sentence, the error manifested as a probability decrease in two calculation steps, and it would be successfully detected using the rule that was illustrated in Fig. 3 – Situation 1. As can be seen, the morphologic class sequence probabilities did not show significant decrease, since the sentence is grammatically correct. In the second sentence, the misspelled word "*ručne*" also manifested as a probability decrease in two steps. However, the exact location would not be accurately determined by using the previously mentioned rule. This is due to a false positive at the word "*srećnija*", which caused a probability decrease in the output of the morphologic LM as well, which would classify it as a grammatical error. The reason this happened is the lack of data in the training corpus, which is indicated by back-off to unigram (and bigram for morphologic LM). The third sentence illustrates a situation where the error manifested in only one point, but it can be seen that the probability of the sentence as a whole is very high, indicating that even though back-off was applied, the extreme probability decrease still indicates an error, and the back-off degrees in the following steps confirm the error location. The fourth and the fifth sentence illustrate how a semantic error at the beginning of a sentence can be detected. Namely, the probabilities, being relatively low (even though no back-off was applied), indicate that the first word is uncommon at the start of a sentence. Furthermore, based on the back-off and the low probabilities at second steps, the first two words probably do not constitute a meaningful phrase, which, along with the probabilities in the first steps, indicate the locations of the errors to be at the starts of the sentences.

Many different situations were encountered within the experimental set of sentences, but there is a number of very common situations, which can be handled efficiently.

7. Conclusion and Future Research

Designing an advanced spell checking software is a complex task, which requires extensive experimenting and many iterations of testing and improving the algorithm. The work presented in this paper represents an important step in the development of such an application for Serbian, providing valuable information for further progress. The experiment verified the potential the described LMs have when it comes to semantic error detection.

Further research needs to be focused on dealing with the problems that were observed so far. A much larger textual corpus for training language models is necessary. Furthermore, a large corpus of sentences containing semantic errors needs to be created in order to obtain detailed information on how to adapt the error detection algorithm to a variety of situations. Semantic information extraction is an important issue as well, since it would contribute to the efficiency of the error handling process.

Acknowledgments: This research was supported by the Ministry of Education and Science of the Republic of Serbia under the Research grant TR32035.

References:

Bojanić M. et al. (2012), *A detector of spelling errors for Serbian - anSpellChecker*, Technical Solution: Prototype, Faculty of Technical Sciences and AlfaNum, Novi Sad, Serbia, implemented and verifies in 2012.

Delić V. et al. (2013), *Speech and Language Resources within Speech Recognition and Synthesis Systems for Serbian and Kindred South Slavic Languages*, Lecture notes in computer science, No LNAI 8113, pp. 319-326, ISSN 0302-9743, 15. SPECOM, Speech and Computer, Plzeň: Faculty of Applied Sciences, University of West Bohemia, Plzeň, Czech Republic, St. Petersburg Institute for Informatics and Automation, Russian Academy of Sciences, 1-5 September, pp. 319-326, ISBN 978-3-319-01930-7

De Schryver G-M, Prinsloo D. J. (2004), *Spellcheckers for the South African languages, Part 1: The status quo and options for improvement*, South African Journal of African Languages 24(1), pp. 57-82.

Janev M. et al. (2010), *Eigenvalues Driven Gaussian Selection in Continuous Speech Recognition Using HMMs With Full Covariance Matrices*, Applied Intelligence, pp.107-116.

Liang H. L. (2005.), *Spell Checkers and Correctors: A Unified Treatment*, Master's thesis, University of Pretoria.

Manning C, Schütze H. (1999), Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, May.

Mikolov T. (2012), *Statistical language models based on neural networks*, in PhD Thesis, Brno University of Technology.

Mozgovoy M. (2011), *Dependency-Based Rules for Grammar Checking with LanguageTool*, Proceedings of the Federated Conference on Computer Science and Information Systems, pp. 209–212.

Navarro, Gonzalo (2001), *A guided tour to approximate string matching*, ACM Computing Surveys 33 (1): 31–88. [doi:10.1145/375360.375365].

Ostrogonac S. et al. (2012), *A Language Model for Highly Inflective Non-Agglutinative Languages*, 10. SISY - International Symposium on Intelligent systems and Informatics, Subotica: IEEEXplore, 20-22. September, pp. 177-181, ISBN 978-1-4673-4749-5

Ostrogonac S. et al. (2012), *Impact of training corpus size on the quality of different types of language models for Serbian*, 20. Telecommunications forum TELFOR, Belgrade, 20-22. November.

Ostrogonac S. et al. (2015), *The Use of Statistical Language Models for Grammar and Semantic Error Handling in Spell Checking Applications for Serbian*, 12th International Conference on Electronics, Telecommunications, Automation and Informatics, ETAI 2015, Ohrid, Makedonija, ISBN: 978-9989-630-76-7.

Ostrogonac S. et al. (2015), *Automatic Word Clustering Based on Semantics - an Approach for Serbian*, 3rd International Acoustics and Audio Engineering Conference, TAKTONS 2015, Novi Sad, Serbia, November, pp. 36-37, ISBN: 978-86-7892-758-4.

Sečujski M. (2002), *Accentuation Dictionary for Serbian Intended for Text-to-Speech Technology*, Proceedings of DOGS, pp.17-20, Novi Sad, Serbia.

Sečujski M, Delić V. (2007), *An Overview of the AlfaNum Text-to-Speech Synthesis System*, 12. SPECOM, Speech and Computer, Moskva: Moskovski državni lingvistički univerzitet, 15-18 Oktobar, pp. 3-7, ISBN 6-7452-0110-x

Stolcke A. (2002), *SRILM – an extensible language modeling toolkit*, Proceedings of ICSLP, vol. 2, pp. 901-904, Denver, USA.

Verberne S. (2002), *Context-sensitive spell checking based on word trigram probabilities*, Master's thesis, University of Nijmegen.

Vosse T. (1992), *Detecting and correcting morpho-syntactic errors in real texts*, Proceedings of the third conference on Applied natural language processing, March 31-April 03, Trento, Italy. [doi:10.3115/974499.974519]

Stevan Ostrogonac

AUTOMATSKO PREPOZNAVANJE I ISPRAVLJANJE SEMANTIČKIH GREŠAKA U TEKSTOVIMA NA SRPSKOM

Sažetak: Oruđa za ispravljanje teksta su razvijena za mnoge jezike, ali za većinu njih, uključujući i srpski, takve aplikacije su zasnovane na rečnicima i stoga su primenjive samo na greške takozvanih ne-reči. Ovo istraživanje bavi se razvojem naprednijeg softvera za srpski jezik. Semantičke greške je najteže uočiti te se u radu bavimo statističkim jezičkim modelima zasnovanim na rečima i morfološkim klasama. Predstavljen je jedan algoritam i prototip sistema sa rezultatima evaluacije prototipa.

Ključne reči: semantičke greške, ispravljanje grešaka, N-gram, jezički model, srpski, morfologija.