Unsupervised Spelling Correction for the Slovak Text

Daniel HLADEK, Jan STAS, Jozef JUHAR

Department of Electronics and Multimedia Communications,
Faculty of Electrical Engineering and Informatics, Technical University of Kosice,
Park Komenskeho 13, 042 00 Kosice, Slovakia

daniel.hladek@tuke.sk, jan.stas@tuke.sk, jozef.juhar@tuke.sk

Abstract. *This paper introduces a method to automatically propose and choose a correction for an incorrectly written word in a large text corpus written in Slovak. This task can be described as a process of finding the best matching sequence of correct words to a list of incorrectly spelled words, found in the input. Knowledge base of the classification system - statistics about sequences of correctly typed words and possible corrections for incorrectly typed words can be mathematically described as a hidden Markov model. The best matching sequence of correct words is found using Viterbi algorithm. The system will be evaluated on a manually corrected testing set.*

Keywords

Automatic spelling correction, hidden Markov model, natural language processing.

1. Introduction

Important part of the natural language processing, statistical language modeling and information retrieval is a preparation of text data. The problem is that the text data often contain typographical and grammatical errors that decrease its information value. This paper will focus on the problem of correcting possibly incorrectly typed sentence. Spell-checking refers to the task of identifying and marking incorrectly spelled words in a document written in a natural language [1].

As it is known from the world of word processors, the spell-checker containing a dictionary of correct words is probably able to find incorrect word form and propose a list of corrections. Choosing the best matching correct word depends on the surrounding context. The user then can manually check the selected word, choose one of the proposed word forms, correct the word manually or proclaim that the correction is not necessary, because the word is truly correct.

The algorithm for correction of the text using an incorporated spell-checker then can be described as:

- check if the highlighted word is really incorrect,
- if it is, check if the correction is in the list of the proposed corrections,
- if a correction is in the list, choose the best possible one,
- if a correction is not available, then manually rewrite the word to its correct form.

In the case of a very large textual data, such as training corpora or web search indices it is simply not possible to use this method that requires human intervention. In order to improve the data it is necessary to reduce human work and do it as much as it is possible in an unsupervised way. One part of the work can be done using a rule based system that can solve the simplest correction by replacing by its usual correction. In the case when there are more corrections possible, a statistical approach is necessary in order to choose the best correction from the list according to the surrounding context.

2. The State of the Art

The technique of finding misspelled word and providing a list of possible corrections is known for a long time in common word processing software, such as Microsoft Word or OpenOffice Writer. It is common that these applications also provide other tools, checking not only spelling, but also the grammar using a rule-based system. However, proposing a correction autonomously is still not common in the office area.

According to [1], [2], errors related to the misspelled words can be categorized into two basic classes:

- **non-word errors** - where the misspelled word is not a valid word in a language,
- **real-word errors** - where the word in question is valid yet inappropriate in the context, and hence not giving the intended meaning.

Paper [1] states that human typing leads to non-word errors that can arise due to three major factors:

- **typographic errors** - a result of motor coordination slips and are related to keyboard mis-punches (e.g. "the" - "teh", "spell" - "speel"),
- **cognitive errors** - caused by the writer's misconceptions (e.g. "receive" - "recieve", "conspiracy" - "conspiricy"),
- **phonetic errors** - a result of substituting a phonetically equivalent sequence of letters (e.g. "seperate" - "separate").

Grobbelaar and Kinyua [3] proposed a system capable of providing corrections for the South African language. Sirts [4] deals with spelling errors, caused by learners of Estonian. This approach utilizes hidden Markov model (HMM) and part-of-speech (POS) tagging in order to choose the best correction of a word. In [5], Li, Duan and Zhai focused on the correction of web search queries and used HMMs with discriminative training. Lund and Ringer [6] used a decision lists in order to improve OCR recognition accuracy. Rodphon et al. [7] used bigram and trigram probability to select the best word in the OCR recognition result in the Thai language. In [8], Zhou et al. used a tribayes method to propose the best correction.

3. The Proposed Approach

The most simple correction mechanism possible is to use search and replace, where a manually created a list of incorrect expressions and their corrections are used to search and replace common errors. However, this approach have its limitations and have to be supplemented by some more sophisticated methods.

If the word is out-of-vocabulary (OOV), it can be one of:

- **word tokens** - regular word, proper name, foreign word,
- **non-word tokens** - numbers, web links, incorrectly typed words.

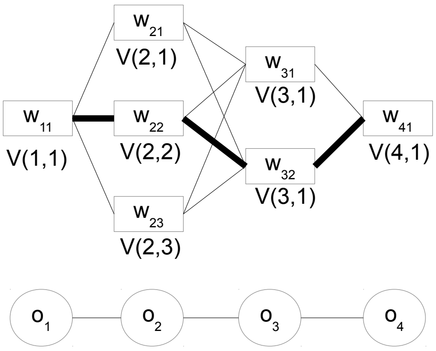

Fig. 1: Viterbi trellis for a sentence.

The first step of the automatic correction should decide, if the OOV word is feasible for automatic word correction. This step ensures that possibly useful OOV word are not "destroyed" and incorrectly replaced by one of the vocabulary words. The following heuristic rules are proposed to identify an incorrect word. The word probably contains typing error if:

- it is not in the vocabulary of the correct words,
- it is lowercase,
- it does not contain features characteristic for foreign word, such as strange letters, numerals, uncommon letter combinations.

After the words qualified for automatic correction are identified, the spell-checking can be described as a process of finding the best possible sequence of correct words, given to the list of possibly incorrect words.

The most common method for classification of the time dependent (sequence) data is hidden Markov model and the Viterbi algorithm. In this case, words distorted by errors in the sentence can be described as a sequence of observation o (bottom part of Fig. 1). Possible corrections w for these words o are hidden states. The Viterbi algorithm then can assign the most probable sequence of the corrected words to the given list of possibly incorrect words (marked by the bold lines).

To make correct representation of this problem using the HMM framework in a way similar to [9], it is necessary to correctly express basics components of the HMM: observation matrix $P(o|w)$ and state transition matrix $P(w|h)$.

Observation matrix is a probability of the occurrence of the incorrect form of the word according to its correct form. The first problem is that the number of possible incorrect word forms is infinite. Then the method of maximum likelihood estimation is impossible to use and have to be estimated in a different way. A heuristic procedure for probability $P(o|w_j)$ of observation o, according to the state w_j is then estimated as

$$P(o|w_j) = \frac{C-j}{\sum_{i=0}^{C}(C-i)}, \tag{1}$$

where j is order of the word proposed by the spell-checking module. C is a number of proposed corrections by the spell-checking module and $\sum_{i=0}^{C}(C-i)$ is a normalizing constant, so that the first proposed correction of the spell-checker has the highest probability. It assumes that the first proposed word by the spell-checking module has the highest probability of a match with current observation. Also note that this expression is not mathematically correct, because it does not ensure that sum of probabilities for all possible bad forms for the inspected state is one.

State set of a HMM is a list of all correct forms of words, given by the manually checked dictionary. The state-transition matrix is a language model of the target language and expresses probability of occurrence of a word according to the given context. The Slovak language is characterized by many possible morphological word forms. As a consequence, every operation that considers statistical information about sequences of words, such as word spelling correction, needs to have a proper method of training to improve the performance of the system [10].

In the case of n-gram language model, the maximum likelihood estimation $P(w|h)$ of the word w according to the context h is given by

$$P(w|h) = \frac{C(h,w)}{C(h)}, \tag{2}$$

where $C(h,w)$ is count of the sequence (h,w) and $C(h)$ is count of the context h in the training corpus.

However, this formula cannot be used in practice, because in the case of insufficient training data, counts of n-grams are often zero. This type of language model then incorrectly gives zero probability also for those situations that are perfectly valid in the given language. From this reason a method of adjusting resulting probability (smoothing) have to be chosen to estimate the probability of events that had not been observed in the training corpus as it is in [11].

If incorrect words, state and observation probabilities can be estimated, the most probable sequence of the corrections can be calculated using Viterbi algorithm. For every possible correct word form a trellis is constructed and every node has assigned a certain value $V(t,i)$ that is used to find the best possible sequence as a path in this trellis.

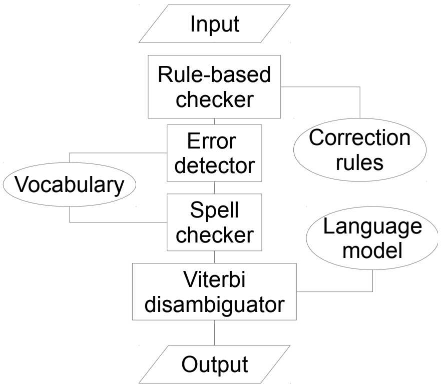

Fig. 2: The proposed system structure.

The Viterbi value of a node, representing a transition a correct word form to another (a state-transition) can be calculated using a recursive formula:

$$V(t,i) = P(o_t|w_{t_i}) \max_{j \in S_{t-1}} V(t-1,j)P(w_{t_i}|w_{t-1,j}), \tag{3}$$

where $V(t,i)$ is Viterbi value for word w_i in time t, $P(o_t|w_{t_i})$ is observation probability similar to the Eq. (1), $V(t-1,j)$ is Viterbi value of word w_j in time $(t-1)$ and $P(w_{t_i}|w_{t-1,j})$ is transition probability given by the language model, as it is described by the Eq. (2). The whole process is depicted in the Fig. 1, where each word o_t in a given sentence has some possible corrections $w_i(t)$. Then a value $V_i(t)$ is assigned for every possible correction and using backtracking from the last value it is possible to derive the best sequence of corrections for the given sentence. After all nodes in the Viterbi trellis are evaluated, the best path (the best sequence of correct words) can be found using backtracking, taking paths with the best Viterbi evaluation.

To summarize, the automatic correction algorithm, processes a given sentence:

1. applies rule based corrections in order to deal with the most simple errors,
2. identifies possibly incorrect words,
3. for every incorrect word proposes a list of corrections,
4. constructs a Viterbi trellis, evaluating all possible transitions between correct states,
5. using backtracking proposes the best sequence of correct states.

4. The Experimental Part

The most important part of any natural language processing task is a proper training corpus preparation.

Tab. 1: Evaluation of bad words detection on the testing set.

Result/Class	**Word correct**	**Contains error**
true	28 699	128
false	99	393
	Prec: 0,24	*Rec*: 0,56

Tab. 2: Evaluation of detected errors and their corrections.

Correct corrections	93	72,6 %
Bad correction proposal	15	11,7 %
Bad classifier decision	20	15,6 %
Classifier corrects	73	78,4 %

It is required the language model for the unattended spelling corrector is composed of the grammatically and typographically correct text. As a training set, a corpus created from Slovak fiction books have been used, as it is considered to be manually checked and unaffected by the typographic errors. The training corpus contains 49 592 554 tokens in 2 910 180 sentences.

The training corpus preparation step includes word tokenization, sentence boundary identification, sentence and token filtering, transcription and training sentences correction as it is noted in [12]. The list of words have been constrained to the biggest manually checked the dictionary of the Slovak words (6, 5 mil. of unique words) [13].

It is important to say that both training text for the classifier's language model and input text containing errors have very similar structure - both should be a result of the same text preparation process. For this purpose, the same method for the testing set preparation has been used on a selection of the Slovak blog web pages [12], because it is considered to contain typing errors and its style is close to the fiction, used in the training corpus. Testing part of the blog corpus has 29 319 tokens and 1 507 sentences.

In the first experiment, success rate of the bad word detection has been evaluated. Results summarized in the Tab. 2 show that the weakest part of the proposed system is identification of words where spell-check should be applied. Heuristic procedure presented above seems to be insufficient. In order to improve efficiency of the system it is required to extend dictionary to reduce the number of words that should not have proposed corrections. Otherwise, too large number of words that are good but have incorrectly assigned correction is present in the output of the system.

The second evaluation (see Tab. 1) focused on the words that were truly incorrect and the system found a correction for them. Results are comparable to the ones presented in [4], [8] that claim accuracy of spelling correction around 85 %.

It is possible to say that in some cases correction failed, because the spell-check module was not able to propose correct form and classiffier had to choose only from incorrect forms. If a correct form had been available, sometimes Viterbi algorithm chosen wrong. Anyway, it is possible to conclude that the described classifier is sufficient and the biggest room for improvement is in the dictionary, that contains a list of words, that are not qualified for correction.

The results of the experiments show that error in the assigning a correction can be made if: a) the given rules and dictionaries contain errors; b) the heuristic rules for recognition of possibly incorrect word fail; c) the estimated observation probability is not sufficiently correct; d) the given language model is not good enough; e) the spell-checking module propose wrong possible corrections. If all of these weak points are implemented in a plausible way, then the result of the human unattended correction process might be acceptable. If the output of the automatic correction is gathered, processed and manually checked by human supervisor, the whole system can be improved by analyzing the result, updating the list of good words and correction rules.

5. Conclusion

This approach can utilize human effort as much as possible and large amounts of text can be corrected in an unsupervised way. Alternatively, this approach can be used in a semi-automatic way, where the system autonomously proposes a correction and human operator can approve it. Decision then can be remembered and used in latter cases. As a result, probability of occurrence of OOV word is reduced and the well corrected training corpus should produce a better results.

This contribution is the first for the Slovak language and thus it can not be directly compared to other approaches. Still it can be used as a starting point for future research and it seems to be valuable for other languages. The presented results can be improved by better detection of the incorrect word and by improving the vocabulary. It also might be useful to try other classification techniques.

Acknowledgment

The research presented in this paper was supported by the Ministry of Education, Science, Research and Sport of the Slovak Republic under the research project MS SR 3928/2010-11 (50 %) and Research and Development Operational Program funded by the ERDF under the project ITMS-26220220141 (50 %).

References

[1] JAYALATHARACHCHI, E., A. WASALA and R. WEERASINGHE. Data-driven spell checking: The synergy of two algorithms for spelling error detection and correction. In: *International Conference on Advances in ICT for Emerging Regions (ICTer), 2012*. Colombo: IEEE, 2012, pp. 7–13. ISBN 978-1-4673-5529-2. DOI: 10.1109/ICTer.2012.6422063.

[2] KUKICH, K. Techniques for automatically correcting words in text. *ACM Computing Surveys*. 1992, vol. 24, iss. 4, pp. 377–439. ISSN 0360-0300. DOI: 10.1145/146370.146380.

[3] GROBBELAAR, L. and D. J. M. KINYUA. A spell checker and corrector for the native South African language, South Sotho. In: *Proceedings of 2009 Annual Conference of the Southern African Computer Lecturers' Association, SACLA 2009*. Mpekweni Beach Resort: ACM, 2009, pp. 50–59. ISBN 978-1-60558-683-0. DOI: 10.1145/1562741.1562747.

[4] SIRTS, K. Noisy-channel spelling correction models for Estonian learner language corpus lemmatisation. In: *The 5th International Conference Human Language Technologies–The Baltic Perspective, HLT 2012*. Tartu: IOS Press, 2012, pp. 213–220. ISBN 978-1-61499-132-8. DOI: 10.3233/978-1-61499-133-5-213.

[5] LI, Y., H. DUAN and C.-X. ZHAI. A generalized hidden Markov model with discriminative training for query spelling correction. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 12*. Portland: ACM, 2012, pp. 611–620. ISBN 978-1-4503-1472-5. DOI: 10.1145/2348283.2348365.

[6] LUND, W. B. and E. K. RINGGER. Error correction with in-domain training across multiple OCR system outputs. In: *11th International Conference on Document Analysis and Recognition, ICDAR 2011*. Beijing: IEEE, 2011, pp. 658–662. ISBN 978-1-4577-1350-7. DOI: 10.1109/ICDAR.2011.138.

[7] RODPHON, M., K. SIRIBOON and B. KRUATRACHUE. Thai OCR error correction using token passing algorithm. In: *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, PACRIM 2001*. Victoria: IEEE, 2001, pp. 599–602. ISBN 0-7803-7080-5. DOI: 10.1109/PACRIM.2001.953704.

[8] ZHOU, Y., S. JING, G. HUANG, S. LIU, and Y. ZHANG. A correcting model based on tribayes for real-word errors in English essays. In: *Fifth International Symposium on Computational Intelligence and Design (ISCID), 2012*. Hangzhou: IEEE, 2012, pp. 407–410. ISBN 978-1-4673-2646-9. DOI: 10.1109/ISCID.2012.108.

[9] HLADEK, D., J. STAS and J. JUHAR. Dagger: The Slovak morphological classiffer. In: *Proceedings of 54th International Symposium ELMAR 2012*. Zadar: IEEE, 2012, pp. 195–198. ISBN 978-1-4673-1243-1.

[10] STAS, J., D. HLADEK, M. PLEVA and J. JUHAR. Slovak language model from internet text data. In: *Proceedings of the Third COST 2102 international training school conference on Toward autonomous, adaptive, and context-aware multimodal interfaces: theoretical and practical issues*. Berlin: Springer, 2011, pp. 340–346. ISBN 978-3-642-18183-2. DOI: 10.1007/978-3-642-18184-9_29.

[11] STAS, J., D. HLADEK and J. JUHAR. Language model adaptation for Slovak LVCSR. In: *Proceedings of the International Conference on Applied Electrical Engineering and Informatics, AEI 2010*. Venice: ACM, 2010, pp. 101–106. ISBN 978-80-553-0519-6.

[12] HLADEK, D. and J. STAS. Text gathering and processing agent for language modeling corpus. In: *12th International Conference on Research in Telecommunication Technologies, RTT 2010*. Velke Losiny: VSB–Technical University of Ostrava, 2010, pp. 137–140. ISBN 978-80-248-2261-7.

[13] KRAJCI, S., M. MATI and R. NOVOTNY. Morphonary: A Slovak language dictionary, tools for acquisition, organisation and presenting of information and knowledge. In: *Proceedings in Informatics and Information Technologies*. Bratislava: STU, 2006, pp. 162–165. ISBN 80-227-2468-8.

About Authors

Daniel HLADEK was born in Kosice, Slovakia in 1982. He received his M.Sc. (Ing.) from Artificial Intelligence in 2007 and Ph.D. in 2009. He is currently working as a post-doctoral researcher at the Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics, Technical University of Kosice with a focus on natural language processing, speech and audio processing and intelligent decision methods. He is an author and co-author of more than 30 conference and journal papers from this area.

Jan STAS was born in Bardejov, Slovakia in

1984. He received his M.Sc. from Telecommunications in 2007 and Ph.D. in 2011. He is currently working as a post-doctoral researcher at the Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics, Technical University of Kosice with a focus on natural language processing and understanding and language modeling. He is an author and co-author of more than 30 conference and journal papers in this area.

Jozef JUHAR was born in Poproc, Slovakia in 1956. He received his M.Sc. (Ing.) in Radioelectronics from in 1980 and Ph.D. in 1991. He is currently working as a Full Professor at the Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics, Technical University of Kosice. He is an author and co-author of more than 220 scientific papers. His research interests include digital speech and audio processing, speech synthesis and development in spoken dialogue and speech recognition systems.