AN IMPROVED LEVENSHTEIN ALGORITHM FOR SPELLING CORRECTION WORD CANDIDATE LIST GENERATION

HANAN NAJM ABDULKHUDHUR

COLLEGE OF ARTS AND SCIENCE

UNIVERSITI UTARA MALAYSIA

2016

Permission to Use

In presenting this dissertation in fulfillment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this dissertation in any manner, in whole or in part, for the scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this dissertation or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my dissertation.

Requests for permission to copy or to make other use of materials in this dissertation, in whole or in part should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Senarai calon terhasil dalam pembetulan ejaan adalah satu proses untuk mencari kata-kata dari leksikon yang hampir sama dengan perkataan yang tidak tepat. Algoritma paling banyak digunakan untuk menjana senarai calon untuk kata-kata yang tidak tepat adalah berdasarkan jarak Levenshtein. Walau bagaimanapun, algoritma ini mengambil masa yang terlalu lama apabila terdapat bilangan besar kesilapan ejaan. Sebabnya ialah bahawa pengiraan algoritma Levenshtein termasuk operasi yang menghasilkan jajaran dan pengisian sel-sel jajaran dengan membandingkan huruf-huruf perkataan yang tidak betul dengan huruf-huruf perkataan dari leksikon. Oleh kerana kebanyakan leksikon mengandungi berjuta-juta perkataan, maka operasi ini akan diulang berjuta-juta kali bagi setiap perkataan tidak tepat untuk menjana senarai calonnya. Kajian ini men ambahbaikkan algoritma Levenshtein dengan merekabentuk teknik operasi yang telah dimasukkan dalam algoritma ini. Teknik operasi yang dicadangkan meningkatkan algoritma Levenshtein dari segi masa pemprosesan perlaksanaannya tanpa menjejaskan ketepatannya. Ia mengurangkan langkah operasi yang diperlukan untuk mengukur nilai sel-sel dalam baris dan lajur pertama, baris dan lajur kedua serta baris dan lajur ketiga dalam jajaran Levenshtein. Algoritma Levenshtein yang telah tingkatkan telah dibandingkan dengan algoritma asal. Hasil kajian menunjukkan bahawa prestasi algoritma yang dicadangkan melebihi prestasi algoritma Levenshtein asal dari segi masa pemprosesan, iaitu sebanyak 36.45% manakala ketepatan kedua-dua algoritma adalah masih sama.

Kata Kunci: Algoritma Levenshtein, Masa pemprosesan, Penghasilan senarai calon,

Pmbetulan Kesilapan.

Abstract

Candidates' list generation in spelling correction is a process of finding words from a lexicon that should be close to the incorrect word. The most widely used algorithm for generating candidates' list for incorrect words is based on Levenshtein distance. However, this algorithm takes too much time when there is a large number of spelling errors. The reason is that calculating Levenshtein algorithm includes operations that create an array and fill the cells of this array by comparing the characters of an incorrect word with the characters of a word from a lexicon. Since most lexicons contain millions of words, then these operations will be repeated millions of times for each incorrect word to generate its candidates list. This dissertation improved Levenshtein algorithm by designing an operational technique that has been included in this algorithm. The proposed operational technique enhances Levenshtein algorithm in terms of the processing time of its executing without affecting its accuracy. It reduces the operations required to measure cells' values in the first row, first column, second row, second column, third row, and third column in Levenshtein array. The improved Levenshtein algorithm was evaluated against the original algorithm. Experimental results show that the proposed algorithm outperforms Levenshtein algorithm in terms of the processing time by 36.45% while the accuracy of both algorithms is still the same.

Keywords: Levenshtein Algorithm, Processing time, Candidates' list generation, Errors correction.

Acknowledgement

Each part of this dissertation is guided, inspired, and supported by many people. Firstly, I would like to thank all the members of my family especially my husband and my parents for their unconditional support. My goal would not be achieved without them. The most important support and guidance were from my research supervisors Dr. Shahrul Azmi Mohd Yusof and Prof. Madya Dr. Yuhanis Yusof. Thank you very much for your great help and support. It is an honor for me to do a research under your supervisions. I would like to thank all the academic and technical staff in School of Computing of Utara Universiti Malaysia for their help in the study process and providing all the excellent facilities. Finally, I would like to thank all my friends for their support.

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

LA	Levenshtein algorithm
HD	Hamming distance
DD	Damerau distance
LCS	Longest common subsequence
OT	Operational technique
ILA-OT	Improved Levenshtein algorithm by using the proposed OT
MVFRFC	Measure values of the first row and first column
MVSRSC	Measure values of the second row and second column
PT	Processing time
PD	Percentage decrease

CHAPTER ONE
INTRODUCTION

1.1 Background

Spelling correction is the process of detecting and repairing spelling errors in a text. Research in spelling correction is not new; it started in the mid of 1960, and many algorithms for spelling correction have been suggested since then (Mahdi, 2012). Spelling correction can be either manual or automatic. The first type allows intervention of humans in the correction process. The second type, a system will decide the correction to replace an incorrect word by choosing the best candidate word without human's intervention (Bassil & Alwani, 2012b).

Most methods of automatic spelling correction have three functions: error detection, generation of candidates, and error correction (Naseem & Hussain, 2007). The first function is to find incorrect words in the output text. The second function is to generate candidate words from a lexicon for each of the incorrect words. Candidate list generation is a process of finding words from a lexicon that should be close to the incorrect word. For example, the candidates' list generated from a lexicon for the incorrect word "*czp*" are "*cup*", "*cap*", and "*cop*". The last function is to correct all incorrect words by selecting the best candidate to replace with each incorrect word.

The process of generating candidates list can be achieved by using a specific algorithm. An algorithm is a set of operations that will be performed on some data to solve a specific problem. In general, algorithms can be classified according to their optimal solution into two categories: exact and approximate. In execution, exact algorithms will reach an optimal solution while approximation algorithms can be

The contents of the thesis is for internal user only

REFERENCES

Adhitama, P., Kim, S. H., & Na, I. S. (2014). Lexicon-Driven Word Recognition Based on Levenshtein Distance. *International Journal of Software Engineering and Its Applications, 8*(2), 11-20.

Ahmed, B. (2015). *Lexical Normalisation of Twitter Data.* Paper presented at the Science and Information Conference (SAI), 2015.

Al-Bakry, A. M., & Al-Rikaby, M. K. (2015). *Enhanced Levenshtein Edit Distance Method functioning as a String-to-String Similarity Measure.* Paper presented at the Proceedings of the Networks Security and Distributed Systems (NSDS'2015).

Al-Masoudi, A. F. R., & Al-Obeidi, H. S. R. (2015). Smoothing Techniques Evaluation of N-gram Language Model for Arabic OCR Post-processing. *Journal of Theoretical and Applied Information Technology, 82*(3), 432-439.

Al-Zaydi, Z. Q., & Salam, H. (2015). Multiple Outputs Techniques Evaluation for Arabic Character Recognition. *International Journal of Computer Techniques (IJCT), 2*(5), 1-7.

Alobaedy, M. M. T. (2015). *Hybrid Ant Colony System Algorithm For Static And Dynamic Job Scheduling In Grid Computing.* (PhD thesis, Universiti Utara Malaysia, Kedah, Malaysia). Retrieved from http://etd.uum.edu.my/id/eprint/5382

Andoni, A., & Krauthgamer, R. (2012). The smoothed complexity of edit distance. *ACM Transactions on Algorithms (TALG), 8*(4), 44.

Andoni, A., & Onak, K. (2012). Approximating edit distance in near-linear time. *SIAM Journal on Computing, 41*(6), 1635-1648.

Attia, M., Pecina, P., Samih, Y., Shaalan, K. F., & van Genabith, J. (2012). *Improved Spelling Error Detection and Correction for Arabic.* Paper presented at the International Conference on Computational Linguistics (COLING), Mumbai, India.

Bard, G. V. (2007). *Spelling-error tolerant, order-independent pass-phrases via the Damerau-Levenshtein string-edit distance metric.* Paper presented at the Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68.

Bassil, Y., & Alwani, M. (2012a). Context-sensitive Spelling Correction Using Google Web 1T 5-Gram Information. *Computer and Information Science, 5*(3), 37-48.

Bassil, Y., & Alwani, M. (2012b). Ocr post-processing error correction algorithm using google online spelling suggestion. *Journal of Emerging Trends in Computing and Information Sciences, 3*(1), 90-99.

Batawi, Y., & Abulnaja, O. (2012). Accuracy Evaluation of Arabic Optical Character Recognition Voting Technique: Experimental Study. *IJECS: International Journal of Electrical & Computer Sciences, 12*(1), 29-33.

Bergroth, L., Hakonen, H., & Raita, T. (2000). *A survey of longest common subsequence algorithms.* Paper presented at the Proceedings of the Seventh International Symposium on String Processing Information Retrieval (SPIRE'00).

Burkhardt, S., & Kärkkäinen, J. (2002). *One-gapped q-gram filters for Levenshtein distance.* Paper presented at the Proceedings of the Combinatorial Pattern Matching.

Cooper, L., & Cooper, M. W. (1981). *Introduction to dynamic programming*: Pergamon Press New York.

Crumrine, K. T., Ritschel, J. D., & White, E. (2014). Earned Schedule 10 Years Later Analyzing Military Programs: DTIC Document.

Daðason, J. F. (2012). *Post-Correction of Icelandic OCR Text.* (Master's thesis), University of Iceland, Reykjavik, Iceland.

Federico, M., & Cettolo, M. (2007). *Efficient handling of n-gram language models for statistical machine translation.* Paper presented at the Proceedings of the Second Workshop on Statistical Machine Translation.

Formiga Fanals, L., & Rodríguez Fonollosa, J. A. (2012). *Dealing with input noise in statistical machine translation.* Paper presented at the International Conference on Computational Linguistics (COLING), Mumbai, India.

Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. *International Journal of Computer Applications, 68*(13).

Google. (2015). Google Ngram Retrieved september 05, 2015, from http://storage.googleapis.com/books/ngrams/books/datasetsv2.html

Grzebala, P. B. (2016). *Private Record Linkage: A Comparison of Selected Techniques for Name Matching.* Master thesis, Wright State University, Ohio, United States.

Gupta, B., Bhatt, G., & Mittal, A. (2016). Language Identification and Disambiguation in Indian Mixed-Script. *Distributed Computing and Internet Technology* (pp. 113-121): Springer.

Habeeb, I. Q., Yusof, S. A., & Ahmad, F. B. (2014). Two Bigrams Based Language Model for Auto Correction of Arabic OCR Errors. *International Journal of Digital Content Technology and its Applications, 8*(1), 72 - 80.

Haldar, R., & Mukhopadhyay, D. (2011). Levenshtein distance technique in dictionary lookup methods: An improved approach. *arXiv preprint arXiv:1101.1232*.

Hassan, A., Noeman, S., & Hassan, H. (2008). *Language Independent Text Correction using Finite State Automata.* Paper presented at the Third International Joint Conference on Natural Language Processing (IJCNLP), Hyderabad, India.

Huldén, M. (2009). Fast approximate string matching with finite automata. *Procesamiento del lenguaje natural, 43*, 57-64.

Islam, A., & Inkpen, D. (2009). *Real-word spelling correction using Google Web 1T n-gram with backoff.* Paper presented at the International Conference on Natural Language Processing and Knowledge Engineering.

Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (2nd ed.): Pearson Education India.

Levenshtein, V. I. (1966). *Binary codes capable of correcting deletions, insertions and reversals.* Paper presented at the Soviet physics doklady.

Li, M., Zhang, Y., Zhu, M., & Zhou, M. (2006). *Exploring distributional similarity based models for query spelling correction.* Paper presented at the Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics.

Lounis, O., Guermeche, B., Eddine, S., Saoudi, L., & Benaicha, S. E. (2014). *A new algorithm for detecting SQL injection attack in Web application.* Paper presented at the Science and Information Conference (SAI), 2014.

Lu, W., Du, X., Hadjieleftheriou, M., & Ooi, B. C. (2014). Efficiently Supporting Edit Distance based String Similarity Search Using B-trees. *IEEE Transactions on Knowledge and Data Engineering, 26*(12), 2983-2996.

Lund, W. B. (2014). *Ensemble Methods for Historical Machine-Printed Document Recognition.* (PhD dissertation, Brigham Young University, Utah, United States).

Ma, D., & Agam, G. (2013). *A super resolution framework for low resolution document image OCR.* Paper presented at the IS&T/SPIE Electronic Imaging.

Maarif, H., Akmeliawati, R., Htike, Z., & Gunawan, T. S. (2014). *Complexity Algorithm Analysis for Edit Distance.* Paper presented at the International Conference on Computer and Communication Engineering (ICCCE), 2014

Mahdi, A. A. M. (2012). *Spell checking and correction for Arabic text recognition.* (Master thesis), King Fahd university of petroleum and minerals, Saudi Srabia.

Mainsah, B. O., Morton, K. D., Collins, L. M., Sellers, E. W., & Throckmorton, C. S. (2015). Moving away from error-related potentials to achieve spelling correction in P300 spellers. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 23*(5), 737-743.

Mihov, S., & Schulz, K. U. (2004). Fast approximate search in large dictionaries. *Computational Linguistics, 30*(4), 451-477.

Musa, S. M., Opara, E. U., Shayib, M. A., & Oliver, J. (2016). Measurement and Test Performance for Integrated Digital Loop Carrier for White Noise Impairment Using Interleaved Mode. *Communications of the IIMA, 14*(3), 4.

Naseem, T. (2004). *A Hybrid Approach for Urdu Spell Checking.* (Master thesis), National University of Computer & Emerging Sciences, Islamabad, Pakistan.

Naseem, T., & Hussain, S. (2007). A novel approach for ranking spelling error corrections for Urdu. *Language Resources and Evaluation, 41*(2), 117-128. doi: 10.1007/s10579-007-9028-6

Navarro, G. (2001). A Guided tour to approximate string matching. *ACM Computing Surveys (CSUR), 33*(1), 31-88.

Navarro, G., Grabowski, S., Mäkinen, V., & Deorowicz, S. (2005). Improved time and space complexities for transposition invariant string matching *Technical Report TR/DCC-2005-4, Department of Computer Science*: University of Chile.

Pal, S., & Rajasekaran, S. (2015). *Improved algorithms for finding edit distance based motifs.* Paper presented at the IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2015.

Phillips, C. R. (2015). *Employing an Efficient and Scalable Implementation of the Cost Sensitive Alternating Decision Tree Algorithm to Efficiently Link Person Records.* Master thesis, Texas State University, San Marcos, United States.

Polyanovsky, V., Roytberg, M. A., & Tumanyan, V. G. (2011). Comparative analysis of the quality of a global algorithm and a local algorithm for alignment of two sequences. *Algorithms for Molecular Biology, 6*, 25.

Popescu, O., & Vo, N. P. A. (2014). *Fast and Accurate Misspelling Correction in Large Corpora.* Paper presented at the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Pradhan, N., Gyanchandani, M., & Wadhvani, R. (2015). A Review on Text Similarity Technique used in IR and its Application. *International Journal of Computer Applications, 120*(9).

Rardin, R. L., & Uzsoy, R. (2001). Experimental evaluation of heuristic optimization algorithms: A tutorial. *Journal of Heuristics, 7*(3), 261-304.

Rieck, K., & Wressnegger, C. (2016). Harry: A Tool for Measuring String Similarity. *Journal of Machine Learning Research, 17*(9), 1-5.

Ruoro, S. W. (2009). *A Parallel Corpus Based Translation Using Sentence Similarity.* PhD dissertation, University of Nairobi, Nairobi, Kenya.

Sheng, C., Tao, Y., & Li, J. (2012). Exact and approximate algorithms for the most connected vertex problem. *ACM Transactions on Database Systems (TODS), 37*(2), 12.

Siklósi, B., Novák, A., & Prószéky, G. (2016). Context-aware correction of spelling errors in Hungarian medical documents. *Computer Speech & Language, 35*, 219-233.

Singh, S. P., Kumar, A., Darbari, H., Chauhan, S., Srivastava, N., & Singh, P. (2015). Evaluation of Similarity metrics for translation retrieval in the Hindi-English Translation Memory. *International Journal of Advanced Research in Computer and Communication Engineering, 4*(8).

Ukkonen, E. (1985). Finding approximate patterns in strings. *Journal of algorithms, 6*(1), 132-137.

Vargas, S. G. J. (2008). *A Knowledge-Based information Extraction Prototype for Data-Rich Documents in the Information Technology Domain.* (Master dissertation), National University of Colombia, Bogotá, Colombia.

Wick, M. L., Ross, M. G., & Learned-Miller, E. G. (2007). *Context-sensitive error correction: Using topic models to improve OCR.* Paper presented at the Ninth International Conference on Document Analysis and Recognition (ICDAR).