Arib@QALB-2015 Shared Task: A Hybrid Cascade Model for Arabic Spelling Error Detection and Correction

Nouf AlShenaifi[1], Rehab AlNefie[2], Maha Al-Yahya[3] and Hend Al-Khalifa[4]
[1,2]Computer Science Department and [3,4]Information Technology Department
College of Computer and Information Sciences
King Saud University
Riyadh, Saudi Arabia
{[1]noalshenaifi|[3]malyahya|[4]hendk}@ksu.edu.sa,
{[2]rehhb@hotmail.com}

Abstract

In this paper we present the Arib system for Arabic spelling error detection and correction as part of the second Shared Task on Automatic Arabic Error Correction. Our system contains many components that address various types of spelling error and applies a combination of approaches including rule based, statistical based, and lexicon based in a cascade fashion. We also employed two core models, namely a probabilistic-based model and a distance-based model. Our results on the development and test set indicate that using the correction components in cascaded way yields the best results. The overall recall of our system is 0.51, with a precision of 0.67 and an F1 score of 0.58.

1 Introduction

In last year's shared task on Automatic Arabic Error Correction of the Arabic NLP Workshop (QALB-2014 shared task), a diverse set of approaches were presented including pipeline, hybrid and cascade. These approaches used different techniques such as supervised learning, rule and/or lexicon based, and statistical language modeling. Furthermore, systems presented used several external resources, namely, Arabic Gigaword, AraComLex dictionary, Arabic Wikipedia and Aljazeera articles, to name but a few.

The QALB-2015 shared task is an extension of the first QALB-2014 shared task [1] that occurred last year. QALB-2014 handled errors in comments written by Arabic native speakers in Aljazeera articles [2]. This year's competition includes two subtasks, and, in addition to Arabic native speakers errors, also includes correction of texts written by new learners of Arabic language [3]. The test written by Arabic native subtask includes Alj-train-2014, Alj-dev-2014, Alj-test-2014 texts from QALB-2014. The L2 subtask includes L2-train-2015 and L2-dev-2015. This data was released for the development of the systems.

To build on the previous efforts, we present in this paper, the design and implementation of the Arib system to address the problem of Arabic spelling errors detection and correction. Hence, the name Arib [أريب] is an Arabic word that means a person who is bright, skilled, intelligent and insightful.

Arib will employ a hybrid cascade model as an approach with distance and probability-based techniques that reuse a large scale dataset complied from different external resources.

This paper is organized as follows: section 2 presents related work, section 3 shows how we compiled the necessary language resources for our system, section 4 highlights the main components of our proposed system, section 5 presents our experiments on the system, section 6 reports the obtained results and section 7 concludes the paper with final remarks and future directions.

2 Related Work

The task of Arabic spelling errors detection and correction generally addresses errors such as edit errors, add, split, merge, punctuation, orthographical, dialectal, and other error types. Depending on the techniques used for the task, systems designed for the error detection and correction task utilize language resources such as textual corpora and dictionaries.

One of the earliest studies on Arabic spelling detection and correction is the work conducted by Al-Fedaghi and Amin [4]. The system built

Proceedings of the Second Workshop on Arabic Natural Language Processing, pages 127–132,
Beijing, China, July 26-31, 2015.

detects all four error types edit, add, split, and merge and employs the technique of reducing the words to their original roots to identify spelling errors. Dictionaries used in this system are arranged according to Arabic word roots. The work presented in [5] describes a system which uses an Arabic morphological analyzer, lexicon, and heuristics to detect five types of errors: reading, hearing, touch-type, morphological errors and editing errors. Another similar system that uses the Arabic Web Dictionary (AWD) is presented in [6]. The system used dictionary lookup, morphological analysis and regular expressions to detect the four error types as well as punctuation errors. Other dictionaries used for the Arabic spelling errors detection and correction task include: Ayaspell [7], and AraComLex [8] [9].

Arabic language corpora have been used for spelling error detection and correction. Using a corpus to support the task by providing a resource for training machine-learning based spell-checking systems. Popular corpora used in Arabic spelling error detection and correction systems include: QALB corpus [10], Muaidi [11], and the Arabic Gigaword. The QALB corpus is a large Arabic corpus of manually corrected sentences, it is considered as a "Spelling-error corpus" for Arabic. Systems which used the corpus for the task of error detection and correction include [12], [13], [14], [7], [15], [8], and [16]. The Muaidi corpus has been used in the work presented in [17]. The corpus is a personally built corpus containing a set of 101,987 word types. The Arabic Gigaword corpus is a large corpus of Arabic text from Arabic news sources, developed by the Linguistic Data Consortium. The work described in [9] uses the Gigaword corpus to support the task of spelling error detection and correction.

Techniques and tools reported in the literature for supporting the Arabic spelling errors detection and correction task include morphological analysis [12] [5] [4] [6] [18] [15] [16], regular expressions [6] [13], heuristics (rules) [5] [14] [7] [15] [8] [16], finite state transducer with edit distance [9] [8], statistical character level transformation [14], N-gram scores [17] [8], conditional random fields [14] [8], and Naïve Base [15].

Similar to systems described in the literature, Arib utilizes language resources such as dictionaries and corpora as well as the application of different techniques to support the task of Arabic spelling error detection and correction.

3 Language Resources

An important component of any spelling errors detection and correction system is the compilation of a large scale dictionary that can be used to cover most Arabic words for the sake of detecting the misspelled word. So in order to build this dictionary we reverse-engineered the QALB corpus by replacing the wrong words from the annotated text with the correct words in the final text. We also used several other corpuses, namely: KSU corpus of classical Arabic [19], Open Source Arabic Corpora (OSAC) [20], Al-Sulaiti Corpus [21], and KACST Arabic Corpus [22]. These corpuses were compiled into one complete corpus, we then used KHAWAS tool (KACST Arabic Corpora Processing Tool) [23] to extract the words with their frequencies. This final step helped in building a huge dictionary that was used later on in our system (See Fig.1).

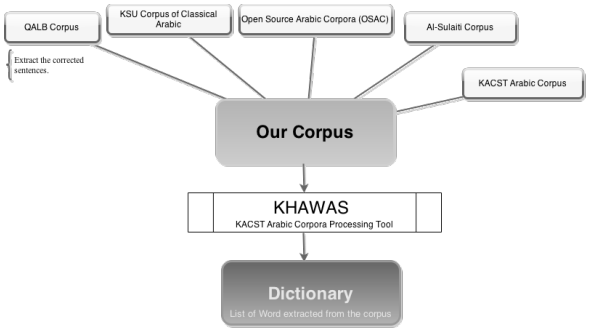

Fig. 1: Dictionary List of Arib.

4 Our Approach

The design of Arib is based on a hybrid cascade approach to spelling errors detection and correction. By cascade we mean that the original Arabic text passes through several components before a final result is returned. Each component participates in identifying spelling errors and recommending a correction. The final result is a compiled collection of all spelling errors identified and the suggested corrections. Our system can cover a range of spelling errors. Errors that are discovered by Arib include: edit, add, split, merge, punctuation, phonological, and common mistakes. The general architecture and major components of Arib system are shown in Fig. 2.

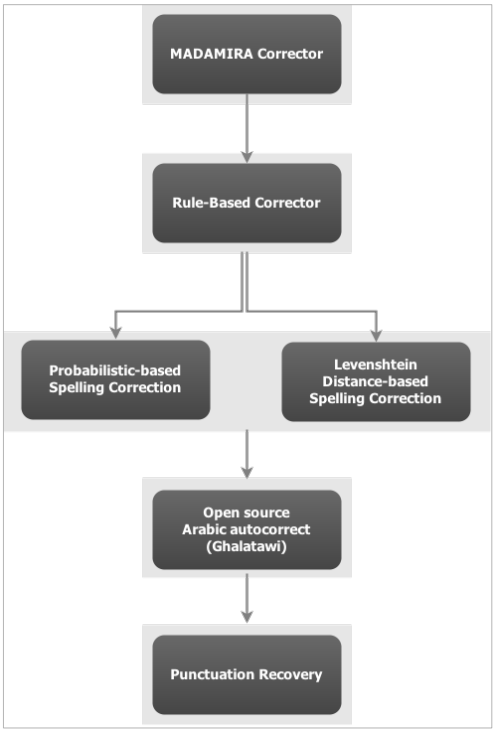

Fig. 2: The general architecture of Arib.

4.1 MADAMIRA Corrector

MADAMIRA [24] is a system developed for morphological analysis and Disambiguation of Arabic text. Since the organizers of the shared task provided the data pre-processed with MADAMIRA, we used the features generated by MADAMIRA to support the spelling error detection and correction. The output of MADAMIRA includes an analysis and correction of the spelling mistakes in the word (Alf)(أ) and terminal (Yaa)(ى). Spelling errors of this type can easily and accurately be detected and corrected using this component.

4.2 Rule-Based Corrector

In this component knowledge of common spelling error patterns are represented as rules that can be applied to provide a correction. All rules are applied to the misspelled word to generate possible corrections. These rules were created through analysis of samples of the QALB Shared Task Dataset and from Arabic language expert who summarized common misspellings of Arabic new learners.

Examples of the extracted rules:

- Replace the English punctuation marks by the Arabic ones (e.g.: replace ‘,’ by ‘،’).
- All numbers are separated from words.
- Fix the Speech effects characters.
- Remove extra characters by eliminating a sequence of three or more of the same characters. (e.g.: replace ‘آمييييييين’ (Āmyyyyyyyn) by ‘آمين’ (Āmyn)).
- Insert a space after all words end by a Ta-Marbouta characters (ة)(p) if it is attached to the following word.
- Insert a space after “ElY, ALY” (على، إلى) (On, For) preposition if it is attached to the following word.
- Merge the lone occurrences of the conjunction “W, FA” (and) (و، ف) with the following word.

4.3 Probabilistic-Based Spelling Correction

This component scans the text for spelling errors using Bayes probability theory, and is based on the algorithm by Peter Norvig for spell checking [25], [26]. It is classified as a probabilistic technique, thus it computes the probability that a given word is the correction for a misspelled work. This component uses our customized dictionary, with word frequencies extracted from KHAWAS to enumerate all possible corrections for the misspelled word. In order to find a correction of misspelled word from all possible corrections we chose the candidate word with the highest probability. For example, the misspelled non-word "تزاب" “tzAb” could be corrected to "تراب" “trAb” (Soil) or "تراث" “trAv” (Heritage), in this component we suggest the correction based on the probabilities.

4.4 Levenshtein-Distance-based Spelling Correction

This component implements the Symmetric Delete Spelling Correction (FAROO) algorithm, a robust algorithm for error detection and correction based on the edit distance using (Damerau-Levenshtein) distance measure [27]. A dictionary entry is selected to be the correction based on its edit distance to the misspelled word. The algorithm works by generating words with an edit distance of <=2 from each dictionary word, and adds them both to the dictionary. Words are generated with an edit distance of <=2 from the input words, and they are searched in the dictionary.

4.5 Open Source Arabic autocorrect (Ghaltawi)

Ghalatawi [28] is an open source Arabic spelling errors detection and correction system available online [28]. The system discovers common spelling errors and uses a dictionary lookup and regular expressions. It is written in Python and has been integrated as a cascade within our development.

4.6 Puctuation Recovery

This component runs a set of rules against the input to determine the absence of periods, semicolon and commas in a given Arabic text. Rules on punctuation are extracted from Arabic language resources and modeled within this component. Previous works mentioned that it is always better to keep the existing punctuation marks in the text [15], so we keep the current punctuation marks (period, comma, question mark, exclamation mark, colon, semicolon, parentheses, and quotation mark) and attempt only to insert the missing marks. The output of this component is the final output of the system.

5 System Experiments

As we previously mentioned, Arib consists of several components designed to tackle different types of errors. For the submissions to the second shared task, we submitted three versions of the system. We refer to these as Arib-1, Arib-2, and Arib-3.

Table.1 shows the component of our system and which components are incorporated in each version.

Component	**System Run**		
	Arib-1	**Arib-2**	**Arib-3**
MADAMIRA	•	•	•
Rule-Based	•	•	•
Probabilistic		•	•
Distance	•		•
Ghaltawi	•	•	•
Punctuation	•	•	•

Table.1: The three output runs of Arib.

6 Results and Discussion

With a view to evaluate the performance of our system, we used the M2 Scorer [29], the official scorer of the shared task.

Table.2 reports the performance results of Arib on the development and test set Alj-dev-2014, Alj-test-2014, L2-dev-2015, and L2-test-2014.

Table.3 reports the performance results of Arib as each system component is added.

	Precision	**Recall**	**F-measure**
Arib Result	0.6658	0.5108	0.5781

Table.2: Performance results of Arib.

Component	**System Performance**		
	Precision	**Recall**	**F-measure**
MADAMIRA	0.6615	0.3671	0.4722
+Rule-Based	0.6719	0.4212	0.5178
+Probabilistic	0.6521	0.4471	0.5305
+Distance	0.6650	0.5092	0.5768
+Ghaltawi	0.6658	0.5108	0.5781

Table.3: Performance results of Arib with the respect of each system component.

Table.4 reports the performance results of Arib on the test set for the 2nd QALB Shared Task Alj-test-2015 and L2-test-2015.

System	**Test Set**	**Precision**	**Recall**	**F-measure**
Arib-1	Alj-test-2015	64.50	56.50	60.23
Arib-2		67.56	51.61	58.51
Arib-2	L2-test-2015	50.08	22.30	30.86
Arib-3		48.79	24.57	32.68

Table.4: Performance results of Arib on Alj-test-2015 and L2-test-2015

Results from the evaluation show that the Arib performed well as each component is added to the system.

7 Conclusion and Further Research

In this paper, we described a hybrid cascade approach for Arabic Spelling detection and correction system for participation in the second shared task on Automatic Arabic Error Correction. Our approach combines rule-based linguistic techniques with probabilistic-based and Distance-based Spelling Correction techniques. We experiment with our system using different configurations of the developed components. Results of the experiments show encouraging results.

Future work involves further enhancements to the system including developing more intelligent techniques to correct split and merge errors. Moreover, use more advanced techniques for the sake of punctuation corrector including machine learning techniques and semantic text analysis technology.

Reference

[1] B. Mohit, A. Rozovskaya, N. Habash, W. Zaghouani, and O. Obeid., "The First QALB Shared Task on Automatic Text Correction for Arabic," in Proceedings of EMNLP Workshop on Arabic Natural Language Processing, Doha, Qatar, 2014.

[2] W. Zaghouani, B. Mohit, N. Habash, O. Obeid, N. Tomeh, A. Rozovskaya, N. Farra and S. Alkuhlani, and K. Oflazer., "Large Scale Arabic Error Annotation: Guidelines and Framework," in Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), 2014.

[3] W. Zaghouani, N. Habash, H. Bouamor, A. Rozovskaya, B. Mohit, A. Heider, and K. Oflazer., "Correction Annotation for Non-Native Arabic Texts: Guidelines and Corpus," in Proceedings of the Ninth Linguistic Annotation Workshop, Association for Computational Linguistics, 2015, pp. 129–139.

[4] Al-Fedaghi, S. and Amin, A., "Automatic correction of spelling errors in Arabic," J-Univ. KUWAIT Sci., vol. 19, no. 2, p. 175, 1992.

[5] K. Shaalan, Amin Allam, and Abdullah Gomah, "Towards automatic spell checking for Arabic," 2003.

[6] Rachidi T., M. Bouzoubaa, L. ElMortaji, B. Boussouab, and A. Bensaid, "Arabic user search Query correction and expansion," in Proc. of COPSTIC'03, Rabat December 11--13, 2003.

[7] Djamel Mostefa, Omar Asbayou, and Ramzi Abbes, "TECHLIMED system description for the Shared Task on Automatic Arabic Error Correction," in EMNLP, Doha, Qatar.

[8] Mona Diab, Mohammed Attia, and Mohamed Al-Badrashiny, "GWU-HASP: Hybrid Arabic Spelling and Punctuation Corrector," in EMNLP, Doha, Qatar, 2014.

[9] Attia, Mohammed, Pavel Pecina, and Younes Samih, "Improved Spelling Error Detection and Correction for Arabic," in COLING, 2012.

[10] B. Mohit, "QALB: Qatar Arabic language bank," Qatar Found. Annu. Res. Forum Proc., p. ICTP 032, Nov. 2013.

[11] Muaidi H, "Extraction Of Arabic Word Roots: An Ap- proach Based on Computational Model and Multi-ackpropagation Neural Networks.," PhD Thesis, De Mont- fort University, UK, 2008.

[12] Y. Hassan, M. Aly, and A. Atiya, "Arabic Spelling Correction using Supervised Learning," ArXiv14098309 Cs, Sep. 2014.

[13] Taha Zerrouki, Khaled Alhowaity, and Amar Balla, "Auto-correction of arabic common errors for large text corpus," in EMNLP, Doha, Qatar, 2014.

[14] H. Mubarak and K. Darwish, "Automatic Correction of Arabic Text: a Cascaded Approach," ANLP 2014, p. 132, 2014.

[15] A. R. N. Habash and R. E. N. F. W. Salloum, "The Columbia System in the QALB-2014 Shared Task on Arabic Error Correction," ANLP 2014, p. 160, 2014.

[16] Michael Nawar and Moheb Ragheb, "Fast and Robust Arabic Error Correction System," in EMNLP, Doha, Qatar, 2014.

[17] H. Muaidi and R. Al-Tarawneh, "Towards Arabic Spell-Checker Based on N-Grams Scores," Int. J. Comput. Appl., vol. 53, no. 3, pp. 12–16, Sep. 2012.

[18] B. Haddad and M. Yaseen, "Detection and Correction of Non-Words in Arabic: A Hybrid Approach," Int. J. Comput. Process. Lang., vol. 20, no. 04, pp. 237–257, Dec. 2007.

[19] Alrabiah, M., Al-Salman A. and Atwell, E., "The design and construction of the 50 million words KSUCCA King Saud University Corpus of Classical Arabic," in Proceedings of the Second Workshop on Arabic Corpus Linguistics (WACL-2), Lancaster University, UK, 2013.

[20] Saad, M. and Ashour, W., "OSAC: Open Source Arabic Corpora," in Proceedings of the 6th International Symposium on Electrical and Electronics Engineering and Computer Science (EEECS'10), European University of Lefke, Cyprus, pp. 118-123, 2010.

[21] Al-Sulaiti, Latifa; Atwell, Eric. “The design of a corpus of contemporary Arabic,” International Journal of Corpus Linguistics, vol. 11, pp. 135-171, 2006.

[22] Al-Thubaity, A., “A 700M+ Arabic corpus: KACST Arabic corpus design and construction,” Language Resources and Evaluation, pp. 1-31, 2014.

[23] “Ghawwas: An open source system for Arabic corpora processing.”[Online]. Available: http://sourceforge.net/projects/ghawwasv4/. [Accessed: 1-June-2015].

[24] A. Pasha, M. Al-Badrashiny, M. T. Diab, A. E. Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, and R. Roth, “MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic,” in Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014, 2014, pp. 1094–1101.

[25] “How to Write a Spelling Corrector.” [Online]. Available: http://norvig.com/spell-correct.html. [Accessed: 28-May-2015].

[26] T. Segaran and J. Hammerbacher, Beautiful Data: The Stories Behind Elegant Data Solutions. O’Reilly Media, Inc., 2009.

[27] “1000x Faster Spelling Correction algorithm | FAROO Blog.” [Online]. Available: http://blog.faroo.com/2012/06/07/improved-edit-distance-based-spelling-correction/. [Accessed: 28-May-2015].

[28] “غلطاوي:Ghalatawi الـعربـي الـتـلقـائي الـتـصـحـيـح.” [Online]. Available: http://ghalatawi.sourceforge.net/. [Accessed: 28-May-2015].

[29] D. Daniel, and H. Ng., “Better evaluation for grammatical error correction,” in Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2012.

[30] A. Rozovskaya, H. Bouamor N. Habash, W. Zaghouani, O. Obeid, and B. Mohit., “The Second QALB Shared Task on Automatic Text Correction for Arabic,” in Proceedings of ACL Workshop on Arabic Natural Language Processing, Beijing, China, 2015