SAHSOH@QALB-2015 Shared Task: A Rule-Based Correction Method of Common Arabic Native and Non-Native Speakers' Errors

Wajdi Zaghouani
Carnegie Mellon University,
Doha, Qatar
wajdiz@cmu.edu

Taha Zerrouki
Bouira University,
Bouira, Algeria
t_zerrouki@esi.dz

Amar Balla
The National Computer
Science Engineering School
(ESI), Algiers, Algeria
a_balla@esi.dz

Abstract

This paper describes our participation in the QALB-2015 Automatic Correction of Arabic Text shared task. We employed various tools and external resources to build a rule based correction method. Hand written linguistic rules were added by using existing lexicons and regular expressions. We handled specific errors with dedicated rules reserved for non-native speakers. The system is simple as it does not employ any sophisticated machine learning methods and it does not correct punctuation errors. The system achieved results comparable to other approaches when the punctuation errors are ignored with an F1 of 66.9% for native speakers' data and an F1 of 31.72% for the non-native speakers' data.

1 Introduction

The Automatic Error Correction (AEC) is an interesting and challenging problem in Natural Language Processing. The existing methods that attempt to solve this problem are generally based on deep linguistic and statistical analysis. AEC tools can assist in solving multiple natural language processing (NLP) tasks like Machine Translation or Natural Language Generation. However, the main application of AEC is the building of automated spell checkers to be used as writing assistant tools (e.g. word-processing) or even for applications such as Mobile auto-completion and auto correction programs, post-processing optical character recognition tools or with the correction of large content site such as Wikipedia. Conventional spelling correction tools detect typing errors simply by comparing each token of a text against a dictionary of words that are known to be correctly spelled. Any token that matches an element of the dictionary, possibly after some minimal morphological analysis, is deemed to be correctly spelled; any token that matches no element is flagged as a possible error, with near-matches displayed as suggested corrections (Hirst 2005).

In this paper we describe our participation in the QALB-2015 shared task (Rozovskaya 2015) which is an extension of the first QALB shared task (Mohit et al. 2014) that took place last year. The QALB-2014 shared task was reserved to errors in comments written to Aljazeera articles by native Arabic speakers (Zaghouani et al. 2014; Obeid et al. 2013). The 2015 competition includes two tracks. The first track is dedicated to errors produced by native speakers and the second track includes correction of texts written by learners of Arabic as a foreign language (L2) (Zaghouani et al. 2015). The native track includes Alj-train-2014, Alj-dev-2014, Alj-test-2014 texts from QALB-2014. The L2 track includes L2-train-2015 and L2-dev-2015. This data was released for the development of the systems. The systems were scored on blind test sets Alj-test-2015 and L2-test-2015.

Our pipeline approach is based on a combination of pre-existing tools, hand written contextual rules and lexicons. Detecting and correcting such complex errors within the scope of a rule based approach require specific rules to be written in order to correctly analyze the dependencies between words in a given sentence. The remainder of this paper is organized as follows: Section 2 describes the related works. Section 3 presents our approach including the tools and resources used and finally in Section 4 we report the results obtained on the Development set.

Proceedings of the Second Workshop on Arabic Natural Language Processing, pages 155–160,
Beijing, China, July 26-31, 2015.

2 Related Works

The task of automatic error correction has been explored widely by many researchers in the past years especially for the English language. Many approaches have been used to build systems (hybrid, rule base, supervised and unsupervised machine learning…). These systems used various NLP tools and resources including pre-existing lexicons, morphological analyzers and Part of Speech Taggers. We cite for the English language early works done by (Church and Gale, 1991; Kukich, 1992; Golding, 1995; Golding and Roth, 1996). Later on we find (Brill and Moore, 2000; Fossati and Di Eugenio, 2007) and more recently Han and Baldwin, 2011; Dahlmeier and Ng 2012; Wu et al., 2013). For Arabic, this problem has been investigated in a couple of papers as in Shaalan et al. (2003) who presented his work on the specification and classification of spelling errors in Arabic. Later on, Haddad and Yaseen (2007) built a hybrid approach that used rules and some morphological features to correct non-words using contextual clues and Hassan et al. (2008) presented a language independent text correction method using Finite State Automata. More recently, Alkanhal et al. (2012) wrote a paper about a stochastic approach used for word spelling correction and Attia et al. (2012) created a dictionary of 9 million entries fully inflected Arabic words using a morphological transducer. Later on, they used a dictionary to build an error model by analyzing the various error types in the data. Moreover, Shaalan et al. (2012) created a model using unigrams to correct Arabic spelling errors and recently, (Pasha et al., 2014) created MADAMIRA, a morphological analyzer and a disambiguation tool for Arabic. Finally, Alfaifi and Atwell (2012) created a native and non-native Arabic learner's corpus and an error coding correction taxonomy made available for research purpose.

3 Our Approach

Our correction approach watches out for certain predefined "errors" as the user types, replacing them with a suggested "correction" depending on the corpus type L1 or L2. Therefore an error analysis was performed on the provided data set to find the most frequent error types per data set. We also located some external freely available resources listed in (Zaghouani 2014) such as Alfaifi L1 and L2 corpus (Alfaifi and Atwell 2013), The JRC-Names names (Steinberger et al. 2011) and the Attia list (Attia 2012).

3.1 Corpus Error Analysis

In order to better write our correction rules and to better understand the nature of errors in the L1 and L2 data, we performed a manual inspection on a sample taken from the Dev Sets of the shared task and we obtained the errors distribution shown in Table 1. While the errors committed by L1 speakers are mostly spelling errors such as the Hamza and Ta-Marbuta confusion, L2 speakers tend more to have difficulties with the following issues: the definiteness structure, the words agreement, the preposition usage and the correct word choice in the sentence. We used this analysis to optimize our rules for each corpus.

Rank	Native L1	Non-Native L2
#1	Hamza	Definiteness
#2	Ta-Marbuta / Ha Alif-Maqsura/Ya	Agreement
#3	Case Endings	Prrnaleposition
#4	Verbal Inflection	Hamza
#5	Conjunctions	Word Choice

Table 1: Most frequent errors observed in the Dev sets of the L1 and L2 Corpus. The errors are sorted from the most frequent to the least frequent

In Arabic, spelling confusion in Hamza forms is frequently found, e.g. the word إستعمال IstEmAl[1] "usage" must be written by a simple Alef ا, not Alef with Hamza below إ . This error can be classified as a kind of errors and not a simple error in a word as reported by (Shaalan, 2003, Habash, 2011). While typical common errors based on wrong letter spelling such as the confusion in the form of Hamza همزة, Daad ضادand Za ظاء, and the omission dots with Yeh ياء and Teh تاء are generally relatively easy to handle, the task is more challenging for grammatical and semantic errors. Previously, we created an Arabic auto correction tool to correct common mistakes in Wikipedia articles. The idea is to create a script that detects common spelling errors using a set of regular expressions and a word replacement list[2].

In a similar way, the system we are presenting in this paper is based primarily on:

[1] Buckwalter transliteration

[2] The script is named AkhtaBot, which is applied to Arabic wikipedia, the Akhtabot is available on http://ar.wikipedia.org/wiki/مستخدم:AkhtaBot

- Regular expressions used to identify errors and give a replacement.

- Replacement list that contains the misspelled word and the exact correction needed for each particular case. Furthermore, we used the following combination of tools and resources:

- **Arabic word list for spell checking**: This list contains 9 million Arabic words from AraComLex, an open-source finite state transducer (Attia 2012). The list[3] was validated against Microsoft Word spell checker tool. This list was used to check and replace wrongly spelled words.
- **JRC-Names**[4]: a list of 1.18 million person and 6,700 organization names (Steinberger et al. 2011). We used the list to correct and replace wrongly spelled named entities in the data set.
- **Alfaifi L1 and L2 corpus:** Used to observe the errors in context and to study the patterns of spelling errors made by native and non-native speakers. The corpus was created by (Alfaifi and Atwell 2013) and freely available[5].
- **A Python script** to generate the errors.
- **Hunspell** spellchecker program[6] combined with **Ayaspell**[7] dictionary (Hadjir 2009, Zerrouki, 2013).
- **Ghalatawi**[8] : Our spelling correction tool
- A **task dedicated script** to select the best suggestion from Hunspell correction suggestions to generate customized autocorrected list customized for each data set.

3.1 Regular Expressions

We use regular expression patterns to detect errors by using the Arabic verb forms (الأوزان AlOwzAn) and affixes. For example we can detect very common Hamza spelling errors with the Arabic verbs form VII which expresses a submission to an action or an effect as in the case of an animate being, it could mean an involuntary submission. This form reflects the meaning on two levels: reflexive (to let oneself be put through) and an agentless passive (non-reciprocal of form I). Using such a rule with a word such as INFIAL it انفعال should be written with Hamza Wasl, as the form إنفعال InfEAl is wrongly spelled. Moreover, we represent all forms with all possible affixes as shown in Table 2 and Table 3

Suffixes	Form	Prefixes
...ين، ات، ي، ان، ه، ها، هما، ك، كما kmA, k, hmA, h, An, y, At, yn…	انفعال	...ب، ال، و، ف f, w, Al, b

Table 2: Infi'aal verb forms with affixes

Example of rules for انفعال
ur'\b(إن|ان)(ال|)(ف|و)(\w)ا(\w\w)(\w)(ة|ات|تين|ين)(إي|)\b'
ur'\b(إن|ان)(لل|)(ف|و)(\w)ا(\w\w)(\w)(ة|ات|تين|ين)(إي|)\b'
ur'\b(إن|ان)(ف|و)(\w)ا(\w\w)(\w)(إي|أ|آ|ا)\b'

Table 3: Sample rules for the Infi'al verb form

Furthermore, we have modeled the following spelling errors cases using regular expressions (c.f Table 4):
(1) words with the verb forms infi'al and ifti'al انفعال وافتعال ; (2) words with the letter Alif Maqsura followed by Hamza, for example سىئ} will be corrected as سيء sy'. (3) words with Teh Marbuta misplaced or incorrectly merged, like in مدرسةالعلم mdrspAlElm to be corrected to مدرسة العلم mdrsp AlElm "school of knowledge".

Regular expressions	Replacements
# removing kashida (Tatweel)	
ur'([\u0621-\u063F\u0641-\u064A])\u0640+([\u0621-\u063F\u0641-\u064A])'	ur'\1\2'
# rules for انفعال	
ur'\b(إن|ان)(ال|)(ف|و)(\w)ا(\w\w)(\w)(إي|)(ين|ات|ة|تين)\b'	ur'\1\2\3\4ان\5\6\7'

Table 4: Sample rules expressed by regular expressions.

3.2 Wordlist

Many common mistakes cannot be corrected using regular expressions only, such as confusion between the letter Dhad and the letter Za, and omitted dots on letter Teh and letter Yeh, as in the المكتبه* Almktbh "the library" and * فى fY "in", So we resort to build a list of common misspelled words.

To build our word list, we used the existing lexicons of correctly spelled 9M words by Attia

[3] The list available at: http: //sourceforge.net/projects/arabic-wordlist/
[4] The list is available at : https://ec.europa.eu/jrc/en/language-technologies/jrc-names
[5] The corpus is available at http://www.arabiclearnercorpus.com
[6] Available at http://hunspell.sourceforge.net/
[7] Available at http://ayaspell.sourceforge.net.
[8] The Ghalatawi autocorrect program is available as an open source program at http://ghalatawi.sourceforge.net

(2012) and the JRC-Names named entities corpus (Steinberger et al. 2011) by generating errors for common letters errors, then filtering the results to obtain an autocorrected words list with no ambiguity. In order to build the list, first, we take a correct word list than we select candidate words from words starting with Hamza Qat' or Wasl , words ending by Yeh or Teh marbuta or Words containing the letter Dhad or Zah. Than we generate errors on words by replacing candidate letters by errors on purpose. Finally we check the spelling and eliminate the corrected words, because some modified words can be correct, for example, if we take the word ضلّ Dla , then modify it to ظل Zl, the modified word exists in the dictionary, then we exclude it from the auto corrected wordlist, and we keep only misspelled modified words as the examples in the word إسلام IslAm "islam", it can be written as اسلام AslAm "islam" by mistake since it has the same phonological construction.

3.3 Customized Wordlist for L1 and L2 Texts

We generated a case specific auto correction list for each corpus (L1 or L2). The following algorithm is applied to generate customized list from each corpus:

(1) Extract misspelled words from dataset by using Hunspell spellchecker. **(2)** Generate suggestions given by Hunspell. **(3)** Observe the suggestions to choose the best one in hypothesis that words have common errors on letters according to modified letters. **(4)** Exclude ambiguous cases. **(5)** The automatically generated word list is used to autocorrect the dataset instead of the default word list.

4 Evaluation

In order to evaluate the performance of our system, we used the data set provided in the shared task test (Alj-dev-2014 and L2-dev-2015). For this evaluation we have used two autocorrected word lists:
- A generic word list generated from Attia wordlist and the JRC corpus, this wordlist is used for general correction purposes.
- A customized wordlist based on each dataset L2-dev-2015, L2-test-2015, Alj-dev-2014 and Alj-test-2015 by generating a special word list according to each data set, in order to improve the results and avoid unnecessary replacement. The customized auto correction word list is built in the same way as the generic one, by replacing the source dictionary by misspelled words from QALB corpus (Zaghouani, 2014). We submitted only one run for each corpus type and the official results obtained on the Development sets and the Test sets are shown in Table 5 by using the M2 scorer (Dahlmeier et al 2012):

Data set	Precision	Recall	F1
Alj-dev-2014	71.40	32.10	44.30
Alj-test-2014	82.63	41.89	55.59
Alj-test-2015	81.88	40.24	53.97
L2-dev-2015	60.30	11.30	19.00
L2-test-2015	59.75	15.90	25.12

Table 5: Results on the Dev and Test sets

The relatively low results obtained were expected since we decided to ignore the punctuation errors and therefore our system is penalized by this decision. We estimate that punctuation errors represent more than 38% of the errors in the QALB data sets (L1 and L2). When the punctuation errors were removed from the evaluation, we noticed a significant improvement of the recall and the F1 score for L1 (+13 points) and for L2 (+6.6 points) as seen in table 6.

Data set	Precision	Recall	F1
Alj-test-2015	83.85	55.65	**66.90**
L2-test-2015	58.95	21.70	31.72

Table 6: Official Results on the Dev and Test sets with with punctuation errors ignored

5 Error Analysis

Our system failed to find the appropriate correction in many cases due to the limitations of the rule based systems in general. In this section, we will highlight some of the main errors not corrected by our system for both data sets. We will not discuss punctuation related errors as they are not handled by our system.

5.1 L1 Errors

- **Split and Merge errors:** Such as والجزيرةنت wAljzyrpnt "AljazeeraNet" it is not obvious to detect where the words should be split as in نت والجزيرة wAljzyrp nt "Aljazeera Net". Other words that should be merged are hard to detect as both words produced can be valid entries such as طيني الفلس Alfls Tyny that should be corrected to الفلسطيني AlflsTyny "the Palestinian" but both

words wrongly produced are acceptable in this case.

- **Wrong Hamza spelling:** Such as أن On "indeed" and إن In "indeed". For these particular examples advanced rules may be required.
- **Ta-Marbuta / Ha errors:** These errors are practically frequent for the L1 corpus and they are not always corrected by our system in the cases of named entities.
- **Keyboard Typos:** Keyboard errors are very frequent and our system did not detect most of them due to the complexity of the issue, since the typo word could be correctly spelled like misspelling الباب AlbAb "the door" for البار AlbAr "The bar" .

5.2 L2 Errors

Many L2 detection errors are very similar to the L1 errors listed in the previous section, but some errors are mostly found in L2 texts such as the following:

- **Definiteness:** correcting definite errors with a rule based system could be very challenging without access to a parser. For instance errors such as missing definite article in **منورة** المدينة Almdynp mnwrp "The Madinah Munawwarah" are very frequent in L2 texts and our system failed to detect them most of the time since the word missing the definite article are correct as standalone words.
- **Gender and number agreement:**

The Gender-number agreement is another frequent error type where our system failed frequently to correct it such as in أخلاق سكانه جيد Oxlaq skAnh jyd "morals of its inhabitants is good" with the wrong gender in the word جيد jyd "good" that should be corrected to جيدة jydp instead as it is related the feminine noun أخلاق Oxlaq "morals".

- **Prepositions:** Non-native speakers are frequently confused in the preposition usage in Arabic. An advanced language level is usually required to master this. A frequent confusion in the usage of the wrong preposition في fy "in" in the following example. ذهبت في البيت *hbt fy Albyt "I went in the house" that should be corrected by our system to ذهبت إلى البيت . *hbt IlY Albyt "I went to the house"
- **Wrong Word choice:** L2 speakers have some difficulties with words that may be homophones but spelled in a different way such as in البقار يستريحون AlbqAr ystryHwn "the cow boys are resting" and it is obvious here that it is meant to be الأبقار يستريحون AlObqAr ystryHwn "the cows are resting". Again these cases show another limitation of rule based systems to detect correctly spelled wrong word choices.

6 Conclusion and Discussion

We presented a pipeline rule based approach for correcting Arabic text optimized for two native and non-native text types. We focused mainly on the most common errors made by native and non-native speakers such as the Hamza errors, The Ta-Marbuta and letter Ya. We also used complex regular expressions to correct splitting and merging errors. We also, used lexicons such as the Attia word list and the JRC-names to boost the results of our system. The correction of more complex errors was also tested such as the correction of phonological errors caused by a confusion and similarity of the words. For non-native speakers, we detected and corrected some of the errors related to the misuse of gender and number agreement and also for the wrong usage of the definite article.

The results obtained showed that our systems performs much better with native speakers texts, this is mainly due to the complex nature of some spelling errors of L2 learners. In the future, we plan to handle more complex errors for both native and non-native texts such as grammatical and case ending errors and also wrong word choice errors. We are also planning to integrate the MADAMIRA morphological analyzer in a post processing step to increase our recall.

7 Acknowledgements

This publication was made possible by grants NPRP-4-1058-1-168 from the Qatar National Research Fund (a member of the Qatar Foundation).

References

Alfaifi Abdullah and Atwell Eric. 2012. Arabic Learner Corpora (ALC): a taxonomy of coding errors. In Proceedings of the 8th International Computing Conference in Arabic (ICCA 2012)

Alfaifi, Abdullah and Atwell, Eric. 2013. Arabic Learner Corpus v1: A New Resource for Arabic Language Research. In proceedings of *the Second Workshop on Arabic Corpus Linguistics (WACL-2)*. Lancaster University, UK.

Alkanhal, Mohamed I., Mohamed A. Al-Badrashiny, Mansour M. Alghamdi, and Abdulaziz O. Al-Qabbany. 2012. Automatic Stochastic Arabic Spelling Correction With Emphasis on Space Insertions and Deletions. IEEE Transactions on Audio, Speech, and Language Processing, Vol. 20, No. 7, September 2012.

Attia, Mohammed, Pavel Pecina, Younes Samih, Khaled Shaalan, Josef van Genabith. 2012. Improved Spelling Error Detection and Correction for Arabic. COLING 2012, Bumbai, India.

Dahlmeier, Daniel and Ng, Hwee Tou. 2012. Better evaluation for grammatical error correction. In Proceedings of NAAC-HLT, Montreal, Canada.

Deorowicz S,, Marcin G. Ciura. 2005. Correcting Spelling Errors By Modeling Their Causes. Int. J. Appl. Math. Comput. Sci., 2005, Vol. 15, No. 2, 275–285

Golding and Roth. 1999. A Winnow based approach to Context-Sensitive Spelling Correction. In Machine Learning - Special issue on natural language learning, Volume 34 Issue 1-3, Feb. 1999.

Habash Nizar. 2010. Introduction to Arabic natural language processing. Synthesis Lectures on Human Language Technologies 3.1 (2010): 1-187

Habash Nizar, Ryan M. Roth. 2011. Using Deep Morphology to Improve Automatic Error Detection in Arabic Handwriting Recognition, ACL, page 875-884. The Association for Computer Linguistics, (2011)

Hadjir I .2009 .Towards an open source Arabic spell checker. MA thesis in Natural language processing, scientific and technique research center to Arabic language development.

Hammad M and Mohamed Alhawari. 2010. In Recent improvement of arabic language search, Google Arabia Blog, Google company, 2010 http://google-arabia.blogspot.com/.

Hassan Ahmed, Noeman Sara and Hassan Hany. 2008. Language Independent Text Correction using Finite State Automata. IJCNLP. Hyderabad, India.

Hirst Graeme and Alexander Budanitsky. 2005. Correcting real-word spelling errors by restoring lexical cohesion, Natural Language Engineering 11 (1): 87–111, 2005 Cambridge University Press

Mohit Behrang, Alla Rozovskaya, Wajdi Zaghouani, Ossama Obeid, and Nizar Habash. 2014. The First shared Task on Automatic Text Correction for Arabic. In Proceedings of EMNLP Workshop on Arabic Natural Language Processing, Doha, Qatar.

Obeid Ossama, Wajdi. Zaghouani, Behrang. Mohit, Nizar Habash, Kemal Oflazer and Nadi Tomeh. 2013. A Web-based Annotation Framework For Large-Scale Text Correction. In The Companion Volume of the Proceedings of IJCNLP 2013: System Demonstrations. Asian Federation of Natural Language Processing.

Pasha A., M. Al-Badrashiny, M. Diab, A. El Kholy, R. Eskander, N. Habash, M. Pooleery, O. Rambow, and R. Roth. 2014. 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).

Rozovskaya Alla, Houda Bouamor, Wajdi Zaghouani, Ossama Obeid, and Nizar Habash and Behrang Mohit. 2015. The Second QALB Shared Task on Automatic Text Correction for Arabic. In Proceedings of ACL Workshop on Arabic Natural Language Processing, Beijing, China.

Shaalan, Khaled, Amin Allam and Abdallah Gomah. 2003. Towards automatic spell checking for Arabic. In Proceedings of the Conference on Language Engineering, 2003 - claes.sci.eg

Steinberger, Ralf, Pouliquen, Bruno, Kabadjov, Mijail, Belyaeva, Jenya and van der Goot, Erik. 2011. JRC-NAMES: A Freely Available, Highly Multilingual Named Entity Resource. In Proceedings of the International Conference Recent Advances in Natural Language Processing, Hissar, Bulgaria.

Zaghouani, Wajdi. 2014. Critical survey of the freely available Arabic corpora. In Proceedings of the Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools Workshop, LREC 2014, Reykjavik, Iceland.

Zaghouani Wajdi, Nizar Habash, Houda Bouamor, Alla Rozovskaya, Behrang Mohit, Abeer Heider, and Kemal Oflazer. 2015. Correction annotation for nonnative arabic texts: Guidelines and corpus. In Proceedings of The 9th Linguistic Annotation Workshop, pages 129–139, Denver, Colorado, USA, June. Association for Computational Linguistics.

Zaghouani Wajdi, Behrang Mohit, Nizar Habash, Ossama Obeid, Nadi Tomeh, Alla Rozovskaya, Noura Farra, Sarah Alkuhlani, and Kemal Oflazer. 2014. Large Scale Arabic Error Annotation: Guidelines and Framework. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland.

Zerrouki Taha. 2011. Improving the spell checking dictionary by users feedback. A meeting of experts check the spelling and grammar and composition automation, Higher Institute of Applied Science and Technology of Damascus, the Arab Organization for Education, Science and Culture, Damascus, April 18 to 20, 2011.