Autocorrection Of Arabic Common Errors For Large Text Corpus

QALB-2014 Shared Task

Taha Zerrouki
Bouira University, Bouira, Algeria
The National Computer Science Engineering School (ESI), Algiers, Algeria
t_zerrouki@esi.dz

Khaled Alhawaity
Tabuk University, KSA
al-howity@hotmail.com

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

Abstract

Automatic correction of misspelled words means offering a single proposal to correct a mistake, for example, switching two letters, omitting letter or a key press. In Arabic, there are some typical common errors based on letter errors, such as confusing in the form of Hamza همزة, confusion between Daad ضاد and Za ظاء, and the omission dots with Yeh ياء and Teh تاء .

So we propose in this paper a system description of a mechanism for automatic correction of common errors in Arabic based on rules, by using two methods, a list of words and regular expressions.

Keywords: *AutoCorrect, spell checking, Arabic language processing.*

1 Introduction

Spell check is the most important functions of correct writing, whether manual or assisted by programs, it detects errors and suggests corrections.

Conventional spelling checkers 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).

2 Auto-correction

An auto-correction mechanism watches out for certain predefined "errors" as the user types, replacing them with a "correction" and giving no indication or warning of the change.

Such mechanisms are intended for undoubted typing errors for which only one correction is plausible, such as correcting accomodate* to accommodate (Hirst, 2005).

In Arabic, we found some common errors types, like the confusion in Hamza forms, e.g. the word Isti'maal (إستعمال*) must be written by a simple Alef, not Alef with Hamza below. This error can be classed as a kind of errors and not a simple error in a word (Shaalan, 2003, Habash, 2011).

Spellchecking and autocorrection are widely applicable for tasks such as:

- word- processing
- Post-processing Optical Character Recognition.
- Correction of large content site like Wikipedia.
- Correction of corpora.
- Search queries
- Mobile auto-completion and autocorrection programs.

3 Related works

Current works on autocorrection in Arabic are limited; there are some works on improving spell checking to select one plausible correction especially for correcting large texts like corpus. In English, Deorowicz (2005) had worked on correcting spelling errors by modeling their causes, he propose to classify mis-

Proceedings of the EMNLP 2014 Workshop on Arabic Natural Langauge Processing (ANLP), pages 127–131,
October 25, 2014, Doha, Qatar.

takes causes in order to improve replacement suggestion.

In Arabic, Microsoft office provides an autocorrect word list of common errors, which is limited and not studied.

Google search engine had improved its search algorithm for Arabic query by using some rules on letters which can be mistaken, for better words split based on letters properties, for example if we type [رااائعةالجمال]*, the engine can give results for "Rae'at alJamaal" [رااائعةالجمال]*and [رائعة الجمال]. , some other example: "Altarbia wa alta'lim", "Google", "Jaridat alahraam", [، [قووقل]*، جريدةالاهرام]*، [[التربيةوالتعليم]*.

Google Arabia says in its blog, that "this improvement which looked very simple, enhance search in Arabic language by 10% which is in real an impressive change" (Hammad, 2010).

4 Our approach

We have launched our first project about autocorrection for a special objective to enhance Wikipedia article spell checking. Wikipedia is a large text database written by thousands of persons with different language skill levels and with multiple origins, which make a lot of mistakes. The idea is to provide an automatic script which can detect common errors by using regular expressions and a word replacement list[1].

This objective can be extended to answer other needs for users in office, chat, tweets, etc.

The idea is to use a non-ambiguous regular expressions or word list, to prevent common errors, while writing or as an automated script for large texts data.

As we say above, our method is based on:

- Regular expressions which can be used to identify errors and give one replacement.

- Replacement list which contains the misspelled word, and the exact correction for this case, this way is used for cases which can't be modeled as regular expression.

4.1 Regular Expressions

We use regular expression pattern to detect.errors in words by using word weight (Wazn) and affixes. For example we can detect that words with the weight INFI'AL انفعال must be written by Hamza Wasl, and we consider the form إنفعال* as wrong. Then, we represent all forms of this weight with all possible affixes.

Suffixes	Weight	prefixes
ين، ات، ي، ان، ه، ...ها، هما، ك، كما	انفعال	...ب، ال، و، ف

Table 1 Infi'aal wheight with its affixation

rules for انفعال
ur'\b(ان|ال|ف|ب|ك)(ة|تين|ات|ين)(\w)(\w\w)(و|ف|ي|)\b'
ur'\b(ان|لل|ال|ف|و)(ة|تين|ات|ين)(\w)(\w\w)(و|ف|ي|)\b'
ur'\b(ان|ال|ل|ب|ك)(\w)(\w\w)(و|ف|ي|)(ه|ها|كما|هما|هم|كم|هن|كن|نا|ه|ك|ها|تهما|تكما|تهم|تكم|تهن|تكن|تنا|ته|تها|تك|تكما|تهما|تك|ما|تكما|تهم|تكم|تهن|تكن|تنا|ته|تها|تك)\b'
ur'\b(ان|ال|ف|و)(ة|تين|ات|ين)(\w)(\w\w)(و|ف|ي|)\b'
ur'\b(ان|ال|ف|و)(ة|تين|ات|ين)(\w)(\w\w)(و|ف|ي|)\b'

Table 2 Rules for the Infi'al weight in all forms

By regular expressions we have modeled the following cases (cf.):

- words with weights (infi'al and ifti'al انفعال و افتعال)
- Words with Alef Maksura followed by Hamza, for example سئ will be corrected ad سيء.
- words with Teh Marbuta misplaced, like مدرسةالعلم to be corrected to مدرسة العلم.

Regular expression	**replacement**
# removing kashida (Tatweel)	
ur'([\u0621-\u063F\u0641-\u064A])\u0640+([\u0621-\u063F\u0641-\u064A])'	ur'\1\2'
# rules for انفعال	
ur'\b(ان|ال|ف|ب|ك)(ة|تين|ات|ين)(\w)(\w\w)(و|ف|ي|)\b'	ur'\1\2\3\4ان\5\6\7'
ur'\b(ان|لل|ال|ف|و)(ة|تين|ات|ين)(\w)(\w\w)(و|ف|ي|)\b'	ur'\1\2\3ان\4\5\6'
ur'\b(ان|ال|ل|ب|ك)(\w)(\w\w)(و|ف|ي|)(ه|ها|كما|هما|هم|كم|هن|كن|نا|ه|ك|ها|تهما|تكما|تهم|تكم|تهن|تكن|تنا|ته|تها|تك|تكما|تهما|تك|ما|تكما|تهم|تكم|تهن|تكن|تنا|ته|تها|تك)\b'	ur'\1\2\3ان\4\5\6'
ur'\b(ان|ال|ف|و)(ة|تين|ات|ين)(\w)(\w\w)(و|ف|ي|)\b'	ur'\1\2\3ان\4\5\6'
ur'\b(ان|ال|ف|و)(ة|تين|ات|ين)(\w)(\w\w)(و|ف|ي|)\b'	ur'\1\2\3ان\4\5'

Table 3 Rules expressed by regular expressions.

4.2 Wordlist

Most common mistakes cannot be represented as regular expressions, such as errors in

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

the confusion between the Dhad and Za, and omitted dots on Teh and Yeh, such as in the المكتبه * and *فى, So we resort to build a list of common misspelled words.

To build an autocorrect word list, we suppose to use statistical extraction from a corpus, but we think that's not possible in Arabic language, because the common mistakes can have certain pattern and style, for example, people who can't differentiate between Dhad and Zah, make mistakes in all words containing these letters. Mistakes on Hamzat are not limited to some words, but can be typical and occur according to letters not especially for some words.

For this reason, we propose to build a word list based on Attia (2012) spell-checking word list, by generating errors for common letters errors, then filter resulted word list to obtain an autocorrect word list without ambiguity.

How to build generated word list:

1- take a correct word list

2- select candidate words:

- words start by Hamza Qat' or Wasl.
- words end by Yeh or Teh marbuta.
- Words contain Dhad or Zah.

3- Make errors on words by replacing candidate letters by errors.

4- Spell check the wordlist, and eliminate correct words, because some modified words can be correct, for example, if we take the word ضلَ Dhalla ، then modify it to ظل Zalla , the modified word exists in the dictionary, then we exclude it from autocorrect wordlist, and we keep only misspelled modified words.

words	modified	Spellcheck	Add to word list
بمكتبة	بمكتبه	True	
المكتبة	المكتبه	False	المكتبه
بالمكتبة	بالمكتبه	False	بالمكتبه
وبالمكتبة	وبالمكتبه	False	وبالمكتبه
ومكتبة	ومكتبه	True	

Table 4 Example of word errors generating

For example, if we have the word إسلام Islam, it can be written as اسلام Islam by mistake because that have the same pronociation. We can generate errors on words by appling some rule:

- Alef with Hamza above همزة قطع <=> Alef همزة وصل
- Alef with Hamza below همزة تحت الألف <=> Alef همزة وصل
- Dhah ض <=> Zah ظ
- The Marbuta ة <=> Heh ه
- Yeh ي <=> Alef Maksura ى

We suppose that we have the following word list, this list is chosen to illustrate some cases.

إسلام
ظلام
ظل
مكتبة
المكتبة
إعلام

For every word, we map an mistaken word, then we get a list like this:

Word	candidate word
إسلام	اسلام
ظلام	ضلام
ظل	ضل
مكتبة	مكتبه
المكتبة	المكتبه
إعلام	اعلام

We note that some candidate words are right, then we remove it, and the remaining words consititute the autocorrect wordlist

Word	candidate word
إسلام	اسلام
ظلام	ضلام
المكتبة	المكتبه
إعلام	اعلام

The following list (cf. Table**5**) shows the number of words in each type of errors,

Error type	**Words count**
words started by Hamza Qat'	101853
words ended by Yeh	700198
words ended by Teh marbuta	152210
words contained Dhad	396506
words contained Zah	94395
Total	1445162

Table 5 Errors categories in wordlist

The large number of words is due to the multiple forms per word, which avoids the morphological analysis, in such programs.

Customized Wordlist

Large number of replacement cases in generated autocorrect list encourages us to make an improvement to generate customized list for specific cases in order to reduce list length. We apply the following algorithm to generate customized list from large text data set:

1. Extract misspelled words from dataset by using Hunspell spellchecker.
2. Generate suggestions given by Hunspell

3. Study suggestions to choose the best one in hypothesis that words have common errors on letters according to modified letters.
4. Exclude ambiguous cases.

The automatically generated word list is used to autocorrect the dataset instead of default word list

5 Tools and resources

In our program we have used the following resources:

- Arabic word list for spell checking containing 9 million Arabic words, from Attia works (2012).
- a simple Python script to generate errors.
- Hunspell spellchecker program with Ayaspell dictionary (Hadjir 2009, Zerrouki, 2013). and Attia spellchecking wordlist (2012).
- our autocorrect program named Ghalatawi[2] (cf. a screenshot on Figure 1) ,
- A script to select best suggestion from Hunspell correction suggestions to generate customized autocorrect list.

Example

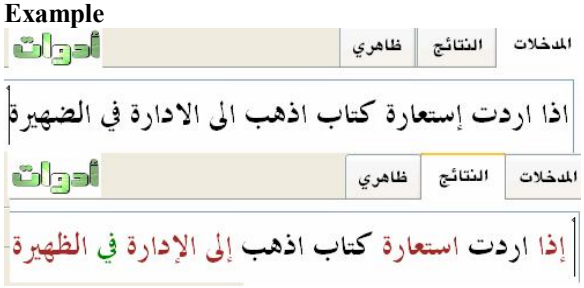

Figure 1 Ghalatawi program, autocorrection example

6 Evaluation

In order to evaluate the performance of automatic correction program, we used the data set provided in the shared task test (Behrang, 2014). After that autocorrect the texts by Galatawi program based on regular expressions and a wordlist.

For this evaluation we have used two autocorrect word lists:

- a generic word list generated from Attia wordlist, this wordlist is used for general purposes. This word list is noted in evaluation as "STANDARD".

- a customized wordlist based on dataset, by generating a special word list according to data set, in order to improve auto correction and avoid unnecessary replacement. this wordlist is noted in evaluation as "CUSTOMIZED".

The customized autocorrect word list is built in the same way as STANDARD, by replacing the source dictionary by misspelled words from QALB corpus (Zaghouani, 2014).

How customized list is built from dataset?

1- Hunspell detects 3463 unrepeated misspelled word in the dataset, like

```
للامريكيين*، الاف *
اإثيوبي
, االسف
االشعب
االقاتل
,االمتظاهرين
,االمدعو
,االمدنين، المرسوم
```

2- Hunspell generates suggestions for misspelled words, like

```
@(#) International Ispell Version 3.2.06 (but really Hunspell 1.3.2)
& للامريكيين 1 4: للأمريكيين

& الاف 15 1: الأف، الآف، ألاف، ألاق، ألأف، ألآف، إلاف، إلاق، آلاف، آلأف، آلآف، لافا، للاف، تلاف، غلاف
```

3- the script can select all words with one suggestion, and words with near suggestion as a common error. The script has select only 1727 non ambiguous case (not repeated).

The customized autocorrected list is used in test as CUSTOMIZED.

We got the following results (cf. Table **6**) by using the M2 scorer (Dahlmeier et al 2012):

	Training		Test	
	STAND.	CUST.	STAND.	CUST.
Precision	0.6785	0.7383	0.698	0.7515
Recall	0.1109	0.2280	0.1233	0.2315
F_1.0	0.1906	0.3484	0.2096	0.35

Table 6 Training dataset evaluation

We note that the customized wordlist give us precision and recall better than the use of standard wordlist.

7 Conclusion

AutoCorrect for words is to propose a one correction for common errors in writing.

[2] The Ghalatawi autocorrect program is available as an open source program at http://ghalatawi.sourceforge.net

In Arabic there are the following common mistakes: failure to differentiate between Hamza Wasl and Qat', confusion between the Dhah and Zah, and the omission of dots on Teh and under Yeh.

We have tried in this paper to find a way to adjust these errors automatically without human review, using a list of words and regular expressions to detect and correct errors.

This technique has been tried on the QALB corpus and gave mentioned results.

References

Hadjir‹I› "Towards an open source arabic spell checker", magister in Natural language processing, scientific and technique research center to arabic language development, 2009.

Zerrouki T, "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.

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

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

K Shaalan, A Allah, Towards automatic spell checking for Arabic… - Conference on Language Engineering, 2003 - claes.sci.eg

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

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

Behrang Mohit, Alla Rozovskaya, Wajdi Zaghouani, Ossama Obeid, and Nizar Habash , 2014. The First shared Task on Automatic Text Correction for Arabic.

Wajdi Zaghouani, 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.

Golding and Roth. "A Winnow based approach to Context-Sensitive Spelling Correction". 1999.

Dahlmeier, Daniel and Ng, Hwee Tou. 2012. Better evaluation for grammatical error correction. In Proceedings of NAACL.

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