The Columbia System in the QALB-2014 Shared Task on Arabic Error Correction

Alla Rozovskaya Nizar Habash[†] Ramy Eskander Noura Farra Wael Salloum

Center for Computational Learning Systems, Columbia University
[†]New York University Abu Dhabi

{alla,ramy,noura,wael}@ccls.columbia.edu
[†]nizar.habash@nyu.edu

Abstract

The QALB-2014 shared task focuses on correcting errors in texts written in Modern Standard Arabic. In this paper, we describe the Columbia University entry in the shared task. Our system consists of several components that rely on machine-learning techniques and linguistic knowledge. We submitted three versions of the system: these share several core elements but each version also includes additional components. We describe our underlying approach and the special aspects of the different versions of our submission. Our system ranked first out of nine participating teams.

1 Introduction

The topic of text correction has seen a lot of interest in the past several years, with a focus on correcting grammatical errors made by learners of English as a Second Language (ESL). The two most recent CoNLL shared tasks were devoted to grammatical error correction for non-native writers (Ng et al., 2013; Ng et al., 2014).

The QALB-2014 shared task (Mohit et al., 2014) is the first competition that addresses the problem of text correction in Modern Standard Arabic (MSA) texts. The competition makes use of the recently developed QALB corpus (Zaghouani et al., 2014). The shared task covers all types of mistakes that occur in the data.

Our system consists of statistical models, linguistic resources, and rule-based modules that address different types of errors.

We briefly discuss the task in Section 2. Section 3 gives an overview of the Columbia system and describes the system components. In Section 4, we evaluate the complete system on the development data and show the results obtained on test. Section 5 concludes.

2 Task Description

The QALB-2014 shared task addresses the problem of correcting errors in texts written in Modern Standard Arabic (MSA). The task organizers released training, development, and test data. All of the data comes from online commentaries written to Aljazeera articles.[1] The training data contains 1.2 million words; the development and the test data contain about 50,000 words each. The data was annotated and corrected by native Arabic speakers. For more detail on the QALB corpus, we refer the reader to Zaghouani et al. (2014). The results in the subsequent sections are reported on the development set.

It should be noted that in the annotation process, the annotators did not assign error categories but only specified an appropriate correction. In spite of this, it is possible, to isolate certain error types automatically, by using the corrections in coordination with the input words. The first type concerns punctuation errors. Errors involving punctuation account for about 39% of all errors in the data. In addition to punctuation mistakes, another very common source of errors refers to suboptimal spelling for two groups of letters – *Alif* (and its *Hamzated versions*) and *Ya* (and its *undotted* or *Alif Maqsura versions*). For more detail on this and other Arabic phenomena, we refer the reader to Habash (2010; Buckwalter (2007; El Kholy and Habash (2012). Mistakes associated with *Alif* and

[1] http://www.aljazeera.net/

Proceedings of the EMNLP 2014 Workshop on Arabic Natural Langauge Processing (ANLP), pages 160–164,
October 25, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

Component	System		
	CLMB-1	CLMB-2	CLMB-3
MADAMIRA	✓	✓	
MLE	✓	✓	
Naïve Bayes	✓		
GSEC			✓
MLE-unigram			✓
Punctuation	✓	✓	✓
Dialectal		✓	
Patterns	✓	✓	✓

Table 1: **The three versions of the Columbia system and their components.**

Ya spelling constitute almost 30% of all errors.

3 System Overview

The Columbia University system consists of several components designed to address different types of errors. We submitted three versions of the system. We refer to these as CLMB-1, CLMB-2, and CLMB-3. Table 1 lists all of the components and indicates which components are included in each version. The components are applied in the order shown in the table. Below we describe each component in more detail.

3.1 MADAMIRA Corrector

MADAMIRA (Pasha et al., 2014) is a tool designed for morphological analysis and disambiguation of Modern Standard Arabic. MADAMIRA performs morphological analysis in context. This is a knowledge-rich resource that requires a morphological analyzer and a large corpus where every word is marked with its morphological features. The task organizers provided the shared task data pre-processed with MADAMIRA, including all of the features generated by the tool for every word. In addition to the morphological analysis and contextual morphological disambiguation, MADAMIRA also performs *Alif* and *Ya* spelling correction for the phenomena associated with these letters discussed in Section 2. The corrected form was included among the features and can be used for correcting the input. We use the corrections proposed by MADAMIRA and apply them to the data. As we show in Section 4, while the form proposed by MADAMIRA may not necessarily be correct, MADAMIRA performs at a very high precision. MADAMIRA corrector is used in the CLMB-1 and CLMB-2 systems.

3.2 Maximum Likelihood Model

The Maximum Likelihood Estimator (MLE) is a supervised component that is trained on the training data of the shared task. Given the annotated training data, a map is defined that specifies for every word n-gram in the source text the most likely n-gram corresponding to it in the target text. The MLE model considers source n-grams of lengths between 1 to 3; the MLE-unigram model that is part of the CLMB-3 version only considers n-grams of length 1.

The MLE approach performs well on errors that have been observed in the training data and can be unambiguously corrected without using the surrounding context, i.e. do not have many alternative corrections. Consequently, MLE fails on words that have many possible corrections, as well as words not seen in training.

3.3 Naïve Bayes for Unseen Words

The Naïve Bayes component addresses errors for words that were not seen in training. The system uses the approach proposed in Rozovskaya and Roth (2011) that proved to be successful for correcting errors made by English as a Second Language learners. The model operates at the word level and targets word replacement errors that involve single tokens. Candidate corrections are generated using a character confusion table that is based on the training data. The model is a Naïve Bayes classifier trained on the Arabic Gigaword corpus (Parker et al., 2011) with word n-gram features in the 4-word window around the word to be corrected. The Naïve Bayes component is used in the CLMB-1 system.

3.4 The GSEC Model

The CLMB-3 system implements a Generalized Character-Level Error Correction model (GSEC) proposed in Farra et al. (2014). GSEC is a supervised model that operates at the character level. Because of this, the source and the target side of the training data need to be aligned at the character level. We use the alignment tool Sclite (Fiscus, 1998). The alignment maps each source character to itself, a different character, a pair of characters, or an empty string. For the shared task, punctuation corrections are ignored since punctuation errors are handled by the punctuation corrector described in the following section. It should also be noted that the model was not trained to insert missing characters. The model is a multi-class SVM classifier (Kudo, 2005) that makes use of character-level features using a window of four characters that may occur within the word boundaries as well as in the surrounding context. Due to a long training time, GSEC was trained on a quarter of the training data. The system is post-processed with a unigram word-level maximum-likelihood model described in Section 3.2. For more detail on the GSEC approach, we refer the reader to Farra et al. (2014).

3.5 Punctuation Corrector

The shared task data contains a large number of punctuation mistakes. Punctuation errors, such as missing periods and commas, account for about 30% of all errors in the data. Most of these errors involve incorrectly omitting a punctuation symbol. Our punctuation corrector is a statistical model that inserts periods and commas. The system is a decision tree model trained on the shared task training data using WEKA (Hall et al., 2009). For punctuation insertion, every space that is not followed or preceded by a punctuation mark is considered.

To generate features, we use a window of size three around the target space. The features are defined as follows:

- The part-of-speech of the previous word
- The existence of a conjunctive or connective proclitic in the following word; that is a “w” or “f” proclitic that is either a conjunction, a sub-conjunction or a connective particle

The part-of-speech and proclitic information is obtained by running MADAMIRA on the text.

We also ran experiments where the model is trained with a complete list of features produced by MADAMIRA; that is part-of-speech, gender, number, person, aspect, voice, case, mood, state, proclitics and enclitics. This was done for two preceding words and two following words. However, this model did not perform as well as the one described above, which we used in the final system.

Note that the punctuation model predicts presence or absence of a punctuation mark in a specific location and is applied to the source data from which all punctuation marks have been removed. However, when we apply our punctuation model in the correction pipeline, we find that it is always better to keep the already existing periods and commas in the input text instead of overwriting them with the model prediction. In other words, we only attempt to add missing punctuation.

3.6 Dialectal Usage Corrector

Even though the shared task data is written in MSA, MSA is not a native language for Arabic speakers. Typically, an Arabic speaker has a native proficiency in one of the many Arabic dialects and learns to write and read MSA in a formal setting. For this reason, even in MSA texts produced by native Arabic speakers, one typically finds words and linguistic features specific to the writer’s native dialect that are not found in the standard language.

To address such errors, we use Elissa (Salloum and Habash, 2012), which is Dialectal to Standard Arabic Machine Translation System. Elissa uses a rule-based approach that relies on the existence of a dialectal morphological analyzer (Salloum and Habash, 2011), a list of hand-written transfer rules, and dialectal-to-standard Arabic lexicons. Elissa uses different dialect identification techniques to select dialectal words and phrases (dialectal multi-word expressions) that need to be handled. Then equivalent MSA paraphrases of the selected words/phrases are generated and an MSA lattice for each input sentence is constructed. The paraphrases within the lattice are then ranked using language models and the n-best sentences are extracted from lattice. We use 5-gram language models trained using SRILM (Stolcke, 2002) on about 200 million untokenized, *Alif*/*Ya* normalized words extracted from Arabic GigaWord. This component is employed in the CLMB-2 system.

3.7 Pattern-Based Corrector

We created a set of rules that account for very common phenomena involving incorrectly split or merged tokens. The MADAMIRA corrector described above does not handle splits and merges; however, some of the cases are handled in the MLE method. Note that the MLE method is restrictive since it does not correct words not seen in training, while the pattern-based corrector is more general. The rules were created through analysis of samples of the QALB Shared Task training data. Some of the rules use regular expressions, while others make use of the rule-based Standard Arabic Morphological Analyzer (SAMA) (Maamouri et al., 2010), the same out-of-context analyzer used inside of MADAMIRA.

Rules for splitting words

- All digits are separated from words.
- A space is added after all word medial *Ta-Marbuta* characters.
- A space is added after the very common “ElY” ‘at/about/on’ preposition if it is attached to the following word.
- If a word has a morphological analysis that includes “lmA” (as negation particle, relative pronoun or pseudo verb), “hA” (a demonstrative pronoun), or “Ebd” and “>bw” in proper nouns, a space is inserted after those parts of the analysis.
- If a word has no morphological analysis, but starts with a set of commonly mis-attached words, and the rest of the word has an analysis, the word is split after the mis-attached word sequence.

Rules for merging words

- All lone occurrences of the conjunction *w* ‘and’ are attached to the following word.
- All sequences of the punctuation marks (., ?, !) that occur between two and six times are merged: e.g ! ! ! → !!!.

4 Experimental Results

In Section 3, we described the individual system components that address different types of errors. In this section, we show how the system improves when each component is added into the system. System output is scored with the M2 scorer (Dahlmeier and Ng, 2012), the official scorer of the shared task.

Table 2 reports performance results of each version of the Columbia system on the development data. Table 3 shows the performance results for the best-performing system, CLMB-1, as each system component is added.

System	P	R	F1
CLMB-1	**72.22**	**62.79**	**67.18**
CLMB-2	69.49	61.72	65.38
CLMB-3	69.71	59.42	64.15

Table 2: **Performance of the Columbia systems on the development data.**

System	P	R	F1
MADAMIRA	83.33	32.94	47.21
+ MLE	86.52	42.52	57.02
+ NB	85.80	43.27	57.53
+ Punc.	73.66	59.51	65.83
+ Patterns	72.22	62.79	67.18

Table 3: **Performance of the CLMB-1 system on the development data and the contribution of its components.**

System	P	R	F1
CLMB-1	**73.34**	**63.23**	**67.91**
CLMB-2	70.86	62.21	66.25
CLMB-3	71.45	60.00	65.22

Table 4: **Performance of the Columbia systems on the test data.**

Finally, Table 4 reports results obtained on the test data. These results are comparable to the performance observed on the development data. In particular, CLMB-1 achieves the highest score.

5 Conclusion

We have described the Columbia University system that participated in the first shared task on grammatical error correction for Arabic and ranked first out of nine participating teams. We have presented three versions of the system; all of these incorporate several components that target different types of mistakes, which we presented and evaluated in this paper.

Acknowledgments

This material is based on research funded by grant NPRP-4-1058-1-168 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors. Nizar Habash performed most of his contribution to this paper while he was at the Center for Computational Learning Systems at Columbia University.

References

T. Buckwalter. 2007. Issues in Arabic Morphological Analysis. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

D. Dahlmeier and H. T. Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of NAACL*.

A. El Kholy and N. Habash. 2012. Orthographic and morphological processing for English–Arabic statistical machine translation. *Machine Translation*, 26(1-2).

N. Farra, N. Tomeh, A. Rozovskaya, and N. Habash. 2014. Generalized character-level spelling error correction. In *Proceedings of ACL*.

J. Fiscus. 1998. Sclite scoring package version 1.5. US National Institute of Standard Technology (NIST), URL http://www. itl. nist. gov/iaui/894.01/tools.

N. Y. Habash. 2010. *Introduction to Arabic natural language processing*. Synthesis Lectures on Human Language Technologies 3.1.

M. Hall, F. Eibe, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. 2009. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18.

T. Kudo. 2005. YamCha: Yet another multipurpose chunk annotator. http://chasen.org/ taku/software/.

M. Maamouri, D. Graff, B. Bouziri, S. Krouna, A. Bies, and S. Kulick. 2010. *LDC Standard Arabic Morphological Analyzer (SAMA) Version 3.1*. Linguistic Data Consortium.

B. Mohit, A. Rozovskaya, N. Habash, W. Zaghouani, and O. Obeid. 2014. The first QALB shared task on automatic text correction for Arabic. In *Proceedings of EMNLP Workshop on Arabic Natural Language Processing*.

H. T. Ng, S. M. Wu, Y. Wu, Ch. Hadiwinoto, and J. Tetreault. 2013. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of CoNLL: Shared Task*.

H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of CoNLL: Shared Task*.

R. Parker, D. Graff, K. Chen, J. Kong, and K. Maeda. 2011. *Arabic Gigaword Fifth Edition*. Linguistic Data Consortium.

A. Pasha, M. Al-Badrashiny, A. E. Kholy, R. Eskander, M. Diab, 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 LREC*.

A. Rozovskaya and D. Roth. 2011. Algorithm selection and model adaptation for ESL correction tasks. In *Proceedings of ACL*.

W. Salloum and N. Habash. 2011. Dialectal to standard arabic paraphrasing to improve arabic-english statistical machine translation. In *Proceedings of the First Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*.

W. Salloum and N. Habash. 2012. Elissa: A dialectal to standard arabic machine translation system. In *Proceedings of COLING (Demos)*.

A. Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*.

W. Zaghouani, B. Mohit, N. Habash, O. Obeid, N. Tomeh, A. Rozovskaya, N. Farra, S. Alkuhlani, and K. Oflazer. 2014. Large scale arabic error annotation: Guidelines and framework. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.