From Arabic User-Generated Content to Machine Translation: Integrating Automatic Error Correction

Haithem Afli[1], Walid Aransa[2], Pintu Lohar[1] and Andy Way[1]

[1] ADAPT Centre
School of Computing,
Dublin City University,
Dublin, Ireland.
{hafli, plohar, away}@computing.dcu.ie

[2] Université du Maine,
Avenue Olivier Messiaen F-72085 - LE MANS, France.
walid.aransa@lium.univ-lemans.fr

Abstract. With the wide spread of the social media and online forums, individual users have been able to actively participate in the generation of online content in different languages and dialects. Arabic is one of the fastest growing languages used on Internet, but dialects (like Egyptian and Saudi Arabian) have a big share of the Arabic online content. There are many differences between Dialectal Arabic and Modern Standard Arabic which cause many challenges for Machine Translation of informal Arabic language. In this paper, we investigate the use of Automatic Error Correction method to improve the quality of Arabic User-Generated texts and its automatic translation. Our experiments show that the new system with automatic correction module outperforms the baseline system by nearly 22.59% of relative improvement.

Keywords: Automatic Error Correction, Machine translation, pre-processing, Arabic User-Generated content.

1 Introduction

User-generated Content (UGC) texts such as the posts, threads and comments found on the social media and web forums have different challenges on informal Arabic language processing than formal Modern Standard Arabic (MSA) texts (e.g. news). This comes from the fact that the majority of Natural Language Processing (NLP) tools for Arabic language are designed for MSA, while most of the online Arabic users are writing using Dialectal Arabic (DA) and informal style. Although DA and MSA are related but there are some lexical, phonological and morphological differences between them [11, 4, 3].

That's why it is widely accepted that machine translation systems are still imperfect in translating Arabic UGC [31]. This is reflected by errors contained in the original Arabic informal text (see for example in Table 1) which is typically measured with automatic quality metrics such as BLEU [26], or TER [28]. These scores alone, however, only reflect the overall translation quality of the text, and do not provide any insight in what exactly the origin of the translation problem. We can see in the Table 1, we have often Arabic words in the Machine Translation (MT) output which called Out Of Vocabulary (OOV). These words are a spelling errors of MSA words like قنات instead of قناة (which means 'channel') in the second example, or a Dialectal words like the Levantine (Syrian/Lebanese) word هيك converted to هكذا (which means "that", "like","that kind" or "such") in MSA. The challenge is that online users are linguistically switching between MSA and DA, either in the course of a single sentence or across different sentences. In [8], the authors found that 98.13% of sentences crawled from Egyptian DA discussion forums for the COLABA corpus contain intrasentential code switching. In this context, it is appropriate to ask the following question:

How can we automatically improve the informal UGC Arabic text translation?

In this paper we explore the effectiveness of Automatic Error Correction (AEC) of Arabic UGC and its impact on automatic translation. The combination uses a DA-to-MSA normalisation and an MSA correction system based on statistical approach. The proposed algorithm used to "convert" an informal text to its MSA version based on statistical methods. We investigate the use of Arabic tokenisations for improving the errors detecting.

The paper is organised as follows: Section 2 presents the background and the related work in UGC error correction; Section 3 provides the proposed error correction method and UGC-to-MT framework; Section 4 describes the technical details of using the statistical model with different tokenization and the feasibility experiments conducted; Section 5 reports the results and discuss the final Arabic UGC translation; and section 6 concludes the paper and provides possible future directions.

2 Background

2.1 Arabic User-Generated Errors

In Arabic UGC texts, we have several challenges including:

Arabic: . وهيك معارضة بدها هيك تعامل امني
MT output: and هيك opposition needs such behaviour
Ref Arabic: . وهكذا معارضة يلزمها هكذا تعامل امني
Ref Translation: and that opposition needs that kind of security dealings.
Arabic: . اللي العاملين في قنات الجزيرة
MT output: and اللي working for قنات Al-Jazeera.
Ref Arabic: . إلى العاملين في قناة الجزيرة
Ref Translation: for those who are working in Al-Jazeera channel.

Table 1. UGC Error examples with poor MT output. The first example represents a sentence in Levantine Dialect (without spelling error) mixed with MSA words and translated with a standard Arabic to English system. The second example represents an MSA sentence with spilling errors (قنات and اللي) translated by the same MT system.

- OOV problem: this happen if the word does not exist in the MSA dictionary *e.g.*قنات instead of قناة (which means channel). This happen because of a spelling mistake or because the word is dialectal word.
- Segmentation: extra space error can divide the word and generate a segmentation problem *e.g.*اة قن instead of قناة.
- Punctuation: in informal language, commas or points could be in wrong places or missing.
- Character format: the use of some Farsi/Urdu Characters like *e.g.*ڤ instead of ق or ڤ instead of ف in Tunisian, Algerian and Moroccan Dialects.
- Word sense ambiguity: some words are shared between DA and MSA but have different meaning *e.g.*بقي in Egyptian Dialect means 'become' while it means 'remain' in MSA i.e. أصبح.

2.2 Automatic Error Correction of Arabic UGC

The goal of Automatic Error Correction of UGC is to detect and correct misspellings and DA words in the text before translation. The obvious way to correct OCR errors is to edit the output text manually by linguists. This method requires a continuous manual human intervention which is to some degree regarded

as a costly and time-consuming practice.

There are two main existing approaches to automatically correct texts. The first approach is based on a lexical error correction [22, 13, 5]. In this method, a lexicon is used to spell check words and correct them if they are not present on the dictionary. Although this technique is easy to implement and use, it still have various limitations that prevent it to being the perfect solution for UGC Arabic texts. The first one is that it requires a wide-ranging dictionary that covers every single word in the language. However, a morphologically complex language such Arabic (MSA) and DA, have enormous numbers of possible words estimated to 60 billion possible surface forms. Another limitation is that conventional dictionary do not support names of regions, geographical locations, some technical keyword and domain specific terms. They normally target a single specific language in a given period, and thus, cannot support actual news day per day which are more discussed in UGC texts.

The second type of approach is the context-based error correction. Those techniques are founded on statistical language modelling and word n-grams. It aims at calculating the likelihood of a word sequence to appear [30, 19]. Using this technique, the candidate correction of an error might be successfully found using the "Noisy Channel Model" [20]. Accordingly, for each source word w we are looking for the word c that is the most likely spelling correction for that word (which may indeed be the original word itself). However, some words are more likely corrected than others because they are more frequent like *e.g.*the stop words, which can result in erroneous corrections. Also when many consecutive corrupted words are encountered in a sentence then it is difficult to consider good candidate words.

For the task of Arabic morphological analysis and disambiguation, [27] created MADAMIRA, which can be used to improve the accuracy of spelling checking system especially with Hamza spelling correction. [14] presented a statistical machine translation (SMT) model to train an error correction system. In contrast to these two approaches, we used MADA [10] and the SMT model to train our correction systems presented in this paper.

3 Statistical Conversion System

3.1 Basic Idea

This technique centres on using a conversion system trained on UGC texts which have been post-edited and manually corrected and normalised to MSA. It is like a UGC-to-MSA Machine Translation system with spelling Error correction. The translation systems handle the conversion process as the transformation of a sequence of symbols in a source language into another sequence of symbols in a target language. Generally the symbols dealt with are the words in the two languages. We consider that our system will translate UGC error words to a corrected MSA words in the same language following the work of [9, 1, 2].

In fact, using the standard approach of Statistical Machine Translation we are given a sentence (a sequence of Arabic words in informal form) $s^M = s_1...s_M$ of size M which is to be translated into a corrected MSA sentence $t^N = t_1...t_N$ of size N in the same language (Arabic in our case). The statistical approach aims at determining the translation t^* which maximizes the posterior probability given the source sentence. Formally, by using Bayes' rule, the fundamental equation is (1):

$$t^* = \arg\max_t Pr(t|s) = \arg\max_t Pr(s|t)Pr(t) \tag{1}$$

It can be decomposed, as in the original work of [6], into a language model probability $Pr(t)$, and a translation model probability $Pr(s|t)$. The language model is trained on a large quantity of MSA corrected data and the translation model is trained using a bilingual text aligned at sentence (segment) level, *i.e.* a UGC for a segment and its ground-truth in MSA obtained by manual annotation. As in most current state-of-the-art systems, the translation probability is modelled using the log-linear model [24] in (2):

$$P(t|s) = \sum_{i=0}^{N} \lambda_i h_i(s,t) \tag{2}$$

where $h_i(s,t)$ is the i$^t h$ feature function and λ_i its weight (determined by an optimisation process).

We assume that we do not need to use the reordering model (RM) in the task of spelling error correction in the same language as demonstrated by [1]. However, in our task we are also normalising DA to MSA which can lead to a transformation of the order of the words. That's why we kept using the RM in our conversion system.

3.2 Arabic tokenization

In the special case of Arabic language, Arabic words are often ambiguous in their morphological analysis. This is due to Arabic's rich system of affixation and clitics and the omission of disambiguating short vowels and other orthographic diacritics in standard orthography ("undiacritized") [10]. We believe that this complexity of the language can affect the detection of the spilling errors. That's why we proposed to use two different systems in order to verify this assumption:

- First System (called *Sys1*): trained with cleaned data without any tokenization;
- Second system (called *Sys2*): trained using MADA [10] tokenisation and;

An example of an Arabic UGC sentence and its MSA correction with the different tokenisations used in this paper can be seen in Table 2. The source and reference sentence are also spelled into Latin letter using Buckwalter code [7].

Source UGC sentence without tokenisation :
بعد إللي شفته وسمعته أنا جمال باشا السفاح أصدرت فرماني بنصب ٦١ مشنقة في ساحة المرجة ليشنق عليها أعمدة النظام السوري وذلك بعد العيد مباشرة . لا محاكم ولا لاهاي ولا تضييع وقت
Reference MSA sentence without tokenisation:
بعد الذي رأيته وسمعته ، أنا جمال باشا السفاح ، أصدرت فرماني بنصب 16 مشنقة في ساحة المرجة ليشنق عليها أعمدة النظام السوري ، وذلك بعد العيد مباشرة ، . لا محاكم ، ولا لاهاي ، ولا تضييع وقت
Source UGC sentence with MADA tokenisation :
بعد اللي شفت +ه و+ سمعت +ه انا جمال باشا السفاح اصدرت ف+ رما +ني ب+ نصب 16 مشنقة في ساحة المرجة ل+ يشنق علي +ها اعمدة النظام السوري و+ ذلك بعد العيد مباشرة . لا محاكم و+ لا لاهاي و+ لا تضييع وقت
Reference MSA sentence with MADA tokenisation :
بعد الذي رايت +ه و+ سمعت +ه ، انا جمال باشا السفاح ، اصدرت ف+ رما +ني ب+ نصب 16 مشنقة في ساحة المرجة ل+ يشنق علي +ها اعمدة النظام السوري ، و+ ذلك بعد العيد مباشرة ، . لا محاكم ، و+ لا لاهاي ، و+ لا تضييع وقت
Source UGC sentence in Buckwalter spelling :
bEd <lly $fth wsmEth >nA jmAl bA$A AlsfAH >Sdrt frmAny bnSb ٦١ m$nqp fy sAHp Almrjp ly$nq ElyhA >Emdp AlnZAm Alswry w*lk bEd AlEyd mbA$rp lA mHA km wlA lAhAy wlA tDyyE wqt .
Reference MSA sentence in Buckwalter spelling :
bEd Al*y r>yth wsmEth , >nA jmAl bA$A AlsfAH , >Sdrt frmAny bnSb 16 m$nqp fy sAHp Almrjp ly$nq ElyhA >Emdp AlnZAm Alswry , w*lk bEd AlEyd mbA$rp , lA mHAkm , wlA lAhAy , wlA tDyyE wqt .

Table 2. Example of UGC sentence and its MSA correction in the different tokenisation forms used in our experiments.

3.3 UGC-to-MT framework

The basic system architecture is depicted in Figure 1. We can distinguish three steps: automatic tokenization and cleaning, error correction (Sys Correction) and machine translation (MT). We begin by cleaning the original documents in language L1 (Arabic UGC in our case) and generates an automatic tokenization. This text is then corrected by the statistical conversion method described in the previous section.

The final corrected text in L1 forms the input to the MT system. We anticipate that the automatic correction will improve the quality of the final translation to the language L2 (English in our case). Accordingly, this framework sets out to address the question of whether a shared and novel integration of language processing components from a corrected UGC can significantly improve the final translation quality of informal texts.

3.4 Impact of Error Correction on Automatic Translation

The proposed UGC-to-MT framework raises several issues. Each step can introduce a certain number of errors. It is important to highlight the feasibility of the approach and the impact of each module on the final automatic translation. Thus, we conducted three different types of experiments, described in Figure 2.

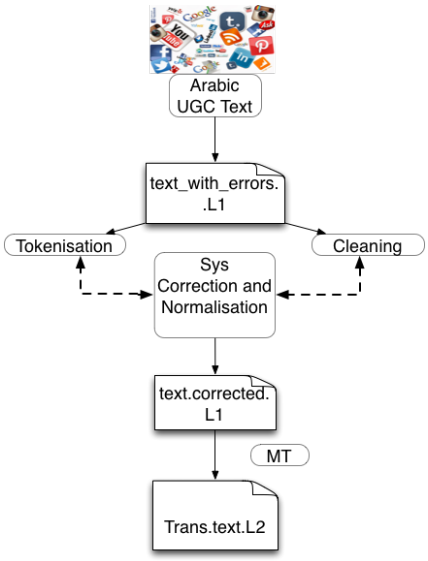

Fig. 1. The proposed UGC-to-MT framework.

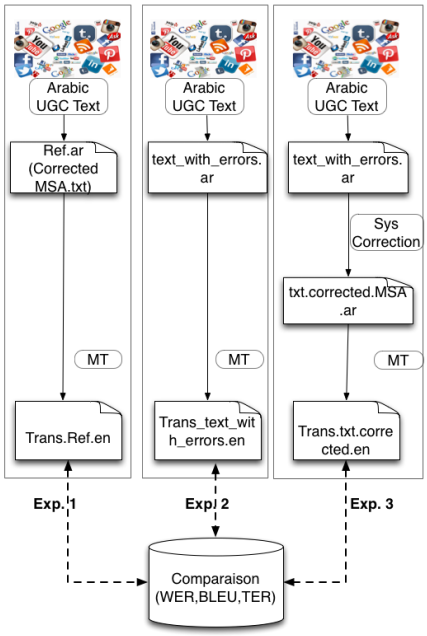

Fig. 2. Different experiments to analyze the impact of the correction module.

Source UGC sentence: اعداد القتلى في صفوف الارهابيين بالمئات و من الصعب حصر اعداد القتلى بشكل دقيق بسبب الحرب الشاملة التي يشنها الجيش العربي السوري في اماكن تواجدهم .
A 0 1 edit أعداد REQUIRED -NONE- 0 A 4 5 edit الإرهابيين REQUIRED -NONE- 0 A 6 6 add_token_before ، REQUIRED -NONE- 0 A 6 8 merge ومن REQUIRED -NONE- 0 A 10 11 edit أعداد REQUIRED -NONE- 0 A 23 24 edit أماكن REQUIRED -NONE- 0
Corrected MSA sentence generated automatically: أعداد القتلى في صفوف الإرهابيين بالمئات، ومن الصعب حصر أعداد القتلى بشكل دقيق بسبب الحرب الشاملة التي يشنها الجيش العربي السوري في أماكن تواجدهم .

Table 3. Corrected Arabic User-Generated Content Error example from QALB corpus. The first correction replaces token with ID 0 with the word (أعداد) which is exactly like second, fifth and sixth modifications called (edit). The third correction (add_token) specifies an insertion of an arabic comma (،) in front of token with tokenID 6. The fourth correction merges tokens 6 and 7.

In the first experiment (*Exp. 1*) we use the MSA reference (*Ref.ar*) as input to the MT system. This is the most favourable condition, as it simulates the case where the Error Correction systems do not commit any error. Accordingly, we consider this as the reference during the automatic evaluation process. In the second experiment (*Exp. 2*) – the baseline experiment – we use the UGT text (*text_with_errors.ar*) directly as input to the MT system without any correction. Finally, the third experiment represents the complete proposed framework, described in Section 3.3.

4 Data and systems description

4.1 Data description

To train our correction models, a 1,3 million Arabic UGC words were obtained from QALB corpus [32]. The segments were then manually corrected by annotators. An example of this annotation can be seen in the Table 3. Based on this manual correction, we generated the corrected MSA version of the training, development and test data. Next, the UGC sentences and the MSA version were aligned at sentence level and tokenised using MADA.

Statistics of the corpus used for the automatic error correction in our experiments are presented in Table 4.

bitexts	# UGC tokens	# ref MSA tokens
train_raw	1.22M	1.31M
train_MADA_tok	14.8M	15.6M
dev_raw	3 64.6K	69.5K
dev_MADA_tok	78.0K	83.4K
test_raw	61.2K	65.9K
test_MADA_tok	74.1K	79.4K

Table 4. Statistics of training (train), development (dev) and test data available to build the correction system.

The language model used on the correction system was built using the KenLM[1] toolkit [12] with Kneser-Ney smoothing [15] and default backoff.

4.2 Arabic to English Machine Translation System

The language pair of the SMT system for UGC is MSA and Arabic dialects into the English. The system is a standard phrase-based system trained using Moses toolkit [16], SRILM [29], KenLM [12], and GIZA++ [25]. Log linear weights are optimized using MERT [23].

The SMT system is trained using the bilingual training corpora listed in Table 5 from LDC. The size of the tuning set is 111.8K and 138.2K of Arabic and English tokens. All Arabic data are tokenized using MADA-ARZ version 0.4 [11]. We used data selection method based on [21] to select the relevant monolingual data for our 4-gram back-off language model.

5 Results

5.1 Automatic Error Correction

In order to evaluate the effectiveness of error correction, we used Word Error Rate (WER) which is derived from Levenshtein distance [17]. We compare results on the test data of the different systems used in our experiments, against the baseline results which represent scores between the original UGC arabic text and the corrected MSA reference (called *UGC-Baseline*).

Table 6 reports on the percentage of Correctness, Accuracy and WER of different system outputs. We can see that the two models trained with the same method described in section 3.1, with/without tokenization, were able to decrease the Baseline Errors in the UGC text.

[1] https://kheafield.com/code/kenlm

Corpus	Arabic genre	Arabic tokens	English tokens
bolt	Egyptian dialect	1.70M	2.05M
thy		282k	362k
bbnturk		1.52M	1.58M
bbnegy		514k	588k
gale	MSA	4.28M	5.01 M
fouo		717 k	791k
ummah		3.61M	3.72M
iraqi	Iraqi dialect	1M	1.14M
bbnlev	Levantine dialect	1.59M	1.81M
Total		15.2M	17M

Table 5. The sizes and the genres of bilingual training corpora.

Systems	Correctness	Accuracy	WER
UGC-Baseline	71.79	70.38	29.62
Sys1	86.44	82.48	17.52
UGC-Baseline	70.61	55.99	44.01
Sys2	84.05	77.92	22.08

Table 6. Word Error Rate (WER), Accuracy and Correctness results on test UGC-corrected data using Sys1 and Sys2 compared to the UGC-Baseline.

5.2 Machine translation

For the translation evaluation we used BLEU-4 score [26], Smoothed BLEU [18] and TER [28] calculated between the output of *Exp.1* (our reference) and *Exp. 2* output (the baseline) or *Exp.3* output (our proposed framework).

Table 7 lists the results of the two translation outputs from *Exp. 2* and *Exp. 3* with different systems used in this experiments, compared to *Exp.1* output. It shows that our proposed framework is very capable of correcting the final translation of the Arabic UGC text. The best system (*Sys2*) increases near 4 points in the BLEU-4 and 3 points in the Smooth BLEU scores. This results are confirmed with the decrease of 1.72 TER score points.

Systems	BLEU-4	Smooth BLEU	TER
Exp. 2	64.41	64.42	23.17
Exp. 3 Sys1	67.30	66.53	21.94
Exp. 3 Sys2	**68.31**	**67.42**	**21.45**

Table 7. BLEU-4, Smooth BLEU and TER results on test translated UGC-data corrected by Sys1 and Sys2.

6 Discussion

6.1 Systems comparaison

In order to analyze the degree of the agreement between the different systems, We transformed the Sys1 outputs to the same tokenisation of Sys2 and we scored all of them comparing to the MSA correction reference transformed in MADA tokenisation (of Sys2), and using WER metric.

Systems	Correctness	Accuracy	WER
UGC-Baseline	67.24	59.54	40.46
Sys1	70.31	63.18	36.82
Sys2	**74.81**	**68.68**	**31.32**

Table 8. Word Error Rate (WER), Accuracy and Correctness results on test UGC-corrected data.

We can see that the best improvements of the correction results are obtained with the Sys2 using the MADA tokenisation. This model was able to decrease 9.14% of the UGC word errors, which means 22.59% of relative improvements.

6.2 Analysis

The experiments performed in this paper showed that the integration of an error correction module as pre-processing is very helpful to improve the translation results of User-Generated Content Arabic Data. This method can resolve the problem of MT system adaptation to different dialects by the normalisation to the MSA form of the Arabic language which represents a challenge for Arabic SMT translation. Our correction results showed also that all different systems with/without tokenisation can decrease the word errors of the Arabic UGC texts. This suggest to apply combination of these different systems using a confusion network combination method in order to select the best correction of each sentence. This assumption will be one of our future investigation in the processing of Arabic UGC data. Lastly, we are considering branching into application areas other than MT that can benefit from this correction framework, in particular, Arabic information retrieval, sentiment analysis and language learning.

7 Conclusion

In this paper, we presented a new framework of Arabic User-Generated Content translation. The proposed method consists of the integration of a new error correction system prior to the translation phase *per se*. We validate the feasibility of our approach using a set of experiments to analyze the impact of error

correction module on the final translation. The use of this approach allows the system to correct spilling errors and convert Dialectal words to Standard Arabic. We have shown that such systems are able to improve the final translation. Our best model outperforms the UGC-Base (baseline) by up to 4 BLEU points and 1.72 TER points (when computed at tokenised version), which represents a good improvement.

Nowadays User-Generated Content of some complex morphological languages like Arabic represent a challenge for many translation projects. The morphological complexity of such languages, which have billions surface forms (*e.g.*60 billions for Arabic), complicates others correction methods like dictionary-based [19]. This is mainly because listing all the possible words is not an easy task. That is why we believe that our new method can be a good way to resolve this kind of problems. We plan to test it on other different languages and types of data. As future work, we would like to investigate the robustness of our systems and their combination with other methods.

References

1. Haithem Afli, Loïc Barrault, and Holger Schwenk. OCR Error Correction Using Statistical Machine Translation. *16th International Conference on Intelligent Text Processing and Computational Linguistics*, Cairo, Egypt 2015.
2. Haithem Afli, Zhengwei Qiu, Andy Way, and Páraic Sheridan. Using SMT for OCR error correction of historical texts. In *Proceedings of LREC-2016*, (to appear), Portorož, Slovenia 2016.
3. Mohammed Attia, Pavel Pecina, Younes Samih, Khaled Shaalan, and Josef van Genabith. Improved spelling error detection and correction for arabic. In *Proceedings of COLING 2012*, pages 103–112, Mumbai, India, 2012.
4. Mahmoud Azab, Houda Bouamor, Behrang Mohit, and Kemal Oflazer. Dudley north visits north london: Learning when to transliterate to arabic. In *Proceedings of NAACL-HLT 2013*, pages 439–444, Atlanta, Georgia, USA 2013.
5. Youssef Bassil and Mohammad Alwani. OCR Post-Processing Error Correction Algorithm Using Google's Online Spelling Suggestion. *Journal of Emerging Trends in Computing and Information Sciences*, 3:90–99, 2012.
6. Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19:263–311, 1993.
7. Tim Buckwalter. Buckwalter arabic morphological analyzer version 1.0. Technical Report LDC Catalog No.: LDC2002L49, Linguistic Data Consortium, University of Pennsylvania, 2002.
8. Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Yassin Benajiba. . colaba: Arabic dialect annotation and processing. In *Proceedings of LREC Workshop on Semitic Language Processing*, pages 66–74, Malta, 2010.
9. Federico Fancellu, Andy Way, and Morgan O'Brien. Standard language variety conversion for content localisation via SMT. *17th Annual Conference of the European Association for Machine Translation*, pages 143–149, Dubrovnik, Croatia 2014.

10. Nizar Habash and Owen Rambow. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 573–580, Ann Arbor, USA 2005.
11. Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh. Morphological analysis and disambiguation for dialectal arabic. In *Proceedings of NAACL-HLT 2013*, pages 426–432, Atlanta, Georgia, USA 2013.
12. Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the 6th Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, UK, 2011.
13. Tao Hong. *Degraded Text Recognition Using Visual and Linguistic Context.* PhD thesis, University of New York, NY, USA 1995.
14. Serena Jeblee, Houda Bouamor, Wajdi Zaghouani, and Kemal Oflazer. CMUQ@QALB-2014: An SMT based system for automatic arabic error correction. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 137–142, Doha, Qatar, 2014.
15. Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 181–184, Detroit, Michigan, USA, 1995.
16. Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Meeting of the Association for Computational Linguistics*, pages 177–180, 2007.
17. V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710, 1966.
18. Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 605–612, Barcelona, Spain 2004.
19. Walid Magdy and Kareem Darwish. Arabic OCR Error Correction Using Character Segment Correction, Language Modeling, and Shallow Morphology. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 408–414, Sydney, Australia 2006.
20. Eric Mays, Fred J. Damerau, and Robert L. Mercer. Context based spelling correction. *Information Processing and Management*, 27(5):517–522, 1991.
21. Robert C. Moore and William Lewis. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Stroudsburg, PA, USA, 2010.
22. Hisao Niwa, Kazuhiro Kayashima, and Yasuham Shimeki. Postprocessing for character recognition using keyword information. In *IAPR Workshop on Machine Vision Applications*, volume MVA'92, pages 519–522, Tokyo, Japan 1992.
23. Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA, 2003.
24. Franz Josef Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 295–302, Philadelphia, USA, 2002.
25. Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29:19–51, March 2003.

26. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania 2002.
27. Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan 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'14)*, pages 1094–1101, Reykjavik, Iceland, 2014.
28. S. Snover, B. Dorr, R. Schwartz, M. Micciulla, and J. Makhoul. A study of translation edit rate with targeted human annotation. *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA 2006.
29. Andreas Stolcke. Srilm - an extensible language modeling toolkit. *In Proceesings of the 7th International Conference on Spoken Language Processing (ICSLP 2002*, pages 901–904, 2002.
30. Mikael Tillenius. Efficient generation and ranking of spelling error corrections. Technical report, Royal Institute of Technology, Stockholm, Sweden 1996.
31. Marlies van der Wees, Arianna Bisazza, and Christof Monz. Five Shades of Noise: Analyzing Machine Translation Errors in User-Generated Text. In *Proceedings of the ACL 2015 Workshop on Noisy User-generated Text*, pages 28–37, Beijing, China, 2013.
32. Wajdi Zaghouani, Behrang Mohit, Nizar Habash, Ossama Obeid, Nadi Tomeh, Alla Rozovskaya, Noura Farra, Sarah Alkuhlani, and Kemal Oflazer. Large scale arabic error annotation: Guidelines and framework. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation LREC 2014*, pages 2362–2369, Reykjavik, Iceland, 2014.