Sentence Level Dialect Identification for Machine Translation System Selection

Wael Salloum, Heba Elfardy, Linda Alamir-Salloum, Nizar Habash and **Mona Diab**†
Center for Computational Learning Systems, Columbia University, New York, USA
{wael,heba,habash}@ccls.columbia.edu
†Department of Computer Science, The George Washington University, Washington DC, USA
†mtdiab@email.gwu.edu

Abstract

In this paper we study the use of sentence-level dialect identification in optimizing machine translation system selection when translating mixed dialect input. We test our approach on Arabic, a prototypical diglossic language; and we optimize the combination of four different machine translation systems. Our best result improves over the best single MT system baseline by 1.0% BLEU and over a strong system selection baseline by 0.6% BLEU on a blind test set.

1 Introduction

A language can be described as a set of dialects, among which one "standard variety" has a special representative status.[1] Despite being increasingly ubiquitous in informal written genres such as social media, most non-standard dialects are resource-poor compared to their standard variety. For statistical machine translation (MT), which relies on the existence of parallel data, translating from non-standard dialects is a challenge. In this paper we study the use of sentence-level dialect identification together with various linguistic features in optimizing the selection of outputs of four different MT systems on input text that includes a mix of dialects.

We test our approach on Arabic, a prototypical diglossic language (Ferguson, 1959) where the standard form of the language, Modern Standard Arabic (MSA) and the regional dialects (DA) live side-by-side and are closely related. MSA is the language used in education, scripted speech and official settings while DA is the primarily spoken native vernacular. We consider two DAs: Egyptian and Levantine Arabic in addition to MSA. Our best system selection approach improves over our best baseline single MT system by 1.0% absolute BLEU point on a blind test set.

[1]This paper presents work supported by the Defense Advanced Research Projects Agency (DARPA) contract No. HR0011-12-C-0014. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA.

2 Related Work

Arabic Dialect Machine Translation. Two approaches have emerged to alleviate the problem of DA-English parallel data scarcity: using MSA as a bridge language (Sawaf, 2010; Salloum and Habash, 2011; Salloum and Habash, 2013; Sajjad et al., 2013), and using crowd sourcing to acquire parallel data (Zbib et al., 2012). Sawaf (2010) and Salloum and Habash (2013) used hybrid solutions that combine rule-based algorithms and resources such as lexicons and morphological analyzers with statistical models to map DA to MSA before using MSA-to-English MT systems. Zbib et al. (2012) obtained a 1.5M word parallel corpus of DA-English using crowd sourcing. Applied on a DA test set, a system trained on their 1.5M word corpus outperformed a system that added 150M words of MSA-English data, as well as outperforming a system with oracle DA-to-MSA pivot.

In this paper we use four MT systems that translate from DA to English in different ways. Similar to Zbib et al. (2012), we use DA-English, MSA-English and DA+MSA-English systems. Our DA-English data includes the 1.5M words created by Zbib et al. (2012). Our fourth MT system uses ELISSA, the DA-to-MSA MT tool by Salloum and Habash (2013), to produce an MSA pivot.

Dialect Identification. There has been a number of efforts on dialect identification (Biadsy et al., 2009; Zaidan and Callison-Burch, 2011; Akbacak et al., 2011; Elfardy et al., 2013; Elfardy and Diab, 2013). Elfardy et al. (2013) performed token-level dialect ID by casting the problem as a code-switching problem and treating MSA and Egyptian as two different languages. They later

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 772–778,
Baltimore, Maryland, USA, June 23-25 2014.

used features from their token-level system to train a classifier that performs sentence-level dialect ID (Elfardy and Diab, 2013). In this paper, we use AIDA, the system of Elfardy and Diab (2013), to provide a variety of dialect ID features to train classifiers that select, for a given sentence, the MT system that produces the best translation.

System Selection and Combination in Machine Translation. The most popular approach to MT system combination involves building confusion networks from the outputs of different MT systems and decoding them to generate new translations (Rosti et al., 2007; Karakos et al., 2008; He et al., 2008; Xu et al., 2011). Other researchers explored the idea of re-ranking the n-best output of MT systems using different types of syntactic models (Och et al., 2004; Hasan et al., 2006; Ma and McKeown, 2013). While most researchers use target language features in training their re-rankers, others considered source language features (Ma and McKeown, 2013).

Most MT system combination work uses MT systems employing different techniques to train on the same data. However, in this paper, we use the same MT algorithms for training, tuning, and testing, but vary the training data, specifically in terms of the degree of source language dialectness. Our approach runs a classifier trained only on source language features to decide which system should translate each sentence in the test set, which means that each sentence goes through one MT system only. Since we do not combine the output of the MT systems on the phrase level, we call our approach "*system selection*" to avoid confusion.

3 Machine Translation Experiments

In this section, we present our MT experimental setup and the four baseline systems we built, and we evaluate their performance and the potential of their combination. In the next section we present and evaluate the system selection approach.

MT Tools and Settings. We use the open-source Moses toolkit (Koehn et al., 2007) to build four Arabic-English phrase-based statistical machine translation systems (SMT). Our systems use a standard phrase-based architecture. The parallel corpora are word-aligned using GIZA++ (Och and Ney, 2003). The language model for our systems is trained on English Gigaword (Graff and Cieri, 2003). We use SRILM Toolkit (Stolcke, 2002) to build a 5-gram language model with modified Kneser-Ney smoothing. Feature weights are tuned to maximize BLEU on tuning sets using Minimum Error Rate Training (Och, 2003). Results are presented in terms of BLEU (Papineni et al., 2002). All evaluation results are case *insensitive*. The English data is tokenized using simple punctuation-based rules. The MSA portion of the Arabic side is segmented according to the Arabic Treebank (ATB) tokenization scheme (Maamouri et al., 2004; Sadat and Habash, 2006) using the MADA+TOKAN morphological analyzer and tokenizer v3.1 (Roth et al., 2008), while the DA portion is ATB-tokenized with MADA-ARZ (Habash et al., 2013). The Arabic text is also Alif/Ya normalized. For more details on processing Arabic, see (Habash, 2010).

MT Train/Tune/Test Data. We have two parallel corpora. The first is a *DA-English* corpus of 5M tokenized words of Egyptian (∼3.5M) and Levantine (∼1.5M). This corpus is part of BOLT data. The second is an *MSA-English* corpus of 57M tokenized words obtained from several LDC corpora (10 times the size of the DA-English data). We work with eight standard MT test sets: three MSA sets from NIST MTEval with four references (MT06, MT08, and MT09), four Egyptian sets from LDC BOLT data with two references (EgyDevV1, EgyDevV2, EgyDevV3, and EgyTestV2), and one Levantine set from BBN (Zbib et al., 2012) with one reference which we split into LevDev and LevTest. We used MT08 and EgyDevV3 to tune SMT systems while we divided the remaining sets among classifier training data (5,562 sentences), dev (1,802 sentences) and blind test (1,804 sentences) sets to ensure each of these new sets has a variety of dialects and genres (weblog and newswire).

MT Systems. We build four MT systems.

(1) *DA-Only*. This system is trained on the DA-English data and tuned on EgyDevV3.

(2) *MSA-Only*. This system is trained on the MSA-English data and tuned on MT08.

(3) *DA+MSA*. This system is trained on the combination of both corpora (resulting in 62M tokenized[2] words on the Arabic side) and tuned on

[2]Since the *DA+MSA* system is intended for DA data and DA morphology, as far as tokenization is concerned, is more complex, we tokenized the training data with dialect awareness (DA with MADA-ARZ and MSA with MADA) since MADA-ARZ does a lot better than MADA on DA (Habash et al., 2013). Tuning and Test data, however, are tokenized by MADA-ARZ since we do not assume any knowledge of the dialect of a test sentence.

EgyDevV3.

(4) *MSA-Pivot*. This MSA-pivoting system uses Salloum and Habash (2013)'s DA-MSA MT system followed by an Arabic-English SMT system which is trained on both corpora augmented with the DA-English where the DA side is preprocessed with the same DA-MSA MT system then tokenized with MADA-ARZ. The result is 67M tokenized words on the Arabic side. EgyDevV3 was similarly preprocessed with the DA-MSA MT system and MADA-ARZ and used for tuning the system parameters. Test sets are similarly preprocessed before decoding with the SMT system.

Baseline MT System Results. We report the results of our dev set on the four MT systems we built in Table 1. The *MSA-Pivot* system produces the best singleton result among all systems. All differences in BLEU scores between the four systems are statistically significant above the 95% level. Statistical significance is computed using paired bootstrap re-sampling (Koehn, 2004).

System Name	Training Data (TD)				BLEU
	DA-En	MSA-En	DA^T-En	**TD Size**	
1. *DA-Only*	5M			5M	26.6
2. *MSA-Only*		57M		57M	32.7
3. *DA+MSA*	5M	57M		62M	33.6
4. *MSA-Pivot*	5M	57M	5M	67M	**33.9**
Oracle System Selection					**39.3**

Table 1: Results from the baseline MT systems and their oracle system selection. The training data west used in different MT systems are also indicated. DA^T (in the fourth column) is the DA part of the 5M word DA-En parallel data processed with the DA-MSA MT system.

Oracle System Selection. We also report in Table 1 an oracle system selection where we pick, for each sentence, the English translation that yields the best BLEU score. This oracle indicates that the upper bound for improvement achievable from system selection is 5.4% BLEU. Excluding different systems from the combination lowered the overall score between 0.9% and 1.8%, suggesting the systems are indeed complementary.

4 MT System Selection

The approach we take in this paper benefits from the techniques and conclusions of previous papers in that we build different MT systems similar to those discussed above but instead of trying to find which one is the best, we try to leverage the use of all of them by automatically deciding what sentences should go to which system. Our hypothesis is that these systems complement each other in interesting ways where the combination of their selections could lead to better overall performance stipulating that our approach could benefit from the strengths while avoiding the weaknesses of each individual system.

4.1 Dialect ID Binary Classification

For baseline system selection, we use the classification decision of Elfardy and Diab (2013)'s sentence-level dialect identification system to decide on the target MT system. Since the decision is binary (DA or MSA) and we have four MT systems, we considered all possible configurations and determined empirically that the best configuration is to select *MSA-Only* for the MSA tag and *MSA-Pivot* for the DA tag. We do not report other configuration results due to space restrictions.

4.2 Feature-based Four-Class Classification

For our main approach, we train a four-class classifier to predict the target MT system to select for each sentence using only source-language features. We experimented with different classifiers in the Weka Data Mining Tool (Hall et al., 2009) for training and testing our system selection approach. The best performing classifier was Naive Bayes (with Weka's default settings).

Training Data Class Labels. We run the 5,562 sentences of the classification training data through our four MT systems and produce sentence-level BLEU scores (with length penalty). We pick the name of the MT system with the highest BLEU score as the class label for that sentence. When there is a tie in BLEU scores, we pick the system label that yields better overall BLEU scores from the systems tied.

Training Data Source-Language Features. We use two sources of features extracted from untokenized sentences to train our four-class classifiers: *basic* and *extended features*.

A. Basic Features

These are the same set of features that were used by the dialect ID tool together with the class label generated by this tool.

i. Token-Level Features. These features rely on language models, MSA and Egyptian morphological analyzers and a Highly Dialectal Egyptian lexicon to decide whether each word is MSA, Egyptian, Both, or Out of Vocabulary.

ii. Perplexity Features. These are two features that measure the perplexity of a sentence against

two language models: MSA and Egyptian.

iii. Meta Features. Features that do not directly relate to the dialectalness of words in the given sentence but rather estimate how informal the sentence is and include: percentage of tokens, punctuation, and Latin words, number of tokens, average word length, whether the sentence has any words that have word-lengthening effects or not, whether the sentence has any diacritized words or not, whether the sentence has emoticons or not, whether the sentence has consecutive repeated punctuation or not, whether the sentence has a question mark or not, and whether the sentence has an exclamation mark or not.

iv. The Dialect-Class Feature. We run the sentence through the Dialect ID binary classifier and we use the predicted class label (DA or MSA) as a feature in our system. Since the Dialect ID system was trained on a different data set, we think its decision may provide additional information to our classifiers.

B. Extended Features

We add features extracted from two sources.

i. MSA-Pivoting Features. Salloum and Habash (2013) DA-MSA MT system produces intermediate files used for diagnosis or debugging purposes. We exploit one file in which the system identifies (or, "selects") dialectal words and phrases that need to be translated to MSA. We extract confidence indicating features. These features are: sentence length (in words), percentage of selected words and phrases, number of selected words, number of selected phrases, number of words morphologically selected as dialectal by a mainly Levantine morphological analyzer, number of words selected as dialectal by the tool's DA-MSA lexicons, number of OOV words against the *MSA-Pivot* system training data, number of words in the sentences that appeared less than 5 times in the training data, number of words in the sentences that appeared between 5 and 10 times in the training data, number of words in the sentences that appeared between 10 and 15 times in the training data, number of words that have spelling errors and corrected by this tool (e.g., word-lengthening), number of punctuation marks, and number of words that are written in Latin script.

ii. MT Training Data Source-Side LM Perplexity Features. The second set of features uses perplexity against language models built from the source-side of the training data of each of the four baseline systems. These four features may tell the classifier which system is more suitable to translate a given sentence.

4.3 System Selection Evaluation

Development Set. The first part of Table 2 repeats the best baseline system and the four-system oracle combination from Table 1 for convenience. The third row shows the result of running our system selection baseline that uses the Dialect ID binary decision on the Dev set sentences to decide on the target MT system. It improves over the best single system baseline (*MSA-Pivot*) by a statistically significant 0.5% BLEU. Crucially, we should note that this is a deterministic process.

System	BLEU	Diff.
Best Single MT System Baseline	**33.9**	0.0
Oracle	*39.3*	*5.4*
Dialect ID Binary Selection Baseline	**34.4**	0.5
Four-Class Classification		
Basic Features	35.1	1.2
Extended Features	34.8	0.9
Basic + Extended Features	**35.2**	1.3

Table 2: Results of baselines and system selection systems on the Dev set in terms of BLEU. The best single MT system baseline is *MSA-Pivot*.

The second part of Table 2 shows the results of our four-class Naive Bayes classifiers trained on the classification training data we created. The first column shows the source of sentence level features employed. As mentioned earlier, we use the Basic features alone, the Extended features alone, and then their combination. The classifier that uses both feature sources simultaneously as feature vectors is our best performer. It improves over our best baseline single MT system by 1.3% BLEU and over the Dialect ID Binary Classification system selection baseline by 0.8% BLEU. Improvements are statistically significant.

System	BLEU	Diff.
DA-Only	26.6	
MSA-Only	30.7	
DA+MSA	32.4	
MSA-Pivot	**32.5**	
Four-System Oracle Combination	*38.0*	*5.5*
Best Dialect ID Binary Classifier	32.9	0.4
Best Classifier: Basic + Extended Features	**33.5**	**1.0**

Table 3: Results of baselines and system selection systems on the Blind test set in terms of BLEU.

Blind Test Set. Table 3 shows the results on our Blind Test set. The first part of the table shows the results of our four baseline MT systems. The systems have the same rank as on the Dev set and

System	All	Dialect	MSA
DA-Only	26.6	19.3	33.2
MSA-Only	32.7	14.7	**50.0**
DA+MSA	33.6	19.4	46.3
MSA-Pivot	33.9	**19.6**	46.4
Four-System Oracle Combination	*39.3*	*24.4*	*52.1*
Best Performing Classifier	**35.2**	**19.8**	**50.0**

Table 4: Dialect breakdown of performance on the Dev set for our best performing classifier against our four baselines and their oracle combination. Our classifier does not know of these subsets, it runs on the set as a whole; therefore, we repeat its results in the second column for convenience.

MSA-Pivot is also the best performer. The differences in BLEU are statistically significant. The second part shows the four-system oracle combination which shows a 5.5% BLEU upper bound on improvements. The third part shows the results of the Dialect ID Binary Classification which improves by 0.4% BLEU. The last row shows the four-class classifier results which improves by 1.0% BLEU over the best single MT system baseline and by 0.6% BLEU over the Dialect ID Binary Classification. Results on the Blind Test set are consistent with the Dev set results.

5 Discussion and Error Analysis

DA versus MSA Performance. In Table 4, column **All** illustrates the results over the entire Dev set, while columns **DA** and **MSA** show system performance on the DA and MSA subsets of the Dev set, respectively. The best single baseline MT system for DA is *MSA-Pivot* has a large room for improvement given the oracle upper bound (4.8% BLEU absolute). However, our best system selection approach improves over *MSA-Pivot* by a small margin of 0.2% BLEU absolute only, albeit a statistically significant improvement. The MSA column oracle shows a smaller improvement of 2.1% BLEU absolute over the best single *MSA-Only* MT system. Furthermore, when translating MSA with our best system selection performer we get the same results as the best baseline MT system for MSA even though our system does not know the dialect of the sentences a priori. If we consider the breakdown of the performance in our best overall (33.9% BLEU) single baseline MT system (*MSA-Pivot*), we observe that the performance on MSA is about 3.6% absolute BLEU points below our best results; this suggests that most of the system selection gain over the best single baseline is on MSA selection.

Manual Error Analysis. We performed manual error analysis on a Dev set sample of 250 sentences distributed among the different dialects and genres. Our best performing classifier selected the best system in 48% of the DA cases and 52% of the MSA cases. We did a detailed manual error analysis for the cases where the classifier failed to predict the best MT system. The sources of errors we found cover 89% of the cases. In 21% of the error cases, our classifier predicted a better translation than the one considered gold by BLEU due to BLEU bias, e.g., severe sentence-level length penalty due to an extra punctuation in a short sentence. Also, 3% of errors are due to bad references, e.g., a dialectal sentence in an MSA set that the human translators did not understand.

A group of error sources resulted from MSA sentences classified correctly as *MSA-Only*; however, one of the other three systems produced better translations for two reasons. First, since the MSA training data is from an older time span than the DA data, 10% of errors are due to MSA sentences that use recent terminology (e.g., Egyptian revolution 2011: places, politicians, etc.) that appear in the DA training data. Also, web writing styles in MSA sentences such as blog style (e.g., rhetorical questions), blog punctuation marks (e.g., "..", "???!!"), and formal MSA forum greetings resulted in 23%, 16%, and 6% of the cases, respectively.

Finally, in 10% of the cases our classifier is confused by a code-switched sentence, e.g., a dialectal proverb in an MSA sentence or a weak MSA literal translation of dialectal words and phrases. Some of these cases may be solved by adding more features to our classifier, e.g., blog style writing features, while others need a radical change to our technique such as word and phrase level dialect identification for MT system combination of code-switched sentences.

6 Conclusion and Future Work

We presented a sentence-level classification approach for MT system selection for diglossic languages. We got a 1.0% BLEU improvement over the best baseline single MT system. In the future we plan to add more training data to see the effect on the accuracy of system selection. We plan to give different weights to different training examples based on the drop in BLEU score the example can cause if classified incorrectly. We also plan to explore confusion-network combination and reranking techniques based on target language features.

References

Murat Akbacak, Dimitra Vergyri, Andreas Stolcke, Nicolas Scheffer, and Arindam Mandal. 2011. Effective arabic dialect classification using diverse phonotactic models. In *INTERSPEECH*, volume 11, pages 737–740.

Fadi Biadsy, Julia Hirschberg, and Nizar Habash. 2009. Spoken arabic dialect identification using phonotactic modeling. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages at the meeting of the European Association for Computational Linguistics (EACL), Athens, Greece.*

Heba Elfardy and Mona Diab. 2013. Sentence Level Dialect Identification in Arabic. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (ACL-13)*, Sofia, Bulgaria.

Heba Elfardy, Mohamed Al-Badrashiny, and Mona Diab. 2013. Code switch point detection in arabic. In *Proceedings of the 18th International Conference on Application of Natural Language to Information Systems (NLDB2013)*, MediaCity, UK.

Charles F Ferguson. 1959. Diglossia. *Word*, 15(2):325–340.

David Graff and Christopher Cieri. 2003. English Gigaword, LDC Catalog No.: LDC2003T05. Linguistic Data Consortium, University of Pennsylvania.

Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh. 2013. Morphological Analysis and Disambiguation for Dialectal Arabic. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Atlanta, GA.

Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18.

S. Hasan, O. Bender, and H. Ney. 2006. Reranking translation hypotheses using structural properties. In *EACL'06 Workshop on Learning Structured Information in Natural Language Applications*.

Xiaodong He, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore. 2008. Indirect-hmm-based hypothesis alignment for combining outputs from machine translation systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 98–107. Association for Computational Linguistics.

Damianos Karakos, Jason Eisner, Sanjeev Khudanpur, and Markus Dreyer. 2008. Machine translation system combination using itg-based alignments. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 81–84. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Christopher Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christopher Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Wei-Yun Ma and Kathleen McKeown. 2013. Using a supertagged dependency language model to select a good translation in system combination. In *Proceedings of NAACL-HLT*, pages 433–438.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Franz Josef Och. 2004. A smorgasbord of features for statistical machine translation. In *Meeting of the North American chapter of the Association for Computational Linguistics.*

Franz Josef Och. 2003. Minimum Error Rate Training for Statistical Machine Translation. In *Proceedings of the 41st Annual Conference of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

Antti-Veikko Rosti, Spyros Matsoukas, and Richard Schwartz. 2007. Improved word-level system combination for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 312–319, Prague, Czech Republic, June. Association for Computational Linguistics.

Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic Morphological Tagging, Diacritization, and Lemmatization Using Lexeme Models and Feature Ranking. In *Proceedings of ACL-08: HLT, Short Papers*, pages 117–120, Columbus, Ohio.

Fatiha Sadat and Nizar Habash. 2006. Combination of Arabic preprocessing schemes for statistical ma-

chine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1–8, Sydney, Australia, July. Association for Computational Linguistics.

Hassan Sajjad, Kareem Darwish, and Yonatan Belinkov. 2013. Translating dialectal arabic to english. In *The 51st Annual Meeting of the Association for Computational Linguistics - Short Papers (ACL Short Papers 2013), Sofia, Bulgaria*.

Wael Salloum and Nizar 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*, pages 10–21, Edinburgh, Scotland.

Wael Salloum and Nizar Habash. 2013. Dialectal Arabic to English Machine Translation: Pivoting through Modern Standard Arabic. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Atlanta, GA.

Hassan Sawaf. 2010. Arabic dialect handling in hybrid machine translation. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*, Denver, Colorado.

Andreas Stolcke. 2002. SRILM an Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*.

Daguang Xu, Yuan Cao, and Damianos Karakos. 2011. Description of the jhu system combination scheme for wmt 2011. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 171–176. Association for Computational Linguistics.

Omar F Zaidan and Chris Callison-Burch. 2011. The arabic online commentary dataset: an annotated dataset of informal arabic with high dialectal content. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-11)*, pages 37–41.

Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F. Zaidan, and Chris Callison-Burch. 2012. Machine Translation of Arabic Dialects. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 49–59, Montréal, Canada, June. Association for Computational Linguistics.