Arabic Diacritization in the Context of Statistical Machine Translation

Mona Diab, Mahmoud Ghoneim, Nizar Habash
Center for Computational Learning Systems
Columbia University
475 Riverside Drive, New York, NY 10115
mdiab,mghoneim,habash@cs.columbia.edu

Abstract

Diacritics in Arabic are optional orthographic symbols typically representing short vowels. Most Arabic text is underspecified for diacritics. However, we do observe partial diacritization depending on genre and domain. In this paper, we investigate the impact of Arabic diacritization on statistical machine translation (SMT). We define several diacritization schemes ranging from full to partial diacritization. We explore the impact of the defined schemes on SMT in two different modes which tease apart the effect of diacritization on the alignment and its consequences on decoding. Our results show that none of the partial diacritization schemes significantly varies in performance from the no-diacritization baseline despite the increase in the number of types in the data. However, a full diacritization scheme performs significantly worse than no diacritization. Crucially, our research suggests that the SMT performance is positively correlated with the increase in the number of tokens correctly affected by a diacritization scheme and the high F-score of the automatic assignment of the particular diacritic.

1 Introduction

Modern standard Arabic (MSA) is written with an orthography that includes *optional* diacritical marks (henceforth, diacritics). Diacritics are extremely useful for readability and understanding. Their absence in Arabic text adds another layer of lexical and morphological ambiguity. Naturally occurring Arabic text has some percentage of these diacritics present depending on genre and domain. They are there to aid the reader disambiguate the text or simply to articulate it correctly. For instance, religious text such as the Quran is fully diacritized to minimize the chances of reciting it incorrectly. So are childrens' educational texts. Classical poetry tends to be diacritized as well. However, news text and other genre are sparsely diacritized (e.g., around 1.5% of tokens in the `United Nations` Arabic corpus bear at least one diacritic).

In speech technology, full diacritization has improved state-of-the-art Arabic automatic speech recognition (ASR) systems especially in cross dialectal speech modeling (Kirchhoff and Vergyri, 2005). However, no studies have investigated the optimal level of diacritization sufficient to yield the best ASR results. To date, no systematic study of the impact of diacritization on other NLP applications has been reported. Guided by the utility of diacritization for readability and its presence in naturally occurring text, we introduce the notion of partial diacritization for natural language processing (NLP). We define several diacritization schemes ranging from full diacritization to partial diacritization. The schemes vary in representation from the inflectional to the lexical. We also investigate the impact of both partial and full diacritization on statistical machine translation (SMT). In the SMT pipeline, we consider two different modes that tease apart the effect diacritization has on alignment and its consequences on decoding. This investigation falls within approaches to preprocessing source language text, which typically attempt to reduce word sparsity through morphological preprocessing and orthographic normalization as a means of improving translation quality. However, our work differs from previous approaches in that it maintains the same preprocessing tokenization throughout all the experiments and only varies some of the word forms, effectively increasing the word types in our vocabulary adding to the complexity of our text rather than reducing it. Our results show that none of the partial diacritization schemes significantly varies in performance from the no-diacritization baseline despite the increase in the number of types in the data. However, a full diacritization scheme performs significantly worse than no diacritization. Crucially, our research suggests that the SMT performance is positively correlated with the number of tokens affected by a diacritization scheme coupled with the high F-score of the specific automatic diacritic assignment. Hence, the larger the size of the accurately affected tokens by the application of a diacritization scheme, the higher the yielded SMT score.

This paper is organized as follows: Section 2 reviews related work; Section 3 presents Arabic diacritic linguistic facts; Section 4 describes the experimental setup; Section 5 presents the experimental results and discussion. Finally, we conclude with Section 6 with a possible future directions.

2 Related Work

Arabic Diacritization Much work has been done on Arabic diacritization (*aka* vowelization, diacritic/vowel restoration). We refer to the literature review in (Zi-

touni et al., 2006) for an excellent general review. Zitouni et al. (2006) use a maximum entropy classifier to assign diacritics to the letters of each word. Vergyri and Kirchhoff (2004) and Ananthakrishnan et al. (2005) present work on diacritization targeted toward improving ASR. Both systems exploit the Buckwalter Arabic Morphological Analysis (BAMA) system. Vergyri and Kirchhoff (2004) use a single tagger to select amongst the diacritized analyses; whereas Ananthakrishnan et al. (2005) use a language-modeling approach. Habash and Rambow (2007) introduce a system MADA that also uses BAMA, but they use 14 taggers and a lexeme-based language model. MADA is the best performing system to date achieving a word error rate of 14.9% and a diacritic error rate of 4.8%. We use MADA in implementing our diacritization schemes.[1]

Diacritization as Preprocessing Relevant research on the effect of morphological preprocessing on SMT quality focuses on morphologically rich languages such as German (Nießen and Ney, 2004); Spanish, Catalan, and Serbian (Popović and Ney, 2004); and Czech (Goldwater and McClosky, 2005). They all studied the effects of various kinds of tokenization, lemmatization and POS tagging and show a positive impact on SMT quality. Specifically for Arabic, Lee (2004) investigated the use of automatic alignment of POS tagged English and affix-stem segmented Arabic to determine appropriate tokenizations. Habash and Sadat (2006) investigated a wide set of possible tokenization schemes for Arabic. Results from both research investigations show that morphological preprocessing helps, but only for the smaller corpora. As token size increases, the benefits diminish. Habash and Sadat (2006) showed that a more linguistically informed approach to tokenization yields better results than simple heuristics. The work presented here differs from previous approaches to preprocessing in that we maintain the same preprocessing tokenization throughout all our experiments and only vary some of the word forms, effectively increasing the word types in our vocabulary.

Diacritization in Natural Language Preprocessing Regardless of the level of diacritization, to date, there have not been any systematic investigations of the impact of different types of Arabic diacritization on SMT (or any other NLP application, for that matter). One exception is the work of Kirchhoff and Vergyri (2005) on ASR. They show that full diacritization improves state-of-the-art Arabic ASR, in particular, in the context of cross dialectal modeling.

3 Arabic Diacritics: A Linguistic Description

Arabic script consists of two classes of symbols: letters and diacritics. Letters are always written whereas diacritics are optional: written Arabic may be fully diacritized, may have some diacritics, or may be entirely undiacritized. In this section, Arabic diacritics are described in terms of their form and function before discussing how they are actually used in practice.

3.1 Arabic Diacritic Forms

There are three types of diacritics: vowel, nunation, and *shadda* (gemination). Vowel diacritics represent Arabic's three short vowels and a diacritic indicating the absence of any vowel. The following are the four vowel-diacritics exemplified in conjunction with the letter ب *b*[2]: بَ *ba* (*fatha*), بُ *bu* (*damma*), بِ *bi* (*kasra*), and بْ *bo* (no vowel *aka sukuun*). Nunation diacritics can only occur in word final positions in nominals (nouns, adjectives and adverbs). They indicate a short vowel followed by an unwritten *n* sound: باً[3] *bAF*[4], بٌ *bN* and بٍ *bK*. Nunation is an indicator of nominal indefiniteness. The *shadda* is a consonant doubling diacritic: بّ *b*∼ (/bb/). The *shadda* can combine with vowel or nunation diacritics: بُّ *b*∼*u* or بٌّ *b*∼*N*.

Additional diacritical marks in Arabic include the *hamza*, which appears in conjunction with a small number of letters (e.g., أ, إ, آ, ؤ , ئ). Since most Arabic encodings do not count the hamza a diacritic, but rather a part of the letter (like the dot on the lower-case Roman *i* or under the Arabic *b*: ب), we do not consider it here as part of the diacritic set.

3.2 Arabic Diacritic Functions

Functionally, diacritics can be split into two different kinds: **lexical** diacritics and **inflectional** diacritics.

3.2.1 Lexical Diacritics

Lexical diacritics distinguish between two lexemes.[5] We refer to a lexeme with its **citation form**; Arabic lexeme citation forms are third masculine singular perfective for verbs and masculine singular (or feminine singular if no masculine is possible) for nouns and adjectives. For example, the diacritization difference between the lexemes كَاتِب *kAtib* 'writer' and كَاتَب *kAtab* 'to correspond' distinguishes between the meanings of the word (lexical disambiguation) rather than their inflections. Any of the diacritics may be used to mark lexical variation. A common example with the *shadda* (gemination) diacritic is the distinction between Form I and Form II of Arabic

[1] The specific version of MADA we use does not include the lexeme language models.

[2] We use the Buckwalter transliteration to romanize Arabic examples (Buckwalter, 2002).

[3] Arabic orthography calls for adding a silent Alif (ا) in conjunction with ـً in words ending with a consonant.

[4] Buckwalter's transliteration symbols for nunation, *F*, *N* and *K*, are pronounced */an/*, */un/* and */in/*, respectively.

[5] A **lexeme** is an abstraction over inflected word forms which groups together all those word forms that differ only in terms of one of the inflectional morphological categories such as number, gender, aspect, voice, etc. A **lemma** is a citation form.

verb derivation. Form II, indicates, in most cases, added causativity to the Form I meaning. Form II is marked by doubling the second radical of the root used in Form I: اكَل *Akal* 'ate' versus اكّل *Ak∼al* 'fed'. Generally speaking, however, deriving word meaning through lexical diacritic placement is largely unpredictable and they are not specifically associated with any particular part of speech.

3.2.2 Inflectional Diacritics

Inflectional diacritics distinguish different inflected forms of the *same* lexeme. For instance, the final diacritics in كِتابُ *kitAbu* 'book [nominative]' and كِتابَ *kitAba* 'book [accusative]' distinguish the morphological case of 'book' (e.g., whether the word is subject or object of a verb). Additional inflectional features marked through diacritic change, in addition to case, include voice, mood, and definiteness. Inflectional diacritics are predictable in their positional placement in a word. Moreover, they are associated with certain parts of speech.

3.3 Arabic Diacritics in Practice

Typically, Arabic text is undiacritized except in religious texts and children educational texts. Some diacritics are indicated in modern written Arabic to help readers disambiguate certain words. In the Penn Arabic Treebank (ATB) III ver.2 (Maamouri et al., 2004), 1.6% of all word tokens have some diacritic(s) occurring naturally in the text. Among these, the most common diacritics are the nunation diacritics (*F, K* and *N*), accounting for 73.4% of the naturally occurring diacritics in the ATB. Majority (96%) of these nunation markers are used inflectionally to mark nominals (nouns, adjectives, proper nouns) indicating case assignment: *F* diacritic marks accusative case, as in كتابًا *kitAbAF* 'a book [accusative]', *K* diacritic marks genitive case, and *N* diacritic marks nominative case.

The second most frequent diacritic is the *shadda* or gemination diacritic. It comprises 20.8% of the naturally occurring diacritics in the ATB. It occurs 56.7% of the time with verbs, where it is often used to distinguish between the derivationally related Form I and Form II as explained earlier. In some cases it is important to explicitly mark this distinction since the context in both the geminated and the non-geminated forms could be the same. The rest of the *shadda* distribution is as follows: 33% occur in nominals, and 9.8% occur in function words such as pronouns and prepositions. Hence, in practice, the *shadda* diacritic is a lexical diacritic that has a direct impact on semantic disambiguation.

The third most frequent diacritic is the *damma* (*u*). It constitutes 3% of the naturally occurring diacritics. The majority (74.4%) of its usage is for designating the passive form of verbs, e.g., كُتب *kutib* 'was written' versus كتب *katab* 'wrote'. The need often arises to explicitly distinguish both forms since Arabic allows for both subject-verb-object (SVO) and verb-subject-object (VSO) orders. Determining whether an un-explicitly marked verb is passive or active may require reading ahead to figure out the number of noun phrases following the verb in the VSO order. Explicitly indicating the single distinctive diacritic marking passivization saves the reader a garden-path reading. Moreover, the *damma* marks nominative case for nominals, e.g., الجدرانُ *AljdrAnu* 'the walls'; As well as it marks indicative mood for verbs, e.g., يعتقدُ *yEtqdu* 'he believes'.

The *fatha* (*a*) and *kasra* (*i*) diacritics are less common. They are both used word finally as an inflectional diacritic to mark case in nominals. Moreover, the *fatha* is used to inflect verbs for subjunctive mood. Both *kasra* and *fatha* could be used for semantic disambiguation anywhere in the word. They make up a small portion of the naturally occurring diacritics, roughly 2%. The majority (95%) of them, however, are inflectional markers for case and mood.

Finally, accounting for less than 1% of the diacritics in natural text is the *sukuun*, which is used to explicitly mark the absence of a vowel. The *sukuun* diacritic (*o*) appears word medially or word finally only. When used lexically, it marks the end of consonant-vowel-consonant syllables, e.g., عبْر *Ebor* 'across' (as opposed to عبَر *Ebar* 'he crossed'). The majority (99.4%) of the *sukuun* occurrences in the ATB are lexical. But less often, as observed in the ATB, (0.6%), the *sukuun* marks jussive mood for verbs, e.g., يردْ *yrdo* 'want [jussive]'.

To our knowledge, the naturally occurring diacritics are never used in diacritization restoration. We do not make use of them either. However, their natural distribution and functional use guide our definition of the partial diacritization schemes that are used to mark specific linguistic phenomena in our experiments.

4 Experimental Setup

For purposes of our investigation, we define different diacritization schemes (DS) highlighting the different linguistic phenomena observed in natural text. We preprocess the Arabic source text in the context of phrase-based SMT using these different DSs. We also explore two different alignment modes, where a diacritization scheme is either used or not used for alignment purposes.

4.1 Diacritization Schemes

We define six different diacritization schemes that are inspired by our observations of the relevant naturally occurring diacritics. For all of the schemes, we use the MADA system for Arabic disambiguation (Habash and Rambow, 2005; Habash and Rambow, 2007). The fully disambiguated form of a word is marked for all its morphological features and is also fully diacritized. For each scheme, we selectively delete diacritics that are irrelevant to that scheme given the scheme's defined features of interest. The following are the defined diacritization schemes:

• **NONE**: This is the baseline DS, in which all diacritics are absent, including the naturally occurring ones;

• **PASS**: This is an inflectional DS which marks the verb passivization (*u*) only. It is only used on verbs marked by MADA as passive and where the (*u*) is explicitly present;[6]

• **C-M**: This is an inflectional DS encoding both case and mood. The (*a, i, u, F, K, N*) mark CASE on nominals. The (*a, o, u*) diacritics mark subjunctive, jussive and indicative MOOD on verbs, respectively. Only words marked by MADA with an explicit case or mood feature are diacritized with the relevant diacritic;

• **GEM**: This is a lexical DS which marks the words in the data with the *shadda* diacritic (∼). Only words that have a gemination diacritic, in the underlying lemma form, as deemed by MADA, are explicitly marked with the (∼) diacritic;

• **SUK**: This is a lexical DS which marks words in the data with the *sukuun* diacritic (*o*) (the no vowel marker). Only words that have a *sukuun* in the underlying lemma form, as deemed by MADA, are marked with the (*o*) diacritic. Hence, the inflectional diacritic (*o*) (marking case or mood) is not used in this scheme;

• **FULL**: This DS fully specifies all the diacritics in a word as produced by the MADA system.

DS	Example	Transliteration
NONE	سترمم الجدران	*strmm AljdrAn*
PASS	سُترمم الجدران	*sturmm AljdrAn*
C-M	سترممُ الجدرانُ	*strmmu AljdrAnu*
GEM	سترمّم الجدران	*strm∼m AljdrAn*
SUK	سترمم الجدْران	*strmm AljdorAn*
FULL	سَتُرَمِّمُ الجِدُرانُ	*saturam∼mu AljidorAnu*

Table 1: Contrasting Diacritization Schemes

In Table 1, the same sentence 'the walls will be restored' is illustrated using the different diacritization schemes explored in our experiments. For NONE, we note the absence of any diacritic, which is contrasted with FULL where all the diacritics are explicitly present. PASS only exhibits the *damma* or (*u*) passivization diacritic. C-M exhibits the indicative mood marking the verb سترممُ *strmmu* 'will be restored' with the verb final *damma* (*u*) and الجدرانُ *AljdrAnu* 'the walls' is marked with the nominative case diacritic (*u*) indicating that it is the subject. In GEM, we preserve only the gemination lexical diacritic (∼). Likewise, for SUK, only the *sukuun* diacritic (*o*) in الجدْران *AljdorAn* is preserved.

It is worth noting that using the lexical DSs, GEM and SUK, is, in effect, explicitly marking different senses of the same underlying undiacritized forms, hence, applying such lexical schemes constitutes a simple form of word sense disambiguation on the orthographic level.

[6]There are some verbs that passivize without the use of the *damma* (*u*) such as in the verb قيل *qyl* 'was said'.

4.2 Alignment Strategies

We investigate two different alignment strategies that we expect to respond differently to change in DS. In the first strategy (ALIGNBASIC), we run word alignment with Giza++ (Och and Ney, 2003) with the training data marked for one of the diacritization schemes. In the second strategy (ALIGNREMAP), we always align with the NONE scheme and then map the source text in the alignments to the text with the DS of interest. The intuition for ALIGNREMAP is that word alignment with any of the DS, apart from NONE, will suffer from sparsity issues due to the increase in the number of word types. NONE DS is more robust with respect to alignments since the number of word types is smaller (relative to the other DS) over the whole corpus. By mapping the NONE alignments to the DS of interest in an ALIGNREMAP strategy, we get the benefits of augmenting the phrase tables with more phrases than those present in ALIGNBASIC for that same DS circumventing the sparseness problem introduced by the diacritization scheme at alignment time in ALIGNBASIC. For example, in our setup, comparing the phrase tables for NONE and FULL-ALIGNBASIC, the size of the phrase table expands by 6.66%. Also the ambiguity decreases - the number of source phrases corresponding to a target phrase - decreases from 1.426 for NONE to 1.407 for FULL-ALIGNBASIC. However, for FULL-ALIGNREMAP, an increase of 3.63% in the size of the phrase table is observed, corresponding to 1.410 ambiguity (contrasted against 1.426 for NONE).

For both modes, ALIGNBASIC and ALIGNREMAP, the phrase table extraction and decoding proceeds in the typical phrase-based SMT manner. Crucially, the test and tuning data is marked with the same DS used for training data in ALIGNBASIC or ALIGNREMAP.

5 Experiments

5.1 Experimental Data

We use an Arabic-English parallel news wire corpus of about 5 million words for the translation model training data. The parallel text includes: Arabic News (LDC2004T17), eTIRR (LDC2004E72), Arabic Treebank and its translation (LDC2005E46), and Ummah (LDC2004T18). We create the English language model from the English side of the training data together with 340 million words from the English Gigaword corpus (LDC2005T12).

English preprocessing simply includes lower-casing, separating punctuation from words, and splitting off "'s". The same preprocessing is used on the English data for all experiments. Arabic preprocessing is varied only with respect to diacritization. We assume the same tokenization scheme (ATB style clitic tokenization) for all the Arabic data. The decoding weight optimization is performed using a set of 200 sentences from the 2002 NIST

		NONE	PASS	C-M	GEM	SUK	FULL
MT03	ALIGNBASIC	0.4495	0.4507	0.4354	0.4341	0.4482	0.4293
	ALIGNREMAP		0.4538	0.4411	0.4444	0.4536	0.4307
MT04	ALIGNBASIC	0.4195	0.4202	0.3977	0.4128	0.4173	0.3898
	ALIGNREMAP		0.4141	0.4047	0.4059	0.4195	0.3938
MT05	ALIGNBASIC	0.4389	0.4416	0.4217	0.4310	0.4410	0.4177
	ALIGNREMAP		0.4389	0.4290	0.4304	0.4392	0.4183

Table 2: BLEU score results obtained for all the diacritization schemes in both alignment strategies on 3 different test sets, MT03, MT04 and MT05

MT evaluation test set (MT02). We report results on the 2003, 2004 and 2005 NIST MT evaluation test sets.

5.2 SMT System

In all our experiments, we use an off-the-shelf phrase-based SMT system, Pharaoh (Koehn, 2004). Trigram language models are implemented using the SRILM toolkit (Stolcke, 2002). Decoding weights are optimized using Och's algorithm (Och, 2003). The weights are optimized over the BLEU metric (Papineni et al., 2002), which is the evaluation metric we use.[7] We use the BLEU metric in a mode insensitive to casing.

For each of the diacritization schemes described above, we train two systems per each alignment strategy. The results are described in the next section.

5.3 Results and Discussion

Table 2 illustrates the BLEU scores obtained for the different diacritization schemes (DS) and the different alignment strategies ALIGNBASIC and ALIGNREMAP. From Table 2, the worst results are those obtained with the FULL condition across the three evaluation sets for both alignment strategies ALIGNBASIC and ALIGNREMAP. C-M is the second worse performing condition. The differences between NONE and C-M are at least one BLEU point for both alignment strategies and all test sets. All PASS conditions outperform NONE for all test sets except PASS-ALIGNREMAP MT04. The differences between the PASS conditions and the NONE condition are not statistically signifi cant, however. SUK-ALIGNREMAP conditions outperform NONE for all test sets (albeit, not statistically signifi cant).

We note a consistent, anecdotal nonetheless, improvement from ALIGNBASIC to ALIGNREMAP for the three test sets in diacritization schemes FULL and C-M. We also note a slight improvement in the MT03 test set for PASS, GEM and SUK from ALIGNBASIC to ALIGNREMAP. Moreover, SUK's performance increases from 0.4173 to 0.4195 from ALIGNBASIC to ALIGNREMAP in the MT04 test set. However, the rest of the conditions yield worse results, albeit not statistically signifi - cant, for ALIGNREMAP when compared to ALIGNBASIC, for MT04 and MT05.

The overall results for MT03 are the best for all conditions and all alignment strategies indicating that the MT03 set is the easiest. MT04 set yields the worst BLEU scores including the scores for the baseline, indicating that it is the hardest test set. The MT04 test set has the largest disparity from the newswire training data in genre and domain compared to the MT03 and MT05 test sets.

We experimented with different sizes of the training data to measure the impact of diacritization in the absence of large amounts of training data (in the same spirit of previous work on morphological preprocessing, *see Section 2*). We observe the same trends in the results for both alignment modes and all the test sets when using only 10% and 50% of the training data.

The overall results obtained suggest that adding diacritization does not change the behavior of the SMT system much from the baseline (NONE). And in the case of adding too many diacritics such as in FULL and C-M, the results are signifi cantly worse than NONE. It is worth noting that the results are expected to be worse for FULL and C-M since they each signifi cantly increase the number of types in the data by 60% and 52%, respectively. Moreover, FULL and C-M perform at a relatively low MADA F-score. Table 3 illustrates the percentage increase in number of types over NONE for each of the diacritization schemes. The maximum decrease in performance in FULL is by 2.97 BLEU points and for C-M the decrease is by 2.18 BLEU points.

DS	Type Increase	MADA F-score
PASS	1.3%	62.0%
C-M	52.0%	79.4%
GEM	1.3%	95.6%
SUK	3.9%	96.6%
FULL	60.0%	82.5%

Table 3: Percentage increase in type size for each of the diacritization schemes over the baseline NONE and the affected MADA F-score

Table 3 also reflects the performance of the MADA system in the form of the affected F-score. The affected

[7] We are aware of the caveats of using the BLEU metric as our evaluation metric and we intend to run a more comprehensive set of evaluations for follow on work (Callison-Burch et al., 2006).

F-score measures the quality of the specific diacritization scheme. In Table 3, MADA's performance was measured against the ATB gold data to determine if the specific diacritic is rendered correctly. MADA's overall diacritic error rate is around 5%, however, when we examine its performance in terms of the partial diacritization task, the results are significantly worse. This suggests perhaps that a future direction to diacritization research should adopt an F-score-like metric rather than accuracy. In Table 3, Comparing the MADA F-scores for GEM and PASS, where both have the same type size increase of 1.3% over NONE, GEM yields a MADA F-score of 95.6% while PASS yields a MADA F-score of 62%, but PASS outperforms GEM in all conditions as illustrated in Table 2. This could be explained by the fact that only 0.4% of the tokens are affected by the passivization diacritization scheme PASS while 16.85% of the tokens are affected by GEM, suggesting that GEM is more sensitive to the MADA error rate. On the other hand, in spite of the fact that SUK has more word types than both NONE and PASS, it is either comparable or outperforms them both for the three test sets, in SUK-ALIGNREMAP. Moreover, of all the diacritization schemes, it is the closest one to the highest performing scheme, be it the baseline NONE or PASS in SUK-ALIGNBASIC. It is also worth noting that SUK for both modes significantly outperforms GEM in both modes for all three test sets (for example, in MT03 SUK-ALIGNBASIC yields 0.4482 vs GEM-ALIGNBASIC yields 0.4354). This is interesting in light of the fact that the type increase for SUK is 3.9% compared to GEM which only has a type increase of 1.3% and their MADA F-scores are quite similar (SUK is 96.6% and GEM is 95.6%). However, the number of tokens correctly affected by the SUK diacritization scheme is significantly higher than those affected by the GEM diacritization scheme (25.3% vs. 16.85%, respectively), as well as the type increase from SUK to GEM is at least doubled. This suggests that there is a positive correlation between the quality of the automatic diacritization system, type increase and the number of affected tokens and the BLEU score. Hence, the better the quality of the MADA output for a specific scheme, and the higher the number of correctly affected tokens, the better the performance of the SMT system.

It is worth noting that the overall type increase in the lexical DSs, GEM and SUK, is relatively small which is an artifact of the lack of variation in genre and domain in the training data, i.e., many of the naturally lexically polysemous words in the language that could have the *sukuun* diacritic for the SUK DS or the *shadda* diacritic for the GEM DS, will either be with or without the diacritic in the training data. Hence, a more diversified training set in terms of genre and domain should really be the test for these fine tuned types of partial diacritization schemes.

In an attempt to understand the performance of the system better, we observe the out of vocabulary (OOV) rates between the source test and source training data within a certain scheme. Table 4 presents the OOV rates for the different schemes for the different test sets. C-M and FULL have the highest OOV rates across all the evaluation sets. Since, C-M and FULL are also the worst performing DSs, this confirms the known inverse correlation between SMT performance and OOV rate. However, SUK has worse OOV rates than NONE across the three test sets, but the differences in yielded BLEU scores is insignificant. More interestingly however, SUK has the same or worse OOV rates than GEM across the three test sets, yet SUK is close to significance in outperforming GEM on all test sets and on all conditions. This strongly suggests that the sheer size of tokens affected by the SUK DS (>25%) coupled with the increase in word types (3.9% for SUK vs. 1.3% for GEM) and the high MADA F-score on SUK scheme make it robust enough to the relatively high OOV rate when compared to GEM.

DS	MT03	MT04	MT05
NONE	1.14%	1.54%	1.64%
PASS	1.14%	1.57%	1.66%
C-M	1.49%	2.32%	2.09%
GEM	1.17%	1.57%	1.66%
SUK	1.17%	1.62%	1.68%
FULL	1.57%	2.44%	2.16%

Table 4: OOV rate for all DS in ALIGNBASIC for MT03, MT04 and MT05

6 Conclusion and Future Directions

We present the novel idea of partial diacritization for NLP. We define several diacritization schemes guided by observations of the distribution of naturally occurring diacritics in Arabic text. We test the utility of applying partial and full diacritization in the context of phrase-based SMT. Our results confirm that full diacritization is not useful for SMT; however, none of the other conditions of partial diacritization vary significantly from the baseline condition of no diacritization, NONE. In fact, two of the partial diacritization schemes (PASS and SUK) perform slightly better than NONE on two of the test sets. Crucially, our research strongly suggests that the SMT performance is positively correlated with the number of tokens accurately affected by a diacritization scheme to the extent that it would make it robust to slightly high OOV rates. We believe research that targets specific diacritization phenomena such as passivization or lexical disambiguation diacritics such as the *shadda* (gemination diacritic) is needed before we can see a significant effect on NLP applications such as SMT.

For future work, we plan to experiment with a new scheme that mimics the naturally occurring diacritic distribution, where only the distinguishing diacritics are marked on the words. We plan to perform more semantically oriented error analysis on the output to qualita-

tively assess the impact of these different diacritization schemes. Moreover, we would like to test the robustness of the system by varying the genre and domain of the training data coupled with our different partial diacritization schemes. Finally, we would also like to explore using partial diacritization in the context of other NLP applications, ASR in particular.

Acknowledgements

This work was funded by DARPA Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

References

S. Ananthakrishnan, S. Narayanan, and S. Bangalore. 2005. Automatic diacritization of arabic transcripts for asr. In *Proceedings of ICON*, Kanpur, India.

Tim Buckwalter. 2002. Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Catalog No.: LDC2002L49.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the Role of BLEU in Machine Translation Research. In *Proceedings of the 11th conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*, pages 249–256, Trento, Italy.

Sharon Goldwater and David McClosky. 2005. Improving Statistical MT Through Morphological Analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 676–683, Vancouver, Canada.

Nizar Habash and Owen Rambow. 2005. Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 573–580, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Nizar Habash and Owen Rambow. 2007. Arabic Diacritization through Full Morphological Tagging. In *Proceedings of the 8th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL07)*.

Nizar Habash and Fatiha Sadat. 2006. Arabic Preprocessing Schemes for Statistical Machine Translation. In *Proceedings of the 7th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL06)*, pages 49–52, New York, NY.

Katrin Kirchhoff and Dimitra Vergyri. 2005. Cross-dialectal data sharing for acoustic modeling in arabic speech recognition. *Speech Communication*, (46):37–51.

Philipp Koehn. 2004. Pharaoh: a Beam Search Decoder for Phrase-based Statistical Machine Translation Models. In *Proceedings of the Association for Machine Translation in the Americas*, pages 115–124.

Young-Suk Lee. 2004. Morphological Analysis for Statistical Machine Translation. In *Proceedings of the 5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04)*, pages 57–60, Boston, MA.

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.

Sonja Nießen and Hermann Ney. 2004. Statistical Machine Translation with Scarce Resources using Morpho-syntactic Information. *Computational Linguistics*, 30(2).

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–52.

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, July.

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.

Maja Popović and Hermann Ney. 2004. Towards the Use of Word Stems and Suffixes for Statistical Machine Translation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 1585–1588, Lisbon, Portugal, May.

Andreas Stolcke. 2002. SRILM - an Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 901–904.

Dimitra Vergyri and Katrin Kirchhoff. 2004. Automatic Diacritization of Arabic for Acoustic Modeling in Speech Recognition. In Ali Farghaly and Karine Megerdoomian, editors, *COLING 2004 Workshop on Computational Approaches to Arabic Script-based Languages*, pages 66–73, Geneva, Switzerland.

Imed Zitouni, Jeffrey S. Sorensen, and Ruhi Sarikaya. 2006. Maximum Entropy Based Restoration of Arabic Diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 577–584, Sydney, Australia.