Automatic Diacritization for Urdu

Abbas Raza Ali[1], Sarmad Hussain[2]
[1]*National University of Computer and Emerging Sciences, Lahore,*
[2]*Center for Language Engineering, Al-Khawarizmi Institute of Computer Science, University of Engineering and Technology, Lahore*
abbas.raza@nu.edu.pk; sarmad@cantab.net

Abstract

Urdu language is written in Arabic script. In this script, the consonantal context is clearly represented, but the vocalic sounds are represented (mostly) by marks or diacritics, which are optional and normally not written. Readers can guess the diacritics and thus can pronounce words correctly, based on their knowledge of the language. But un-diacritized Urdu text creates ambiguity for novice learners and computational systems that require pronunciation. In this paper, a statistical approach is used to mark diacritics for Urdu automatically. Use of multiple knowledge sources is also integrated with the statistical techniques to investigate their effects to the process. These knowledge sources include stemming, part-of-speech tagging, pronunciation lexicons, and word bigrams. The experimental results show that letter-level trigram model performs best and achieves 95.37% overall accuracy by applying all knowledge sources.

1. Introduction

Urdu is spoken by more than 100 million people in South Asia and the Middle East [1]. It is the national language of Pakistan and one of the scheduled languages of India. Urdu is Indo-Aryan language written with Arabic script in Nastalique writing style [2]. Arabic writing system is a consonantal system (also called Abjad [3]), i.e. the consonants are written but the vowels are not always written explicitly, for example the word *diacritization* would be written as *dcrtztn*, and both *ball* ("twist") and *bill* ("bill") will be written as *bl*. The vowels are realized through small marks above or below the preceding consonant but are optional and normally not written. Native speakers can normally recreate these unwritten vowels through context (using their knowledge of Urdu vocabulary).

Though native speakers may not need the explicit marking of vowel diacritical marks, these are essential for computational processing of Urdu. For example, it would not be possible to create a text to speech system of Urdu using this under-specified input text. In addition the ambiguity makes it hard for Machine Translation system, which needs the full specification to resolve ambiguity for proper translation. Thus, it is essential to devise a method to automatically diacritize text with vowel marks to develop effective computational techniques for Urdu.

This paper presents details of Urdu writing system and details of a hybrid technique which uses knowledge sources coupled with a statistical method to automatically predict the diacritics. Section 2 presents the Urdu writing system in detail, highlighting the scope of the problem. Though there is little work on Urdu, Section 3 discusses some similar work on Arabic and other languages. Section 4 gives details of the methodology of the work undertaken for Urdu, including the statistical model developed. Section 5 covers the knowledge sources used in the work, and Section 6 presents the results achieved. Section 7 discusses issues and future work to further improve the work.

2. Urdu Writing System

Urdu uses extended Arabic character set. Figure 1 below shows Urdu consonantal inventory [4] of the language.

آا ب بھ پ پھ ت تھ ٹ ٹھ ث ج جھ
چ چھ ح خ د دھ ڈ ڈھ ذ ر رھ ڑ ڑھ ز ژ س

ش ص ض ط ظ ع غ ف ق ک کھ گ
گھ ل م ن ں و ہ ء ی ے[1]

Figure 1: Urdu Consonants

In addition, there are seventeen vowels in Urdu, including seven long oral vowels, seven long nasal vowels and three short vowels. As the number of vowels in Urdu is considerably greater, the three marks Fatha, Kasra and Damma (in Urdu referred to as Zabar, Zer and Pesh respectively) are insufficient. Urdu uses these marks to represent the three short vowels and a combination of these marks with base characters ا، و، ی، ے to indicate the long vowels[2] [5]. In addition, the nasalization characters ں، ن are used to indicate the nasalized vowels. Thus, short vowels are represented by a mark on the preceding base consonant, long vowels are represented by a combination of the mark on the preceding base consonant plus an additional vowel base character (ا، و، ی، ے), and long nasal vowels are additionally represented by following nasalization character. This is summarized in Table 1 below, adapted from [5].

Table 1: Urdu Vowel System and its Writing System [5]

Bay + Zabar	بَ	ə
Bay + Zer	بِ	ɪ
Bay + Pesh	بُ	ʊ
Bay + NULL + Alef	با	ɑ
Bay + NULL + Alef + Noon Ghunna	باں	ɑ̃
Bay + NULL + Vao	بو	o
Bay + NULL + Vao + Noon Ghunna	بوں	õ
Bay + Zabar + Vao	بَو	ɔ
Bay + Zabar + Vao + Noon Ghunna	بَوں	ɔ̃
Bay + Pesh + Vao	بُو	u
Bay + Pesh + Vao + Noon Ghunna	بُوں	ũ
Bay + NULL + Yay	بے	e
Bay + NULL + Yay + Noon Ghunna	بیں	ẽ
Bay + Zabar + Yay	بَے	æ
Bay + Zabar + Yay + Noon Ghunna	بَیں	æ̃
Bay + (NULL | Zer)[3] + Yay	بی	i
Bay + (Null | Zer) + Yay + Noon Ghunna (see Footnote 2)	بِیں	ĩ

Thus, Urdu vowel marks are significantly conditioned by the consonantal context. As is seen in Table 1, it is not possible to have Zabar before ی, Zer before ا، ے, or Pesh before ا، ی، ے etc. for long vowels. Similarly, ں can only come after ا، و، ی. See [5] for further details regarding how the writing system relates to the pronunciation system of Urdu.

Urdu vowel marks always anchor on the preceding base consonant and cannot be written otherwise. Urdu allows for syllables to start with vowels [6]. In such cases, there is no base (onset) consonant available for the marks to anchor. In such cases, dummy base consonants are inserted, which are not pronounced. Work initially, ا is inserted as a place-holder for the vowel mark, and word medially, ء is inserted as a place-holder for the vowel mark for syllables without onset consonants [5].

Urdu follows its own set of rules for writing vowels Though models developed for Arabic language can be useful, they are not sufficient for Urdu.

3. Literature Review

This section discusses some statistical approaches that are used for automatic diacritization of different languages.

Mihalcea and Nastase [7] performed experimentation on four different languages: Czech, Hungarian, Polish and Romanian. The data is

[1] Different dictionaries may vary in the character set.

[2] و،ی also act as consonantal characters.

[3] NULL or Zer. It is controversial whether Zer is present for the representation of vowel /i/. One solution is to process both cases till the diction controversy is solved.

collected over the internet, newspapers, and electronic literature. Corpus of 14,60,000 words for Czech, 17,20,000 words for Hungarian, 25,00,000 words for Polish, and about 30,00,000 words for Romanian out of which 50,000 examples are used for testing and rest is used for training. Instance based learning technique is used at letter-level for diacritics restoration. The technique requires no additional tagging tools or resources other than raw text, which makes it language independent, particularly appealing for the languages for which there are few resources available. The maximum accuracy determined for all four languages is 98.17%.

Vergyri and Kirchhoff [8] worked on Arabic and used two transcribed corpora: FBIS consisting of 240,000 words and LDC consisting of 160,000 words for training, and 48,000 words for testing purpose. Three techniques for Arabic diacritization are used; first combines acoustic, morphological and contextual information to predict the correct form, the second ignores contextual information, and the third is fully acoustics based. The best accuracy of 88.46% is recorded by combining acoustic, morphological and contextual information at character-level.

Ananthakrishnan et al. [9] used generative techniques for recovering vowels and other diacritics that are contextually appropriate. The Arabic Treebank, consisting of 5,54,000 words, is used for training and testing purpose. The training set contains 5,41,000 words and test set is about 13,300 words.

Table 2: Sample Normal and Diacritized Text in Urdu and Ambiguities

NORMAL TEXT

پاکستان کے شمالی علاقے سربلند چوٹیوں سرسبزوشاداب وادیوں پہاڑوں کو چیرتی آبشاروں رومانی جھیلوں

دیو قامت گلیشیرز بل کھاتے دریاؤں اور گھنے جنگلوں جیسے قدرتی حسن سے مالا مال ہیں۔

DIACRITIZED TEXT

پَاکِستَان کے شُمَالِی عِلَاقے سَربُلَند چوٹِیوں سَرسَبزوشَادَاب وَادِیوں پہَاڑوں کو چِیرتی آبشَاروں رُومَانی جھِیلوں

دیو قَامَت گلیشِیرز بَل کھَاتے دَریَاؤں اَور گھَنے جَنگلوں جَیسے قُدرَتِی حُسن سے مَالَا مَال ہَیں۔

AMBIGUITIES IN THE TEXT

WORD	IPA	POS	WORD	IPA	POS
جھیلُوں	/ ‘ʤʰe. l ũ/	VB	جھِیلوں	/’ ʤʰi. lõ/	NN
بِل	/’ bɪl/	NN	بَل	/’ bəl/	NN
حَسَن	/hə.’ sən/	PN	حُسن	/’ hʊsn/	NN

Maximum likelihood and knowledge-base approaches are applied at word and character-level to solve automatic Arabic diacritization problem. Using trigram word-level model, tetra-gram character-level model, and part-of-speech knowledge 86.50% accuracy is recorded.

Nelken and Shieber [10] solve the problem of Arabic diacritization by using probabilistic finite-state transducers trained on the Arabic Treebank. The probabilistic technique is integrated with maximum likelihood based word and letter-level language models. LDC corpus for training and testing purpose with the ratio of 90% and 10% is used. Using trigram word-level, clitic concatenation and tetra-gram character-level model a maximum of 92.67% accuracy is achieved by the system.

Elshafei, Muhtaseb, and Alghamdi [11] train the system based on domain knowledge e.g., sports, weather, local news, international news, business, economics, religion, etc. Fully diacritized transcript of The Holy Quran in Arabic consist of 78,679 words is used for testing purpose. Hidden Markov Model approach is used to solve the problem of automatic generation of diacritical marks of Arabic text. The baseline algorithm achieved 95.9% accuracy and improvements like preprocessing and trigram language models for selected number of words, achieved about 97.5% accuracy.

Kirchhoff and Vergyri [12] used the same corpora and also split training and test data same as (Vergyri et. al., 2004). A standard trigram model is used but true morphological tag assignment was not known, only set of possible tags for each word were available during training, so that the probabilities and tag sequence models were updated iteratively. Overall accuracy achieved by applying the above mentioned techniques on Arabic text is 95%.

Zitouni, Sorensen, and Sarikaya [13] used Maximum Entropy based approach for restoring diacritics in Arabic text. The language model integrated with a wide array of lexical, segment based and part-speech tag features. LDC, and An-Nahar corpus is used in their work for training and testing purpose with the ratio of 85% and 15%. By combining lexical, segment based and part-of-speech features a maximum of 94.9% accuracy is achieved.

4. Methodology

Based on the review of existing literature, it is clear that knowledge sources combined with statistical techniques provide the best solution. Therefore, a hybrid technique is developed for automatic diacritization of Urdu.

The automatic diacritization is divided into three categories of problems. In the first category, many words of Urdu can be diacritized by directly looking up the diacritics from lexical resources, e.g. كتاب (kitaab, “Book”) to كِتاب.

Secondly, there are Urdu words in the lexicon which have same base form but can be assigned different sets of diacritics (e.g. see Table 2), for example بن is ambiguous between three possible actual words: بِن (bin; *harf*[4]: “of” used with family name), بَن (ban; verb: “to be made”) بُن (bun; verb: “knit”). In this second category this ambiguity is partially[5] solved using part-of-speech tagging of the words.

Finally, lookup will not work for words which are not in the lexical data. These words form the third category, and are also sub-divided into two parts. Some such words are morphological variations of lemmas in the lexicon. Such words can be stemmed and their stems and affixes can be separately looked up in the lexicon and then combined for diacritization. However, there are words which can neither themselves nor through stemming process be found in the lexicon. For these unknown or out-of-vocabulary words, the diacritics are guessed using a statistical method discussed later in this section. The complete process is summarized in Figure 2 below.

Details of the lexical and other linguistic resources used are given in the next section. In addition, a statistical diacritic tagger is developed based on the following mathematical model.

[4] Harf is a POS category which includes most of the closed class words.

[5] Complete disambiguation would require semantic level analysis.

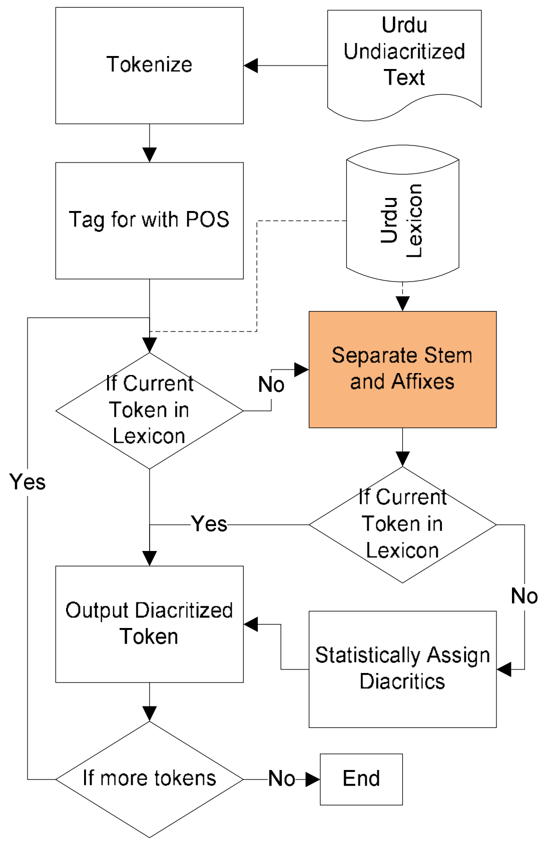

Figure 2: Process Flow Diagram for the Urdu Diacritization System

Assuming a word *W* is a sequence of base characters c_i, for i= 1 to *n*, where *n* is the number of characters in the word, it can be represented as in (1).

$$W = c_1.c_2 \ ... c_n \qquad (1)$$

Each character c_i has an associated diacritic d_i. Thus, each word *W* has a diacritic sequence D_W associated with it, which can be written as follows.

$$D_W = d_1.d_2 \ ... d_n \qquad (2)$$

We are given the word *W*, and have to determine the corresponding diacritic sequence D_W. If we consider all possible diacritic sequences, the most likely sequence $\widehat{D}_W$ is the one with maximum probability for the word sequence *W*, as in (3).

$$\widehat{D}_W = argmax_{D_W} \ P \ (D_W|W) \qquad (3)$$

Using Bayes rule, equation in (3) can be rewritten as (4):

$$\widehat{D}_W = argmax_{D_W} \ \frac{P \ (W|D_W)P(D_W)}{P(W)} \qquad (4)$$

As the probability of the word in the denominator is constant across all cases, the maximum value is dependent only on the numerator of equation in (4), as re-written in (5).

$$\widehat{D}_W = argmax_{D_W} \ P \ (W|D_W)P(D_W) \qquad (5)$$

Equation in (5) can be expanded as in (6):

$$\widehat{D}_W = argmax_{D_W} \ P \ (c_1.c_2 \ ... c_n|d_1.d_2 \ ... d_n)$$

$$. \ P(d_1.d_2 \ ... d_n) \qquad (6)$$

The first factor, using the chain rule of conditional probability, can be re-written as (7).

$$P(c_1.c_2 \ ... c_n|d_1.d_2 \ ... d_n)$$
$$= P(c_1|d_1.d_2 \ ... d_n).P(c_2|d_1.d_2 \ ... d_n, c_1) \ ...$$
$$. \ P(c_n|d_1.d_2 \ ... d_n, c_1.c_2 \ ... c_{n-1}) \qquad (7)$$

If we assume that a character is independent of the characters before itself, (7) can be simplified to (8):

$$P(c_1.c_2 \ ... c_n|d_1.d_2 \ ... d_n)$$
$$= P(c_1|d_1.d_2 \ ... d_n).P(c_2|d_1.d_2 \ ... d_n) \ ...$$
$$P(c_n|d_1.d_2 \ ... d_n) \qquad (8)$$

Further, if we also assume that for any character, only the diacritic resting on it is relevant and others can be ignored, the equation in (8) simplifies to (9) below:

$$P(c_1.c_2 \ ... c_n|d_1.d_2 \ ... d_n)$$
$$= P(c_1|d_1).P(c_2| \ d_2) \ ... P(c_n|d_n)$$
$$= \prod_{i=1}^{n} P(c_i|d_i) \qquad (9)$$

Second factor of the equation in (6) can also be simplified if we make Markov assumption that the diacritic is only dependent on recent history. If we make bigram assumption, i.e. the diacritic is only dependent on the previous one, we get the equation in (10).

$$P(d_1.d_2\ \ldots d_n) = P(d_1).P(d_2|d_1).P(d_3|d_1d_2)\ldots$$

$$P(d_n|d_1d_2\ldots d_{n-1})$$

$$= (d_1).P(d_2|d_1).P(d_3|d_2)\ldots P(d_n|d_{n-1})$$

$$= \prod_{i=1}^{n} P(d_i|d_{i-1}) \qquad (10)$$

In case trigram assumption is taken, the diacritic will depend on the context of previous two diacritics. Combining (9) and (10), we can write (3) as follows:

$$\widehat{D}_W = argmax_{D_W}\ P\ (D_W|W) \cong$$
$$argmax_{D_W}\ P\ (W|D_W)P(D_W) \cong$$
$$argmax_{D_W} \prod_{i=1}^{n} P(c_i|d_i)P(d_i|d_{i-1}) \qquad (11)$$

Given a fully diacritized text corpus for Urdu, the first factor in (11) can be estimated as the ratio of the the number of times the character c_i occurs with the diacritic d_i, and the number of times the diacritic d_i occurs:

$$P(c_i|d_i) = \frac{count\ (c_i,d_i)}{count\ (d_i)} \qquad (12)$$

The second factor in (11) can be estimated as the ratio of the number of times the diacritic d_{i-1} occurs with the diacritic d_i, and the number of times the diacritic d_{i-1} occurs:

$$P(d_i|d_{i-1}) = \frac{count\ (d_{i-1},d_i)}{count\ (d_{i-1})} \qquad (13)$$

These counts are obtained from the corpus. Witten-Bell smoothing technique is used for low frequency occurrences for these factors. Hidden Markov Model is developed and a variation of Viterbi algorithm is modified to calculate the best sequence of diacritics for the unknown words given these counts.

5. Linguistic Resources

For the lookup a comprehensive lexicon is needed. Though there are a variety of lexical resources available, they have been developed for different tasks and thus do not conform to a single underlying structure.

Three different lexical resources[6] are used to derive the lexicon for the current task. The first lexicon has been developed for the text-to-speech system for Urdu[7]. The lexicon contains 85,000 words, and each lexical entry is annotated with diacritics, pronunciation and part-of-speech. The POS tagset is limited to noun, verb, adjective, adverb, pronoun, and harf. In addition, 81,000 from online Urdu Dictionary[8] are also incorporated. These words are annotated with a lot of linguistic information. Information relevant to the current context includes pronunciation, root word, etymology, and part-of-speech. The lexicon also has the same six parts-of-speech marked. Furthermore, the corpus based lexicon is of 50,000 common words and 53,000 proper nouns is also used [15]. The words are annotated with pronunciation, part-of-speech, lemma, phonetic transcription and grammatical feature. It is using eleven part-of-speech tags including noun, Verb, adjective, adverb, pronoun, numerals, post-position, conjunction, auxiliary, case markers, and harf.

The final lexicon consists of orthography, pronunciation, part-of-speech tag and root language information of each word. Using these sources a consolidated annotated lexicon of 165,000 unique words is derived.

A subset of an existing Urdu corpus [15] consisting of 100,000 words, which has been tagged with POS [16], is manually annotated with diacritics. Five different diacritics are marked: Zer, Zabar, Pesh, Jazm and Null. Though Null diacritic may be generally confused with Jazm (which marks absence of Zer, Zabar or Pesh on consonants), the two are quite distinct. Null is used to indicated absence of a diacritic on a consonant before long vowels, thus on onset consonants in a syllable, whereas Jazm is used on consonants to indicate that they are coda position in a syllable (see [6] for discussion on Urdu syllable structure). Space is also used before the initial and after the final character of the word to mark word

[6] All these lexica have been internally developed at Center for Research in Urdu Language Processing (www.crulp.org).

[7] Developed through the Urdu Localization Project [14].

[8] www.crulp.org/oud.

boundaries. This space is also marked with a Null2 diacritic, which is different from the Null diacritic discussed above.

Different POS taggers for Urdu are available (e.g. [16, 17, 18]). These taggers use an extended POS tagset (in some cases more than 40 tags). Such fine distinction is not needed for the disambiguation task for the diacritics. Therefore, it is collapsed to the six tags including noun, verb, adjective, adverb, pronoun, and harf. This also reduces sparseness and improves accuracy of the tagging model. A tagged corpus of 250,000 words is developed and is used to train the POS tagger on this tagset.

In addition to the POS tagger, an Urdu stemmer [19] is also used. The stemmer takes in an Urdu word and returns its stem, prefixes and suffixes. It processes a total of 174 prefixes and 712 suffixes. The stemmer gives an accuracy of 91.18%. This stemmer is integrated into the process. The list of prefixes and suffixes of Urdu are diacritized and also added to the lexicon for eventual lookup.

6. Results

A total of 10,143 diacritized and part-of-speech tagged words are held-out as testing data. The tagging accuracy of 95.6% is achieved using the smaller tagset. Various experiments are performed to gauge the best combination of procedures. Initially only the statistical method is employed without consulting any linguistic resources. This baseline experiment gives an overall accuracy of 81.13% using bigrams and 84.07% using trigrams. The accuracy is improved to 90.86% and 91.83% if POS based lexical lookup is also done before the statistical tagging, and to 89.06% and 90.75% respectively if the words are looked up in the diacritized corpus. If the words are stemmed and then diacritized, the accuracy also goes up to 88.35% and 90.15% for bigram and trigram analyses respectively. Other combination of techniques are also evaluated. The results are given in Table 3 below.

These results show that using all knowledge sources and then statistically tagging the remaining untagged words gives the best results. Accuracies of 95.20% and 95.37% are achieved using bigram vs. trigram statistical tagging.

Table 3: Test Results for Diacritization using a Combination of Different Techniques

TECHNIQUE/SOURCE	BIGRAM	TRIGRAM
Base line	81.13 %	84.07 %
POS based lexical lookup	90.86 %	91.83 %
Bigram lookup from corpus	89.06 %	90.75 %
Stemming	88.35 %	90.15 %
Bigram lookup from corpus + Stemming	91.91 %	92.77 %
POS based lexical lookup + Bigram lookup from corpus	93.86 %	94.35 %
POS based lexical lookup + Stemming	92.77 %	93.18 %
POS based lexical lookup + Bigram lookup from corpus + Stemming	**95.20 %**	**95.37 %**

7. Discussion

The results show that using knowledge sources significantly improves the accuracy of statistical tagging. This is because lexical lookup is much more reliable and POS helps resolve any look up ambiguities (except rarer ambiguities based on semantics, instead of word classes). Simple word bigram look up from diacritized corpus also fairly accurate and thus increasing the size of this corpus in the future can increase the accuracy even further. Stemming is a relatively light process but contributes significantly to the accuracy. This is because Urdu is morphologically rich and not all word forms can be found in the corpus or in the lexicon. Separating root and affixes allows more words to be looked up from the lexicon, and affixes is a closed set which can be pre-stored in an affix lexicon to add to the accuracy. The results achieved are comparable to the best work on Arabic language, even with smaller corpus for statistical tagging and more diacritics.

References

[1] M. P. Lewis (Ed.), (2009). *Ethnologue: Languages of the World, 16th Edition.* SIL, USA.

[2] S. Hussain (2003), "www.LICT4D.asia/Fonts/Nafees_Nastalique," in the *Proceedings of 12th AMIC Annual Conference on E-Worlds: Governments, Business and Civil Society,* Asian Media Information Center, Singapore.

[3] Writing systems

[4] Urdu Dictionary Board (2008). *Urdu Lughat.* Urdu Dictionary Board, Karachi, Pakistan.

[5] S. Hussain (2004), "Letter to Sound Rules for Urdu Text to Speech Sytem," in the *Proceedings of Workshop on Computational Approaches to Arabic Script-based Languages, COLING,* Geneva, Switzerland.

[6] S. Hussain (2005), "Phonological Processing for Urdu Text to Speech System," in *Contemporary Issues in Nepalese Linguistics* (eds. Yadava, Bhattarai, Lohani, Prasain and Parajuli), Linguistics Society of Nepal, Kathmandu, Nepal.

[7] R. Mihalcea and V. Nastase (2002), "Letter Level Learning for Language Independent Diacritics Restoration," in the Proceedings of 6th Workshop on Computational Language Learning, CoNLL.

[8] D. Vergyri and K. Kirchhoff (2004), "Automatic Diacritization of Arabic for Acoustic Modeling in Speech Recognition," in the *Proceedings of Workshop on Computational Approaches to Arabic Script based Languages, COLING,* Geneva, Switzerland.

[9] S. Ananthakrishnan, S. S. Narayanan, and S. Bangalore (2005), "Automatic Diacritization of Arabic Transcripts for Automatic Speech Recognition," in the *Proceedings of ICON*, Kanpur, India.

[10] R. Nelken and S. M. Shieber (2005), "Arabic Diacritization using Weighted Finite-State Transducers," in the *Workshop on Computational Approaches to Semitic Languages*, USA.

[11] M. Elshafei, H. Al-Muhtaseb, M. Alghamdi (2006), "Statistical Methods for Automatic Diacritization of Arabic Text," intThe *Saudi 18th National Computer Conference*, Riyadh, Saudi Arabia.

[12] K. Kirchhoff, K. and D. Vergyri, (2004), "Cross-dialectal acoustic data sharing for Arabic speech recognition," in the *Proceedings of ICASSP '04*.

[13] I. Zitouni, J. S. Sorensen and R. Sarikaya (2006), "Maximum Entropy Based Restoration of Arabic Diacritics," in the *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia.

[14] S. Hussain (2004), "Urdu Localization Project: Lexicon, MT and TTS", In the *Proceedings of Workshop on Computational Approaches to Arabic Script-based Languages, COLING*, Geneva, Switzerland.

[15] M. Ijaz and S. Hussain (2007), "Corpus Based Urdu Lexicon Development," in the *Proceedings of Conference on Language Technology (CLT07),* University of Peshawar, Pakistan.

[16] H. Sajjad (2007), *Statistical Part of Speech Tagger for Urdu,* unpublished MS Thesis, National University of Computer and Emerging Sciences, Lahore, Pakistan.

[17] H. Sajjad and H. Schmid (2009), "Tagging Urdu Text with Parts of Speech: A Tagger Comparison," in the *Proceedings of the 12th Conference of the European Chapter of the ACL,* pp. 692–700, Athens, Greece.

[18] A. Muaz, A. Ali, S. Hussain (2009), "Analysis and Development of Urdu POS Tagged Corpora," in the *Proceedings of the 7th Workshop on Asian Language Resources, IJCNLP*, Singapore.

[19] *Q. Akram, A. Naseer,* S. Hussain (2009), "Assas-band, an Affix-Exception-List Based Urdu Stemmer," in the *Proceedings of the 7th Workshop on Asian Language Resources, IJCNLP*, Singapore.