Low-Resource Speech Translation of Urdu to English Using Semi-Supervised Part-of-Speech Tagging and Transliteration

A. Ryan Aminzadeh[†]
Wade Shen

MIT/Lincoln Laboratory, {ryan.aminzadeh,swade}@ll.mit.edu

ABSTRACT

This paper describes the construction of ASR and MT systems for translation of speech from Urdu into English. As both Urdu pronunciation lexicons and Urdu-English bitexts are sparse, we employ several techniques that make use of semi-supervised annotation to improve ASR and MT training. Specifically, we describe 1) the construction of a semi-supervised HMM-based part-of-speech tagger that is used to train factored translation models and 2) the use of an HMM-based transliterator from which we derive a spelling-to-pronunciation model for Urdu used in ASR training. We describe experiments performed for both ASR and MT training in the context of the Urdu-to-English task of the NIST MT08 Evaluation and we compare methods making use of additional annotation with standard statistical MT and ASR baselines.

Index Terms— Low-resource, Transliteration, Part-of-Speech Tagging, Unsupervised learning, Urdu, Speech Translation

1. INTRODUCTION

In recent years the performance of ASR and MT systems has improved significantly. Many of the best performing systems make use of statistical modeling techniques to learn associations between either sounds and words, in the case of ASR, or words across languages. For both technologies, the performance of these modeling methods is highly dependent on the amount of data available to train them. For example, it is not uncommon for ASR systems to be trained with thousands of hours of transcribed data and a number of the discriminative training methods that are currently used [1] crucially rely on having large volumes of data for their improvements. As with ASR, current MT systems make use of statistical phrase-based models that are often trained with hundreds of thousands of parallel sentence pairs.

For many languages and language pairs, the vast resources needed to train statistical models for ASR and MT are not available. In such cases, techniques that make use of limited resources may improve translation and recognition quality. In this paper we describe the construction of a speech translation system for Urdu-English using limited transcribed speech resources and bilingual texts. In addition to standard statistical models, we employ additional source-language annotations, specifically, POS taggers and transliterators trained in a semi-supervised manner using extremely limited quantities of supervised data in tandem with larger unsupervised data sets.

We examine the use of POS taggers to construct factored translation models. Earlier experiments using these models suggest that they can make better use of limited data resources when compared with standard phrase based models [2][3]. In section 4.3, we describe experiments comparing these models against standard phrase-based models on the NIST MT08 Evaluation task [4].

For the construction of our ASR system, we are limited by the lack of pronunciation lexica. In this work, we explore the use of transliteration to derive pronunciation lexicons for ASR training. This is described in detail in section 3.3. We compare this approach with a grapheme-based approach [5].

Both transliteration and POS tagging are done using an HMM-based model. As the data available for training these models is limited, we employ a semi-supervised approach in which HMM parameters are initialized using a small set of supervised data, then multiple iterations of unsupervised training are performed. The training process for each of these models is described in section 3.2 and 4.2.

2. PREVIOUS WORK

Transliteration of proper nouns into English from languages using the Arabic text, particularly Arabic itself, have been examined extensively, due to the importance of preserving such information in translations [6]. The use of transliteration methods to derive pronunciation models is, however, less well explored. Our approach makes use of parallel data to derive transliteration examples (from Urdu to English). The resulting English characters can then be mapped to phones using English spelling-to-pronunciation algorithms i.e. [7][8]. This approach is well suited to resource-poor languages like Urdu.

As our goal is to build a speech translation system with very limited resources, we also examined the use of semi-supervised, HMM part-of-speech taggers to enrich the standard phrase-based MT model. We employ a factored translation model for this task and compare the performance of this model against a baseline system trained with less than 100k sentences of bitext training. For POS tagger training we use unsupervised data in a manner similar to that described in [2]. Following [3], we then use our tagger to construct factored models translating both word and POS phrases. The structure of our model is similar to that reported in [2] but in our case, we make use of very little supervised data to train our POS tagger as Urdu resources are much more limited than the Spanish and Czech experiments those authors report on.

[†] This work is sponsored by the Air Force Research Laboratory under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2008	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Low-Resource Speech Translation of Urdu to Euglish Using Semi-Supervised Part-of-Speech Tagging and Transliteration	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **MIT/Lincoln Laboratory**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**4**	

3. ASR PRONUNCIATION MODELING USING TRANSLITERATION

3.1. Deriving Supervised Data from Parallel Text

The Urdu-English training data from the NIST MT08 Evaluation includes a set of named entity annotations, generated by LDC's simpleNET tool [9]. Many of these named entities contain proper names or persons/places, or words borrowed from English, making them suitable for transliterator training.

We extracted Urdu-English word pairs as candidates for transliteration training using word alignments proposed by GIZA++ [10]. Candidate word pairs are then scored based on a word length ratio shown in Eq. 1, excluding vowels in English, since Urdu words do not contain short vowel characters.

$$WLR\% = \left| \frac{L(Urdu) - L(English_{no-vowels})}{\max\{L(Urdu), L(English_{no-vowels})\}} \right| x100\% \qquad (1)$$

A threshold of 80% is applied to automatically narrow the potential field of actual transliterated pairs from the annotated data. Examining a small sample of the remaining word pairs, we found 90% purity. Though noisy, this set was still usable for training.

One-to-one character level mappings of Urdu to English are then created using the algorithm below, and are important because we employ an HMM that assumes that each observation is generated by exactly one state.

The first step is to normalize several variations of Urdu characters:

KAF & KEHEH → KAF
FARSI YEH & ARABIC YEH → FARSI YEH
HEH & HEH DOCHESHMEH → HEH

Next, all English words are lower cased, and Urdu and English digraphs and trigraphs are mapped to single non-alphabet characters in the ASCII set, and double letters in English are mapped to one capital letter:

mushroom → mu$rOm
پرانیویٹ → پر2ویٹ (private)

Because a number of Urdu vowels are generally not written, we insert a neutral vowel indicator ("_") between consecutive Urdu consonants. In some cases, consecutive consonants are true clusters and the neutral vowel should be empty. In our training data, we add a corresponding empty state character ("{") to the English side of our data in these cases. This is done when the English character associated with the inserted "_" is not a vowel.

At test time, the test data has an "_" inserted between all Urdu consonants, and thus the transliteration process allows for vowelization to take place based upon the different mappings for that state in the trained models, whether to a vowel, vowel digraph, or just an empty place-filler, meaning there are two consonants without a short vowel between:

<E>4ز_ر_ک<S> → <S>kar{z~<E> (Karzai)

Finally, <S> and <E> tags are applied to the words in both English and Urdu, and the pairs are parsed on at the character-level.

This process does not guarantee a one-to-one mapping. As such, all Urdu-English word pairs that contain a mismatched number of characters are removed from the training set. This algorithm yields 398 transliterated word pairs for training, out of about 1,700.

An additional set of data, 200 parallel Farsi names and their English transliterations [11] were also normalized in the manner described above, and added to our training data set.

3.2. Training Procedure

Our tranlisterator uses a standard discrete hidden markov model, with a two-token context for state transitions [12]. Both transition and observation models were learned in a supervised ML manner. Transition probability models were trained separately for each language because the Farsi name list is heavily biased towards names, whereas the Urdu data tends to have many borrowed English words. A trigram LM was used because the insertion of potential empty states destroys some context and forces an expansion of the n-gram size required to capture sufficient context. Linear fusions of these two transition probability models are tested on the training set to find an optimal set of weightings.

Pruning parameters, applied to restrict the scope of the decoder, were tested against the training data in order to find a point of convergence.

In all these cases, optimization was done against the training set through cross validation as the amount of supervised data for testing is extremely limited

To validate the performance of transliteration model we selected a set of 191 transliterated words from the NIST MT08 Evaluation Test Set.

3.3. Transliterator Experiments

The primary metric used to evaluate the stand-alone performance of the transliteration tool is the character error rate (CER) which we compute using SCLITE.

Experiments were first carried out to determine the optimal linear weights for the Urdu and Farsi language models, tuning to the lowest CER. The results are shown in the Table 1 below:

URDU LM	FARSI LM	CER (%)
1	0	36.4
0.9	0.1	35.7
0.75	0.25	35.5
0.5	0.5	35.3
0.25	0.75	35.1
0.1	0.9	35.7
0	1	39.6

Table 1: LM Fusion Optimization

As seen in the Table 1, the optimal LM weights were found to be 0.25 for Urdu and 0.75 for Farsi.

We also experimented with lexical constraints applied after transliterator decoding. In this case, we select hypotheses from the transliterator's n-best list that occur in an existing English lexicon. Using lexical constraints after decoding yields a 4.7% decrease in CER.

We also examined the effect of pruning during our optimization process. Considering the top 3,500 paths from the previous state transition yields optimal results, with a larger search costing too much in compute-time to justify the very marginal 0.1% decrease in CER for every extra 1,500 paths considered per

state transition in the search. The optimized transliterator yields a CER of 30.4% on the test set described above.

Upon analysis of the errors being made, several common themes are observed. Some words have only one or two character errors, with these errors being the wrong vowel being inserted for a short-vowel sound. An example would be the word "Maradabad" in Urdu, which is transliterated as "Muradabad". The other, less prominent class of errors is an effect that "Urdu-izes" borrowed words from English; for example, "airport" in Urdu gets transliterated to "arapurat", which has several hallmarks of how city names are written in Urdu.

3.4. Transliterator as Pronunciation Model and ASR Experiments

We ran a number of experiments using the speech recognition system described in [13]. In all cases, the data used to train the recognizer was derived from the ARL Urdu Speech Corpus from LDC [14]. A partition of 20 speakers (19,000+ utterances) was held out from this corpus for testing. All results reported below are from this held-out test set.

We compare the performance of our transliteration model to a simple grapheme model in which every Urdu character is treated as a single phone. Because short vowels are not written, we introduce a neutral vowel between consecutive consonants. In this case all short vowels are represented by one phone model.

Our transliteration approach works as follows: we apply transliteration to each Urdu word in our vocabulary. We then select the top five hypotheses as potential pronunciations. For each of these transliterations, we map the transliterator state sequence to English phones using a hand-generated table. A forced alignment procedure during training is used to select pronunciations used for training.

Results of these experiments are shown in the Table 2 below:

	Grapheme WER (%)	Transliterations WER (%)
Unadapted + 2g LM	16.3	20.9
Adapted + 2g LM	14.7	17.2
Rescore Lattices 2g LM	14.0	16.0
Rescore Lattices 3g LM	8.8	11.2
Rescore Lattices 4g LM	8.2	10.7

Table 2: ASR Experiment Results

From the results in Table 2, the grapheme approach does significantly better than the transliterator-based approach across the board. The grapheme approach contained the ambiguity of mapping all short vowels to a single grapheme, due to their absence in Urdu orthography. However, this shortcoming is outweighed by the fact that, rather than directly mapping the Urdu characters to sounds, the transliterator-based approach has an intermediary step of generating English letter-sequences to be mapped to sounds, allowing errors to be introduced. These errors can then be propagated into the pronunciations that are assigned to different letter-sequences, negating any potential gain from disambiguation of short vowels.

4. SEMI-SUPERVISED POS TAGGING FOR FACTORED TRANSLATION MODELS

4.1. Data Usage

The NIST MT08 Urdu-English training data includes a small set of POS tagged data. The data consists of 5 tagged documents, with 17 unique POS tags. Four of the documents are used as the supervised training data for these experiments, and the final one is used as a test set. The total amount of supervised POS tag data available to training consists of 247 sentences totaling 4,247 words. An additional 42,202 documents of untagged Urdu data (> 374,000 sentences) was used for unsupervised training.

4.2. Training Procedure with Supervised and Unsupervised Data

As with our transliterator, we use a discrete hidden markov model for POS tagging [15][16][17] with a single-token context. In order to handle OOV words better, we prune singleton words and assign their observations to an unknown word token.

Models are initialized using the supervised data to compute ML estimates of the observation and state transition probabilities.

To make use of the unsupervised data, we employ forward-backward training for parameter re-estimation [18][19][20]. Six iterations of EM training are run on the combined supervised and unsupervised training sets, until the training set likelihoods converge.

For the final iteration of the EM-trained models, posterior probabilities for Urdu words are computed and the distributions examined, as well as POS confusion matrices. This process allows us to check any systematic class labeling errors.

4.3. POS Tagging Results

We evaluate the performance of the stand-alone POS tagger by measuring the tag error rate. Unfortunately, due to limited truth data, we are only able to evaluate our error rate using a single document from two news sources (BBC and Jang news). During training, after every iteration of EM, the test set is tagged and scored. The results are shown below in Figure 1, with the lowest tag error rate = 28.88%:

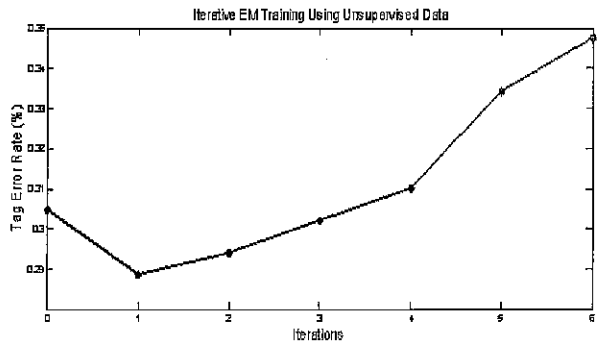

Figure 1: Effects of Iterative EM Training

Prior to EM (0^{th} iteration), only ML training is performed using tagged data from a 4-document supervised set (from BBC/Jang news sources). Each EM iteration after this accumulates training counts via a forward-backward procedure using all of the labeled and unlabelled data (40k+ documents from multiple news sources). It is likely that the increased error rate is due to the relative balance of supervised and unsupervised data.

Despite the increased error rate, the unsupervised EM procedure serves to cluster words unseen in the rather small labeled data set. For the purpose of translation modeling, correct tags may not needed, instead, it may be sufficient that the model generate semantically coherent clusters. The experiments

described in the next section attempt to test usability of the labeling generated by this POS tagger training procedure in an MT context.

4.4. Factored Machine Translation Results

Machine translation experiments were run on the NIST MT08 Evaluation test set using the MIT/AFRL Machine Translation System [21] in order to observe the effect on translation quality of rich annotations such as POS tags.

Our baseline system is a standard phrase-based statistical MT system with added rescoring language models (class-based 7-gram and word-based 6-gram). We compare this system against two factored systems using the Urdu POS tagger. One system made use of English POS tags as generated by the Stanford English POS tagger [22]. Another system used unsupervised word classes trained on the English side of the MT08 Urdu-English training set. In both cases, the factored model configuration shown in Figure 2 was used:

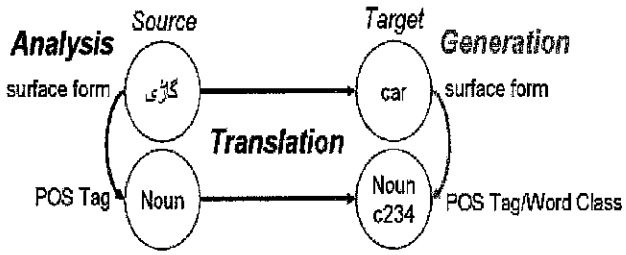

Figure 2: Factored Model Configuration for MT

Results from each system are shown in Table 3 below:

Experiment	BLEU Score
Baseline	16.47
Urdu & Eng POS Tags	16.27
Urdu POS Tags & Eng Classes	16.97

Table 3: MT Experiment Results

From the results in Table 3, it can be seen that using Urdu POS tags and English classes of word-clusters provides a 0.5 increase in BLEU, while using POS tags on both source and target sides actually degrades performance. It is possible that this result stems from a mismatch between the Stanford English POS tagger's training data and the NIST MT08 Evaluation test data. Even though the Urdu POS tags used for this experiment were noisy (given the limited amount of supervised training data), the additional information they provide yields an improved translation model.

5. SUMMARY

There are several important conclusions that can be drawn from these experiments regarding low-resource speech translation. Statistical speech translation performance has been shown to improve when annotations are included in the training processes. However, in these experiments, the volume of data was extremely low for MT, transliterator and POS tagger training.

The ASR experiments presented here show that the transliterator, restricted to a very low-resource training set, generates too much variability in the pronunciation lexicon. As such, in situations where pronunciation lexica are not available, the use of a grapheme approach can outperform the transliterator-based approach (in Urdu, by 2.5% WER). This means that a pronunciation lexicon can be constructed with little or no resources (for languages in which grapheme-based ASR is possible).

The MT experiments, however, show an improvement in performance of 0.5 BLEU when the low-resource trained Urdu POS tagger is applied to generate source-side factors. The conclusion that can be drawn here is that the EM training successfully performs unsupervised clustering of unseen words with very little hand-annotated, supervised data, allowing for improved translation quality using factored machine translation.

6. REFERENCES

[1] Woodland, P.C., Povey, D. "Large Scale Discriminative Training for Speech Recognition." In: Proc. ITRW ASR, ISCA, 2000.
[2] W. Shen, R. Zens, N. Bertoldi, and M. Federico. "The JHU Workshop 2006 IWSLT System." IWSLT, pages 59-63, Kyoto, Japan, November 2006.
[3] Philipp Koehn and Hieu Hoang. "Factored Translation Models." EMNLP, pages 868–876, Prague, Czech Republic, 2007.
[4] http://www.nist.gov/speech/tests/mt/2008/
[5] A. Black, G. Anumanchipalli, and K. Prahallad. "Significance of Early Tagged Contextual Graphemes in Grapheme Based Speech Synthesis and Recognition Systems." IEEE ICASSP, Las Vegas, USA., 2008.
[6] Mehdi M. Kashani, Fred Popowich, and Fatiha Sadat. "Automatic Transliteration of Proper Nouns from Arabic to English." The Challenge of Arabic For NLP/MT, 76-84, 2007.
[7] Riley, M. Tree-based modeling for speech synthesis. In TalkingMachines: Theories, Models, and Designs, Amsterdam, 1992.
[8] Fostler-Lussier,E. "Multi-level Decision Trees for Static and Dynamic Pronunciation Models." Eurospeech 99, ISCA, 1999.
[9]http://projects.ldc.upenn.edu/LCTL/Tools/SimpleNET_20060315.zip
[10] http://www.fjoch.com/GIZA++.html
[11] http://cleo.lcs.psu.edu/boy_names.html
[12] L. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", IEEE, vol. 77, no. 2, pp. 257-286, 1989.
[13] Shen, Wade and Reynolds, Douglas. "A Comparison of Speaker Clustering and Speech Recognition Techniques for Air Situational Awareness." INTERSPEECH 2007, 2421-2424
[14]http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2007S03
[15] J. Kupiec. "Robust Part-of-Speech Tagging Using a Hidden Markov Model." Computer Speech and Language, vol. 6, pp. 225-242, 1992.
[16] B. Merialdo, "Tagging English Text with a Probabilistic Model." Computational Linguistics, vol. 20, no. 2, pp. 155-171, 1994.
[17] A. Ratnaparkhi. "A Maximum Entropy Part-of-Speech Tagger." EMNLP-1996, 1996.
[18] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun. "A Practical Part-of-Speech Tagger." in Proceedings of the Third Conference on Applied Natural Language Processing, 1992.
[19] Q. Wang, D. Lin, and D. Schuurmans. "Improved Estimation for Unsupervised Part-of-Speech Tagging." ACL, 2005.
[20] E. Brill, "Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging." in Proceedings of the Third Workshop on Very Large Corpora, ACL, pp. 1-13, 1995.
[21] W. Shen, B. Delaney and T. Anderson. "The MITLL/AFRL IWSLT 2006 Translation System." IWSLT, 2006, Kyoto, Japan.
[22] Kristina Toutanova and Christopher D. Manning. "Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger." EMNLP/VLC-2000, 2000, pp. 63-70