Errgrams – A Way to Improving ASR for Highly Inflected Dravidian Languages

Kamadev Bhanuprasad
Andhra University, Visakhapatnam
Andhra Pradesh 530 003
India
save.climate@gmail.com

Mats Svenson
University of Macau
Av. Padre Toms Pereira, Taipa
Macau, China
svmats@yahoo.com

Abstract

In this paper, we present results of our experiments with ASR for a highly inflected Dravidian language, Telugu. First, we propose a new metric for evaluating ASR performance for inflectional languages (Inflectional Word Error Rate IWER) which takes into account whether the incorrectly recognized word corresponds to the same lexicon lemma or not. We also present results achieved by applying a novel method – errgrams – to ASR lattice. With respect to confidence scores, the method tries to learn typical error patterns, which are then used for lattice correction, and applied just before standard lattice rescoring. Our confidence measures are based on word posteriors and were improved by applying antimodels trained on anti-examples generated by the standard N-gram language model. For Telugu language, we decreased the WER from 45.2% to 40.4% (by 4.8% absolute), and the IWER from 41.6% to 39.5% (2.1 % absolute), with respect to the baseline performance. All improvements are statistically significant using all three standard NIST significance tests for ASR.

1 Introduction

Speech recognition technologies allow computers equipped with a source of sound input, such as a microphone, to interpret human speech, for example, for transcription or as an alternative method of interacting with a machine. Using constrained grammar recognition (described below), such applications can achieve remarkably high accuracy. Research and development in speech recognition technology has continued to grow as the cost for implementing such voice-activated systems has dropped and the usefulness and efficiency of these systems has improved. Furthermore, speech recognition has enabled the automation of certain applications that are not automatable using push-button interactive voice response (IVR) systems. Speech recognition system are based on simplified stochastic models, so any aspects of the speech that may be important to recognition but are not represented in the models cannot be used to aid in recognition. An essential part of each Automatic Speech Recognition (ASR) system is Language Model (LM) (Rabiner L. and Juang BH., 1993; Huang X., 2001; Jelinek F., 1998). For languages with rich inflection, language modeling is difficult (Ircing P. et al., 2001; Rotovnik T. et al., 2007). To be able to perform Very (300K+) Large Vocabulary Continuous Speech Recognition in real (or at least acceptable) time, nowadays, it is often only possible to use 2-gram LM for the first recognition pass. Using only one word context is usually insufficient in order to achieve good results. To improve performance for off-line ASR, it is possible to rescore output lattice afterward (Chelba and Jelinek, 1999; Richardson F. et al., 1995; Finke et

al., 1999; Ircing P. and Psutka J., 2002). In this paper, we describe our method for reducing error rates, that was applied to improve ASR results for LVCSR of Dravidian languages namely the Telugu language.

2 Telugu and Dravidian languages in general

Since there have not yet been many publications on ASR for the Dravidian languages, we give here some basic information on them. Dravidian languages are spoken by more than 200 million people (Wikipedia, 2007). In phonology, Dravidian languages suffer from the lack of distinction between aspirated and unaspirated stops. While some Dravidian languages have large numbers of loan words from Sanskrit and other Indo-European languages, the words are often mispronounced by monolingual Dravidian speakers. Dravidian languages are also characterized by a three-way distinction between dental, alveoar and retroflex places of articulation as well as large numbers of liquids.

In this work, we show evidence from one particular Dravidian language Telugu. Telugu belongs to the family but with ample influences by the Indo-Arian family and is the official language of the state of Andhra Pradesh, India. It is the Dravidian language with the greatest number of speakers, the second largest spoken language in India after Hindi and one of the 22 official national languages of India.

The Telugu script is believed to descend from the Brahmi script of the Ashokan era. Merchants took the Eastern Chalukyan Script to Southeast Asia where it parented the scripts of Mon, Burmese, Thai, Khmer, C'am, Javanese and Balinese languages. Their similarities to Telugu script can be discerned even today. Its appearance is quite similar to the Kannada script, its closest cousin. Telugu script is written from left to right and consists of sequences of simple and/or complex characters. The script is largely syllabic in nature - the basic units of writing are syllables. Since the number of possible syllables is very large, syllables are composed of more basic units such as vowels (achchu or swar) and consonants (hallu or vyanjan). Consonants in consonant clusters take shapes which are very different from the shapes they take elsewhere. Consonants are presumed to be pure consonants, that is, without any vowel sound in them. However, it is traditional to write and read consonants with an implied 'a' vowel sound. When consonants combine with other vowel signs, the vowel part is indicated orthographically using signs known as vowel maatras. The shapes of vowel maatras are also very different from the shapes of the corresponding vowels. The overall pattern consists of 60 symbols, of which 16 are vowels, 3 vowel modifiers, and 41 consonants. Spaces are used between words as word separators. The sentence ends with either a single (purna virama) or a double bar (deergha virama). They also have a set of symbols for numerals, though Arabic numbers are typically used.

In Telugu, Karta (nominative case or the doer), Karma (object of the verb), and Kriya (action or the verb) follow a sequence. This is one of the several reasons why linguists classify Telugu as a Dravidian Language – this pattern is found in other Dravidian languages but not in Sanskrit. Telugu allows for polyagglutination, the unique feature of being able to add multiple suffixes to words to denote more complex features. Telugu also exhibits one of the rare features that Dravidian languages share with few others: the inclusive and exclusive we. The bifurcation of the First Person Plural pronoun (we in English) into inclusive (manamu) and exclusive (memu) versions can also be found in Tamil and Malayalam. Like all Dravidian languages, Telugu has a base (or of words which are essentially Dravidian in origin.

Telugu pronouns follow the systems for gender and respect also found in other Indian languages. The second person plural 'miru' is used in addressing someone with respect, and there are also respectful third personal pronouns pertaining to both genders. A specialty of the Telugu language, however, is that the third person non-respectful feminine is used to refer to objects, and there is no special 'neuter' gender that is used.

3 Method

3.1 Data

We have recorded a large broadcast news corpus for Telugu. All commercials and stretches of spontaneous speech were removed from the data, since we focus here on ASR for an unexplored language rather than on dealing with automatic audio segmentation and spontaneous speech recognition. Overall, we had at disposal 61.2 hours of pure transcribed speech. It yields 635k word tokens, contained in manual human transcriptions. Due to rich morphology of Dravidian languages, it represents 78k different word forms, with plenty of words appearing just once. We used $\sim$70% of data for training, $\sim$15% for development, and the remainding $\sim$15% for testing.

For language modeling, we used a newspaper corpus containing data from three major Telugu newspapers - Andhra Prabha, Eenadu, and Vaartha. This corpus contains 20M tokens, which corresponds to 615k different word forms.

3.2 Evaluation method

The usual metric to evaluate ASR system performance is Word Error Rate (WER). Unfortunately, as we described in Section 2, Telugu is a highly inflectional language having a really high number of different word forms. Using WER, this cause to underestimate the real system performance, since this metric does not distinguish between confusing word identities and confusing just forms of the same word (lemma). However, it is obvious that these errors do not have the same influence on the usability of automatic transcripts. Taking an example from English, recognizing who instead of whom is not that bad as confusing boom (especially when most Americans or not able to distinguish who and whom anyway).

Thus, we propose to use Inflectional Word Error Rate (IWER), which gives weight 1 to errors confusing lemmas, while only a weight 0.5 when the lemma of the incorrectly recognized word is correct, but the whole word form is not correct. Lemmas corresponding to particular word forms may be obtained using an automatic lemmatization technique.

3.3 Confidence measuring

The key problem for our method (as described below) is to perform appropriate ASR confidence measuring. Confidence measures (CMs) need to be interpreted in order to decide whether a word is probably recognized correct or incorrect. In this paper, we use a confidence measure based on posterior probability formulation. It is well known that the conventional ASR algorithm is usually formulated as a pattern classification problem using the maximum a posterior (MAP) decision rule to find the most likely sequence of words W which achieves the maximum posterior probability $p(W|X)$ given any acoustic observation X.

Obviously, the posterior probability $p(W|X)$ is a good confidence measure for the recognition decision that X is recognized as W. However, most real-world ASR systems simply ignore the term $p(X)$ during the search, since it is constant across different words W. This explains why the raw scores are not usable as confidence scores to reflect recognition reliability. Anyway, after the normalization by $p(X)$, the posterior probability $p(W|X)$ can be employed as a good confidence measure; it represents the absolute quantitative measure of the correspondence between X and W.

In real-world tasks, we have to either employ certain simplifying assumptions or adopt some approximate methods when estimating $p(X)$ in order to obtain the desired posteriors. In the first category, it includes the so-called filler-based methods which try to calculate $p(X)$ from a set of general filler or background models. These approaches are very straightforward and usually can achieve an reasonable performance in many cases. However, we rather used the so-called lattice-based methods which attempt to calculate $p(X)$, then the posterior probability $p(W|X)$ in turn, from a word lattice or graph based on the forwardbackward algorithm, such as Schaaf (Schaaf T. and Kemp T., 1997) and Wessel (Wessel F. et al., 1999) and their colleagues, among others.

Usually, a single word lattice or graph is generated by the ASR decoder for every “utterance”.

Then, the posterior probability of each recognized word or the whole hypothesized stream of words can be calculated based on the word-graph from an additional post-processing stage. Since word graph is a compact and fairly accurate representation of all alternative competing hypotheses of the recognition result which usually dominate the summation when computing $p(X)$ over a variety of hypotheses, the posterior probability calculated from a word graph can approximate the true $p(W|X)$ very well.

In our approach, we extended the lattice based CM by using an *antimodel*. The idea of antimodels has already been proposed for CMs (Rahim M. et al., 1997), however, it has remained unclear what data should be used to estimate these antimodels. In our work, we simply generated anti-examples from our N-gram model. The rationale behind this is very straightforward. LM constraints are very strong in determining the final ASR hypotheses, and may sometimes undesirably wash out correct acoustic posteriors. Also, when you let your LM generate sentences, these sentences correspond well to N-gram probabilities but are definitely neither grammatically nor semantically correct. Thus, these generated sentences can be very well used as anti-examples to train the antimodel. Then, we performed forced-alignment against a random transcript to generate training data for each anti-model.

3.4 Errgrams

The main problem when applying ASR to extremely inflected languages such as Telugu, is the need to use a very large vocabulary, in order to reduce the OOV rate to an acceptable level. However, this causes problems for making the automatic transcription in a time close to the real-time. Since we cannot use such a big dictionary in these task, our first results had quite high WERs and IWERs. However, we analyzed the errors and found that some typical error patterns occur repeatedly. This fact inspired us to design and employ the following method.

First, using HTK large vocabulary speech recognizer (HTK, 2007) and a bigram LM, we generated an N-best ASR output and a scored bigram lattice. Then we statistically analyzed the errors and created so-called *errgrams*. Errgrams are pairs of bigrams, the first member of the pair is the correct bigram and the second member is the recognized bigram. For infrequent bigrams, the method is allowed to back-off to unigrams, using discounting based on common smoothing strategies (such Katz backoff), but the backoff is more penalized since unigram errgrams are much less reliable compared to common language modeling backoffs (such as backoff for training LMs for ASR). Errgrams were not only trained using 1-best ASR output, but to gain more real ASR data, we used 5-best hypothesis for training. For estimating errgram pairs, we also take into account confidence scores - the lower CM, the higher weight is given to a particular errgram example. By this approach, we may achieve better results with using vocabulary of standard size ($< 100k$), since words in "correct" parts of errgrams may include words that are not in the limited size vocabulary used for the first recognition pass. In other words, we can partially reduce the OOV problem by this approach. Note that LMs used for lattice rescoring include all such words originally missing in the baseline LM but appearing in errgrams.

The errgrams trained in the above described way, are then applied in the following way during the decoding phase:

1. Using a bigram model, generate an ASR lattice
2. Walk through the lattice and look for bigrams (or unigrams) having a low CM
3. If for such a low CM n-gram we have a corresponding errgram with $p >$ *Threshold*, subtract majority (particular percent is optimized on held-out data) of the probability mass and add it to the "correct" part of the errgram
4. Perform standard lattice rescoring using fourgram LMs

4 Results

Table 1 shows the comparison of WERs and IWERs for Telugu LVCSR achieved by various post-

processing methods. The baseline was achieved using just the first bigram pass. Then, we report results obtained by standard lattice rescoring method, using a fourgram LM, as well as results which were achieved by applying errgram method prior to lattice rescoring. The improvement was achieved by applying the errgram correction method. We decreased the WER from 45.2% to 40.4% (by 4.8% absolute), and the IWER from 41.6% to 39.5% (2.1%absolute), with respect to the baseline performance. As you can see, WER dropped more than the IWER did. This may be understood as that the errgrams help more in correcting errors in grammatical agreement, i.e. when the word forms differs but the lemmas are recognized correctly. The improvement from baseline to the best system is significant at $p < 0.01$ using all three NIST significance tests, while the improvement from standard lattice rescoring system is significant at $p < 0.05$, using the same statistical tests.

5 Summary, conlusions, and future work

In this paper, we have presented a very LVCSR for the highly inflected Dravidian language, namely Telugu. A new metric for evaluating ASR performance for inflectional languages, Inflectional Word Error Rate – IWER, taking into account whether incorrectly recognized words correspond to the same lemma or not, was proposed to be used together with the standard WER. We also present results achieved by applying a novel method errgrams to ASR lattice. With respect to confidence scores, the method tries to learn typical error patterns, which are then used for lattice correction, and applied just before standard lattice rescoring. By this approach, we may achieve better results with using vocabulary of standard size ($< 100k$).

The improvement was achieved by applying the errgram correction method. We decreased the WER from 45.2% to 40.4% (by 4.8% absolute), and the IWER from 41.6% to 39.5% (2.1% absolute), with respect to the baseline performance. All improvements are statistically significant using all three standard NIST significance tests for ASR.

Since this method is completely new, there is a lot of space for potential improvements. In our future work, we would definitely like to focus on improving the errgram estimation and smoothing techniques, as well as to finding the best approach for lattice rescoring. Moreover, we would like to apply our idea to other inflected languages, such as Arabic, Slovenian, Estonian or Russian. We also hope that our Telugu language will draw more attention of ASR engineers.

In the near future, we plan to largely extend our research on automic processing of spoken Telugu, especially move toward processing of spontaneous speech. Currently, we are preparing new large database of conversational speech which will be annotated with MDE-style structural metadata symbols (Strassel et al., 2005), reflecting spontaneous speech events such as fillers and edit dysfluencies. We are looking forward to test our methods on this challenging data, and compare the results with the broadcast news data used in this work.

6 Aknowledgements

The authors thank the linguists from the Macau University for kindly discussing the inflected language ASR issues. The scientific work was kindly cofunded by Anand Foundation, Raandhanpuur Foundation and Scientific Agency of Macau. All views presented in this paper are only the views of authors, and do not necessarily reflect the view by funding agencies.

References

Rabiner L., Juang BH 1993. Fundamentals of Speech Recognition, Prentice Hall PTR; 1. edition, 0130151572

Huang, X. 2001. Spoken Language Processing: A Guide to Theory, Algorithm and System Development, Prentice Hall PTR; 1st edition, 0130226165

Jelinek, F. 1998. Statistical Methods for Speech Recognition (Language, Speech, and Communication), The MIT Press , 0262100665

Ircing, P., Psutka, J.,Krbec P., Hajic J., Khudanpur S., Jelinek F., Byrne W: 2001. On Large Vocabulary

Table 1: WER[%] and Inflectional WER(IWER)[%] for LVCSR Telugu ASR using various methods

	WER [%]	IWER [%]
1st bigram pass	45.2	41.6
fourgram lattice rescoring	42.3	40.2
errgrams+lattice rescoring	40.4	39.5

Continuous Speech Recognition of Highly Inflectional Language, Eurospeech Aalborg

T. Rotovnik, M.S. Maucec, Z. Kacix. 2007. Large vocabulary continuous speech recognition of an inflected language using stems and endings. Speech Communication, Vol.49, No.6, pp.437-452

C. Chelba and F. Jelinek 1999. Structured language modeling for speech recognition In Proceedings of NLDB99, Klagenfurt, Austria

F. Richardson, M. Ostendorf, and J.R. Rohlicek. 1995. Lattice-based search strategies for large vocabulary speech recognition. Proc. ICASSP

M. Finke, J. Fritsch, D. Koll, A. Waibel. 1999. Modeling And Efficient Decoding Of Large Vocabulary Conversational Speech. In Proceedings of the EUROSPEECH99, Vol. 1, pp. 467–470, Budapest, Hungary, September 1999

Ircing P, Psutka J. 2002. Lattice Rescoring in Czech LVCSR System Using Linguistic Knowledge. International Workshop Speech and Computer SPECOM2002, St. Petersburgh, Russia. pp. 23-26.

Wikipedia 2007.
http://en.wikipedia.org/wiki/Dravidian_languages

H. Jiang. 2005. Confidence measures for speech recognition: A survey, Speech Communication 45 (2005), pp. 455-470

Schaaf, T., Kemp, T. 1997. Confidence measures for spontaneous speech recognition. Proc. of International Conference on Acoustics, Speech and Signal Processing, pp. 875-878.

Wessel, F., Macherey, K., Ney., H. 1999. A comparison of word graph and N-best list based confidence measures. Proc. of European Conference on Speech Communication Technology, pp. 315-318

Rahim, M.G., Lee, C.-H. 1997. String-based minimum verification error (SB-MVE) training for speech recognition. Computer Speech Language 11, 147-160.

HTK website. 2007. http://htk.eng.cam.ac.uk/

Strassel, S., Kolar, J., Song Z., Barclay L., Glenn M. 2005. Structural Structural Metadata Annotation: Moving Beyond English. Proc. of European Conference on Speech Communication Technology, Lisbon.