Syllable Lattices as a Basis for a Children's Speech Reading Tracker

Daniel Bolanos [1, 2], *Wayne Ward* [1], *Sarel Van Vuuren*[1] *and Javier Garrido*[2]

[1] Center for Spoken Language Research, University of Colorado at Boulder, USA
[2] HCTLab-Escuela Politécnica Superior, Universidad Autónoma de Madrid, SPAIN
e-mail: {daniel.bolanos, javier.garrido}@uam.es, {whw, sarel}@cslr.colorado.edu

Abstract

In this paper we present an algorithm that makes use of information contained in syllable lattices to significantly reduce the classification error rate of a children's speech reading tracker. The task is to verify whether each word in a reference string was actually spoken. A syllable graph is generated from the reference word string to represent acceptable pronunciation alternatives. A syllable based continuous speech recognizer is used to generate a syllable lattice. The best alignment between the reference graph and the syllable lattice is determined using a dynamic programming algorithm. The speech vectors that are aligned with each syllable are used as features for Support Vector Machine classifiers that accept or reject each syllable in the aligned path.

Experimental results over three children's speech corpora show that this algorithm can substantially reduce the classification error rate over the standard word based tracker and over a simple best-path syllable based tracker.

Index Terms: speech recognition, children's speech, reading tracker, token passing, SVM.

1. Introduction

Children's speech computer-based reading tracking systems have proven to be an effective and low cost way to teach beginning and early readers to read accurately and fluently. Oral reading tracking systems generally use a speech recognizer to determine whether a child has read a known text passage correctly. By aligning the sequence of word hypothesis produced by the recognizer against the reference text, i.e. the text passage to be read, text passage's words can be tagged as correctly or incorrectly read. Additional information, such as confidence scores, can be attached to each word.

When dealing with children's continuous speech recognition, it is difficult to obtain satisfactory acoustic models due to the great variability of children's speech. In the context of a reading tracker is possible to cope with lack of adequate acoustic models by taking advantage of adaptive language models that reflect what the child is supposed to be reading [1]. While the use of such adaptive language models improves the Word Error Rate of the recognizer, the technique makes rejection of errors difficult.

This paper presents a method that significantly reduces the Classification Error Rate (CER) of a state-of-the-art word-based reading tracker by avoiding the drawbacks inherent to the use of dynamic adaptive language models. The method proposed comprises three steps:

1) A syllable lattice is generated using a syllable trigram language model.
2) The reference text is represented as a syllable graph. A Dynamic Programming algorithm uses Minimum Edit Distance to traverse the lattice and find the path closest to the reference graph. This path gives an alignment between phones (comprising syllables) in a reference path and speech vectors in the input.
3) The speech vectors that are aligned with each phone are used as features for Support Vector Machine (SVM) classifiers that classify each frame as belonging to the reference phone or not. These frame level decisions are combined to make syllable and then word level accept/reject decisions.

Evaluation is based on Classification Error Rate, the rate at which words in the reference string are correctly classified as present or not.

2. Background

In this section we describe the systems used as baselines in the experimental evaluation of the proposed algorithm.

2.1. Word-based Reading Tracker

The baseline system is the word-based reading tracker algorithm used in the Foundations to Literacy system [1]. The Sonic speech recognizer [2] is used to produce a single best hypothesis for what the child read. The recognizer uses a word lexicon and word based trigram language model. The reference string is aligned to the recognition hypothesis. Those words in the reference string that are aligned to the same word in the hypothesis are classified as present. Words in the reference string that are aligned to a different word in the hypothesis are classified as not present. Due to the large variability in children's speech, acoustic models are typically of relatively poor quality, so stronger language models are used to compensate. In this application, the text to be read is known. Very low perplexity adaptive language models based on the words in the text that are currently being read are used during the decoding process. Position-sensitive trigram language models are generated, partitioning the training text into overlapping regions [1]. After decoding each utterance, the position-sensitive language model that gives a higher probability to the last recognized words is selected for the first pass decoding of the subsequent utterance. This results in a very low perplexity language model. Since most words in the corpus are read correctly, using this low perplexity language model and weighting it heavily produces the lowest Word Error Rate, compared to hand generated transcripts of the speech. However, this strategy produces a very sparse word lattice and compromises the ability of the system to assign confidence scores to each word.

2.2. Syllable-based Reading Tracker

One solution to avoid the problems caused by lexical and language model constraints is to use a decoder based on subword units, phones or syllables. The decoder is then more able to produce sequences that match the acoustics of the signal. The lattice contents can then be aligned against the reference string to produce features for classifying the presence or absence of each word. Our analysis of children's speech corpora showed that most of the disfluencies like repetitions and self-corrections occur at the syllable level. For example *"when his **ow-** owners got him as **a** a puppy"*. Since working at the syllable level seems to be very promising for disfluency detection, we built a syllable based speech recognizer using Sonic [2]. The lexicon of the system consisted of a set of 2314 syllables, each with a sequence of one or more phones as its reference pronunciation. Context dependent phone models were trained for a 55-phoneme symbol set. The syllabification process was carried out over a multiple pronunciations version of the lexicon using the syllabification software available from NIST [3]. A back-off trigram syllable language model was trained in which multiple pronunciations of each word are used in the training process. This system is then used to generate a syllable lattice for each utterance.

3. Path Alignment

The reference text, i.e. the text passage that the child supposedly read, is represented in the form of a graph of syllables that encodes the different pronunciations observed for the reference words in the training data. An example of these graphs can be seen in Figure 1. We evaluated two strategies for using paths in the syllable lattice:

1. Comparing the best path (highest score) in the lattice to the reference graph.
2. Finding the path in the syllable lattice that is closest, in terms of Minimum Edit Distance, to the reference text.

The algorithm for finding the MED path is based on the token passing paradigm that is often used in keyword spotting phone-lattice search approaches [4][5]. In the case of aligning a lattice, that is just an acyclic oriented graph, against a reference graph, the number of token expansions grows exponentially. For this reason, and given that a reading tracker is by nature a real time application, beam pruning techniques are applied to reduce the search.

3.1. Lattice Generation

The lattice generation process is carried out using a Viterbi syllable recognizer that decodes each utterance into a syllable lattice. These lattices are transformed into syllable graphs whose density is adjusted as a tradeoff between the detection rate and the real time performance of the algorithm. The decoding search parameters (insertion penalty, language model weight and beam width) were optimized using a development set.

For efficiency of alignment, lattices are constructed so that only one end node </s> is present and no node in the lattice is allowed to have more than one descendant associated with the same syllable.

3.2. Alignment Algorithm

This section describes the algorithm used to find the path in the lattice that is closest match to some path in the reference graph. Let $[t,s,H]$ be defined as a token where t represents the minimum number of MED operations, i.e. insertions, deletions, substitutions and matches necessary to propagate a token to the lattice node where the token is held, and with same reference syllable in the last position of the token history. The value s is the token accumulated MED score and H is the token history. The value t can be easily computed dynamically and will be used to prune tokens globally instead of using the typical node localized V-best token merging procedure. Since tokens with same value of t represent comparable partial alignments of paths in the lattice vs. paths in the reference graph, they can be compared by score and pruned using a beam.

Let $D(G,n)$ be defined as a function that returns the list of nodes that are descendants of the node n in the reference graph G. The alignment algorithm is:

```
1. $t' = 0$
2. Pass an empty token to the lattice root element (<s>).
3. While there are tokens in other nodes rather than the final node of the lattice do:
    a. For each node $i$ in the lattice do:
        i. Merge node tokens with history ending at the same reference node keeping only the best scoring one.
        ii. For each token in the node with $t = t'$ do TokenPropagation($D(G,n), C_i, C_d, C_s$) where $n$ is the last reference node present in the history of token $i$, $C_i$ is the insertion cost function, $C_d$ is the deletion cost function and $C_s$ is the substitution cost function.
    b. $t' = t' + 1$
    c. Apply beam pruning for all tokens in the lattice with $t = t'$ using a beam-width of $B$.
4. Take the best scoring token in the final state (</s>) and return its history as the final hypothesis.
```

4. Syllable Rejection

Once the best matching path from the lattice (relative to a path in the reference graph) is found, it is used to reject syllables in the reference string. If aligned syllables agree, then the syllable in the reference is classified as present. If the aligned syllables do not agree, the syllable in the reference is classified as absent. Classifications produced by this procedure typically have a low False Rejection rate but are subject to higher False Acceptance rates. This motivates the use of a confidence measure acting as a syllable rejection mechanism. Support Vector Machines have been used successfully in recent years for phonetic classification [6].

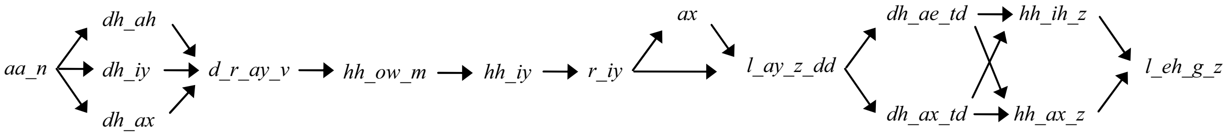

Figure 1. Syllable graph corresponding to the utterance "*...on the drive home he realized that his legs...*".

An SVM learns the decision boundary between samples belonging to two classes by mapping the training sample vectors into a higher dimensional space and then determining an optimal separating hyper-plane [7]. In our case, the alignment of paths in the lattice to paths in the reference string associates a sequence of vectors in the speech signal with each syllable in the reference string. These vectors are used by SVM classifiers to accept or reject the syllable.

4.1. Training and Parameter Selection

For every speech segment present in the training set, 39-dimensional feature vectors, consisting of 12 Mel Frequency Cepstral Coefficients and energy plus first and second order derivatives, have been extracted. The children's speech corpora available are tagged at the word level only so phone boundaries are obtained using a Viterbi-based phonetic alignment against the transcriptions.

SVM's are well suited for two-class separation tasks, however for n-class (n>2) separation tasks, like building a phonetic classifier, n SVMs need to be trained. In this case we have selected a "one vs. all" approach in which one SVM classifier [8] is trained for each of the 55 phonetic symbols used. For the training of each SVM, half of the data points (positive samples) belong to the actual phone while the rest belong to the remaining phones (negative samples).

A radial basis function (RBF) kernel is used for which the parameters *C (cost)* and γ are estimated over the training set doing a "grid-search" process using 5-fold cross validation.

4.2. Confidence measures estimation

For each syllable in a given reference string, a confidence measure (1) is calculated using an arithmetic mean of the confidence measures calculated for each of its phones (2). The later is computed as the posterior probability of a phone given the acoustic observation sequence *O* to which it is aligned.

$$CM_s = \frac{1}{N}\sum_{i=1}^{N} CM_{p_i} \qquad (1)$$

$$CM_{p_i} = P(p_i \mid O) = \frac{P(O \mid p_i)P(p_i)}{\sum_j P(O \mid p_j)P(p_j)} \qquad (2)$$

The probability *P(O|p_i)* is estimated using the trained set of SVMs. To estimate this probability, each speech vector aligned with a phone is passed to the one vs. all SVM classifier associated with the phone. The SVM returns either a positive (vector is an instance of the phone) or negative classification. The probability is the ratio of the number of positive classifications to the total numbers of vectors aligned with the phone. Once (1) is calculated, the decision whether to classify the syllable as "reliable" or "unreliable" is taken based on a fixed threshold *η* previously trained (3).

$$s = \begin{cases} accept \text{ if } CM_s(s) > \eta \\ reject \text{ otherwise} \end{cases} \qquad (3)$$

A final step of syllables to words mapping is necessary. In this step sometimes not all the syllables of a given reference word are present in the hypothesis. These cases have been solved by considering "partially read" words as correctly read when at least 50% of the corresponding syllables were present, otherwise the word is classified as incorrectly read.

5. Experimental Procedure

Experiments were carried out to evaluate the performance of the alignment algorithm proposed and to compare the resulting syllable based system against a tuned word based one.

5.1. Speech material

We present experimental results on a corpus composed of the CU Prompted and Read Children's Speech Corpus [9], the OGI Kid's speech corpus [10] and the CU Read and Summarized Story Corpus [11]. Children's acoustic models are estimated from over 62 hours of audio from the CU Prompted and Read Children's Speech Corpus, the OGI Kids' speech corpus grade K through 5, and data from 1st and 2nd graders found in the CU Read and Summarized Story Corpus. Reading tracking systems are evaluated on the 106 3rd, 4th and 5th graders from the CU Read and Summarized Story Corpus, each speaker reading one out of 10 stories with an average length of 1,054 words per story.

5.2. Evaluation procedure

To evaluate the performance of a reading tracker we use the Classification Error Rate (CER) defined as the percent of words in the reference text that have been correctly tagged, as present (correctly) or absent (incorrectly) read, by the reading tracker. The reference classifications are generated by aligning the reference (prompt) string against hand generated transcriptions of the corresponding speech. Each word in the reference string aligned with the same word in the hand transcription is marked as present (read correctly). Words in the reference not aligned with the same word in the hand transcript are marked as absent (not read correctly). This classification string is the gold standard against which automatic classification output is scored.

To score an automatically classified string, the string is aligned with the gold standard and the classifications for each word are compared. The CER is the percent of words for which the classifications do not agree.

5.3. Results

The first experiment conducted compares performance of a reading tracking system using different parameterizations of the alignment algorithm. These are compared against two tuned baseline systems, the word-based reading tracker described in section 2.1 and a syllable-based reading tracker that uses the single best scoring path from the syllable lattice. Each configuration in the path alignment algorithm is represented by the following notation PAA[SGD,C_s,*B*] where SGD is the syllable graph density, defined as the total number of syllable graph edges divided by the number of actually spoken syllables. *B* is the pruning beamwidth and C_s is the MED cost for substitutions. The value of C_i and C_d is fixed and equal to 1 in all the configurations.

For comparison, performance given by aligning the lattices against the hand transcribed text is shown. This number gives an indication of lattice quality. Performance of this system, marked with an asterisk, shows the potential of lattice-search in the CER minimization task.

Results can be seen in table 1, where performance is measured in terms of CER, false acceptations and false rejections. Note that the parameter combinations are chosen so that the corresponding system is fast enough to perform in real time.

Configuration	CER	FA	FR
syllable-based (best path)	4.70	2.15	2.55
PAA[8,1,2]	4.61	2.91	1.7
PAA[8,1.1,2]	4.17	2.70	1.47
PAA[8,1,3]	4.22	2.65	1.57
PAA[8,1.1,3]	3.97	2.57	1.40
PAA[12,1.1,3]	3.91	2.53	1.38
PAA[8,1,3]*	3.03	1.38	1.65
word-based (best path)	4.48	1.88	2.60

Table 1. Performance of different parameterizations of the algorithm.

Results show that the alignment algorithm outperforms (in CER) both syllable-based and word-based baseline systems. The number of false rejections are reduced at the cost of an increase in the number of false acceptances. Different parameterizations of the algorithm show that, as expected, increasing the lattice density and the beamwidth produce a reduction in the CER. In addition, penalizing substitutions higher than deletions and insertions performs better. Note that these results do not use the SVM classifiers to reject syllables with poor acoustic matches. The hope is that using the SVMs will reduce the False Acceptance rate.

Finally, an experiment to evaluate the discriminative power of the syllable rejection module and how it complements the alignment algorithm by reducing the number of false acceptances was carried out. Each of the syllables present in the lattice path produced by the algorithm is scored using the confidence measure CM_s described in section 4.2 and tagged as “reliable” or “unreliable” using the fixed threshold η. Subsequently, when aligning the hypothesis against the reference during the CER calculation process, “unreliable” syllables will produce incorrectly read words.

Results for this experiment are shown in figure 2 in the form of an ROC curve.

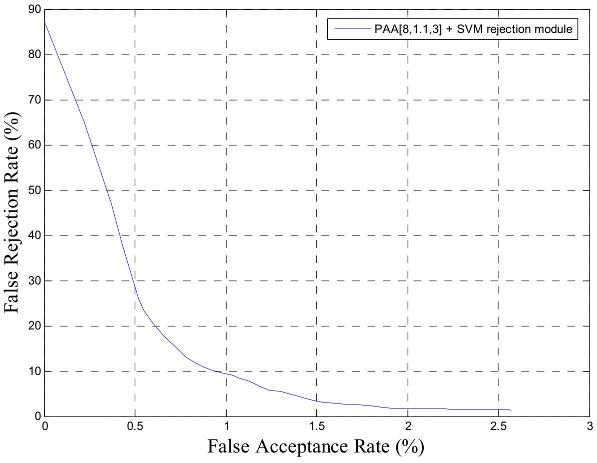

Figure 2. ROC curve for the system PAA[8,1.1,3] after applying the syllable rejection mechanism.

The SVM rejection algorithm was applied to alignments produced by the PAA[8,1.1,3] condition. The initial performance of this condition was CER= 3.97, FA= 2.57, FR= 1.40. As the threshold is increased, the SVM increasingly reduces the FA rate at the cost of increasing the FR rate. The minimum CER is 3.67 which has a FA= 1.99 and FR= 1.68.

6. Conclusions

We have presented a mechanism for a reading tracker task that uses syllable based decoding to avoid the problems associated with using word lexicons and word based language models. The model presented is very preliminary in that it has not been optimized in a number of ways: No allowance has been made for disfluencies in the syllable language model, no disfluency acoustic models are used, the method of estimating the phone probabilities from the individual speech vector probabilities and the syllable and word confidences from these is clearly sub-optimal. Also there is much information in the dense syllable lattices that is not being used. Even so, the method seems very promising in that it reduced CER over a current state-of-the-art word based system from 4.48 to 3.67, a relative reduction in error rate of 18%. It also provides a mechanism for trading off False Alarms against False Rejections to optimize these for a specific application. This general technique can be applied to verifying any hypothesis, not just reading tracking tasks. For example, it can be used to assign confidence scores to words in hypotheses generated by a speech recognizer.

7. References

[1] A. Hagen, B. Pellom, S. Van Vuuren and R. Cole “Advances in Children’s Speech Recognition within an interactive literacy tutor”, HLT 2004.

[2] B. Pellom, "Sonic: The University of Colorado Continuous Speech Recognizer", Technical Report TR-CSLR-2001-01, CSLR, University of Colorado, March 2001.

[3] Fisher, B., tsylb2-1.1 syllabification software, National Institute of Standards and Technology, http://www.nist.gov/speech/, 1996.

[4] S. J. Young and M. G. Brown, “Acoustic indexing for multimedia retrieval and browsing,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1997, vol. 1, pp. 199–202.

[5] K. Thambiratnam and S. Sridharan, “Rapid yet accurate speech indexing using dynamic match lattice spotting”, *IEEE Transactions on Audio, Speech and Language Processing.*, January 2007,vol. 15,no 1.

[6] Clarkson, P. and Moreno, P. J., “On the use of support vector machines for phonetic classification,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.,* 1999.

[7] Vapnick, V. T*he Nature of statistical Learning Theory.* Springer-Verlag, New York, 1995.

[8] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, 2001.

[9] R. Cole, P. Hossom, and B. Pellom. University of Colorado prompted and read children's speech corpus. Technical Report TR-CSLR-2006-02, University of Colorado, 2006.

[10] K. Shobaki, J.-P. Hosom, and R. Cole. The OGI kids' speech corpus and recognizers. In *6th International Conference on Spoken Language Processing*, Beijing, China, 2000.

[11] R. Cole and B. Pellom. University of Colorado read and summarized story corpus. Technical Report TR-CSLR-2006-03, University of Colorado, 2006.