SEGMENTATION USING A MAXIMUM ENTROPY APPROACH

K. Papineni *S. Dharanipragada*

IBM T.J. Watson Research Center
PO Box 218, Yorktown Heights, NY, 10598, USA
kishore@watson.ibm.com dsatya@watson.ibm.com

ABSTRACT

Consider generating phonetic baseforms from orthographic spellings. Availability of a segmentation (grouping) of the characters can be exploited to achieve better phonetic translation. We are interested in building segmentation models without using explicit segmentation or alignment information during training. The heart of our segmentation algorithm is a conditional probabilistic model that predicts whether there are less, equal, or more phones than characters in the word. We use just this contraction-expansion information on whole words for training the model. The model has three components: a prior model, a set of features, and weights of the features. The features are selected and weights assigned in maximum entropy framework. Even though the model is trained on whole words, we effectively localize it on substrings to induce segmentation of the word to be segmented. Segmentation is also aided by considering substrings in both forward and backward directions.

1. INTRODUCTION

In many systems such as a speech recognition system or a spelling to phonetic baseform generation system we generate an output sequence of symbols (words or phones) from an input sequence of symbols (acoustic vectors or characters). Typically, the symbol spaces are different and the lengths of the input and output sequences are different. Here, unlike in a natural language translation system, segments of input symbols are chronologically aligned to segments in the output sequence. Clearly, segmentation of one stream is induced by the other stream.

One of the goals in these systems is to discover the segmentation in the input symbol stream which can be exploited in the generation of the output sequence. For example, in the phonetic baseform generation problem, we would like to generate the segmentations "d ou bt f u ll y", "th o r ough". Even though the output sequences induce the segmentation on the input sequences, we do not have the output sequence available during segmentation of a given input sequence. We are interested in building segmentation models without any explicit segmentation or alignment information in training data. We restrict our attention to the specific problem of segmentation of orthographic spellings that can later be used for phonetic translation. There are many ways to generate segmentation and phonetic translation, e.g., see [1] - [2]. We are interested in building models using as little detailed information as possible.

We start with a corpus of pairs of words and their phonetic baseforms. We then create a training data of (history, future) pairs. History is a full word, and future is whether there are more, equal, or less phones in the phonetic baseform. That is, the future space consists of just three symbols: $\{-1, 0, 1\}$. This is a macro-level description of the phonetic translation data: the training data indicates only whether or not there is a contraction, but not how many contractions or where they occur. The same is true for expansions. Sometimes, a word can contain one contraction that is offset by an expansion; e.g. the word "excess" has an expansion due to "x"(as K S) and a contraction due to "ss". The macro-level information is that there is neither a contraction nor an expansion for this word. However, our goal is to discover the local expansion and the contraction to induce the segmentation "e x c e ss".

The idea is to use such a simple global description to effectively deduce local details such as segmentation, and if necessary even alignments. We first build a probabilistic model to predict one of the three futures, given a string of characters. This model, described in the next section, predicts whether there is a contraction or not in any substring of the word. The goal is to use it on many substrings to locate the contractions in the full word. The novelty of this work is that we train the model on full words, but use it on substrings of a given word to induce segmentation of the word.

There are two issues: 1. expansions and contractions mask each other in a string. 2. From the model's point of view, multiple contractions appear as one, perhaps with stronger probability. The number of contractions cannot be inferred from the value of the contraction probability of the full word.

The crux of the problem lies in effective localization by a proper choice of substrings of the word as the conditioning variable for the model. In this paper, we present an approach to localization and give initial results.

2. THE CONTRACTION MODEL

Our goal is to build a conditional probabilistic model to predict if there are more (expansion), less (contraction)

or equal number of phones than characters in the orthographic spelling. In this conditional model $P(f|h)$, the history h is the sequence of input characters and the future, f is either 0, 1 or -1.

Our conditional model, $P(f|h)$, has three components: a fixed prior probability model, $P_0(f|h)$, a vector of binary feature functions, $\phi(f,h)$, and their corresponding weights, λ:

$$P(f|h) = P_0(f|h)\frac{e^{\lambda\phi(h,f)}}{\sum_f P_0(f|h)e^{\lambda\phi(h,f)}} \quad (1)$$

The prior model captures prior knowledge, if any. If there is no prior knowledge, one can set the prior to be uniform. Given a set of features, λ is chosen to maximize the likelihood of a training corpus, using the Improved Iterative Scaling algorithm [3]. If the prior is uniform, the optimal model also maximizes conditional entropy subject to the constraint that the model's expectation of the feature vector matches that of the empirical distribution. Our prior is uniform. Features themselves are also selected based on their contribution to the likelihood.

We next describe the training data creation, and feature creation.

2.1. Training data

We obtain our training data from a corpus of matched pairs of input and output symbol sequences without explicit segmentation or alignments. In the baseform problem we have a corpus of orthographic spellings and their phonetic baseforms. From this corpus we generate training data as (history, future) pairs. Here, history is the orthographic spelling and future is 1 if the number of input characters is greater than the number of phones, -1 if the number of input characters is less than the number of phones and 0 else. We have about 30000 such pairs in the training data.

2.2. Features

For simplicity we consider only binary-valued feature functions here. Features map (history, future) pairs to 0 or 1 and are essentially questions on presence of character n-grams in the history and whether the future is 1 or -1. The features are automatically generated by considering combinations of all futures with all character n-grams with count 3 or more. We have about 12000 features in the pool. We start with the uniform prior and rank each feature in the pool by its contribution to the likelihood on the training data. We add the top 2 features to the current model and train the resulting model. We then rank the remaining pool with respect to the current model and add top 2 features to the model. This iterative feature selection is continued until the relative contribution of the additional features is insignificant. Some of the top features selected are displayed below, along with their weights.

Char n-gram	Future	α
ng	contraction	11
ll	contraction	47
ch	contraction	73
th	contraction	42
ex	expansion	37

3. SEGMENTATION PROCEDURE

Segmentation means breaking up the given word into groups of consecutive characters. This is equivalent to inserting spaces at appropriate positions in the word. However, we can also view segmentation as locating contractions in a (space-separated) stream of characters comprising the word. The contraction model can predict the probabilities of expansion and contraction for any substring of the word. A typical contraction probability profile on substrings of increasing length from the left for the word "faithfully" is shown in Figure 1 (solid line). That is, the solid line displays the contraction probabilities for "f", "fa", "fai", and so on.

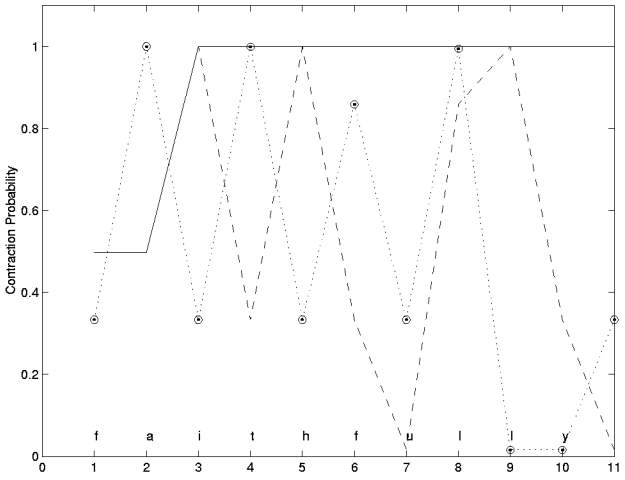

Figure 1: Contraction probability profiles

Clearly, the model predicts the contraction "ai"; however, the contractions "th" and "ll" cannot be so easily deciphered from the profile. This is the case of multiple contractions appearing as one. The increase in contraction probability after the first contraction can be very small. We solve the above problem by discarding the characters leading upto the previous contraction from the conditioning variable of the model. In the figure, the dashed line is the profile of contraction probabilities obtained this way. That is, the dashed line displays the contraction probabilities for "f", "fa", "fai", "it", "ith", "hf" and so on. The modified contraction probability peaks are at positions 3, 5, 8, and 9. If we place contractions at these positions we get the segmentation "f ai th f ull y". There is a spurious contraction between 'u' and 'l'.

Locating contractions using the contraction probability profile computed in the forward direction only is prone

to error. We reduce these spurious contraction errors by validating contractions proposed as above with those proposed by a contraction profile computed in the backward direction. The backward profile is computed on substrings "y" (plotted at position 10), "ly" (position 9), "lly" (8), "ul" (7), "ful" (6) and so on. This is shown as a dotted line in Figure 1. Using the common contractions proposed by the profiles, we get the correct segmentation "f ai th f u ll y". We also ensure that expansions do not mask contractions by similarly discarding characters leading upto the previous expansion from the conditioning variable of the model.

4. RESULTS

The data is divided into four parts. Training data consists of about 30000 events, held-out data of 100 events, development data of 1000 events, and test data of another 1000 events. While training data does not have any segmentation information, all the words in held-out, development, and test sets are segmented by hand. The development data also contains the contraction-expansion information for each of the words.

The basic contraction model predicts the future $\{-1, 0, 1\}$ with an accuracy of 93.8% at the word level on the development set.

The held-out set was used to select appropriate thresholds to determine contraction, and to discard characters based on contraction probability. The same thresholds were used subsequently for models of all sizes.

The segmentation performance is reported using the standard miss and false-alarm rates. The total error rate (TER) is the ratio of sum of misses and false alarms to the number of all possible errors.

The performance on the development set of models of various sizes (number of features) is shown below.

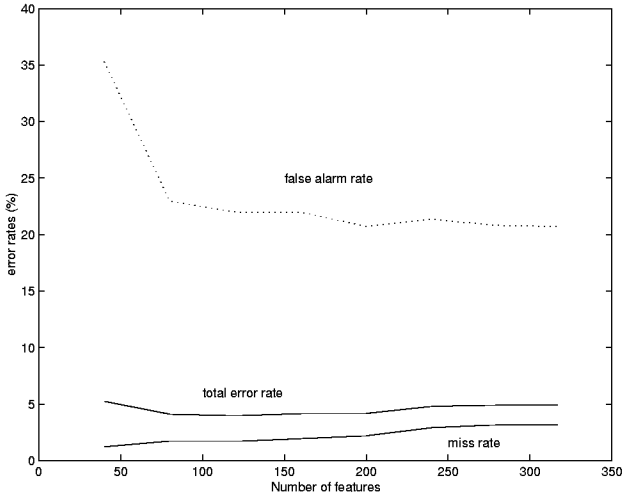

Figure 2: Miss, false alarm, and total error rates

We selected the model with 120 features as our preferred model (M120). We also used it to segment using only the forward contraction probability profile and also using only the backward contraction probability profile. The results on the development set are shown below:

Direction	Miss	FalseAlarm	TER
Forward Only	4.06	23.5	5.91
Backward Only	4.0	17.92	5.18
Both	1.7	22.0	3.97

We used M120 to segment the test set. The results are shown below along with those of a few other schemes. One scheme (Gem) simply segments at each character except between geminations. Another (Rand) segments randomly. The third (Rules) is a rule-based segmenter with about 30 rules.

Model	Miss	FalseAlarm	TER
Rand	15.9	83.4	22.6
Gem	0.03	75.6	9.8
Rules	0.33	9.75	1.5
M120	1.85	21.1	4.1

5. CONCLUSIONS

We considered building a segmentation model based on as little detailed information in training data as possible. The segmentation algorithm presented here uses a basic contraction-expansion prediction model that is developed in maximum entropy framework.

6. REFERENCES

1. J. M. Lucassen and R. L. Mercer, "An information-theoretic approach to the automatic determination of phonetic baseforms," Proceedings of the ICASSP-84, pp. 42.5.1-42.5.4, 1984.
2. S. Deligne et al, "Variable-length sequence matching for phonetic transcription using joint multigrams," Proceedings of Eurospeech-95, vol 3, pp. 2243-2246, Sept 1995.
3. S. Della Pietra et al, "Inducing features of random fields," CMU Technical Report CMU-CS-95-144, 1995.