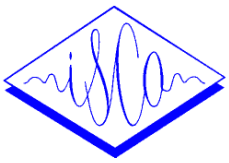

4th European Conference on
Speech Communication and Technology
EUROSPEECH '95
Madrid, Spain, September 18-21, 1995

THE APPLICATION OF DYNAMIC PROGRAMMING TECHNIQUES TO NON-WORD BASED TOPIC SPOTTING

P. Nowell & R.K. Moore
e-mail: nowell@signal.dra.hmg.gb
e-mail: moore@signal.dra.hmg.gb
DRA Malvern
Malvern, Worcs.
ENGLAND

ABSTRACT

This paper describes the application of dynamic programming (DP) techniques to the problems of building and testing non-word based topic spotters. We use a DP algorithm to find sets of similar phoneme sequence fragments, which we call DP-ngrams, and to detect their occurrences in the training and test data. The ability to use partial matches means that that the fragments are longer and more meaningful than the phoneme ngrams that have been tried previously. Detection probabilities of over 90% with less than 10% probability of a false alarm are achieved for seven target categories. Reports about bridges and pontoons are detected with 90% probability at a probability of false alarm of less than 1%.

10.21437/Eurospeech.1995-356

1. INTRODUCTION

Topic spotting is usually performed by first determining a small number of keywords that are best able to discriminate between the topic and non-topic speech [1]. Occurrences of the keywords are detected in the test data and 'usefulness' scores reflecting the discriminative power of the keyword are accumulated. The total is thresholded at regular intervals to give an indication as to the presence or absence of the topic.

One disadvantage to this approach is that orthographically transcribed material is needed in order to determine the best keywords and their usefulness scores. Transcribing speech by hand is laborious and time consuming whilst automatically generating transcriptions by means of a large vocabulary speech recogniser [2] is prone to error. Automatic speech recognisers also need a pronunciation dictionary containing all of the words to be recognised. Unfortunately, the most useful keywords are by definition unusual and are therefore are less likely to be in a general purpose dictionary.

A word based approach also pre-supposes that words are the best units for topic spotting. Any potentially useful information below the level of whole words is not available to the topic spotter. Useful information at the sub-word level could include information such as the speaker and their speaking style. Whilst not strictly topic spotting the use of such information can also be useful in classifying the speech signal as being 'wanted' or 'unwanted'.

Our first experiments in building non-word based topic spotters [3] used the output of a phoneme transcriber, thereby eliminating the need for orthographic transcriptions and/or pronunciation dictionaries. The key-fragments were determined by analysing the output of the phoneme transcriber to generate the set of context independent ngrams (CI-ngrams). Discriminative 'usefulness' scores were calculated for the CI-ngrams and the accumulated usefulness scores on the test data were thresholded.

However, the algorithm which extracts CI-ngrams from the training data uses conventional string matching techniques to find identical sequences of phonemes. The requirement for exact matches means that the sequences that are found tend to be short due to a combination of errors made by the phoneme transcriber and variations in pronunciation. Even if words and phrases are repeated a number times it is unlikely that the underlying phoneme sequence appears as two occurrences of the same CI-ngram. A single word or phrase typically gives rise to a number of short CI-ngrams and the usefulness of any one is low since the semantic content is small.

This paper describes the application of dynamic programming techniques to overcome these limitations. Instead of looking for exact matches when extracting phoneme sequence fragments we look for partial matches where portions of the two transcriptions are similar but not necessarily identical. Such an approach is able to recognise fragments representing the same underlying word or phrase as being equivalent even though the phoneme transcriptions of those words or phrases differ. The output contains longer fragments which are more semantically meaningful and of more use for a topic spotter.

2. A DP ALGORITHM TO FIND SIMILAR PORTIONS OF TWO PHONEME SEQUENCES

The DP algorithm outlined below was originally developed to find similar (homologous) portions of two gene sequences [4]. This algorithm uses dynamic programming techniques to find optimal local alignments between two sequences. The optimal local alignments correspond to regions of maximum similarity between the two sequences.

The algorithm resembles those used for template based word spotting which are used to find portions of one sequence (the speech vectors) that are similar to another (the word template). These algorithms typically use negative scores to penalise insertions, deletions, and mis-matches. Putative hits are generated whenever the alignment score exceeds a predefined threshold value.

A direct application of this approach would involve aligning the two speech sequences and then searching for partial alignments with high scores. However, in the absence of any constraint to keep sequences long, a maximum score of zero is always obtainable with trivial alignments of a single item from one sequence with an identical item from the other.

This problem is overcome by replacing the penalties by scores $s(a_i,b_j)$ that can be positive or negative. Positive scores are used for identity substitutions and negative scores are used for insertions, deletions, and mis-substitutions. Values for the various scores can be preset, obtained from a confusion matrix, or calculated on the fly using acoustic distance measures. In the experiments described in this paper fixed values of +1, -2, -1 and -2 were used.

The DP algorithm finds the optimal local alignment between two sequences $a = a_1,\ldots,a_m$ and $b = b_1,\ldots,b_n$ as follows. A quality score q_{ij} is defined as the maximum score of any local alignment upto and including the point (a_i,b_j). This score is calculated for all $i = 1,\ldots,m$ and $j = 1,\ldots,m$ using the recursion :-

$$q_{ij} = \max\begin{cases} q_{i-1,j} + s(a_i,\phi) \\ q_{i,j-1} + s(\phi,b_j) \\ q_{i-1,j-1} + s(a_i,b_j) \\ 0 \end{cases}$$

Note the additional constraint which ensures that the minimum score of any cell in the DP matrix is zero. This has the effect of eliminating the penalties of preceding mis-matches that are not part of the locally optimal alignment.

Backtracking pointers b_{ij} are also maintained at each step according to the expression which maximises q_{ij}.

$$b_{ij} = \begin{cases} i-1, j \\ i, j-1 \\ i-1, j-1 \\ 0,0 \end{cases}$$

The optimal local alignment is recovered by backtracking from the point at which q_{ij} attains it's maximum value and until b_{ij} equals (0,0)

2.1 Finding Multiple Portions

The algorithm as described returns the optimal local alignment representing the portion of the two sequences that are most similar to one another. In general there will be many portions that are sufficiently similar to be interesting. However, repeatedly backtracking from points in order of decreasing similarity scores produces sequences that differ in trivial ways from those that have already been obtained.

This problem is overcome by marking each cell along the optimal local alignment as 'used' whilst backtracking. The DP alignment algorithm can then be repeated to find the next most similar fragment provided that the score of any cell that has been previously marked as used is set to zero. This prevents the same alignment being detected as well as any minor variations. Backtracking proceeds as before from the point with the highest quality score and the cells along the path are marked as used. Repeated iterations produces local alignments in order of decreasing similarity.

2.2 Post-processing

Each invocation of the DP algorithm produces pairs of phoneme sequence fragments in order of decreasing similarity. Many of these pairs contain fragments that are also similar to those of other pairs due to the presence of common words and phrases. For example, if the two sequences contain two examples of the same word then four paired alignments are generated in addition to any fortuitous matches. We therefore group these and other similar fragments are together in an attempt to avoid duplication of effort and multiple counting of fragments representing the same underlying word or phrase.

The grouping of similar fragments is achieved through an agglomerative clustering algorithm [5]. The clustering algorithm starts by assigning each fragment to a single cluster and then repeatedly merges the two clusters and with the lowest distance score. Values for distance scores between clusters are initially calculated

using a DP algorithm similar to the one described. The score of a merged cluster is simply the average of the scores of the two clusters. Clustering proceeds until the lowest distance score exceeds a predefined threshold value (in this case 0).

The clustering algorithm produces a number of clusters each of which contain fragments that are similar to one another. A single most representative fragment (i.e. the centroid of the cluster) is determined by finding the fragment whose total distance score between itself and all of the other fragments in the cluster is the lowest (again using the same DP algorithm). These cluster centroids replace keywords or CI-ngrams as the units of representation in the topic spotter.

Discriminative 'usefulness' scores U_k are then calculated for each cluster centroid c_k using the number of in-topic occurrences n_k and the conditional probabilities of occurrence in the topic and non-topic training data.

$$U_k = n_k \log\left[\frac{p(c_k \mid \text{topic})}{p(c_k \mid \text{non-topic})}\right]$$

The fragments with the highest 'usefulness' scores are selected for use in the topic spottter.

3. TOPIC SPOTTING WITH DP-NGRAMS

A phoneme transcriber was implemented using a continuous speech recogniser, along with 48 sixteen component mixture, task independent, monophone hidden Markov models (HMMs) and a phoneme syntax of the English syllable [6]. The HMMs were reestimated on the male subset of the SRU-SCRIBE database [7] which is sampled at 19.98kHz and transformed to produce 20 channel vectors containing 8 mel-frequency cepstral coefficients, power, variable frame rate count, and the first order differences thereof.

The training and test data for the topic spotter was taken from the airborne reconnaissance mission (ARM) database. Each ARM report describes one of seven different target categories, these being bridges and pontoons, railway depots, airstrips, air defences, communication sites, repair and supply sites, and mechanized infantry. The first 32 reports in each target category were used for training and the last 23 were used for testing, none of the reports in the training set were present in the test set.

Seven experiments were performed where the topic was defined in turn to be one of the seven target categories. The in-topic subset of the training data was processed using the DP algorithm to generate a list of similar fragment pairs. A similarity threshold of 4 was used to limit the amount of computation and the number of fragment pairs to a manageable level. These fragments were then clustered and discriminative 'usefulness' scores were calculated for each of the cluster centroids.

Another DP algorithm was then used to detect fragments in the topic and non-topic training data that were similar to the cluster centroids. The occurrence counts were then used to calculate the discriminative 'usefulness' scores. The fragments were ranked in order of decreasing 'usefulness' and the 100 most useful fragments were selected for use in the topic spotter.

The topic spotter used the same DP algorithm to detect fragments in the test reports that were sufficiently similar to the selected fragments. As fragments were detected the 'usefulness' scores were accumulated and the total score was thresholded at the end of each report. Reports where the score exceeds the threshold were labeled as topic and otherwise as non-topic. The output of the topic spotter was scored to give detection probabilities and probabilities of false alarm for a range of threshold values.

4. RESULTS

A small number of the most 'useful' fragments for spotting reports about bridges and pontoons are listed below.

Rank	CI-ngram	DP-ngram
1	r	#k@p{stif
2	w	pQndtUn#
3	z<	#k@p{s@tintrir
4	u	stVm#k@p{s
5	p{s	k{tiriwQntdU#
6	l	Um##
7	z<?	#k@p{s@
8	k@p	QntUn
9	{sti	griwAntdU
10	sti	riwaIntU#
11	d	QntUn##r
12	U	wQmdtUn#p
13	ntu	tUn#
14	t	@p{s
15	r@U	itifit#
16	{st	@pAs
17	@p{s	Un#r
18	>k@p	iwVntU##
19	@p{	stitrUlOInz
20	vre	p{stifOleIns#

Table 1: Best Fragments Ranked by Bayesian Usefulness for Spotting ARM Reports About Bridges and Pontoons

The fragments are longer and more meaningful than the CI-ngrams used previously. Phoneme sequences representing words and phrases are often recognisable and these corresponded to the words and phrases that one would expect to be useful in spotting the topic. The two most useful fragments correspond to the words 'capacity' and 'pontoon' whilst the phrase(s) defining the target category are represented by fragments ranked 5, 8, 9, 10 etc.

The table below shows the detection probabilities for each the seven target categories when the probability of a false alarm was fixed at 10%. Results for both CI-ngrams and DP-ngrams are shown for comparison.

Target Category	CI-ngrams	DP-ngrams
bridges and pontoons	84%	100%
railway depots	41%	78%
airstrips	78%	100%
air defences	40%	96%
communication sites	48%	71%
repair and supply sites	15%	92%
mechanized infantry	50%	77%

Table 2: Detection probabilities at 10% probability of false alarm

The performance of topic spotters which use the 100 best DP-ngrams were better than that of topic spotters which use the 1000 best CI-ngrams for all seven ARM target categories. Detection probabilities of 90% or more were obtained with a probability of false alarm less than 1% for reports about bridges and pontoons and with a probability of false alarm less than 10% for three other target categories. These figures are better than those of an earlier word based topic spotter which gave 90% detection probability at 10% probability of false alarm when spotting reports about bridges and pontoons.

5. CONCLUSIONS

The paper has described the application of a dynamic programming algorithm to the problems of non-word based topic spotting. Whereas conventional ngram techniques are confounded by the variability of the phoneme transcriber output, the DP algorithm is able to overcome these problems by using partial matches. The DP-ngrams are therefore longer, more meaningful, and of greater usefulness for the topic spotter. Results show that a topic spotter using 100 DP-ngrams performs better than one using 1000 CI-ngrams on all of the seven ARM target categories.

References

[1] E.S.Parris and M.J.Carey. *Final report on identification of topics in speech.* ENSIGMA Ltd, 18 March 1992.

[2] L.Gillick et al. Application of large vocabulary continuous speech Recognition to topic and speaker Identification using telephone speech. Proc. ICASSP, April 1993.

[3] P.Nowell and R.K.Moore. *A non-word based approach to topic spotting in speech.* DRA Memorandum No. 4815, Oct 1993

[4] D.Sankoff and J.B.Kruskal. An anthology of algorithms and concepts for sequence comparison. In D. Sankoff and J.B.Kruskal, editors, *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison.* Addison-Wesley Publishing Company, Reading, MA, 1983.

[5] Helmut Spath. *Cluster analysis algorithms for data reduction and classification of objects.* Ellis Horwood Publishers, 1980.

[6] R.K.Moore. *PHONOTAX (a phonemic syntax for the English syllable).* CSE1 Research Note 248, DRA Malvern, April 1993.

[7] S.R.Browning, J.McQuillan, M.J.Russell, and M.J.Tomlinson. *Texts of material recorded in the SI89 speech corpus.* SP4 Research Note 142, DRA Malvern, Feb. 1991.