Data Driven Example Based Continuous Speech Recognition

Mathias De Wachter, Kris Demuynck, Dirk Van Compernolle and Patrick Wambacq

Katholieke Universiteit Leuven, ESAT - PSI
Kasteelpark Arenberg 10, 3001 Heverlee (Leuven), Belgium
{mathias.dewachter,kris.demuynck,dirk.vancompernolle,patrick.wambacq}@esat.kuleuven.ac.be

Abstract

The dominant acoustic modeling methodology based on Hidden Markov Models is known to have certain weaknesses. Partial solutions to these flaws have been presented, but the fundamental problem remains: compression of the data to a compact HMM discards useful information such as time dependencies and speaker information. In this paper, we look at pure example based recognition as a solution to this problem. By replacing the HMM with the underlying examples, all information in the training data is retained. We show how information about speaker and environment can be used, introducing a new interpretation of adaptation. The basis for the recognizer is the well-known DTW algorithm, which has often been used for small tasks. However, large vocabulary speech recognition introduces new demands, resulting in an explosion of the search space. We show how this problem can be tackled using a data driven approach which selects appropriate speech examples as candidates for DTW-alignment.

1. Introduction

The great advances in large vocabulary speech recognition systems over the past two decades were largely due to the power and flexibility of the HMM framework. HMMs could simultaneously exploit the continuous increase in available training data and computing power. The most important scalability knobs in the HMM framework were the very detailed multi-gaussian density modeling and the context-dependent acoustic modeling. At the same time a basic criticism of the HMM concept —i.e. the first order Markov assumption— was countered by adding derivatives in the feature input, the context-dependent phone concept and speaker adaptation. In the last few years progress seems finally to be running out of steam. Model based statistical pattern recognition leads to optimal recognition under the condition that the model is correct. But we all agree that even after 20 years of patches the model is still *NOT* correct.

One alternative to improve the model further is to rely on discriminative and minimum error training. We take a drastically different approach. We do away with the model altogether. Yes, we do recognition straight from the data, reminiscent of old-fashioned dynamic time warping. Our motivation is multifold. The primary one is that we are not succeeding in designing appropriate models for the transient nature of speech. Either we keep it simple and it works to some extent (e.g. context-dependent models and time derivatives) or we make it complex and we fail (e.g. trajectory models). Other clear evidence of our poor understanding of speech and transients in particular is the quality jump achieved by fully concatenative speech synthesis by applying the motto: "no modeling, just data". So why not do the same in recognition in a much broader scope: "no modeling of acoustic-phonetics, speakers, acoustic environments, etc.— just data".

Of course simple DTW doesn't solve our problem as we want to tackle large vocabulary continuous speech. First of all we need to adjust the template concept to work with sub-word units. Based on the experience gained with concatenative speech synthesis, we expect that the implementation of maximum continuity in the unit selection will be critical to success. Hereby we can make use of very large acoustic-phonetic contexts but we also have implicit control over speaker properties (gender, dialect, vocal tract length (VTL),...) and acoustics. Using this "non-verbal" information may give a new interpretation to adaptation. But the foremost problem is the computational load which at first glance seems unsurmountable. The reference database is already gigantic, the search space is even worse. Gigabytes of memory and teraflops will help, but won't do the job alone. Part of the solution here will be a considerable bottom-up component for which we find motivation in the psychoacoustic and physiologic literature [1].

In this paper we describe the detailed architecture of our example based large vocabulary recognition in Section 2. In Section 3 we describe the bottom-up template selection procedure. In Section 4 we present results on a first set of experiments. While many details need to be filled in before the system is complete, we think we are at a point where we can prove that example based recognition is both feasible and competitive.

2. Overview of the new example based architecture

Example based recognition compares the speech input with concatenations of suitable templates from a database. The search engine picks out lexically legal templates to form words, and those words are further combined to form sentence examples. These concatenations are aligned with the incoming speech by a dynamic time warping algorithm for continuous speech. The example based framework is well-equipped to handle the transient nature of speech, both within sounds (template alignment) and on a higher level (smoothness constraints on template concatenation). The number of all possible concatenations (i.e. the *search space*) turns out to be too large, even for a beam search. That problem leads us to the natural solution of using a bottom-up template selection.

2.1. The template database

All acoustic knowledge of the new recognizer is contained in the template database. An important design parameter is the length of the templates. Typically, former example based connected speech recognizers used whole-word templates [2]. The disadvantage of word templates is obvious, but very short templates

Research funded by the Fund for Scientific Research Flanders (FWO-projects G.0249.03 and G.0008.01).

may prove to be cumbersome as well: the search space grows as the templates get shorter and bottom-up template selection (see section 3) might become less reliable. When the training corpus is large enough, increasing the template size from phonemes to syllables may be an option.

A major difference to previous example based systems is that we will use the entire database. Formerly, prototype examples were extracted from the database by clustering methods to save on memory and computation time [3]. Because we store the entire database in the original order, the complete acoustic-phonetic context of each template will be available.

Another key aspect of the database is the non-verbal information that can be added to each template. Both speaker-dependent (e.g. gender, VTL and dialect) and environmental (SNR, type of noise, etc.) information can be attached, as well as special features such as prosodic annotation.

2.2. DTW for continuous speech

The Dynamic Time Warping algorithm is widely known because of its use in many small vocabulary isolated word recognition tools. The algorithm is typically written as a recursion, incorporating constraints on possible transitions. For within-template alignment, we use:

$$D_t(y^t) = \min \begin{cases} D_{t-1}(y^t) + d_t(y^t)w_1 & \textit{if } y^{t-1} \neq y^{t-2} \\ D_{t-1}(y^t - 1) + d_t(y^t) & \\ D_{t-1}(y^t - 2) + d_t(y^t)w_2 & \end{cases} \quad (1)$$

Here $D_t(y^t)$ is the score of the partial path of length t that reaches frame number y^t and $d_t(y^t)$ is the local distance between the input frame and the template frame. The factors w_1 and w_2 are extra local costs for staying in and for skipping a template frame respectively. The condition in the first line is the Itakura local constraint [2], which ensures that the path “stretches” the template with no more than a factor 2.

The success of DTW for isolated word recognition incited researchers in the eighties to extend the algorithm to continuous speech recognition [4]. Apart from the template-internal transitions expressed in equation 1, partial paths can now also branch to each lexically legal successor template when reaching a template boundary. A path that crosses a template boundary receives an extra cost:

$$\textit{Trans}(\textit{Templ}(y^{t-1}), \textit{Templ}(y^t)) + \textit{LMCost} \quad (2)$$

LMCost is the language model cost when the partial path has crossed a word boundary. The *Templ(.)* “operator” returns the template that contains the given frame number. The term *Trans(A,B)* is the transition cost between the two templates.

2.3. Features and local distance metric

The local distance metric in DTW plays exactly the same role as the emission probabilities in the HMM framework, assuming single gaussians and pooled variances. By use of multigaussians and state dependent variances, HMMs use a non-uniform metric. For the time being we have restricted ourselves to a feature independent Euclidean distance metric. As features we use sine-liftered cepstra and first-order derivatives. Optimization of this part of the recognizer —which is clearly needed— is not part of the currently presented research.

2.4. The template transition cost *Trans(A,B)*

When two templates are concatenated, a suitable cost is added, depending on how “smooth” the resulting path is. The same idea is found in speech synthesis [5]. Next to a term that controls the global insertion rate, additional penalties are added to the template transition cost:

The first penalty is the cost for incorrect acoustic-phonetic context. Since the complete context for each template is available, contextual constraints can be more complex than classical fixed triphones. Furthermore, eliminating the transition cost for the concatenation of two templates that were recorded in that order will introduce a strong preference for longer segments.

A second penalty is based on information about speaker or environment of the templates that are concatenated. The effect of this cost is that paths with consistent non-verbal information are preferred. Although there is no hard constraint, these costs will result in parallel specialized example based models, which can be seen as an *implicit form of adaptation*. For example, when the cost to switch from a female template to a male template is higher than the beam threshold, the path can only expand further in an acoustic space that has been completely adapted to female speech. In reality, the adaptation will remain soft, but it will occur for many different aspects of the non-verbal information at the same time. It is also conceivable that some constraints will be made hard, except for at certain time slices, where a possible speaker-change is indicated by another preprocessing module [6]. And in contrast to explicit adaptation methods, it is possible that paths with different consistent non-verbal information are examined at the same time. Nevertheless, the implicit adaptation mechanism in no way prevents the use of explicit adaptation methods such as VTLN.

The template transition cost turns out to be a very useful vehicle to introduce further knowledge into the recognizer. However, we have yet to devise a method to estimate the suitable values for the penalties. Remark that all this extra power is obtained from labeling the training data, while no estimation of parameters from the test sample is needed as is the case in an HMM recognizer.

2.5. The search space

As mentioned in section 2.1, our aim is to use the entire training database for recognition. A closer look at the DTW algorithm exposes a problem, however. Each time a partial path reaches the end of a template, it can branch to all lexically legal successor templates. Therefore, compared to HMM decoders, the average branching factor of the search space is multiplied by the number of examples of each elementary part. It is clear that the combinatorial effect of this increase on the number of possible paths severely complicates the search. In a normal top-down beam search, constraints on the beamsize would force the pruning of partial paths after a template transition to be based on local distance scores of only one or two further frames.

To all appearance, the vastness of the search space makes top-down beam search for example based large vocabulary continuous speech recognition impossible. Hence it is essential to somehow select interesting templates based on criteria other than partial path scores. This requirement suggests our solution which uses a *bottom-up* (or *data driven*) *template selection*.

3. Bottom-up template selection

The data driven template selection module is designed to suggest interesting templates to the top-down DTW decoder. For each input frame, a list of templates that have high enough probability to match the following piece of input are returned. To gather that information, the bottom-up part performs an *acoustic look-ahead* [7], preceding the DTW search by about the

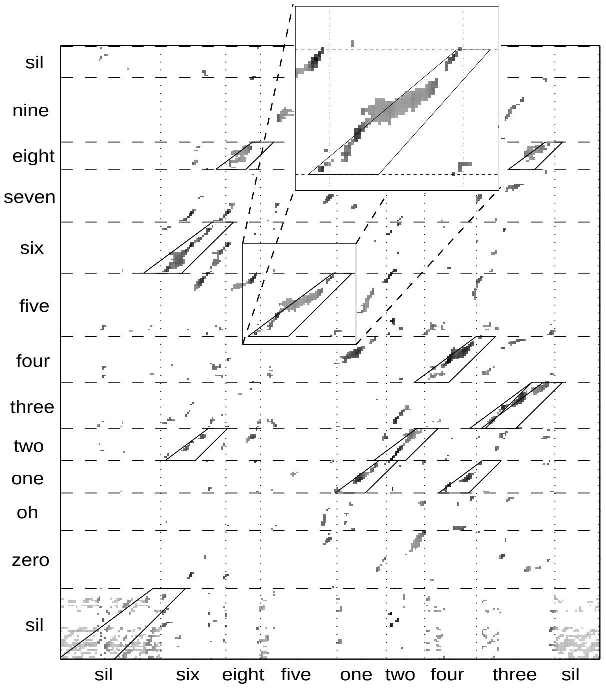

Figure 1: *Time evolution of the k-nearest neighbours (darker dots represent smaller distances) and the output of the time filter. A connected digit string is compared to 13 templates from different speakers. The trapezia show the working of the time filter: the base of each trapezium delimits the time slice in which the template is candidate for path extension in the top-down part. For this very small example, 3% of all distances were calculated. One correct match is magnified.*

length of the longest template in the training set.

The template selection module consists of two blocks. Since the number of acoustic vectors in a large corpus would be prohibitively large for full distance calculation, only a small fraction of acoustic distances in the neighbourhood of the input vector is calculated. Hence a *fast k-nearest neighbour selection* is needed. In a second phase, the time evolution of the nearest neighbours is investigated to find template candidates.

3.1. Fast k-nearest neighbour selection

The problem of finding the k-nearest acoustic vectors given an input vector is a standard, but difficult problem. Most efficient algorithms are only useful in a two-dimensional space or in a very high dimensional space with a limited number of vectors.

Different fast k-nearest neighbours algorithms can be used in our new recognizer. For the prototype system a modified and extended form of the “Roadmap” algorithm [8] was developed with promising results.

3.2. Time filtering

The bottom-up template selection idea is based on the fact that, given a sequence of k-nearest neighbour vectors, it is possible to detect templates that resemble the input. Thus, the *time filter* checks the evolution of the k-nearest neighbours and looks for activation patterns that move through a template at about the same speed as the input. Hence it searches *diagonals* in the distance matrix. Figure 1 shows a very small example, with only one template for each digit. The idea of a heuristic search for an optimal sequence match by looking for diagonals can be found in other research areas as well. One example is sequence similarity search in DNA or protein databases [9].

```
1  function filter()
2  {foreach(nn in k_nearest_neighbours)
3    if(template_head(nn))
4     trigger(nn)
5    elseif(active(nn))
6     if(activation_too_old(nn))
7      cleanup_activation_window(nn)
8     else
9      trigger(nn)
10  endfor
11 }
12 function trigger(nn)
13 {update_score(nn)
14  activate_successors(nn)
15  if(has_template_end_successor(nn))
16   if(template_end_score(nn)>threshold)
17    activate_template(nn)
18 }
```

Figure 2: *Pseudo-code skeleton of the processing of the k-nearest neighbours in the time filter.*

The time filter processes all templates in parallel in a fast, time-synchronous algorithm. Figure 2 gives an outline of the prototype algorithm. A few points need some extra explanation:

line 3: The template head is the first part of the template. Only frames in the template head can trigger without having been activated.

line 6: Activations expire when the cause of the activation happened more than *maxGapSize* input frames ago.

line 7: When encountering an expired activation, the whole window is deactivated. Since activation windows have a fixed size, this is done in constant time.

line 13: The filter score of the triggered frame is based on the score of the source of the activation, the distance score of the nearest neighbour and a weight that penalizes deviation from the diagonal. The weight is similar to the local skipping weights of the DTW algorithm.

line 14: When a filter frame is triggered by a nearest neighbour, its immediate successors (determined by the maximal allowed size of gaps) within the template become active.

line 15: When the last frame of a template becomes active, its score is checked, and if it is above a certain threshold, the template is selected. The threshold is dependent on the length of the template, but can also be made dependent on the template transcription.

Each of the called functions runs in constant time, and since the only loop is over the list of k-nearest neighbours, the time filter’s time complexity is linear in the number of nearest neighbours.

Since the filter does not keep track of when the activation started, and only detects interesting templates when they have been processed completely, the start of the activation has to be guessed based on the end time. For robustness the template is kept active during a certain time window. Furthermore, most often there will be multiple activation paths close to each other, resulting in an even wider activation window. The bases of trapezia in figure 1 denote the region in which a template beginning is activated; the tops are the windows in which the filter score of the template end is high enough for activation.

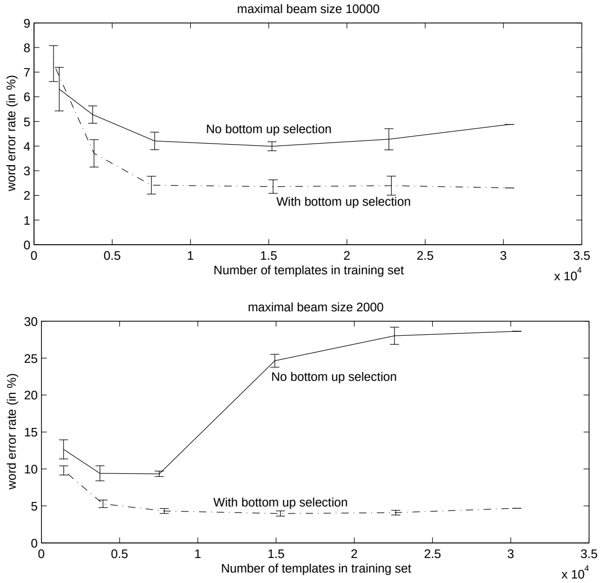

Figure 3: *Effect of database size on recognition performance for the top-down DTW recognizer and the new architecture. The repeated experiments are shown with a one-sigma errorbar.*

4. Experiments

A first set of experiments was performed on the TI-Digits connected digit string recognition task. Since the vocabulary only consists of eleven digits, word templates were used.

The objective of the presented experiments was to validate the concept of bottom-up template selection by comparing it with a strictly top-down DTW recognizer. Our main interest was the behavior of both setups with increasing database size. The only difference between the two setups is that paths in the top-down recognizer can branch to all examples after a partial alignment, while in the bottom-up system only the selected templates are candidate for further expansion. In the template selection module, 1% of all distances were used as nearest neighbours.

Highly optimizing the recognizer for such a simple task by discriminant training or by tweaking the preprocessing unit can yield extremely low error rates [10]. This was not our goal at this point in the development. We have used a simple prototype system with sine-liftered cepstral coefficients and first-order time derivatives. With context-independent templates and without optimization of transition costs, an error rate of about 2% is obtained. This setup is used in the comparative experiments on database size, template selection and beamwidth. It should be noted that the error rate is readily reduced to below 1% when using context-dependent templates.

Figure 3 shows recognition results for different database sizes. The smaller databases were random subsamplings of the complete TI-Digits trainset. To average out effects caused by the random subsampling, each of the tests were run ten times. Obviously, tests using the full training database have been run only once. To limit recognition time, only a small subset of the testset was used.

As can be seen on the figures, the experiments confirm our suspicion that a top-down beam search is unable to deal with large databases. When using a maximal beam width (i.e. the maximal number of parallel paths in the search) of 10000, the top-down setup gradually starts to deteriorate when the database becomes larger than 15000 examples. With a smaller beamsize, results for the top-down setup are catastrophic when the number of examples exceeds 10000. In both cases, the new architecture performs about the same with increasing database size after reaching a near optimum somewhere around 10000 examples. It is not clear why the performance of the bottom-up system does not improve any further, but it can be argued that a few thousand examples is sufficient for such a small vocabulary task.

5. Conclusions

We introduced a new architecture for large vocabulary continuous speech recognition. Several new concepts were introduced. Bottom-up template selection is used to limit the search space. Template concatenation costs result in implicit adaptation and allow for flexible use of contextual information. Only a few parts of the proposed system are implemented today, but the bottom-up template selection has been tested and proved to be a large improvement over top-down example based search. Preliminary experiments on the use of template concatenation costs based on acoustic context also showed a large improvement. We outlined the parts that need further research and are the best candidates to contribute to an improved performance.

6. References

[1] W. D. Marslen-Wilson and A. Welsh, “Processing interactions and lexical structure of spoken language understanding,” *Cognitive Psychology*, vol. 10, pp. 29–63, 1978.

[2] L. R. Rabiner and B.H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, 1993.

[3] L. R. Rabiner, J. G. Wilpon, A. M. Quinn, and S. G. Terrace, “On the application of embedded digit training to speaker independent connected digit recognition,” *IEEE Trans. on ASSP*, vol. 32, no. 2, pp. 272–280, April 1984.

[4] C. Godin and P. Lockwood, “DTW schemes for continuous speech recognition: a unified view,” *Comp. Speech and Lang.*, vol. 3, no. 2, pp. 169–198, 1989.

[5] G. Coorman, J. Fackrell, P. Rutten, and B. Van Coile, “Segment selection in the L&H realspeak laboratory TTS system,” in *Proc. ICSLP*, Beijing, 2000, vol. II, pp. 395–398.

[6] A. Tritschler and R. Gopinath, “Improved speaker segmentation and segments clustering using the bayesian information criterion,” in *Proc. EUROSPEECH*, Budapest, Hungary, 1999, vol. II, pp. 679–682.

[7] S. Ortmanns, A. Eiden, H. Ney, and N. Coenen, “Look-ahead techniques for fast beam search,” in *Proc. ICASSP*, Munich, Germany, Apr. 1997, vol. III, pp. 1783–1786.

[8] D. Povey and P. C. Woodland, “Frame discrimination training of HMMS for large vocabulary speech recognition,” Tech. Rep. CUED/F-INFENG/TR332, Cambridge University Engineering Department, 2000.

[9] S. F. Altschul, T. L. Madden, A. A. Schäffer, J Zhang, Z. Zhang, W Miller, and D. J. Lipman, “Gapped BLAST and PSI-BLAST: a new generation of protein database search programs,” *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[10] Y. Normandin, R. Cardin, and R. De Mori, “High-performance connected digit recognition using maximum mutual information estimation,” *IEEE Trans. on SAP*, vol. 2, no. 2, pp. 229–311, Apr. 1994.