# Classifier combination in speech recognition

*Matti Aksela*

Laboratory of Computer and Information Science
Helsinki University of Technology, Espoo
matti.aksela@hut.fi

## Abstract

In this course project, I will attempt to write up a survey of combination methods used in speech recognition. I will attempt to evaluate them with a more general view of classifier combining, and also consider the usability of some adaptive combination methods that have used in my previous research within the domain of speech recognition. Thus the project consists of two parts, a survey into existing research, and some speculation on my part, based on previous experiences, but leaving out any practical experimentation as that would probably prove to simply take too much time and effort for the scope of a course project.

## 1. Introduction

Classifier combining is an approach often taken in various recognition tasks when seeking further improvements in performance. Generally the main idea is that different recognizers may make different mistakes, and hence by combining the recognizers can result in performance improvement.

There are several notable challenges in taking a classifier combining approach, but for speech recognition in particular, on the word level there is the additional challenge of lining up the recognition results correctly so that any combining can even be performed. And the fact that the vocabularies (number of classes in practice) are so much larger than for many recognition tasks creates a need to view the problem from an entirely different angle; methods using information on the actual labels easily become useless. Hence for someone already rather familiar with combining classifiers in a different application domain this does feel like a very interesting problem indeed.

In the following sections first some combination methods found in literature are described and briefly evaluated. Then the possible applicability, in the domain of speech recognition, of combination methods I have previously examined is examined. Finally some conclusions will be drawn.

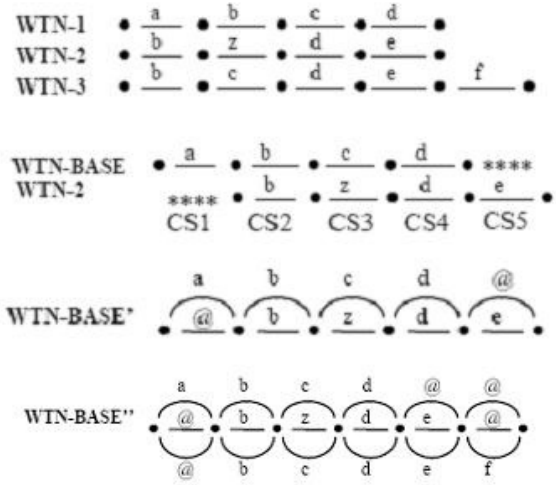



Figure 1: Process of ROVER WTN merging

## 2. Survey of published methods

In this section overviews of some methods presented in literature will be given. Both larger frameworks, such as the ROVER system and confusion networks, and more particular applications have been addressed.

### 2.1. ROVER and its improvements

The NIST Recognition Output Voting Error Reduction (ROVER) system [1] is basically a scheme for combining word-level results. The combination is performed by first aligning the recognition results using a Dynamic Programming algorithm, and then choosing the combiners result with a weighted voting scheme.

The problem of aligning more than two word transition networks (WTNs) is one main problem. As the task is a hyper-dimensional search, the approximation of iteratively combining two WTNs at a time is taken. The downside to this approach is that the resulting combination depends on the order in which the WTNs are combined, a fact that is noted in the paper but is given no more thought. The procedure is illustrated in figure 1.

For the actual combining, a confidence value is ob-

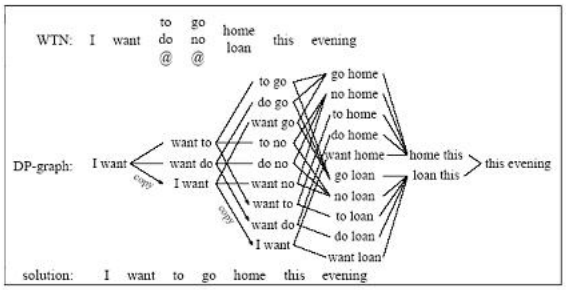



Figure 2: Illustration of the ROVER language model extension

tained from each recognizer. This should, as accurately as possible, reflect how confident the classifier is in its decision. Then the actual scoring of the options is performed based in practice on weighted voting among the results. The author gives three basic ways, frequency of occurrence (just voting, no confidence use), average confidence scores (pretty much the standard way, although scaled a bit funnily) and maximum confidence scores (max for each label).

The author provides results that show word error rate reductions on the LVCSR 1997 HUB-5E from the best members 44.9% to 39.4% word errors.

Also improvements to the ROVER system have been proposed, one of which deals with using a language model in the voting phase [2, 3]. This paper presents a reasonably simple and logical way of incorporating language model information to the ROVER model. The problem of combination order effecting the combined WTN is given a simple solution (which I would expect also the original ROVER had used, but apparently alphabetic ordering was the ordering actually in use. This of course is pretty much as good as any other random ordering with no relevance to the classifiers characteristics... Which surely is suboptimal, as more information is easily available), start with the WTN from the best classifier and combine in the order of decreasing performance, which logically gives the most "weight" to the WTN expected to be best, and is as said a logical choice.

The actual language model incorporation requires a bit of tinkering with the DP algorithm, to make the used trigram context always available, simply put the ties between words are enforced more in order to make a language model applicable (for details see [2]. This is illustrated in figure 2. The language model is used to select the words to minimize the perplexity of the overall sentence. The method is shown to provide better perplexity and sentence error rates, as is to be expected. Also the word error rate is improved in most cases, although not all.

### 2.2. NN combining

Artificial neural networks are often combined to achieve better performance. One example of combining NNs for speech recognition can be found in [4]. The paper proposes combining context-dependant (CD) and context-independent (CI) ANNs by interpolation and/or by collapsing CD network outputs into CI network outputs. The reasoning is that a CI network is usually more robust since more training data is used for each output and a CD network is more precise due to more detailed modeling in each context, combining both types may yield a classifier that is both robust and precise.

Three combination types are described. The first is a homogeneous committee, where networks of the same type (all CD or CI) are combined via linearly combining their outputs. The second approach is a heterogeneous committee, where committees of different types are combined, again through linear combination where the CD network output is first summed over the applicable contexts. The third approach is to join two CD committees by "simulating" a CI network with one, in practice summing over all the contexts for the CD network.

In the experiments two members of four possible are used. The CD member classifiers perform initially better than the CI members, and the best improvements are obtained through combining them by the third "collapsing" approach. Personally I found the results to be somewhat lacking, as all the best results used CD networks (that performed better in the first place), and thus may not really represent the combination methods abilities that well, especially since only two classifiers are combined at a time.

### 2.3. Confusion networks

Confusion networks, produced from word input lattices with multiple possibilities for words, have also been used [5, 6] for disambiguation purposes. However, I was unable to find a paper that actually combined outputs from different classifiers, usually it appears to be used for choosing the correct one from a list of word hypothesis. However, there clearly is no real reason for not applying the method also to a multi-classifier setting. However, the evaluation of the approaches effectiveness cannot be completed as a suitable reference was not discovered, but it is felt that the approach is meaningful enough to be given this brief mention. The basic notion is to form a word lattice into a confusion network format, both of which are shown in figure 3. To my understanding, the confusion network could consist of a number of classifier results, and a combination method could be used to select the word at each location.

### 2.4. Sum, product, min and max rules

Also some of the most standard combination methods, namely the product, sum, min and max rules have been

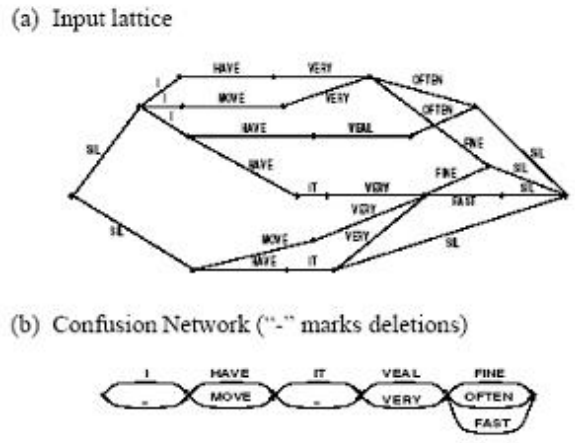



Figure 3: Confusion network example

applied in [7]. In this paper they are used to combine two ANN/HMM hybrids. First of the systems uses 8 log-RASTA-PLP coefficients and their energy and deltas, and the second 9 MFCC coefficients and their deltas.

Their first set of experiments shows that the product and minimum rules, both of which can be seen to implement a "and" - type function as the output is large if both are large, outperform the sum and max rules, which again can be seen as more of an "or" type function, just one being large is enough to provide a large output, clearly.

An approach to using confidence values, in this case defined through computing the entropy of the phone posterior probabilities, averaged over the entire utterance, was also presented. A small additional benefit to accuracy was observed.

### 2.5. Hierarchical combination

[8] presents a refreshingly new look, a hierarchical combination of committees. The paper also shows an interesting comparison between three fundamental approaches: voting, linear combination of classifiers.

In the paper experiments with phonetic classification using the TIMIT acoustic-phonetic corpus are presented. They use a total of 13 different sets of measurements along with mixture diagonal Gaussian classifiers.

The first committee approach presented is standard voting, using classifier ordering for resolving ties. The second method uses a weighted linear combination of classifiers using likelihood ratios for weighting, and the third is the standard product rule (in the paper said to be multiplication of the probabilities based on assuming independence).

Furthermore, an approach of using hierarchical classifiers (where class-specific hierarchical measurement sets for different sounds (vowel, nasal, fricative, stop) are used) in two formations was presented. Firstly, a committee was used as a node in the hierarchical, and secondly by adding a hierarchical classifier to the committee.

The results show that there is no large difference in the performances, with the error rates of 18.6% for the voting approach, 18.5% for the product rule, 18.4% for the linear combination, 18.3% for the hierarchical tree consisting of committees and the best reported result of 18.2% obtained from adding the hierarchical classifier and combining using the product rule.

## 3. Applicability of combination methods

Basically one could see the problem of speech recognition, for the purpose of combining classifiers, as having two levels, phoneme-level and word-level. First my view of the benefits and problems of both will be discussed, and then some combination methods I am more familiar with will be examined.

One notable problem for several combination methods widely used in other domains is the vastness of the label space when using word-based recognition. Another problem with word-level combining is the necessity of aligning the results, as there is no way of knowing even how many words there are in the result stream. And the alignment again produces another problem, it is required that the "input word stream" can be cut into reasonable-sized chunks for the purpose of performing the alignment. For these problems ROVER and it's extensions do provide some solutions, but I did not encounter a truly "good" solution.

Combining recognizers on the phoneme-level is basically much simpler – the dictionary is much more limited, and the entire aligning problem can be circumvented as it really isn't a problem at this level. However, another problem arises in the fact that combining at the phoneme level incorporating more information then the confidences (or something based on the prior performance) of the member classifiers is very hard, as the task of "decoding" the phonemes into actual words still remains. Also one might expect some problems if the phoneme-level combiner makes mistakes in the phonemes, as this might cause a "ripple-effect" for also further phonemes and also further words. (Although that is also the case for word-level combining, naturally).

In the following a few more complex combination methods are presented and the possibility of their application to the speech recognition task is briefly considered.

### 3.1. BKS

In many situations overly simple methods can suffice, but there may be additional gain available from a learning committee structure. One easy example is the Behavior-Knowledge Space (BKS) committee [11]. It is based on a $K$-dimensional discrete space that is used to determine

the class labels. Each dimension of the knowledge space corresponds to the decision of one classifier, and has a discrete value for each of the $m$ class labels. The decision is obtained by first finding the focal unit in the $K$-dimensional space, the unit which is the intersection of the classifiers' decisions for the current input. In the training phase each focal unit collects the count of hits and counts for each class.

During recognition the output of the committee is the class with the highest probability in the focal unit. More precisely, the class $l$ that has accumulated the largest number of samples in the focal unit, and for class $l$ the ratio between the number of samples for that class and all accumulated samples is above a pre-selected threshold, class $l$ is selected as the committee's output.

Such a committee method can learn to correct common mistakes, but one notable downside is the dimensionality involved; $k$ classifiers $\times$ $m$ classes, so the applicabitility to especially word-level combining would hardly be even worth consideration. On the phoneme-level, however, such a method might be worth investigating. Also the incorporation of a language model, or other higher level knowledge, would thus be very difficult.

### 3.2. Dynamically Expanding Context Committee

One can form an adaptive committee based on the Dynamically Expanding Context (DEC) algorithm [12]. The basic approach is that of a committee classifier that creates a set of transformation rules during the processing of the data, with the rule set being created individually for each subject without any prior learning phase. The subject can be eg. a speaker in speech recognition and a writer in handwritten character recognition.

The DEC principle was slightly modified to suit the setting of combining classifiers. For this setting, a list of member classifiers' outputs is taken as a one-sided context for the first member classifier's output. The classifiers are used in the order of decreasing performance on the evaluation set. The errors in the committee's output are corrected by generating transformation rules consisting of a list of member classifier outputs as the inputs and the desired classification result as the output.

Each time an input is fed to the system, the existing rules are first searched through and the most specific applicable rule is used. If no applicable rule is found, the default decision is applied.

For every input the classification is compared to the correct class. If the decision was incorrect, a new rule is created. The created rule always employs the minimal amount of context, ie. member classifier outputs, sufficient to distinguish it from existing rules. To make the rules distinguishable every new rule employs more contextual knowledge, if possible, than the rule causing its creation. Eventually the entire context available will be used and more precise rules can no longer be written. In such situations selection among multiple rules is performed via tracking correctness of the rules' usage.

Means for dealing also with situations where most member classifiers are incorrect are present, although with the rules always being based on previously seen situations, the correction of a mistake requires that the mistake has been made at least once. Still, also the correctness of the member classifiers is of significant importance, as the default decision is created directly from the outputs and each rule is required to have its output included in its inputs. Highly-ranked classifiers with low error rates may also be expected to be favored, as such lead to more general rules of wider applicability.

Also this adaptive committee method should be very usable with application to speech recognition, especially in a situation where run-time adaptation is needed and the re-evaluation of the actual classifiers parameters may be impractical. The committee could be applied on either the phoneme or word level, on the phoneme level the actual performance would be more akin to the BKS or any other method learning the member classifiers typical error patterns and fixing them. On the word level, however, the committee could perhaps be able to learn a language-model like representation, but with the varying context size adding something of a twist... Personally I would find it interesting to see what happens.

### 3.3. Class-Confidence Critic Combining

Generally, a critic-based approach is one in which a separate expert makes a decision on whether the classifier it is examining is correct or not. Critic-driven approaches to classifier combining have been investigated previously, e.g. in a situation where the critic makes its decision based on the same input data as the classifier [**?**]. In our CCCC approach [12] the main idea is to try to produce as good as possible an estimate on the classifier's correctness based on its prior behavior for the same character class.

The basic idea of a critic acting in a committee is to evaluate the probability that the result received from the classifier is correct. In CCCC the probability evaluation results in a confidence value, which is based on the earlier performance of the classifier in similar situations. The similarity of the situations is currently defined by the classifier classifying the input in the same character class.

In CCCC there are two distance distributions for each class stored in each critic. One corresponds to the correct classification results and the other one to the incorrect results. This is illustrated in Figure 4.

The critics update their respective distributions as more data comes, and supply confidence values to the recognition results based on some measure derived from the collected data. The decision is then made based on the member classifiers labels and the critic-supplied confidences.

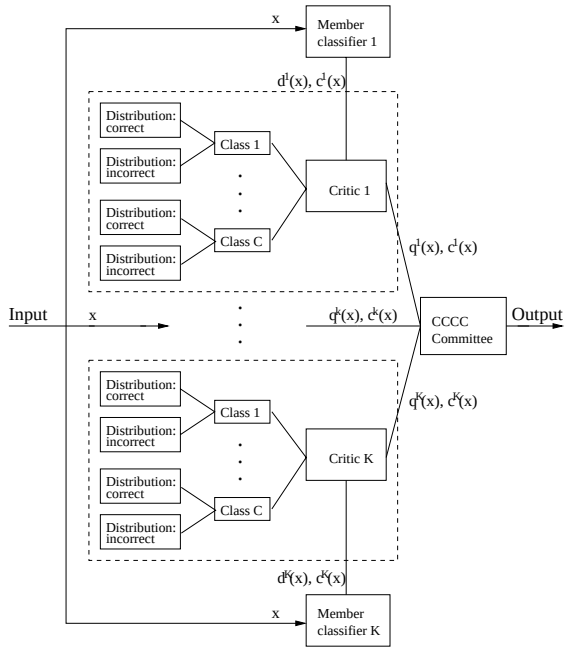



Figure 4: The basic idea of the CCCC committee

Also this type of a combination method can often be very effective, as the main driving force is embracing the fact that the member classifiers are different from each other and trying to maximize the benefits from this by having a individually automatically tailored confidence production scheme for each member classifier. Assuming that sufficiently different member classifiers were used, also a speech recognition application might benefit greatly from this. But here too the critics have distributions that are specific to some label, and thus the applicability of such a method on the word-level would be highly impractical unless a meaningful clustering of the words could be achieved.

## 4. Conclusions

Several common classifier combination methods seem to have been already addressed in the relevant literature, as is to be expected. However, at least I was not able to find references using more elaborate schemes of combination, mostly quite simple approaches were used to provide a way of incorporating a only slightly differing classifiers, most commonly differing mainly on the data they were trained on more than their actual methodology, which may be one reason why most results weren't hugely positive.

A comparison between different combination methods presented is very difficult, as the results were (as is sadly very commonly the case) not really comparable. All in all the ROVER combination framework described in section 2.1 is a much larger piece than just a classifier combining scheme, alignment and other important aspects of the problem are also addressed. The confusion network approach, briefly addressed in section 2.3 is another similar framework of solutions, into which various actual combiner schemes could be plugged in.

The combination schemes discussed in section 2.2 is of interest in the sense that is does combing two approaches of different "level", the context dependent and independent models. But the result that shows that actually creating a context independent model from a context dependent one (in practice just averaging over the contexts) is somewhat mystifying, as it is difficult to see why this would benefit classification. Perhaps it is more due to the fact that the CD networks performed better overall, and thus combining two (with one "falling back" to CI) seems the best option.

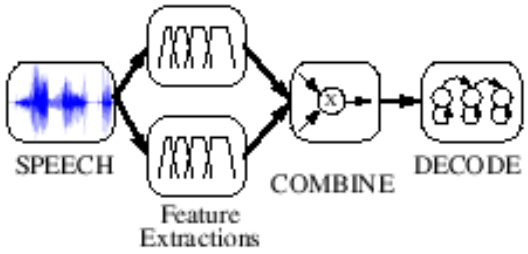
Finally some comparisons could be sought in the papers discussed in sections 2.4 and 2.5, where different combiners are compared on the same data. Sadly both papers do not really provide, in my opinion, a very good representation of the situation; the first combines only two classifiers and not surprisingly valuing just one (product and max rule) is more effective than the more flattening approach (min or sum). This results might be somewhat more interesting with a larger classifier base, though. In the second paper more different methods are compared, but the differences in performance do seem to be a bit small to really definitely say anything about the merits of the respective methods.

All in all, it seems that quite few experiments have been performed with notably different classifiers, although a reviews of several possible methods can be found (for example [9]). Most of the research encountered seemed to focus on somewhat different feature sets, which is of course one approach to getting different classifiers, but perhaps not the best base for classifier combining.

It may be noted, that most of the methods do not address any of the alignment issues, ie. it is expected that correctly segmented frames of data are obtained for the classifiers to work on, and the recognition results from the data are valid for combining.

Also numerous articles concerning various combination of features, more so than actual classifiers, were encountered (sub-band speech recognition being a very common example, for example [10] among others). The main difference here is illustrated in figure 5: the main point is that different sets of features are calculated (the features may be same and from different bands etc) and the weighting (or other combination method) is performed on them and not the results of some classification method. Thus the methodology may actually be very similar (for example a dynamic weighting scheme in [10]), but with the combination being done on the features, such methods were excluded from this survey.

a)




b)

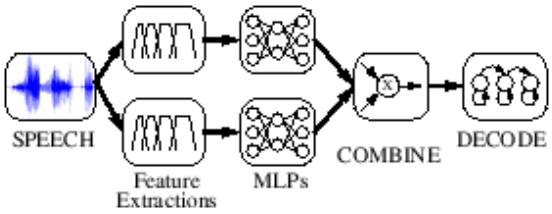



Figure 5: Multiple features (a) vs. multiple classifiers (b)

One notable consideration specific to this domain is that that classifier combining in speech recognition can clearly be divided into two parts, combining on the word level and combining on the phoneme level. It would seem that both have their advantages and drawbacks.

All in all, the process of constructing this small review was a very informative one from my own perspective; I reached my intended goal of familiarizing myself with classifier combination research of the domain, and I feel I have gotten a feel as to what issues may remain. Especially the fact that very few research papers show interesting results using clearly different classifiers of a meaningful amount; just combining a couple of very similar classifiers may present a nice conference paper but is rather unlikely to produce a large breakthrough in the field.

If I have the possibility of performing some own research on the field, I would strive for collecting or creating a diverse enough set of classifiers and experiment with combination schemes that can be more tailored to the situation, such as ones presented in section 3. In my opinion this might provide quite an interesting topic for further study.

## 5. References

[1] J. Fiscus, "A post-processing system to yield reduced word error rates: Recogniser output voting error reduction," 1997. [Online]. Available: citeseer.ist.psu.edu/fiscus97postprocessing.html

[2] H. Schwenk and J. Gauvain, "Improved rover using language model information," 2000. [Online]. Available: citeseer.ist.psu.edu/schwenk00improved.html

[3] J. G. H. Schwenk, "Combining multiple speech recognizers using voting and language model information," in *Proc. ICSLP'2000, Beijing, China*, 2000. [Online]. Available: citeseer.ist.psu.edu/schwenk00combining.html

[4] B. Mak, "Combining anns to improve phone recognition," 1997. [Online]. Available: citeseer.ist.psu.edu/mak97combining.html

[5] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Computer, Speech and Language*, vol. 14, no. 4, pp. 373–400, 2000.

[6] L. Mangu and M. Padmanabhan, "Error corrective mechanisms for speech recognition," 2001. [Online]. Available: citeseer.ist.psu.edu/mangu01error.html

[7] K. Kirchhoff and J. Bilmes, "Dynamic classifier combination in hybrid speech recognition systems using utterance-level confidence values," 1999. [Online]. Available: citeseer.ist.psu.edu/kirchhoff99dynamic.html

[8] A. Halberstadt and J. Glass, "Heterogeneous measurements and multiple classifiers for speech recognition," 1998. [Online]. Available: citeseer.ist.psu.edu/halberstadt98heterogeneous.html

[9] A. Kocsor, L. Tóth, A. Kuba, Jr, K. Kovács, M. Jelasity, T. Gyimóthy, and J. Csirik, "A comparative study of several feature transformation and learning methods for phoneme classification," *International Journal of Speech Technology*, vol. 3, no. 3/4, pp. 263–276, 2000. [Online]. Available: citeseer.ist.psu.edu/kocsor00comparative.html

[10] S. S. Iain Mccowan, "Multi-channel sub-band speech recognition." [Online]. Available: citeseer.ist.psu.edu/640216.html

[11] Y. Huang and C. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 90–94, 1995.

[12] M. Aksela, R. Girdziušas, J. Laaksonen, E. Oja, and J. Kangas, "Methods for adaptive combination of classifiers with application to recognition of handwritten characters," *International Journal of Document Analysis and Recognition*, vol. 6, no. 1, pp. 23–41, 2003.