Integrating Statistical and Rule-Based Knowledge for Continuous German Speech Recognition

René Beutler, Beat Pfister

Speech Processing Group
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
{beutler,pfister}@tik.ee.ethz.ch

Abstract

A new approach to continuous speech recognition (CSR) for German is presented, which integrates both statistical knowledge (at the acoustic-phonetic level) and rule-based knowledge (at the word and sentence levels). We introduce a flexible framework allowing bidirectional processing and virtually any search strategy given an acoustic model and a context-free grammar. An implementation of this class of recognizers by means of a word spotter and an island chart parser is presented. A word recognition accuracy of 93.5% is reported on a speaker dependent recognition task with a 4k words dictionary.

1. Introduction

The German language is known to have some specific properties like relatively free word order, rich morphology, agreement of case, number and gender, discontinuous constituents, long distance dependencies, and last but not least many homonyms. For all these reasons we argue for incorporating detailed linguistic knowledge, particularly from morphology and syntax, in the speech recognition system. Since such linguistic knowledge is rule-based, the recognition system gets inhomogeneous: it includes a statistical and a rule-based part that have to co-operate in some way.

A very simple type of co-operation is sequencing the two subsystems, as can be seen in many so-called speech understanding systems: The statistical subsystem provides some hypotheses, e.g. an N-best word lattice which is subsequently processed by the knowledge-based subsystem. In particular, there is no feedback from the knowledge-based subsystem to the statistical one.

We consider this sequencing inappropriate because it is impossible to determine the number of best hypotheses that have to be provided by the statistical subsystem in advance. It can always happen that a necessary hypothesis is missing and therefore the knowledge-based subsystem cannot find the correct sentence. Increasing the number of hypotheses does not eliminate but only decrease this problem at the costs of a new one, namely the workload of the parser of the knowledge-based subsystem (combinatorial explosion).

A further issue arises from the fact that not all parts of an utterance are equally intelligible. Emphasized syllables are more precisely articulated, whereas others might be pronounced rather carelessly. Additionally, there are coarticulation, pronunciation variations and noise. This motivates to start the recognition at those points where we consider the hypotheses to be reliable using some sort of confidence measure.

All the above considerations motivated us to design an appropriate recognition system architecture that is outlined in Section 2. Sections 3 thru 5 describe the three main parts of this system and how they have been realized. Finally, in Section 6 we report the results of a recognition experiment.

2. System architecture

Based on the above considerations, we propose a speech recognition system architecture which includes a statistical speech units recognizer and a rule-based linguistic processor. It has to be emphasized that these two subsystems work tightly together. The speech unit recognizer provides an initial set of hypotheses to the linguistic processor, that in turn can request additional hypotheses. Such a request can be very specific, e.g. with respect to location or syntactic information. This requires an incremental processing and a bidirectional interaction between these two components, as illustrated in Figure 1.

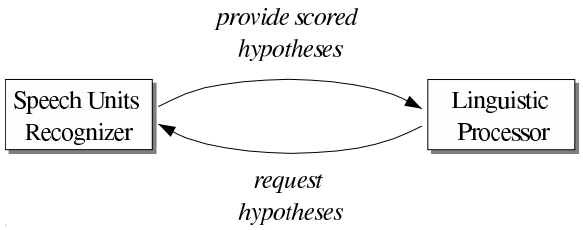

Figure 1: *Fundamental architecture of a speech recognition system that combines statistical speech units recognition and linguistic knowledge processing (bidirectional interaction).*

On top of these two subsystems, an operational system needs some control module. Consequently, the proposed system is composed of the following three sub-systems:

- A statistical speech units recognizer which *i)* provides initial hypotheses, *ii)* is ready to produce additional hypotheses on demand at designated locations and *iii)* is able to score hypotheses proposed by the linguistic processor.
- A rule-based linguistic processor that concatenates small hypotheses to larger ones according to a grammar without any restriction of the parsing direction.
- A control module which includes among other things a strategy component that decides which of the currently possible actions the parser has to execute in order to optimally make progress towards the final solution.

This architecture specifies a broad class of recognizers that can be instantiated in different ways. For our prototype we have chosen some sort of word spotter as speech units recognizer and an island chart parser as linguistic processor.

This prototype has been implemented primarily as a proof of concept. Therefore, the main focus was on exploring principles rather than achieving efficiency.

3. Speech units recognizer

As outlined in Section 2, the speech units recognizer has to provide either hypotheses or has to compute the score of a suggested hypothesis. In both cases some sort of word spotting is used.

3.1. Benefits of word spotting

A word spotter is able to find the best match of a keyword in a signal. For a given keyword, it computes both the location in the signal and a corresponding acoustic score. Note that the spotting algorithm is not restricted to entire words, but can also be used for subword units (e.g. morphemes), multi-word expressions or even whole sentences. This has been exploited threefold:

1. By spotting every word of the recognizer lexicon and sending the best scored words as initial hypotheses to the linguistic processor.
2. The linguistic processor can request additional hypotheses at a specific location in the signal.
3. When the linguistic processor concatenates two hypotheses into a single one, the joined hypothesis can be scored by spotting it in the vicinity of the original ones. Scoring the joined hypothesis is important as its constituting hypotheses often overlap or have a gap in between, as will be explained in more detail in Section 4.2.

3.2. Word spotter

The word spotter is based on the Viterbi decoding algorithm and operates on phoneme models. Phonemes are represented by 40 context-independent monophone HMM/ANN models with a single state, as depicted in Figure 2.

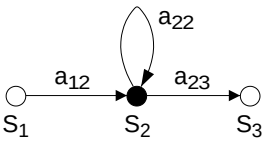

Figure 2: *Single state HMM with non-emitting entry and exit states. The stay state transition probability is denoted as* a_{22}*, the state change probability is* $a_{23} = 1 - a_{22}$*, and* $a_{12} = 1$.

As filler model a variant of the online garbage model in [1] is used. It computes the average of the N best local phoneme scores, whereby the top-value itself is not considered. Thus, the garbage model is never the best one, but is always one of the top candidates.

3.3. Duration model

The geometric duration model of standard HMMs based on static state transition probabilities has been replaced by a more adequate parametric model following a Gamma distribution. Explicit modeling of the state distribution was shown to improve recognition in [2] and Gamma distributions are reported to fit empirical distributions sufficiently well in [3].

Our duration model replaces the static state transition probabilities by dynamic ones which depend on the duration l spent so far in the current state. The state change transition probability $a_{23}(l)$ in function of this duration l is modeled as cumulative Gamma distribution.

$$a_{23}(l) = \int_0^l \frac{\beta^\alpha}{\Gamma(\alpha)} e^{-\beta l} l^{\alpha-1} \, dl \qquad (1)$$

The state self transition probability is then $a_{22} = 1 - a_{23}$. The free parameters of the Gamma distributions have been estimated from the mean and variance of the duration of HMM states from a standard Viterbi segmentation of the training set.[1]

In order to apply the duration model in the word spotter, the Viterbi recursion has been changed slightly by replacing the constant a_{ij} by the variable $a_{ij}(l)$:

$$\delta_t(j) = \left[\max_i \, \delta_{t-1}(i) \, a_{ij}(l)\right] \cdot b_j(\mathbf{o}_t) \qquad (2)$$

where $\delta_t(j)$ is the score for observing the feature vectors $\mathbf{o}_1$ to $\mathbf{o}_t$ and being in state S_j at time t; and $b_j(\mathbf{o}_t)$ is the probability to observe the feature vector $\mathbf{o}_t$ in state S_j.

3.4. Double normalized scores

The speech units hypotheses are scored with a double normalization technique which takes into account the number of frames in each phone and the number of phones in each word (phone-based normalized posterior confidence measure, [4]). According to our experience this measure significantly improves the accuracy of the word spotter score.

4. Linguistic processor

The linguistic processor is realized as an active island chart parser. This section outlines our motivation for this choice and explains further details.

4.1. Island chart parsing

The chart parsing framework conceived by Kaplan (1973) and Kay (1980) is very powerful for parsing natural language. Hypotheses are represented by *edges* which are stored in a data structure called *chart*. The chart represents all syntactic structures that have been found or tried so far. Thus, it shows also the parsing state and it prevents any work from being repeated. Active chart parsing uses an *agenda* to keep track of the grammar rules still to be applied. Depending on the implementation of the agenda different rule invocation strategies (top-down, bottom-up or mixed) and different search strategies (depth-first, breadth-first, best-first etc.) can be adopted. This is attractive as it provides a flexible framework that can be controlled by a strategy component.

An extension of standard chart parsing is island parsing, which enables the recognizer to start at "trustily" recognized constituents and proceed bidirectionally [5, 6]. There are two extremes in choosing the number of islands, either only a single one is selected or every word hypothesis is regarded as an island. Our implementation does the latter. Thus, additional hypotheses provided by the speech units recognizer can be added any time during the parsing process.

[1] shape parameter $\alpha = \mu^2/\sigma^2$, inverse scale parameter $\beta = \mu/\sigma^2$

The properties of island chart parsing conform to the requirements discussed in Sections 1 and 2.

4.2. Parsing text vs. speech

Chart parsing is commonly used for parsing written text. In this case there is always a unique unambiguous boundary between two consecutive words. These boundaries are the vertices to which the edges in the chart are anchored. Only adjoined edges can be combined.

In contrast to that, the word boundaries of hypotheses created by a word spotter are not constrained to be adjoined (typically they overlap or have gaps in between), so chart parsing is not directly applicable. This difference is illustrated in Figure 3.

The simplest solution is to redefine the term of adjacency of two edges. Instead of saying two edges e_1 and e_2 are adjacent if $e_1.end = e_2.start$ we define them to be still adjacent as long as the mismatch of the concerned boundaries is less than τ:

$$\text{adjacent}(e_1, e_2) = \begin{cases} true & \text{if } |e_1.end - e_2.start| < \tau \\ false & \text{else} \end{cases}$$

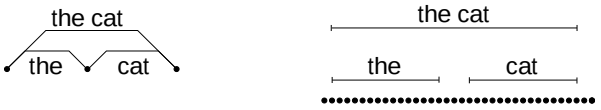

Figure 3: *When parsing text, the vertices of the chart are unique word boundaries (left) and it is thus always clear which words are neighbours and consequently can be concatenated to a larger hypothesis. This contrasts to acoustic hypotheses from a word spotter: generally such hypotheses do not fit; there is often an overlap or a gap (right). Therefore, when parsing speech, the vertices are acoustic frames and words can be connected when they meet some adjacency criterion.*

5. Control module

This section demonstrates the recognition process by showing how the statistical subsystem and the linguistic processor operate under the coordination of the control module. Beforehand, we explain why it makes sense to spot stems instead of full word forms only.

5.1. Spotting stems

Most German words are composed of a stem, an ending and optional prefixes. From one stem typically dozens of correct word forms can be generated and thus the number of full word forms is much higher than the number of stems.

In order to reduce the workload of the word spotter, we spot the stems instead of the full word forms. Only if the control module decides that a stem hypothesis has to be considered for further processing, this stem is expanded to full word forms which are again supplied to the spotter for scoring.

Some words are not subject to this optimization, namely uninflectable words and irregular forms. They are treated as full word forms and are spotted directly.

Consequently, the recognizer shown in Figure 4 has got three lexica that contain *full word forms*, *stems* and *endings*, resp. The full word forms lexicon mainly contains grammatical words, but also some irregular forms. The stem lexicon also informs for each stem which prefixes it can take. Additionally,

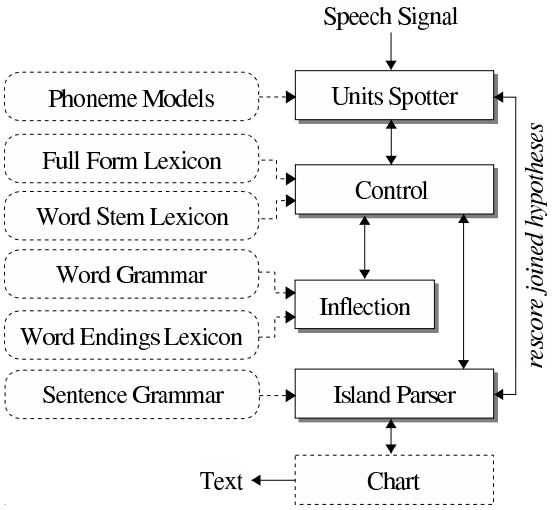

Figure 4: *Block-diagram of the recognizer*

there is a word grammar that tells the inflection module which stems can be combined with which endings to full word forms.

5.2. The recognition process

The recognizer depicted in Figure 4 processes an input speech signal as follows: First and independently of the subsequent processing, the utterance boundaries are detected by computing the best alignment of the model sequence *silence - filler model - silence* using dynamic programming.

Afterwards, all entries of the full form lexicon and the stem lexicon are spotted and the resulting hypotheses are stored in a list. Then the following steps are repeated:

1. Select the best scored hypothesis and remove it from the list.
2. Since a word or a stem can appear more than once in one and the same utterance, it has to be spotted again in the areas it has not already been found. The resulting hypotheses are added to the list. If the hypothesis selected in step 1 is a word, skip step 3.
3. For the *stem* selected in step 1, all possible full word forms are generated and spotted in the signal (near the found stem). The resulting hypotheses are added to the list. Go back to step 1.
4. The *full word form* is passed to the island chart parser that adds an edge representing the word hypothesis to the agenda. Then a full parse is executed, i.e., the rule invocation is repeated until the agenda is empty. In other words, all syntactic structures derivable from the word hypotheses known so far (i.e. that have been passed to the parser) are computed. Note that whenever two edges are combined to a single one, the combined edge has to be rescored by spotting the corresponding word sequence (cf. Section 4.2).
5. If no stop criterion is met, continue with step 1.
6. The best scored hypothesis spanning the whole utterance is printed out.

The stop criterion currently used is a timeout proportional to the utterance length.

5.3. Pruning

Unpromising edges are conservatively pruned. Before an edge is added to the agenda its score is compared to all edges spanning approximately the same part of the utterance. If the score of the edge in question is not among the N best, it is pruned. N depends on the number of words spanned by the edge (refer to Table 1 for exact values).

k	1	2	3	4	5	6	>6
N	100	10	5	4	3	3	2

Table 1: *An edge spanning k words is added to the agenda if its score is within the N best competing edge scores found in the chart, otherwise it is pruned.*

6. Experiments

The utterances of both the training and test set were spoken by a single male speaker in an office environment with low background noise using a head-set microphone sampled at 16 kHz. The feature vector extracted at each frame consists of 14 standard MFCCs plus the log-energy. These vectors are extracted from 25ms windows at a frame rate of 100 Hz.

6.1. Training

40 context-independent monophone HMM/ANN models with a single state have been trained. The neural network is a three layer perceptron with 120 and 80 neurons in the first and second hidden layer respectively. The input is composed by the features of the current frame plus 8 frames context. Thus, the input layer has $15 \cdot 9 = 135$ inputs. Each input neuron transforms the input to have zero mean and standard deviation 1 on the training data. There are 40 neurons in the output layer, one for each phoneme. The network was trained for 10 epochs on 887 newspaper sentences containing 51937 phonemes, corresponding to 101 minutes speech. The samples were chosen randomly such that the phonemes were trained uniformly. The weights are updated after each sample (stochastic learning) using standard back-propagation.

6.2. Testing

The test data consists of 72 sentences taken from a dictation book for pupils. The sentences used for training the HMM/ANN models and the test sentences are disjoint. The utterances contain 3, 6.8 and 11 words in minimum, mean and maximum respectively. Out of the 4067 different word forms that can be recognized, only 232 word forms appear in the test sentences. There are neither out-of-vocabulary words nor out-of-grammar sentences.

The 73 *Definite Clause Grammar* (DCG) rules of the sentence grammar cover the following aspects of the German language. Verb tenses: present tense (Präsens), simple past tense (Präteritum), present perfect tense (Perfekt), past perfect tense (Plusquamperfekt), moods: indicative and subjunctive, main sentences with free word order (affirmative and negative statements) and questions.

The experiments have been conducted for different neural networks (size of hidden layers, number of epochs trained etc.) and garbage model parameter values N, and only the best result is reported here. The choice of the network and N is thus optimized on the test data.

The best scored hypothesis is compared against the reference in terms of substitutions, insertions and deletions.

6.3. Results

A word error rate (WER) of 6.5% was achieved. The recognition results on the word and sentence levels are given in Table 2. The detailed results are: 461 correct words (C), 4 insertions (I), 7 deletions (D) and 21 substitutions (S) on a total number of 489 words (N).

words correct	94.3%
word accuracy	93.5%
sentences correct	66.7%
word error rate (WER)	6.5%

Table 2: *Recognition results in terms of words correct (C/N), word accuracy $(C-I)/N$, word error rate $(I+D+S)/N$ and number of sentences without any error on a speaker dependent recognition task with a 4k words dictionary.*

7. Conclusions and outlook

We have proposed a speech recognizer architecture which integrates both statistical and rule-based linguistic knowledge. As a proof of concept a prototype was implemented and a recognition experiment was carried out. This prototype already achieves a word recognition accuracy of 93.5%. We consider this very promising as the prototype is not yet able to benefit from predicting missing fragments, is based on a simple acoustic model and does not yet have an elaborate strategy component. The results validate the proposed approach and confirm its feasibility.

Future work will concentrate on the top-down prediction, a method to deal with out-of-grammar sentences and the strategy component.

8. Acknowledgement

This work has been the supported by the Swiss authorities in the framework of COST 278.

9. References

[1] H. Bourlard, B. D'hoore, and J.M. Boite. Optimizing recognition and rejection performance in wordspotting systems. In *Proc. of ICASSP*, volume 1, pages 373–376, 1994.

[2] L. Rabiner. Tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, pages 257–286, February 1989.

[3] D. Burshtein. Robust parametric modeling of durations in hidden Markov models. In *Proc. of ICASSP*, volume 1, pages 548–551, Detroit, Michigan U.S.A, May 1995.

[4] G. Bernardis and H. Bourlard. Improving posterior based confidence measures in hybrid HMM/ANN speech recognition systems. In *Proc. of ICSLP*, pages 775–778, Sydney, Australia, 1998.

[5] S. Steel and A. De Roeck. Bidirectional chart parsing. In *Proc. of the 1987 AISB Conference*, pages 223–235, University of Edinburgh, April 1987. John Wiley & Sons.

[6] O. Stock, R. Falcone, and P. Insinnamo. Island parsing and bidirectional charts. In *Proc. of the 12th COLING*, pages 636–641, Budapest, Hungary, 1988.