Using Language Modelling to Integrate Speech Recognition with a Flat Semantic Analysis

Dirk Bühler, Wolfgang Minker, Artha Elciyanti
Department of Information Technology, University of Ulm, Germany
{buehler,minker,artha}@it.e-technik.uni-ulm.de

Abstract

One-stage decoding as an integration of speech recognition and linguistic analysis into one probabilistic process is an interesting trend in speech research. In this paper, we present a simple one-stage decoding scheme that can be realised without the implementation of a specialized decoder, nor the use of complex language models. Instead, we reduce an HMM-based semantic analysis to the problem of deriving annotated versions of the conventional language model, while the acoustic model remains unchanged. We present experiments with the ATIS corpus (Price, 1990) in which the performance of the one-stage method is shown to be comparable with the traditional two-stage approach, while requiring a significantly smaller increase in language model size.

1 Introduction

In a spoken dialogue system, speech recognition and linguistic analysis play a decisive role for the overall performance of the system. Traditionally, word hypotheses produced by the automatic speech recognition (ASR) component are fed into a separate natural language understanding (NLU) module for deriving a semantic meaning representation. These semantic representations are the system's understanding of the user's intentions. Based on this knowledge the dialogue manager has to decide on the system reaction. Because speech recognition is a probabilistic pattern matching problem that ususally does not generate one single possible result, hard decisions taken after the speech recognition process could cause significant loss of information that could be important for the parsing and other subsequent processing steps and may thus lead to avoidable system failures. One common way of avoiding this problem is the use of N-best lists or word lattices as output representations, but these may require more complex NLU processing and/or increased processing times. In this paper, we follow an alternative approach: integrating flat HMM-based semantic analysis with the speech recognition process, resulting in a one-stage recognition system that avoids hard decisions between ASR and NLU. The resulting system produces word hypotheses where each word is annotated with a semantic label from which a frame-based semantic representation may easily be constructed. Fig. 1 sketches the individual processes involved in our integrative approach. The shaded portions in the figure indicate the models and processing steps that will be modified by versions using semantic labels. This will lead to an overall architecture, where a separate semantic decoding step (5) becomes dispensable.

One contribution of this work is to show that compared to other one-stage approaches (Thomae et al., 2003) such an integrated recognition system does not require a specialized decoder or complex language model support. Instead, basic bi-gram language models may be used.

We achieve the integration by "reducing" the NLU part to language modelling whilst enriching the lexicon and language model with semantic information. Conventional basic language modelling techniques are capable of representing this information. We redefine the units used in the language model: instead of using "plain" words, these are annotated with additional information. Such an additional information may consist of semantic labels and context information. For each of these annotated variants of a word, the phonetic transcription of the "plain" word is used. Consequently, the ASR cannot decide which variant to choose on the basis of the acoustic model. No retraining of the acoustic model is necessary. The speech recogniser produces word hypotheses enriched with semantic labels.

The remainder of this paper is structured as follows: In the next section we give a brief overview of the Cam-

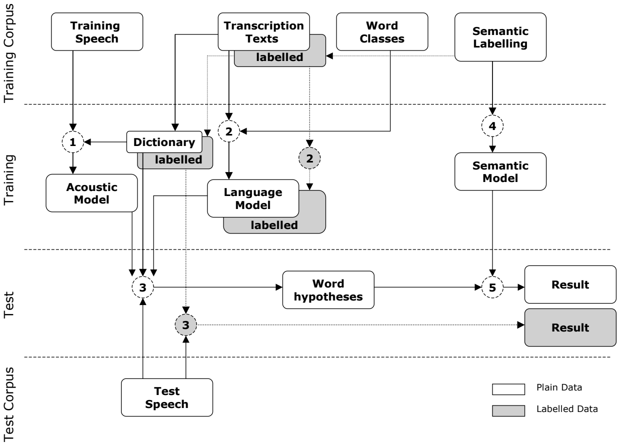

Figure 1: *Principal knowledge sources and models of speech recognition and semantic analysis. Shaded parts constitute the changes when using a one-stage approach. The numbers indicate the following computational steps: (1) acoustic model parameter estimation, (2) language modelling, (3) Viterbi acoustic decoding, (4) semantic model parameter estimation, (5) Viterbi semantic decoding.*

bridge HTK software we used for our experiments with the ATIS corpus. In Section 3 we outline the HMM-based parsing method. The basic approach for adding information into the speech recogniser language model is described in Section 4. In Section 5 we discuss our experiments and present speech recognition results. Finally, we conclude by pointing out further possible improvements.

2 Acoustic Modelling and Speech Recognition Using HTK

Speech recognition may be formulated as an optimisation problem: Given a sequence of observations O consisting of acoustic feature vectors, determine the sequence of words W, such that it maximizes the conditional probability $P(W|O)$. Bayes' rule is used to replace this conditional probability which is not directly computable by the product of two components: $P(O|W)$, the acoustic model, and $P(W)$, the language model.

$$[W]_{opt} = \underset{\mathrm{W}}{\operatorname{argmax}} \{P(W)P(O|W)\} \qquad (1)$$

The Cambridge Hidden Markov Model Toolkit (HTK) (Young et al., 2004) can be used to build robust speaker-independent speech recognition systems. The tied acoustic model parameters are estimated by the forward-backward algorithm.

reference word:	case frame or concept identifier
case frame:	set of cases related to a concept
case:	attribute of a concept
case marker:	surface structure indicator of a case
case system:	complete set of cases of the application

Figure 2: *Semantic case grammar formalism.*

The HTK Viterbi decoder can be used together with a probabilistic word network that may be computed from a finite state grammar or the bi-gram statistics of a text corpus. The decoder's token passing algorithm is able to produce word hypotheses lattices or N-best lists of recognition results. Internally this word network is combined with a phonetic transcription dictionary to produce an expanded network of phoneme states. Usually, one phoneme or triphone is represented by five states.

For our experiments with the ATIS corpus, the acoustic model is constructed in conventional way. We use 4500 utterances to train a triphone recogniser with 8 Gaussian mixtures. A triphone count of 5929 physical triphones expand to 27619 logical ones. The acoustic model is used for both the two-stage and the one-stage experiments.

3 HMM-Based Semantic Case Frame Analysis

In the domain of spoken language information retrieval, spontaneous effects in speech are very important (Minker, 1999). These include false starts, repetitions and ill-formed utterances. Thus it would be improvident to base the semantic extraction exclusively on a syntactic analysis of the input utterance. Parsing failures due to ungrammatical syntactic constructs may be reduced if those phrases containing important semantic information could be extracted whilst ignoring the non-essential or redundant parts of the utterance. Restarts and repeats frequently occur between the phrases. Poorly syntactic constructs often consist of well-formed phrases which are semantically meaningful.

One approach to extract semantic information is based on *case frames*. The original concept of a case frame as described by Fillmore (Fillmore, 1968) is based on a set of universally applicable cases or case values. They express the relationship between a verb and its nouns. Bruce (Bruce, 1975) extended the Fillmore theory to any concept-based system and defined an appropriate semantic grammar whose formalism is given in Fig. 2.

In the example query

> *<you> <get> could you give me a ticket* ***price*** *on [uh] [throat_clear] a flight* ***first*** *class from* ***San Francisco*** *to* ***Dallas*** *please*

a typical semantic case grammar would instantiate the following terminals:

- *price*: this *reference word* identifies the concept airfare (other concepts may be: *book*, *flight*, ...)

- *from*: *case marker* of the *case* from-city corresponding to the departure city *San Francisco*
- *to*: *case marker* of the *case* to-city corresponding to the arrival city *Dallas*
- *class*: *case marker* of the *case* flight-class corresponding to *first*
- *case system*: from, to, class, ...

The parsing process based on a semantic case grammar typically considers less than 50% of the example query to be semantically meaningful. The hesitations and false starts are ignored. The approach therefore appears well suited for natural language understanding components where the need for semantic guidance in parsing is especially relevant.

Case frame analysis may be used in a rule-based case grammar. Here, we apply HMM-based modelling instead (Pieraccini et al., 1992; Minker et al., 1999). In the frame-based representation, the semantic labelling does not consider all the words of the utterance, but only those related to the concept and its cases. However, in order to estimate the model parameters, each word of the utterance must have a corresponding *semantic label*. Thus, the additional label (null) is assigned to those words not used by the case frame analyzer for the specific application. A semantic sequence consists of the basic labels <concept>, (m:case), (v:case) and (null) corresponding respectively to the reference words, case markers, values and irrelevant words.

Relative occurrences of model states and observations are used to establish the Markov Model, whose topology needs to be fixed prior to training and decoding. Semantic labels are defined as the states s_j. All states such as the examples (v:at-city), (null) and <ground-service> shown can follow each other; thus the model is *ergodic*.

In direct analogy to the speech recognition problem (equation 1), the decoding consists of maximizing the conditional probability $P(S|W)$ of some state sequence S given the observation sequence W:

$$[S]_{opt} = \underset{\mathrm{S}}{\operatorname{argmax}} \{P(S)P(W|S)\} \qquad (2)$$

Given the dimensionality of the sequence W, the exact computation of the likelihood $P(W|S)$ is intractable. Again, bi-grams are a common approximation in order to robustly estimate the Markov Model parameters, the state transitions probabilities $P(s_j|s_i)$ and the observation symbol probability distribution $P(w_m|s_j)$ in state j.

In contrast to speech recognition, the computation of the model parameters can be achieved through maximum likelihood estimation, i.e. by counting event occurrences. Usually a back-off and discounting strategy is applied in order to improve robustness in the face of unseen events. An HMM-based parsing module may be conceived as a probabilistic finite state transducer that translates a sequence of words into a sequence of semantic labels. The semantic labels denote word's function in the semantic representation.

Although the flat semantic model has known limitations with respect to the representation of long-term dependencies, for practical applications it is often sufficient. It has been shown that several methods, such as contextual observations and garbage models, exist that enhance the performance of HMM-based stochastic parsing models (Beuschel et al., 2004).

4 Adding Information to the Language Model

As mentioned above, the language model $P(W)$ represents the probability of a state sequence. With the bigram approximation $P(W) \approx P(w_i|w_{i-1})$ this probability becomes a transition probability between words in a word network.

By adding information to the language model (cf. shaded parts of Fig. 1) we modify the word network in a way that instead of "plain" words as nodes the network should now contain "annotated" variants of these original words. The annotation then encodes some additional information that is relevant to the further processing in the dialogue system, but does not affect the pronunciation of the word. By introducing such labelled word variants, it is possible to encode some additional relations that exist between the labels rather than between the words.

Consider the following utterance and a corresponding labelling of each word with additional information:

Show-(null)
me-(null)
ground-<ground-service>
transportation-<ground-service>
for-(null)
Dallas-(v:at-city)

The word network computed from utterances of the latter form instead of plain texts will represent the fact that after a word labelled as (null), a city name labelled as (v:at-city) is much more likely than labelling as (v:from-city) or (v:to-city). In order to compute the modified version of the network it is only necessary to replace the words by their labelled variants in the training corpus and to compute the bi-gram statistics from this modified corpus (cf. step (2) in Fig. 1).

For expanding the word network into a network of phoneme states as required by the speech recognition, it is necessary to modify the phonetic transcription dictionary accordingly: for each labelled variant of a word appearing in the labelled training texts, the respective unlabelled word entry is copied. The Viterbi decoder will now output sequences of annotated words (step (3)).

The language model may not only be enriched by semantic labels. Other information, such as the context of the word may also be used. A language model labelled with a context that consists of one word on the left is essentially a tri-gram model. There is a trade-off between what the network can express and its size. Using too many different labels for each word in the network may quickly result in word networks impractical for real-time use. For our experiments within the ATIS domain,

Table 1: *Word network sizes for different labelling techniques. "Expanded" refers to the phoneme state network.* t *denotes estimated per utterance processing time.*

Method	Words		Expanded		t
	Nodes	Arcs	Nodes	Arcs	[s]
ASR	465	2,939	5,985	15,482	14.1
ASR/Cl	709	4,382	8,775	21,360	67.8
ASR/Co	3,603	14,126	46,043	84,464	15.9
ASR/CC	6,632	18,117	83,207	115,703	117.0
ASR+	1,243	7,108	16,210	38,314	13.4
ASR+Co	2,269	10,272	29,535	59,209	20.8
ASR+N	1,556	7,966	19,516	42,996	14.7

Table 1 summarises the word network sizes for different labelling methods. Here, "ASR" refers to the original base-line unlabelled language model. "ASR/Cl" is a simple class-based language model with manually defined classes. A left context of one word was used in "ASR/Co", and combined in with classes in "ASR/CC". These labelled versions may be used in the two-stage approach to improve the speech recognition results.

"ASR+", "ASR+Co", "ASR+N" refer to semantically labelled language models. "ASR+" is directly trained on the semantically labelled training texts. "ASR+Co" furthermore includes a left context of one *semantic label,* whereas "ASR+N" includes sub-label numbering.

As can be seen from the numbers, word classes as well as the semantic methods only incur a modest increase in network size. The word-based methods, however, significantly inflate the model. Although we have not systematically recorded the time necessary for recognizing the test set with these networks, it is fair to say the time escalates from minutes to hours.

The last column in Table 1 denotes the estimated average per-utterance processing time in seconds. The numbers were obtained on a Pentium 4 with 2,6 GHz speed and 1 GB of RAM running Linux.

5 Speech Recognition Experiments

For our experiments with the ATIS corpus, the stochastic parsing model is computed from 1500 utterances, manually annotated in a bootstrapping process. We use 13 semantic word classes (e.g. /weekday/, /number/, /month/, /cityname/). The semantic representation consists of 70 different labels. Splitting sequences of identical labels into numbered sub-labels results in 174 numbered labels. The semantic representation focuses on the identification of key slots, such as origin, destination and stop over locations, as well as airline names and flight numbers. Word sequences containing temporal information such as constraints on the departure or arrival time are not analysed in detail. Instead, all these words are marked with (v:arrive-time) or (v:depart-time), respectively. The test corpus consists of 418 utterances which were manually annotated with semantic labels.

For the two-stage approach different word-based language models (plain, class-based, context, combined) were used (cf. section 4). An N-best decoding was performed and 20 hypotheses were subsequently labelled by the stand-alone stochastic parser. After that, the result with the maximum combined probability value was chosen. In the one-stage approach, two refinements (context and numbering) were applied to the basic semantic language model.

Table 2: *Word correctness results.*

Method	Correct	Accuracy	Sentence
	[%]	[%]	[%]
ASR	82.66	67.20	20.56
ASR/Co	85.53	72.74	26.61
ASR/Cl	84.33	70.96	24.60
ASR/CC	85.43	72.68	27.42
ASR+	85.04	72.03	25.60
ASR+Co	85.02	71.90	25.60
ASR+N	85.13	72.16	25.81

Tables 2 and 3 present the results of these experiments. They are based on word recognition and concept recognition performance, respectively. The columns titled "Correct" and "Accuracy" refer to word correct rate and word accuracy, as well as to their concept-level equivalents. The "Sentence" column lists the percentage of completely correctly decoded sentences. For the two-stage approach, the numbers in Table 2 denote the performance of the speech recognition system alone (step (3) in Fig. 1). For the one-stage approach, the semantic labels were removed after decoding in order to obtain the plain word sequences. It can be seen that the word-based recognition benefits both from word-based additions to the language model, as well as from semantic labels in about the same rate.

Table 3: *Concept correctness results.*

Method	Correct	Accuracy	Sentence
	[%]	[%]	[%]
NLU	96.97	96.97	85.69
ASR	76.67	60.73	18.18
ASR/Co	78.62	65.92	24.80
ASR/Cl	78.69	66.97	24.60
ASR/CC	78.02	63.77	13.31
ASR+	77.72	64.80	21.57
ASR+Co	77.74	64.69	21.77
ASR+N	77.44	64.58	22.18

Table 3 summarizes the concept-level results. Here, the semantic labels are also compared against the reference. Numbers in sub-labels are ignored, however. The "NLU" row denotes the performance on perfectly recognized data, i.e. on the training transcriptions. One-stage integrated recognition produces competitive recognition rates when compared to the two-stage approach. Even though in the two-stage approach, each stage's representation can be fine-tuned separately.

It seems interesting to note a subtle difference between the decoding procedures of the two-stage and the one-stage architectures. In a stand-alone stochastic parser, Viterbi decoding is used for word-to-label correspondences. The probability of a transition from semantic state s_i to s_j is thus defined as the product $P(w_j|s_j)P(s_j|s_i)$, where $P(w_j|s_j)$ is the probability of observing w_j in state s_j. In contrast, if a labelled language model is used the transition probability is $P(w_j|w_i)$, where w_i and w_j are pairs of the actual words and their associated labels, so the surface form of the last word influences the transition as well (not only its label).

6 Conclusions and Future Work

It can be shown that a flat HMM-based semantic analysis does not require a separate decoding stage. Instead it seems possible to use the speech recogniser's language model to represent the semantic state model, without compromising recognition in terms of word or slot error rate.

For a stand-alone speech recognition component, it seems advantageous to use a class-based or context-based language model, since it improves the word recognition score. For the stochastic parsing, numbered sub-labels provide best results. With N-best decoding, the stochastic parser can be used to select the best overall hypothesis.

A number of improvements and extensions may be considered for the different processing stages. Firstly, instead of representing compound airport and city names such as "New York City" as word sequences, they could be entered in the dictionary as single words, which should avoid certain recognition errors. In addition, an equivalent of a class-based language model should be defined for semantically annotated language models. Also, contextual observations, i.e. the use of a class of manually defined context words could help the stochastic parser to address long-term dependencies that have so far proved difficult. Finally, the ATIS task results in relatively simple semantic structures and yields a limited vocabulary size. It would be interesting to apply our proposed techniques to a more complex domain, such as an appointment scheduling task (Minker et al., 1999), implying a more natural speech-based interaction. This would enable us to validate our approach on larger vocabulary sizes.

References

C. Beuschel, W. Minker, and D. Bühler. 2004. Strategies for Optimizing a Stochastic Spoken Natural Language Parser. In *Proceedings of International Conference of Speech and Language Processing, ICSLP*, pages 2177–2180, Jeju Island (Korea), October.

B. Bruce. 1975. Case Systems for Natural Language. *Artificial Intelligence*, 6:327–360.

Ch. J. Fillmore. 1968. The case for case. In Emmon Bach and Robert T. Harms, editors, *Universals in Linguistic Theory*, pages 1–90. Holt and Rinehart and Winston Inc.

W. Minker, M. Gavaldà, and A. Waibel. 1999. Stochastically-based Semantic Analysis for Machine Translation. *Computer Speech and Language*, 13(2):177–194.

W. Minker. 1999. Stochastically-based semantic analysis for ARISE - Automatic Railway Information Systems for Europe. *Grammars*, 2(2):127–147.

R. Pieraccini, E. Tzoukermann, Z. Gorelov, J.L. Gauvain, E. Levin, C-H. Lee, and J.G. Wilpon. 1992. A Speech Understanding System Based on Statistical Representation of Semantics. In *Proceedings of International Conference on Acoustics Speech and Signal Processing, ICASSP*, pages 193–196, March.

P. Price. 1990. Evaluation of Spoken Language Systems: The ATIS Domain. In *Proceedings of DARPA Speech and Natural Language Workshop*, pages 91–95, June.

M. Thomae, T. Fabian, R. Lieb, and G. Ruske. 2003. A one-stage decoder for the interpretation of natural speech. In *Proceedings of International Conference of Speech and Language Processing, ICSLP*, pages 56–64.

S. Young, G. Evermann, D. Kershaw, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, 2004. *The HTK Book (for HTK Version 3.2)*. Cambridge University Engineering Department.