Stochastic Modelling of Sentence Semantics in Speech Recognition

Włodzimierz Kasprzak and Paweł Przybysz

Abstract. A stochastic approach to spoken sentence recognition is proposed for the purpose of an automatic voice-based dialogue system. Three main tasks are distinguished: word recognition, word chain filtering and sentence recognition. The first task is solved by typical acoustic processing followed by phonetic word recognition with the use of Hidden Markov Models (HMM) and Viterbi search. For the second solution an N-gram model of natural language is applied and a token-passing search is designed for the filtering of important word chains. The third task is solved due to a semantic HMM of sentences. The final sentence is recognized and a meaning is assigned to its elements with respect to given application domain. A particular spoken sentence recognition system has been implemented for train connection queries.

1 Introduction

Spoken sentence analysis is a muli-disciplinary problem in which techniques are involved that originate from signal processing, phonetics, computational linguistics and natural language processing [1]. In engineering disciplines speech processing can be decomposed into stages of acoustic-, phonetic-, lexical-, syntactic-, semantic- and pragmatic analysis. The main application of spoken sentence analysis, that we consider in this paper, are automatic spoken language dialog systems (e.g. automatic railway information system) [2]. This limits our interest to first 4 stages of speech processing and to a limited-scope semantic analysis.

General semantic analysis of natural languages usually requires large data bases. For example, research in psycholinguistics identifies how humans process a natural language. The goal of *WordNet* project [5] was to create a data base for *lexical and semantic memory*, i.e. allowing to search the dictionary by associations

Włodzimierz Kasprzak · Paweł Przybysz
Institute of Control and Computation Engineering
Warsaw University of Technology
ul.Nowowiejska 15/19, 00-665 Warszawa, Poland
e-mail: {W.Kasprzak,P.Przybysz2}@elka.pw.edu.pl

R. Burduk et al. (Eds.): Computer Recognition Systems 4, AISC 95, pp. 737–746.
springerlink.com

originating from the grammar form of a word and/or semantic relations between words in given language. Its use is much broader than automatic dialogue systems - a multi-language WordNet will support automatic language translation and speech understanding (the pragmatic analysis of speech).

The studies on language *ontology* [11], among others, lead to inheritance relations between words based on their meanings. Such relations can then support the pragmatic analysis of sentences, allowing to exchange words with similar meaning depending on the context of their current use.

The syntactic-semantic analysis of spoken sentences is most often based on: case-frame grammars [6], probabilistic grammars [7] and stochastic models [10]. They seem to be useful for specific dialogue systems in which only a limited subset of a natural language need to be considered. The advantage of a case-frame grammar in comparison to context-free formal grammars is the ability to express large number of sentence configurations without a need to generate them all. The idea is to build the sentence ”around” a key word which represents the main use case of a sentence.

The statistical approach to speech recognition is widely recognized. The acquired signal is contaminated by noise, the signal shape is of high variability and depends on the speaker and even the meaning of properly syntactically recognized sentences is often ambiguous. All this motivates the use of stochastic models in speech recognition.

The paper is organized as follows. Section 2 presents and discusses the structure of our system. The next section 3 concentrates on the acoustic-phonetic modelling of spoken words. In section 4 an N-gram estimation approach is presented. In section 5 the *token-passing* search is presented, with the goal to filter possible word sequences by using N-gram models. In section 6 the idea of a HMM for sentence recognition and meaning assignment is introduced. An example is presented in section 7 - the recognition of train departure questions.

2 System Structure

The speech recognition system is structured into man abstraction levels: acoustic analysis, phonetic analysis (word recognition), word chain filtering and sentence recognition. The symbolic part of sentence analysis is split into the syntax-driven detection of word sequences and a semantic-driven sentence recognition. From syntactic point of view a sentence is a chain of words, where each word is again a sequence of phonemes. From system point of view we can distinguish the need of *model creation* and of *model use* (for the purposes of word and sentence recognition).

In Fig.1 the proposed structure of our speech analysis system is outlined. The input signal is converted to a sequence of numeric feature vectors, $\mathbf{Y} = [\mathbf{y_1}, ..., \mathbf{y_n}]$, that represent acoustic features of consecutive signal frames. This is a feature detection step.

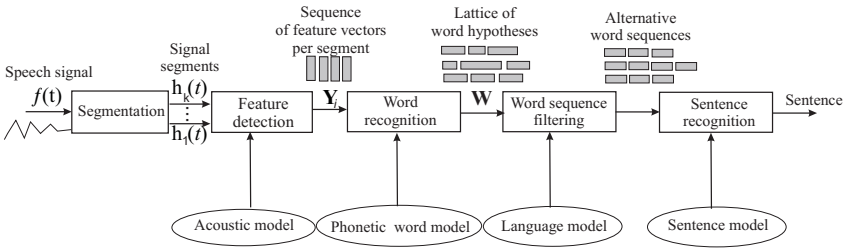

Fig. 1 The speech recognition system

The next step is to recognize a sequence of words, $\mathbf{W} = (\mathbf{w_1}, ..., \mathbf{w_K})$, that in a best way (with highest probability among competitive candidates induced by the language model) matches the measured sequence $\mathbf{Y}$.

The word sequence recognition step (in some works also called as *decoding*) solves the following stochastic task:

$$\hat{W} = \max_{W}[p(W|Y)] \tag{1}$$

But the distribution $p(W|Y)$ represents a diagnostic relation (from observed effect to hidden effect). Hence, it is difficult to obtain its direct model. Applying the Bayes rule we obtain:

$$p(W|Y) = \frac{p(Y|W)\ p(W)}{p(Y)} \tag{2}$$

that means, we express the problem in terms of prior probabilities:

$$\hat{W} = \max_{W}[p(Y|W)]\ p(Y)] \tag{3}$$

For word sequence recognition we shall create two stochastic models:

1. a phonetic model - which gives the conditional probability of signal measurement for given sentence, $p(Y|W)$, and
2. a language model - which gives probabilities of word sequences, $p(W)$.

The Hidden Markow Model (HMM) [10], [12] allows for the transition both from acoustic to phonetic description and from phonemes to words. A sequence of phonetic entities representing a spoken is encoded as a sequence of HMM-states, $(s_1 ... s_m)$, whereas the measured sequence of frame features, $(y_1 ... y_n)$ is a sequence of observation variable values in HMM. The *Viterbi search* [8] is applied to find the best match between such two sequences, or in other words - the best path in the HMM of all words.

The required stochastic language model, $P(W) \sim P(w_1, w_2, ..., w_n)$, should allow to select proper sentences in given language and help to reject wrongly generated

word sequences. For example, P(”W czym mogę Panu pomóc”) should be of high probability value if compared to P(”Jak biegać kot chodzić pies").

In practice it will be difficult to learn the estimates of all individual sentences. We shall rely on shorter word sequences and learn probabilistic models, called N-grams, in which a word's probability is conditioned upon at most $N-1$ direct predecessor words [4], [7], [12]. In this work the language model will be used by a *token-passing* search to perform a filtering of word chains.

Another HMM will be applied for sentence recognition, but it will rather represent the ”semantics” of a sentence in given application. The words will be observed and accepted by HMM states due to their meaning and not their syntactic role. Thus, the HMM states represent key structures of a sentence to that allow to recognize the type of sentence (i.e. question, information, order) and to associate an interpretation (action, answer) to it.

3 Acoustic-Phonetic Model of Spoken Words

3.1 Acoustic Model

Here we apply the well-known scheme of *mel-cepstral* coefficients (MFCC) [1]. In every considered signal segment of around 16 ms duration time a vector of 38 numeric features is detected.

3.2 Phonetic Model

All words in a dictionary are given phonetic transcriptions in terms of 39 phonemes. Each phoneme is divided into 1-, 2- or 3 parts, called three-phones.

The acoustic and phonetic models of spoken words (from given dictionary) are combined into a single HMM. The phonetic model of every word is expressed by the structure of a left-to-right HMM, i.e. its states, s_i, and transition probabilities, a_{ij}, between pairs of states s_i and s_j, where s_j *follows* s_i.

The acoustic feature vector y may be attached to a HMM via possible three-phones and their output probabilities, λ_{jm} (for observed phone m in state j) [10], [12]. In a so called semi-continues probability model, the output probability takes the form of a Gaussian mixture, i.e. the probability of observing in state j a vector y is computed as:

$$b_j(y) = \sum_{m=1}^{M} \lambda_{jm}\, N(y; \mu_{jm}, \sigma_{jm}) \tag{4}$$

3.3 Word Hypothesis Lattice

A slightly modified *Viterbi search* [1] is applied to find a best match between every path in the HMM word model and a sequence of signal segments, that we assume to contain spoken utterances of 1-, 2- or 3-sylab words.

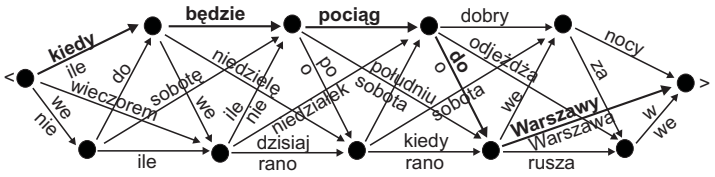

Fig. 2 Illustration of the word hypothesis lattice. Besides the optimal sequence ("Kiedy będzie pociąg do Warszawy") many other competing sequences are still possible.

The competing word sequences can be represented by a lattice-graph, where the nodes are ordered according to the (assumed) sylab (and time) index (Fig. 2). Every arc in this graph represents the best word recognition results achieved for the appropriate signal frame. More than one competing word hypothesis can be stored for a given signal frame, depending on their quality scores.

4 Language Model - N-Grams

In a formal grammar a sentence W belongs or not to some language L. Hence the probability $P(W)$ that the sentence is proper in language L is 0 or 1.

In a probabilistic context-free grammar (PCFG) [7], [12] every production rule has assigned its probability value. Hence, the probability of a sentence is a function of rule probabilities needed to generate this sequence. The key difficulty is to design this function that combines probabilities.

In opposite to PCFG-s stochastic word sequence models, called N-grams, are easy to generate and to modify. But they have to cope with the ”sparse” nature of training data.

4.1 The N-Gram Model

Let $W, W = (w_1 w_2 ... w_n)$, be a word chain. When the word chain satisfies a Markov random process of order $(N-1)$, where $N \leq n$, its N-gram model is:

$$P(W) = \prod_{i=1}^{n} P(w_i | w_{i-N+1}, ..., w_{i-1}) \tag{5}$$

In practice the N-grams are limited to $N = 2, 3, 4$ and such special cases are called as [4]: unigram, $P(w_i)$, bigram, $P(w_i|w_{i-1})$, three-gram, $P(w_i|w_{i-1}, w_{i-2})$, four-gram, $P(w_i|w_{i-1}, w_{i-2}, w_{i-3})$. These distributions are estimated during a learning process that is based on the *maximum likelihood* rule (ML). The elementary learning approach is to count the appearance frequency of given word chain in the training data.

Let $C(w_{i-N+1} ... w_{i-N+N})$ is the number of training set appearances of given word chain of length N. The unigram estimation is simple (w_k means any word in the training set):

$$P(w_i) = \frac{C(w_i)}{\sum_{w_k} w_k} \tag{6}$$

The bigram estimation needs to count the number of appearances of given pair of words and of the first word:

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})} \tag{7}$$

This can be generalized for an N-gram as:

$$P(w_i|w_{i-N+1}, w_{i-N+2}, ..., w_{i-1}) = \frac{C(w_{i-N+1}w_{i-N+2}...w_i)}{C(w_{i-N+1}w_{i-N+2}...wi-1)} \tag{8}$$

4.2 N-Gram Learning with Smoothing

The drawback od elementary N-gram learning approach is the sparsity of available training data, even when the number of words is large. For example in a text that contains several million of words around 50% of three-grams may appear only one time, and around 80% of three-grams - no more than 5 times. Obviously such sparse data set will lead to large N-gram estimation errors.

To eliminate such drawback a common procedure in N-gram learning is to apply a *smoothing* operation on the estimated probability distribution. The simplest smoothing method, called *Laplace smoothing*, is to add 1 to the number a sequence appeared in the training set [4].

4.3 The Katz Smoothing Method

In this work we apply the so called *Katz* method [9]:

1. for frequent N-tuples of words apply the elementary estimation approach,
2. for rare N-tuples of words apply the '*good Turing estimate*,
3. for non-observed N-tuples of words apply a smoothing method by *returning to (N-1)-grams*.

The *good Turing estimate* takes the form (for a bigram):

$$C^*(w_i|w_{i-1}) = \begin{cases} r, & r > k \\ d_r r, & 0 < r \leq k \\ \alpha(w_{i-1})P(w_i), & r = 0 \end{cases} \tag{9}$$

Here r denotes the appearance number, i.e. $r = C(w_{i-1}, w_i)$, d_r - the discount rate, and α - a normalization coefficient. The parameter k, that selects one of the 3 approaches, is set by default - for example it may be set to 5.

$\alpha(w_{i-1}$ is estimated in such a way to satisfy the following condition:

$$\sum_{w_i} C^*(w_{i-1}, w_i) = \sum_{w_i} C(w_{i-1}, w_i) \tag{10}$$

The discount rate is computed as:

$$d_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} \tag{11}$$

for $r = 1, 2, ..., k$. Where n_r denotes the number of N-tuples, that appear exactly r times in the training set, and r^* is the good Turing estimate, i.e.:

$$r_r^* = (r+1)\frac{n_{r+1}}{n_r} \tag{12}$$

Thus the probabilities in a three-gram model are estimated according to the following options:

$$P^*(w_i|w_{i-1}, w_{i-2}) = \begin{cases} \frac{C(w_{i-2}w_{i-1}w_i)}{C(w_{i-2}w_{i-1})}, & r > k \\ d_r \frac{C(w_{i-2}w_{i-1}w_i)}{C(w_{i-2}w_{i-1})}, & 0 < r \leq k \\ \alpha(w_{i-1}, w_{i-2})P(w_i|w_{i-1}), & r = 0 \end{cases} \tag{13}$$

5 Token-Passing Search

We developed a general-purpose search, called *token-passing* search, in order to convert the lattice of word hypotheses into meaningful sentences of words. This is a breadth-first search controlled by evaluations of partial word sequences (paths) and which uses the N-gram model, corresponding to the dictionary of current application domain, to prune paths with inconsistent n-tuples of words. Thus the *token passing* search takes as input: 1) the *dictionary* model, represented by the integrated HMM for words, and 2) the *language* model given by N-grams.

A *token* means a data structure associated with the search tree node that contains: 1) the score (evaluation) of corresponding path, P, and 2) a pointer, *link*, to path description (a structure R).

A search tree node passes its token to its successor nodes. A new structure of type R is created for every successor node that holds: a link to obtained predecessor token, the new added word with its lattice time index and its quality score. Then the score of every new token is modified by the product of N-gram probability (of added word upon the condition of $N-1$ predecessor words) and its quality score.

6 Semantic HMM

A sentence is a sequence of words. We propose the use of another HMM model for representation of a stochastic syntax of sentences in given application. States in this model correspond to word categories, whereas observations can be specific words given in the dictionary.

The categories of words are distinguished from point of view of the application domain. Hence, the states of such HMM represent *semantic* entities, rather than syntactic ones.

Semantic HMM is based on assumption that every sentence is combined from parts containing *atomic* semantic information. For example, an atomic part may be: a question form (*when, where, at what time*), a time period (*at eight a.m., afternoon, at evening*) or destination (*Warszawa*).

Prior probability of a valid sequence of semantic parts is expressed by transition probabilities along the appropriate path in HMM. Posterior probability is found by including the word acceptance (observation) probabilities in states along such path. Specific features of the semantic HMM are:

- It can represent a sentence category (many sentences with the same meaning) rather than a single sentence ;
- During the recognition process a particular sentence from given category is detected along with sentence meaning. Hence, speech recognition system could execute an action according to recognized sentence;
- Semantic parts can be used as a elements of many semantic HMMs.

In a general use of this approach the HMM states would be rather of syntactic meaning, i.e. they may represent:

- the role in a sentence: subject, predicate, object;
- a syntax category: noun, verb, conjunction, adjective, number, etc.

7 Example

The HMM in Fig. 3 allows a very flexible modelling of sentences. It represents the application domain: *question on train departure*. The word dictionary contains at least 38 words in base form, like: from, to, train, hour, minute, day, when, Monday, today, etc. This can be further extended by the names of cities for which train connections are needed.

Some words appear multiple times, but in different grammar forms of common base word. They are included in the word recognition stage. For simplicity of the semantic model, after the word sequence detection they are converted into the base form.

The states of HMM represent following 11 meaning (and not directly syntactic) categories: *question attribute, departure form, day, day-time, train attribute, train, from, to, departure city, destination city, end of sentence*. Every state can emit several words with specific probabilities. For example the state called *train attribute* can accept the following words: "nearest", "last", "fastest", "first".

Here are examples of valid sentences accepted by this model: *When the train from Warszawa to Krakow departs?* (”Kiedy będzie pociąg do Warszawy z Krakowa?”), *When the first train from Krakow to Warszawa departs?* (”Kiedy jest pierwszy pociąg z Krakowa do Warszawy?”), *When the train from Krakow to Warszawa departs tomorrow?* (”Kiedy jest jutro pociąg z Krakowa do Warszawy?”), *When the*

next train to Wrocław departs? (”Kiedy jest najbliższy pociąg do Wrocławia?”), *When the first train from Krakow to Warszawa departs?* (”O której odjeżdża pierwszy pociąg z Krakowa do Warszawy?”), *When the first train from Krakow to Warszawa departs tomorrow afternoon?* (”O której będzie jutro pierwszy pociąg po południu z Krakowa do Warszawy?”), *When tomorrow afternoon the first train from Krakow to Warszawa departs?*, (”O której jutro będzie pierwszy pociąg po południu z Krakowa do Warszawy?”), *When afternoon the first train from Krakow to Warszawa departs a day after tomorrow?* (”O której pojutrze będzie pierwszy pociąg po południu z Krakowa do Warszawy?”), *When the train to Warszawa departs?* (”O której jest pociąg do Warszawy?”), *When tomorrow the train to Grodzisk leaves ?* (”O której jutro odjeżdża pociąg do Grodziska?”).

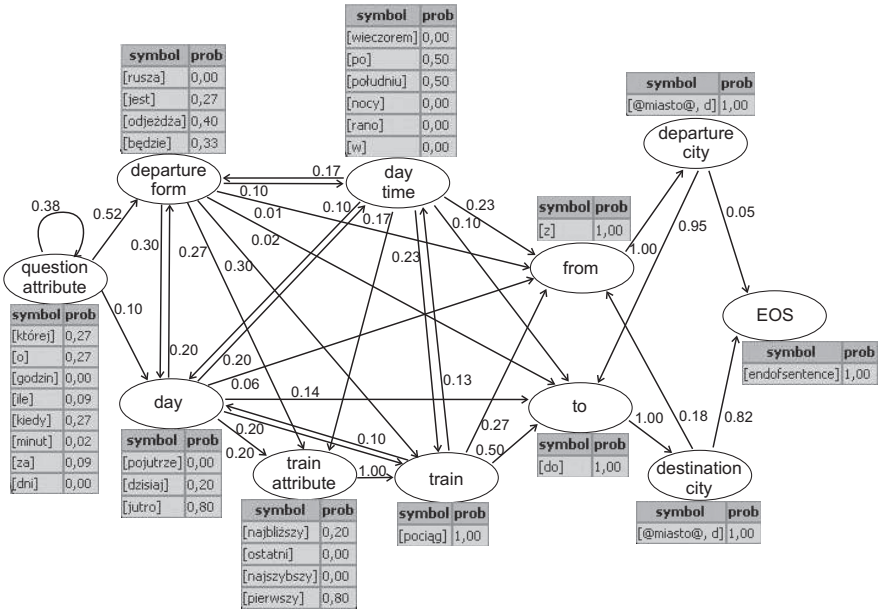

Fig. 3 Illustration of a *semantic* HMM for train departure information.

8 Summary

A design of a spoken sentence recognition system applied for the purpose of man-machine dialogue systems was proposed. The novel elements can be characterized as follows:

1. the stochastic modelling of words and sentences in given language by means of HMMs and N-gram models;
2. a token-passing search as a word filtering stage in this system;
3. a semantic HMM for spoken sentence recognition and meaning (action) association in some application domain (e.g. train departure information).

Acknowledgements. This work was supported by the Polish Ministry of Science and Higher Education by the grant N N514 1287 33.

References

[1] Benesty, J., Sondhi, M.M., Huang, Y. (eds.): Springer Handbook of Speech Processing. Springer, Berlin (2008)
[2] Bennacef, S., Devillers, L., Rosset, S., Lamel, L.: Proceedings of International Conference on Spoken Language Processing, ICSLP 1996, pp. 550–553 (1996)
[3] Chen, S.F., Goodman, J.: Computer Speech and Language 13, 359–393 (1999)
[4] Chen, S.F., Rosenfeld, R.: IEEE Trans. on Speech and Audio Processing 8(1), 37–50 (2000)
[5] Fellbaum, C. (ed.): WordNet. An Electronic Lexical Database. The MIT Press, Cambridge (1998)
[6] Hayes, P.J., Andersen, P.M., Safier, S.: Proceedings of 23rd Annual Meeting of ACL, Chicago, Illinois, pp. 153–160 (1985)
[7] Jelinek, F., Lafferty, J.D., Mercer, R.L.: Basic methods of probabilistic context-free grammars. In: Laface, P., De Mori, R. (eds.) Speech Recognition and Understanding: Recent Advances, Trends, and Applications, pp. 345–360. Springer, Berlin (1992)
[8] Kasprzak, W.: Rozpoznawanie obrazów i sygnałów mowy. Warsaw University of Technology Press, Warszawa (2009)
[9] Katz, S.M.: IEEE Trans. Acoustics, Speech and Signal Proc. ASSP 35, 400–401 (1987)
[10] Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. Prentice Hall, New York (1993)
[11] Russell, S., Norvig, P.: Artificial Intelligence. A Modern Approach. Prentice Hall, New York (2002)
[12] Young, S.: HMMs and related speech recognition technologies. In: Benesty, J., Sondhi, M.M., Huang, Y. (eds.) Springer Handbook of Speech Processing, pp. 539–555. Springer, Berlin (2008)