AN APPROACH TO NATURAL SPEECH UNDERSTANDING BASED ON STOCHASTIC MODELS IN A HIERARCHICAL STRUCTURE

Holger Stahl and Johannes Müller
(names in random order)

ABSTRACT

In this paper, an approach for understanding natural speech by means of two stochastic knowledge bases is presented: Within a given domain, the semantic model generates possible semantic structures, which are semantic representations close to the word level. Corresponding to such a semantic structure, the syntactic model generates word chains using hierarchical Hidden-Markov-Models. Integrated into a speech understanding system, these stochastic knowledge bases can be utilized for a 'top-down'-approach.
Keywords: speech recognition, language understanding, Hidden-Markov-Model, spoken human-machine-dialogue

1 MODEL INTEGRATION INTO A 'TOP-DOWN'-RECOGNITION PROCESS

Speech understanding can be interpreted as mapping a sequence of observation vectors O of an utterance [4] to its semantic content, in this paper represented by the semantic structure S. Given the observation sequence O, the most likely S has to be found. To pursue this goal, the a-posteriori-probability $P(S|O)$ has to be maximized. It can be transformed using the Bayes formula:

$$P(S|O) = \frac{P(S, O)}{P(O)} = \frac{P(O|S) \cdot P(S)}{P(O)} \tag{1}$$

We choose to calculate the probabilities $P(O|S)$ and $P(S)$ by using stochastic methods [2] [5] [6]. Due to the high variety of S and O the conditional probability $P(O|S)$ can not be estimated directly from a set of training data. Therefore, additional representation levels are necessary. Clearly defined is the word level W, which can be used to calculate $P(S|O)$ as follows:

$$P(S|O) = \sum_{W} \frac{P(O|W) \cdot P(W|S) \cdot P(S)}{P(O)} \tag{2}$$

$P(O)$ is irrelevant to the maximization, since it is constant for a given O. Hence, the search problem for the most likely semantic content S_E can be simplified to:

$$S_E = \underset{S}{\operatorname{argmax}} \, [P(O|S) \cdot P(S)] = \underset{S}{\operatorname{argmax}} \sum_{W} [P(O|W) \cdot P(W|S) \cdot P(S)] \tag{3}$$

In the case of searching for the most likely combination of S, W and O, the semantic structure S_E can be determined by the Viterbi decoding algorithm [9] and the sum in eq. (3) is substituted by a maximization:

$$S_E = \underset{S}{\operatorname{argmax}} \, \max_{W} \, [P(O|W) \cdot P(W|S) \cdot P(S)] = \underset{S}{\operatorname{argmax}} \, \max_{W} P(O, W, S) \tag{4}$$

Assuming statistical independence of $P(O|W)$, $P(W|S)$ and $P(S)$, their product represents the joint probability for a certain pattern at each of the representation levels S, W and O. The pattern for the observation sequence O is fixed, since it is given by preprocessing the speech signal of the utterance. Fig. 1 shows the most import modules necessary for top-down decoding the semantic structure S_E of the utterance. Within these modules, additional levels of representation

might be required, for example the phonetic transcription inside the acoustic-phonetic models. The main tasks of the system's hierarchy are listed below:

- *Semantic Model*
  Knowledge base for estimation of $P(S)$, the a-priori probability of the occurrence of the semantic structure S within a particular domain of interest [3].
- *Syntactic Model*
  Knowledge base for estimation of $P(W|S)$, the conditional probability of the occurrence of the word chain *W* given the semantic structure *S* of the utterance.
- *Acoustic-phonetic Models*
  Knowledge base for estimation of $P(O|W)$, the conditional probability of the occurrence of the observation sequence *O* given the word chain *W* [8].

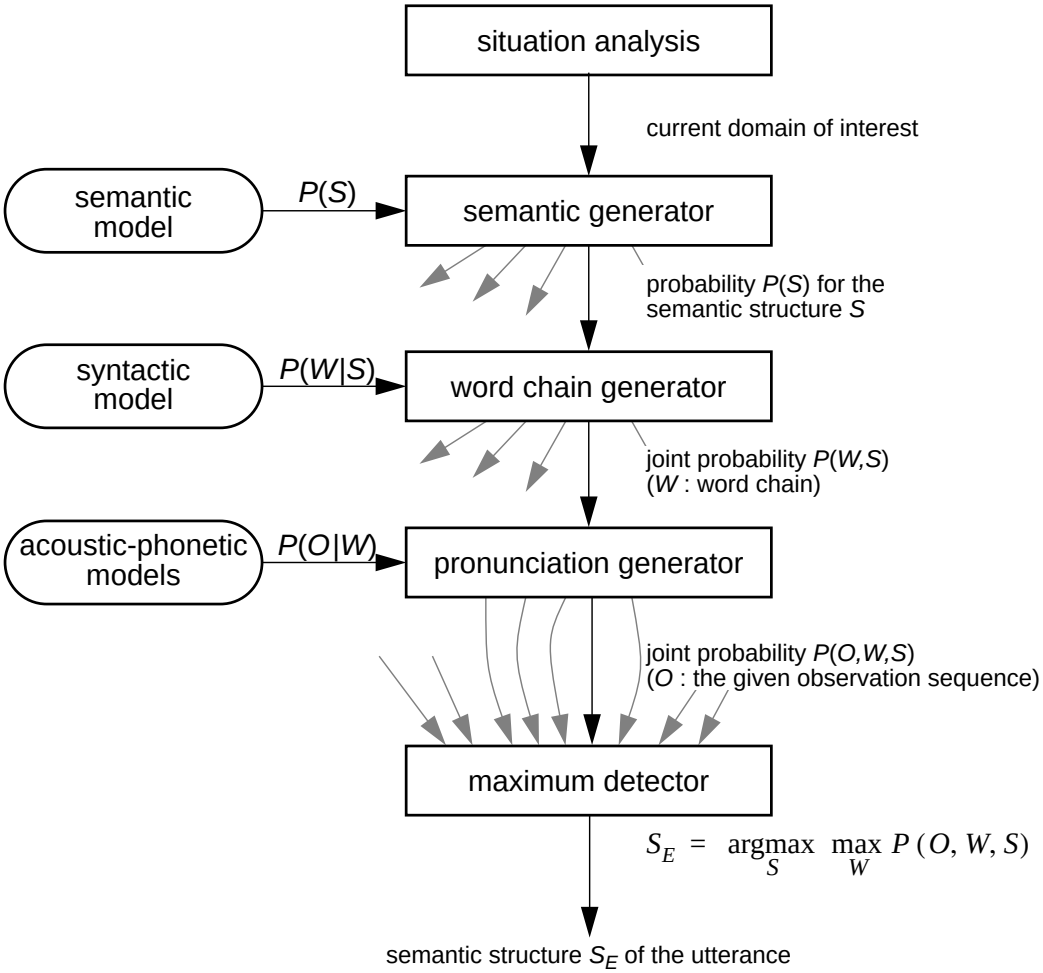


Figure 1: Hierarchy of a system for extracting the semantic structure of an utterance

2 SEMANTIC STRUCTURE

The semantic content of possible utterances is unlimited. Therefore, it is suitable to have the semantic structure S, which is formed of smaller significant units, called semantic units (abbreviated *semuns*), which have limited variety.

2.1 Definition of the semantic structure

The connection of the semuns to a hierarchic tree structure is proposed. A higher level semun should be specified by a lower level semun. As an example, fig. 2 shows the tree of such a semantic structure *S* consisting of $N = 8$ semuns. The nodes are the semuns s_n. The semun in the highest

level is denoted s_1, the others are numerated from 2 to *N* in an optional manner. The edges form the connections between these semuns.

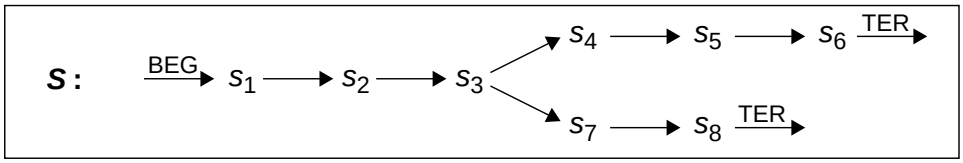


Figure 2: Connection of semuns to the tree of a Semantic Structure *S*

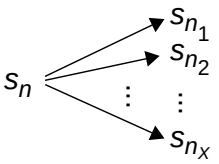
A certain semun has *X* direct successors, connected by the edge '$\longrightarrow$'. '$\xrightarrow{\text{BEG}}$' marks the edge to the semun s_1 in the highest hierarchic level, '$\xrightarrow{\text{TER}}$' a terminal edge. A terminal node does not occur. In the upper example, there is $X = 2$ for s_3 and $X = 1$ for all other semuns (terminal edges count, too).

The connections between the semuns can be explained as follows:

Case $X = 1$:

That part of the semantic structure consisting of the semun s_n and its possible predecessors is described by the semun s_{n_1} and its possible successors.

Case $X = 2$:



The semuns $s_{n_1}, s_{n_2}, \ldots, s_{n_X}$ and their possible successors are related together specified by the semun s_n and describe that part of the semantic structure, which consists of the semun s_n and its possible predecessors.

In the farest sense, a single semun can be seen as an *X*-place predicate logic relation constant [1]. But the connection of single semuns to the semantic structure differs essentially from the representation by predicate logic. However not so exact in the mathematical sense, the semantic structure offers some important advantages:

- The semantic structure is a representation of the semantic content close to the word level. Since every semun s_n in *S* corresponds to one significant word *w* in the word chain *W*, it is possible to design a model for calculating $P(W|S)$ without any more representation levels.
- Connecting the semuns, there is only one mechanism, namely marking other semuns as so-called successors. Designing the models for calculating the probabilities $P(S)$ and $P(W|S)$, only that way to connect the semuns must be taken into account. This restriction is a benefit for designing consistent models.

2.2 Definition of a single semun

A semun contains the semantic information of one significant word. For modelling the semun's meaning separated from the set of possible successors, every semun s_n is represented by its type t_n and its value v_n:

- The *type* $t_n \in \{\tau_1, \ldots, \tau_i, \ldots\}$ lays down the number *X* of successor-types $t_{n_1}, \ldots, t_{n_X}$ and restricts the set of possible successor-types. Furthermore, it makes an efficient selection of the corresponding values.
- The *value* $v_n \in \{\varphi_1, \ldots, \varphi_j, \ldots\}$ shows the proper meaning of the word *w*, which corresponds to the semun s_n. Notice that there could be several words with exactly the same meaning, that the semantic structure S will not change if one of these words are mutated.

For our first investigations, the number *X* of possible successors was limited to two, i.e. for the present, there are semuns only with $X = 1$ or $X = 2$.

2.3 Examples for understanding

For explaining the last two chapters, the semantic structure S is formed for two different utterances, each given as a word chain W in German language. In this example, the utterances are commands to control a simple graphic editor for three-dimensional objects on the screen.

W_1 = 'lösche alle grünen quader'

W_2 = 'zeichne bitte zwei kugeln und mache doch den roten kegel klein'

A set of possible types and values must exist to represent the semantic content of all possible utterances within the current domain of interest as semantic structures. The following table lists some semuns for a graphical editor. The table does not contain all possible semuns, which are necessary for any conceivable utterance. But it elucidates that the semantic content of many utterances can be represented only with a few types and values.

type τ_i	value φ_j	possible successor-types	X	explanation
comm1	create, delete	form, logic	1	action command with one successor
comm2	change	1.: form, logic 2.: colour, size, pos, (terminal)	2	action command with two successors
form	sphere, cylinder, cone, block	quant, colour, size, pos, logic, (terminal)	1	object form
quant	1, 2, 3, ..., many, all	size, pos, colour, (terminal)	1	object quantity
size	big, small, normal	pos, colour, (terminal)	1	object size
pos	middle, up, down, right, left	colour, (terminal)	1	object position
colour	red, green, blue, yellow	(terminal)	1	object colour
logic	and, or	1./2.: form, size, pos, colour, comm1, comm2 (Both the successors must be of the same type!)	2	logical operation

Table 1: Examples for possible types and values for representing utterances to control a graphical editor

With the semuns proposed in tab. 1, the semantic structures S_1 and S_2 explaining the semantic contents of the word chains W_1 and W_2 can be as follows:

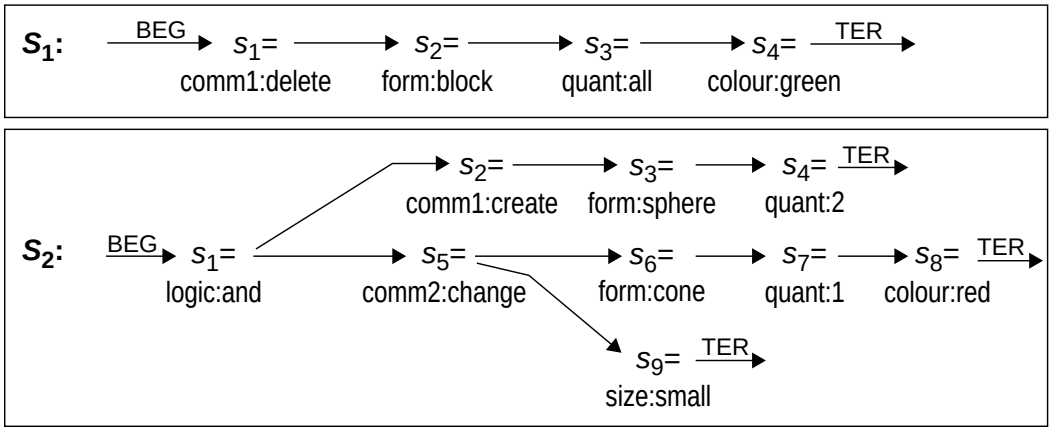


Figure 3: Semantic structures S_1 and S_2 corresponding to the word chains W_1 and W_2. Every semun s_n is represented through a pair of type and value $t_n : v_n$.

For standardisation of the likelihood calculation described in the following chapters, also the successor-type (terminal) has to be included into the set of all existing types $\{\tau_1, \ldots, \tau_i, \ldots\}$. That type indicates an outgoing edge '$\xrightarrow{\text{TER}}$', it does not define a semun, it has no value!

Notice that the insignificant words 'bitte' and 'doch' of the word chain W_2 are not represented in the semantic structure S_2.

3 SEMANTIC MODEL AND SEMANTIC GENERATOR

The semantic generator (see fig. 1) has to generate all possible semantic structures *S* with their a-priori-probabilities $P(S)$ which are imaginable within a certain domain of interest.

3.1 Probabilities of the semantic model

The knowledge base used by the semantic generator is the semantic model, which has to be built of training data (these are many given semantic structures within a certain domain of interest). However, the variety of S is too large to estimate $P(S)$ directly from a training corpus. It is not feasible that all possible semantic structures can be seen in the training. The semantic model must contain a limited number of parameters, which on one hand can be reliably estimated out of the limited training material and on the other hand allow one to calculate the respective a-priori-probability $P(S)$ of an unlimited set of semantic structures S. Hence, the estimation of some first order conditional probabilities is proposed:

- The *begin probability*

$$p_{\text{BEG}}(\tau_i) = P(\text{type} = \tau_i \,|\, \text{in highest hierarchic level}) \tag{5}$$

  indicates the probability that there is a semun of the type τ_i in the highest hierarchic level.

- The *value probability*

$$p_{\tau_i}(\varphi_j) = P(\text{value} = \varphi_j | \text{type} = \tau_i) \tag{6}$$

  indicates the probability that a semun of the type τ_i has the value φ_j.

- The *succession probability*

$$p_{\tau_i}\left(\tau_{i_1}, \tau_{i_2}, \ldots, \tau_{i_X}\right) = P\left(\text{successor-type} = \tau_{i_1}, \tau_{i_2}, \ldots, \tau_{i_X} \,\middle|\, \text{type} = \tau_i\right) \tag{7}$$

  indicates the probability that a semun of the type τ_i has X successor-types $\tau_{i_1}, \ldots, \tau_{i_X}$.

3.2 Calculation of *P(S)*

The estimation of the a-priori-probability $P(S)$ is explained with the following part of a semantic structure.

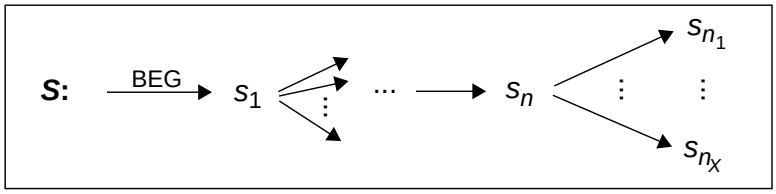


Figure 4: Part of a semantic structure *S*

- The *begin edge* to the semun s_1 has the begin probability

$$f_{\text{BEG}} = p_{\text{BEG}}(t_1). \tag{8}$$

- Every *node*, i.e. every semun s_n, has the value probability

$$e_n = p_{t_n}(v_n) \; . \tag{9}$$

- *All edges* (also terminal ones), which are leaving a certain semun s_n, have the succession probability

$$f_n = p_{t_n}\left(t_{n_1}, \ldots, t_{n_X} \right). \tag{10}$$

Assuming statistical independence of all terms in eq. (8)-(10), the a-priori-probability $P(S)$ is the common product of the begin probability f_{BEG} with all succession and value probabilities of the N semuns within the semantic structure S:

$$\boxed{P(S) = f_{\mathrm{BEG}} \cdot \prod_{n=1}^{N} (e_n \cdot f_n)} \tag{11}$$

Although the assumption of statistical independence is not given in practise, the rather simple modelability justifies the use of eq. (11). $P(S)$ could be specified exactly, if conditional probabilities of n^{th} order with $n \to \infty$ instead of first order are used. Indeed, the number of the parameters which have to be trained grows exponentially with rising order, which, on one hand would aggravate the training, on the other hand would produce a too large hypotheses search space within a top-down semantic recognition.

4 WORD CHAIN GENERATOR AND SYNTACTIC MODEL

Using the syntactic model, the word chain generator has to produce all likely word chains W expressing a certain semantic structure S. The demands made on the word chain generator are listed below:

1. Creation of one significant word w per semun s_n representing its meaning v_n.
2. Insertion of additional, insignificant words.
3. Time alignment of all words in the word chain.

In our approach, a unique syntactic model λ_s exists for every possible semantic structure S, constrained by the condition:

$$P(W|\lambda_S) = P(W|S) \tag{12}$$

The syntactic model represents a Hidden-Markov-Model [7], implying the superimposition of two stochastic processes:

1. Change of states according to a set of transition probabilities.
2. Emission of words from selected states according to emission probabilities.

The probability of a certain state sequence Q, given the model λ_S, is the product of all transition probabilities along Q:

$$P(Q|\lambda_S) = \prod_{\substack{\text{all state transitions} \\ \text{along } Q}} \left(\text{concerned transition probability} \right) \tag{13}$$

The probability of emitting the word chain W along the state sequence Q is given by

$$P(W|Q,\lambda_S) = \prod_{\substack{\text{all states with}\\ \text{word emissions}\\ \text{along } Q}} \Big(\text{concerned emission probability}\Big). \tag{14}$$

If we assume statistical independence of these two stochastic processes, the joint probability for W and Q is simply the product of the above two terms:

$$P(W,Q|\lambda_S) = P(W|Q,\lambda_S) \cdot P(Q|\lambda_S) \tag{15}$$

The probability of W (given the model), which is to be calculated, is obtained by summing this probability over all possible state sequences Q:

$$P(W|\lambda_S) = \sum_{\text{all } Q} P(W,Q|\lambda_S) \tag{16}$$

An infinite set of likely semantic structures is in accordance with an infinite set of syntactic models as well. In contrast, the model parameters have to be extracted from a limited amount of training data (these are word chains describing many semantic structures). To avoid problems caused by the lack of training data, the syntactic model is constructed from a limited set of smaller units, in the following called elementary Hidden-Markov-Models (EHMMs). Starting from the semantic structure as an adequate representation of the semantic content, it seems reasonable to include exactly one EHMM for every semun s_n. The EHMMs have to be linked with each other, associated to the edges in S.

4.1 Parameters of an elementary Hidden-Markov-Model

As shown in fig. 5, the EHMMs contain three or four states $Z_1 \ldots Z_4$, depending on the number of successors X (for our first investigations we limit $X \in \{1, 2\}$). The EHMM appears to be a first order Hidden-Markov-Model, but it contains some substantial distinctions:

- The state Z_1 emits one insignificant word.
- The state Z_2 emits one significant word, which represents the semantic content of the associated semun. This state has to be passed accurately one time. More detailed explanations concerning the word emissions can be found in chap. 2.3.
- The states Z_3 and Z_4 symbolize the EHMMs associated with the successor semuns (one or two). These sub-models have to be entered and left accurately one time.

An EHMM is described by the following parameters:

- *Transition probabilities:*
  a_{ij} is the probability of a transition from state Z_i to Z_j:

  $$a_{ij} = P(\text{next state} = Z_j \,|\, \text{current state} = Z_i) \tag{17}$$

  Transitions to the states Z_3 and Z_4 represent transitions to the start point of the concerned sub-model. Transitions from the states Z_3 and Z_4 represent transitions from the end point of the concerned sub-model.
- *Emission probabilities:*
  $b_i(w)$ is the probability of emitting the word w in state Z_1 or Z_2:

  $$b_i(w) = P(\text{emitted word} = w \,|\, \text{current state} = Z_i), \quad i \in \{1, 2\} \tag{18}$$

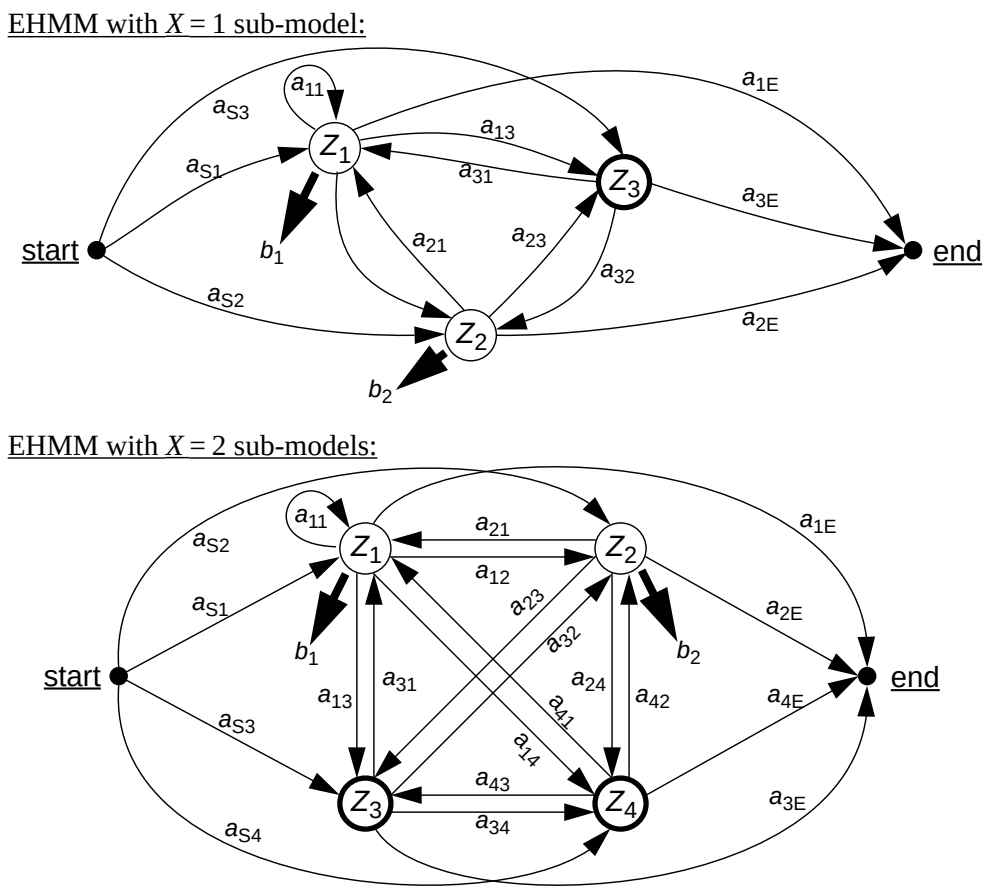


Figure 5: Elementary Hidden-Markov-Models with three or four states

To reduce the number of model parameters to be estimated, we assume that all parameters except the emission probability $b_2(w)$ only depend on the type of the associated semun. Hence, the number of required model parameter sets is equal to the number of semun types τ_i defined for the semantic structure.

4.2 Linking together various EHMMs

The construction of the syntactic model λ_S as a network of linked EHMMs follows the definition of the semantic structure *S*, which consists of single semuns s_n. Hence, λ_S incorporates accurately one EHMM$_n$ for every node (i.e. every semun) in the tree of S. Every edge in the tree of S causes a link of the associated EHMMs. The successors s_{n_1} and s_{n_2} of the semun s_n are integrated into the syntactic model as sub-models EHMM$_{n_1}$ und EHMM$_{n_2}$. The states Z_3 and Z_4 serve as dummies for these sub-models.

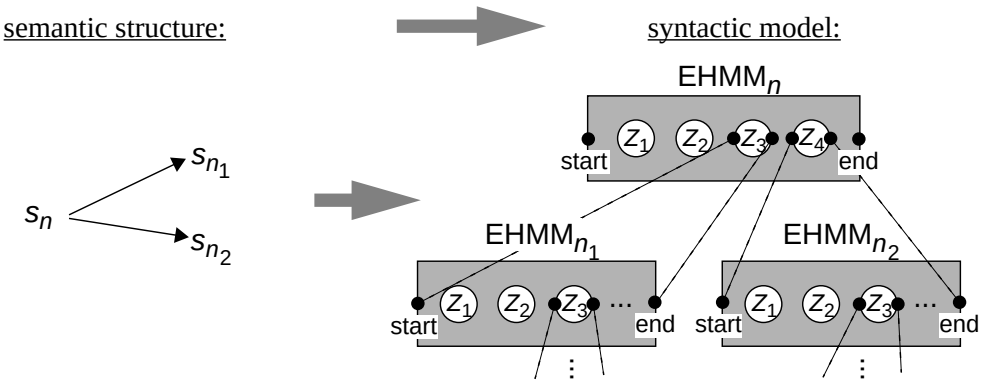


Figure 6: Layout of the syntactic model associated with a given semantic structure

Fig. 6 shows a semun s_n with $X = 2$ successors s_{n_1} and s_{n_2} as a detail of a complete semantic structure. On the right side, the associated part of the syntactic model is illustrated, which consists of the EHMM$_n$ with its sub-models EHMM$_{n_1}$ and EHMM$_{n_2}$.

4.3 Word emissions

- State Z_1 emits exclusively <u>insignificant</u> words. The emission probability $b_1(w)$ only depends on the type t_n of the associated semun s_n.
- State Z_2 emits exclusively <u>significant</u> words. The emission probability $b_1(w)$ depends both on the type t_n and on the value v_n of the associated semun s_n.

Fig. 7 gives an example for possible emissions from the states Z_1 and Z_2, given the type t_n = comm1 (taken from tab. 1):

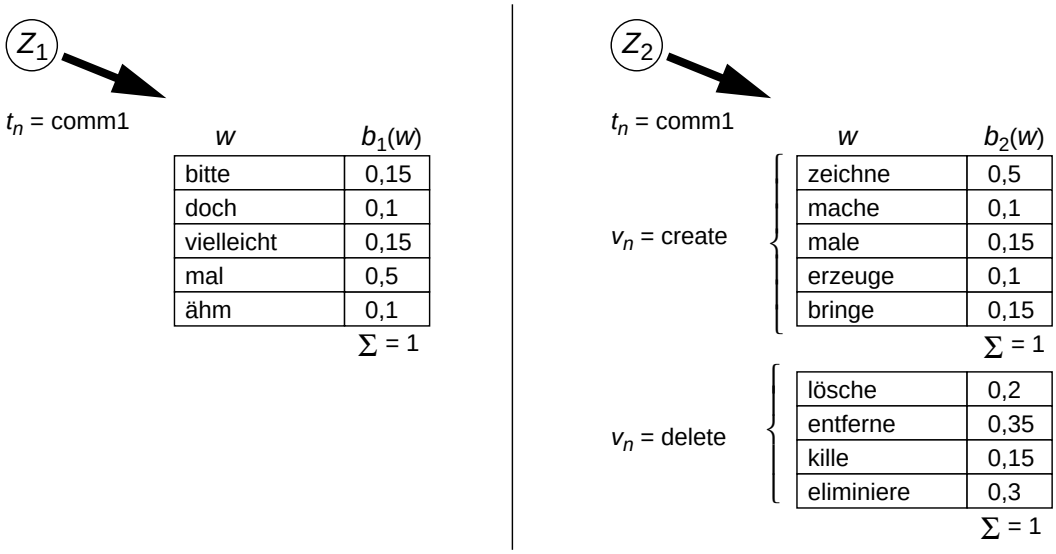


Figure 7: Emissions of significant and insignificant words with associated $b_i(w)$

5 REFERENCES

[1] G. Görz: *Einführung in die künstliche Intelligenz*, Addison-Wesley, 1993

[2] H. Höge: *Statistische Modelle für die Spracherkennung*, Proc. DAGA 1993 (Frankfurt a. M., Germany), pp. 11-30

[3] M. Lang: *Mensch-Maschine-Kommunikation 1*, lecture notes, Technichal University Munich, 1992

[4] M. Lang, H. Stahl: *Spracherkennung für einen ergonomischen Mensch-Maschine-Dialog*, mikroelektronik, vol. 8 (1994), no. 2, pp. 79-82

[5] R. Pieraccini et al.: *A Speech Understanding System Based on Statistical Representation of Semantics*, Proc. ICASSP 1992 (San Francisco, USA), pp. I 193 - I 196

[6] R. Pieraccini, E. Levin, E. Vidal: *Learning how to understand Language*, Proc. Eurospeech 1993 (Berlin, Germany), pp. 1407-1412

[7] L. R. Rabiner: *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proc. IEEE, vol. 77 (1989), no. 2, pp. 257-286

[8] H. Roll, H. Stahl, J. Müller: *Training von Hidden-Markov-Modellen zur Erkennung fließender Sprache*, Internal report, Technical University Munich, 1994

[9] A.J. Viterbi: *Error bounds for convolutional codes and an asymptotical optimal decoding algorithm*, IEEE Trans. Information Theory, vol. 61 (1973), pp. 268-278