FROM J. HAAS ET AL., *SPECOM'98, INTERNATIONAL WORKSHOP SPEECH AND COMPUTER*, ST. PETERSBURG (RUSSIA), 1998, (PP. 151–156).

PROBABILISTIC SEMANTIC ANALYSIS IN RESTRICTED DOMAINS

J. Haas, J. Hornegger, H. Niemann
Universität Erlangen-Nürnberg,
Lehrstuhl für Mustererkennung (Informatik 5)
Martensstraße 3, D-91058 Erlangen, Germany
email: {haas,hornegger,niemann}@informatik.uni-erlangen.de
Tel.: +49/9131/8527873 Fax: +49/9131/303811

ABSTRACT

We present a probabilistic approach to semantic and pragmatic analysis in restricted domains and methods to improve the understanding performance. As framework we use the EVAR system, an automatic dialog system for answering queries on German Intercity train connections over the public telephone network. We introduce the statistical model extracting the semantic content from a word chain as well as our annotation scheme for the semantic of word chains for a specialized task which allows fast and easy annotation. The task of detecting the semantic contents of a word chain is described as the problem of assigning semantic attributes to words. The statistical framework we use has to deal with incomplete data estimation problems which are solved through applying the Expectation Maximization algorithm. The resulting iterative estimation formulas for the desired parameters are presented. The baseline experiment and the obtained recognition results prove the feasibility of our model. We present several methods that are able to support our probabilistic semantic analysis. We present experiments using categorial systems on the lexicon, different initializing methods for probabilistic parameters and combinations of these methods. The results show that the supporting power of the techniques is small when they are used individually but remarkable for the combination of them.

1 INTRODUCTION

For the tasks of semantic analysis, speech understanding and automatic dialog systems it is not that important to recognize the user's utterance with 100% word accuracy but to get the user's intention in order to generate an appropriate system reaction which leads to a successful dialog. In our application of information retrieval dialogs possible reactions are e.g. to start a clarification sub dialog or to give the desired information. Nowadays the syntactic-semantic analysis is mainly realized using rule based systems. These parsers use a bunch of lexical and grammatical knowledge. Hence, the system adaptation to another domain or task requires the encoding of new lexical and grammatical knowledge nearly from scratch. This is a time consuming and tricky work which has to be done by a linguistic expert. Having a probabilistic model for the semantic analysis in a restricted domain which is able to learn the new task from a suitable training set a faster and easier adaptation is possible.

The paper is organized as follows. In section 2 we present the probabilistic model for the semantic analysis and the mathematical details. We present the estimation formulas we obtain for the parameter estimation. The semantic annotation scheme allowing a fast and easy creation of training material is introduced in section 3. For sure this annotation scheme is strongly task dependent. There we also describe the data we use and the first baseline experiment with its results. We examined several methods well known in the speech processing society which should be able to support our probabilistic model. The results which prove the feasibility of the methods individually and in combination are presented in section 4. In section 5 we give a short summary and an outlook on planned activities.

2 PROBABILISTIC MODEL FOR SEMANTIC ANALYSIS

Our probabilistic model is based on the assumption that for each word w_j of a word chain $\boldsymbol{w}$ we have an assignment of one semantic attribute C_i to w_j. These semantic attributes describe parameters which influence the system's reaction. The set $\mathcal{C} = \{C_1, C_2, \ldots C_N\}$ of possible attributes is therefore strongly task dependent. In our case of train timetable information dialogs these attributes are important information slots for the database query like e.g. the source city, the destination or the time of departure, but also markers important for the dialog like e.g. positive or negative feedback which influences the dialog strategy. Obviously not all parts of the word chain are important for its interpretation in the restricted domain or for the dialog continuation. To keep our assignment assumption we have to introduce an additional semantic attribute C_0, the NIL attribute. All words carrying no meaning for the dialog system are attached to C_0. An example illustrating this assumption is given in figure 1.

Each semantic attribute C_i is now identified with a random process that generates words w_j. Thus, each such process defines a density function $P(w_j|C_i)$ over the lexicon of words. The assignment of a semantic attribute C_i to a word w_j is formally described with the assignment function ζ. This function takes as argument the word w_j of the word chain $\boldsymbol{w}$ and gives as result the index of the corresponding attribute $\zeta(w_j) = l_j$. As we want to analyze word chains $\boldsymbol{w}$ we have to extend the assignment function to operate on these:

$$\boldsymbol{\zeta}(\boldsymbol{w}) = (\zeta(w_1), \ldots, \zeta(w_n))^T = (l_1, \ldots, l_n)^T \quad (1)$$

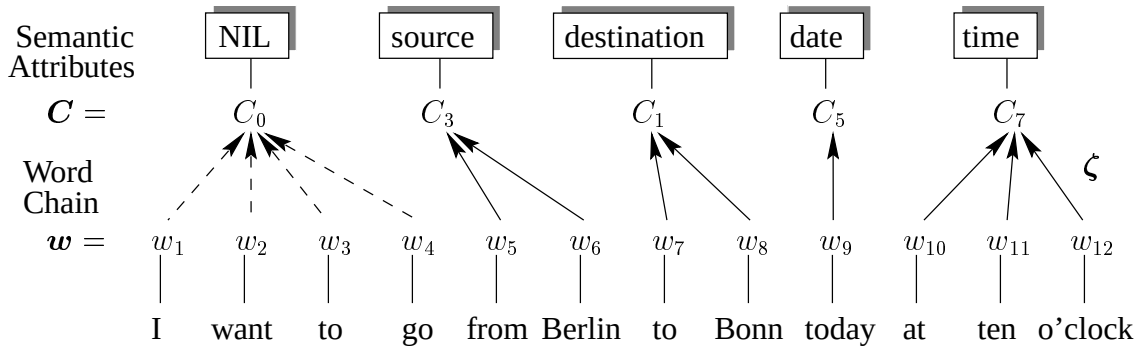

Figure 1. Assignment of semantic attributes C_k to words w_j of a word chain for train timetable information

The semantic of a word chain $\boldsymbol{w}$ in the current task is now described through the set of semantic attributes $\boldsymbol{C}$ that are assigned to the words of $\boldsymbol{w}$. All words corresponding to one attribute from $\boldsymbol{C}$ constitute one semantic segment in the word chain which in a following step can be further analyzed. In comparison to [3] who also defines an assignment from the english to a formal language for the interpretation of word chains, our approach does not make use of the length of semantic segments or the number of assigned attributes for the computation of probabilities. This difference gives us the possibility to use a much easier annotation scheme which will be described in section 4.

The semantic analysis of word chains is now reduced to the problem of finding the best set of semantic attributes that are assigned to the words of $\boldsymbol{w}$. To provide a formalism which enables us to handle the speech understanding task statistically we introduce a model based on two tied probabilistic processes.

1. The first one has to model the appearance of semantic attributes and introduces a probability distribution on the set of possible assignments $\boldsymbol{\zeta}(\boldsymbol{w})$.
2. The second one is the semantic attribute generating words with the density function $P(w_j|C_i)$.

With these two processes the probability $P(\boldsymbol{w}, \boldsymbol{\zeta}|\boldsymbol{C})$ for observing $\boldsymbol{w}$ and the assignment $\boldsymbol{\zeta}$ assuming the semantic described through $\boldsymbol{C}$ is computed. The assignment vector can not be observed as we have only the set of semantic attributes for a word chain but not the explicit alignment. Thus, the probability $P(\boldsymbol{w}|\boldsymbol{C})$ derives from the marginalization of $P(\boldsymbol{w}, \boldsymbol{\zeta}|\boldsymbol{C})$ over all possible assignments, i.e

$$\begin{aligned} P(\boldsymbol{w} \mid \boldsymbol{C}) &= \sum_{\boldsymbol{\zeta}} P(\boldsymbol{w}, \boldsymbol{\zeta}|\boldsymbol{C}) \\ &= \sum_{\boldsymbol{\zeta}} P(\boldsymbol{\zeta}) P(\boldsymbol{w} \mid \boldsymbol{C}, \boldsymbol{\zeta}) \,. \end{aligned} \tag{2}$$

We now claim that the assignments depend only on preceding ones and they are of statistical order g. Then the probability $P(\boldsymbol{\zeta})$ for a special assignment is factorized in conditional probabilities $p(l_k \mid l_{k-g} \dots l_{k-1})$. Renaming these probabilities with $p(l_k \mid l_{k-g} \dots l_{k-1}) = a_{l_{k-g} \dots l_k}$ we get the following:

$$\begin{aligned} P(\boldsymbol{\zeta}) &= p(l_1, l_2, \dots, l_n) \\ &= p(l_1) \cdot p(l_2 \mid l_1) \cdot p(l_3 \mid l_1 l_2) \cdot \dots \\ &\qquad \dots \cdot p(l_n \mid l_1 l_2 \dots l_{n-1}) \\ &= p(l_1) \cdot p(l_2 \mid l_1) \cdot \dots \cdot p(l_g \mid l_1 l_2 \dots l_{g-1}) \cdot \\ &\qquad \prod_{k=g+1}^{n} p(l_k \mid l_{k-g} \dots l_{k-1}) \\ &= a_{l_1} \cdot a_{l_1 l_2} \cdot \dots \cdot a_{l_1 l_2 \dots l_g} \prod_{k=g+1}^{n} a_{l_{k-g} \dots l_k} \end{aligned} \tag{3}$$

The probability $P(\boldsymbol{w} \mid \boldsymbol{C}, \boldsymbol{\zeta})$ is reduced to probabilities for the observation of single words w_j depending only on the actual assignment to the semantic attribute C_{l_j}. With these assumptions the probabilistic semantic analysis problem is reformulated:

$$\begin{aligned} P(\boldsymbol{w} \mid \boldsymbol{C}) &= \sum_{\boldsymbol{\zeta}} p(\boldsymbol{\zeta}) \prod_{j=1}^{n} P(w_j \mid C_{\zeta(w_j)}) \\ &= \sum_{l_1, l_2, \dots, l_n} a_{l_1} \cdot a_{l_1 l_2} \cdot \dots \cdot a_{l_1 l_2 \dots l_g} \\ &\qquad \prod_{k=g+1}^{n} a_{l_{k-g} \dots l_k} \cdot \prod_{j=1}^{n} P(w_j \mid C_{l_j}) \end{aligned} \tag{4}$$

The parameters to be estimated for the stochastic model are the conditional probabilities for the elements of the assignment vector $a_{l_{k-g}, l_{k-g+1}, \dots, l_k}$ and the probabilities $P(w_j|C_{\zeta(w_j)})$ for observing word w_j under the assumption of assigning it to the attribute $C_{\zeta(w_j)}$. As the assignment $\boldsymbol{\zeta}$ is hidden in our training data, we use the Expectation Maximization algorithm (cf. [1]). As result we obtain iterative estimation formulas for the necessary parameters. A detailed derivation for the estimation formulas can be found in [5].

Assume, M test sentences ${}^1\boldsymbol{w}, {}^2\boldsymbol{w}, \dots, {}^M\boldsymbol{w}$ are available for training purposes. The probabilities associated with the assignment function can be iteratively estimated using the following formulas, wherein i and $i+1$ denote the i–th and $(i+1)$–st iteration steps:

$$a_{l_1}^{(i+1)} = \frac{1}{M}\sum_{\varrho=1}^{M}\frac{\sum\limits_{\zeta(\varrho w_1)=l_1} p^{(i)}(\varrho \boldsymbol{w},\boldsymbol{\zeta}|\boldsymbol{C})}{p^{(i)}(\varrho \boldsymbol{w}|\boldsymbol{C})} \quad , (5)$$

$$a_{l_1 l_2}^{(i+1)} = \frac{\sum\limits_{\varrho=1}^{M}\sum\limits_{\zeta(\varrho w_1)=l_1,\zeta(\varrho w_2)=l_2}\frac{p^{(i)}(\varrho \boldsymbol{w},\boldsymbol{\zeta}|\boldsymbol{C})}{p^{(i)}(\varrho \boldsymbol{w}|\boldsymbol{C})}}{\sum\limits_{\varrho=1}^{M}\sum\limits_{\zeta(\varrho w_1)=l_1}\frac{p^{(i)}(\varrho \boldsymbol{O},\boldsymbol{\zeta}|\boldsymbol{C})}{p^{(i)}(\varrho \boldsymbol{w}|\boldsymbol{C})}} \quad (6)$$

$$a_{l_1,l_2,\ldots,l_g}^{(i+1)} = \frac{\sum\limits_{\varrho=1}^{M}\sum\limits_{\zeta(\varrho w_1)=l_1,\zeta(\varrho w_2)=l_2,\ldots,\zeta(\varrho w_g)=l_g}\frac{p^{(i)}(\varrho \boldsymbol{w},\boldsymbol{\zeta}|\boldsymbol{C})}{p^{(i)}(\varrho \boldsymbol{w}|\boldsymbol{C})}}{\sum\limits_{\varrho=1}^{M}\sum\limits_{\zeta(\varrho w_1)=l_1,\zeta(\varrho w_2)=l_2,\ldots,\zeta(\varrho w_{g-1})=l_{g-1}}\frac{p^{(i)}(\varrho \boldsymbol{w},\boldsymbol{\zeta}|\boldsymbol{C})}{p^{(i)}(\varrho \boldsymbol{w}|\boldsymbol{C})}} \quad , \quad (7)$$

$$a_{l_{k-g},\ldots,l_k}^{(i+1)} = \frac{\sum\limits_{\varrho=1}^{M}\sum\limits_{k=g+1}^{\varrho n}\sum\limits_{\zeta(\varrho w_{k-g})=l_{k-g},\ldots,\zeta(\varrho w_k)=l_k}\frac{p^{(i)}(\varrho \boldsymbol{w},\boldsymbol{\zeta}|\boldsymbol{C})}{p^{(i)}(\varrho \boldsymbol{w}|\boldsymbol{C})}}{\sum\limits_{\varrho=1}^{M}\sum\limits_{k=g+1}^{\varrho n}\sum\limits_{\zeta(\varrho w_{k-g})=l_{k-g},\ldots,\zeta(\varrho w_{k-1})=l_{k-1}}\frac{p^{(i)}(\varrho \boldsymbol{w},\boldsymbol{\zeta}|\boldsymbol{C})}{p^{(i)}(\varrho \boldsymbol{w}|\boldsymbol{C})}} \quad . \quad (8)$$

Obviously, these equations are generalizations of the well–known Baum–Welch reestimation formulas for Hidden Markov Models. They can be used for statistical dependencies of arbitrary order g (for $g = 1$ we get those BW-Reestimation formulas for HMM). The same holds for the discrete probabilities which characterize word production, if the semantic attribute C_{l_j} is known ($1 \leq j \leq n$ and $0 \leq l_j \leq N$):

$$p^{(i+1)}(w_j|C_{l_j}) = \frac{\sum\limits_{\varrho=1}^{M}\sum\limits_{k=1}^{\varrho n}\sum\limits_{\zeta(\varrho w_k)=l,\varrho w_k=w_j}\frac{p^{(i)}(\varrho \boldsymbol{w},\boldsymbol{\zeta}|\boldsymbol{C})}{p^{(i)}(\varrho \boldsymbol{w}|\boldsymbol{C})}}{\sum\limits_{\varrho=1}^{M}\sum\limits_{k=1}^{\varrho n}\sum\limits_{\zeta(\varrho w_k)=l}\frac{p^{(i)}(\varrho \boldsymbol{w},\boldsymbol{\zeta}|\boldsymbol{C})}{p^{(i)}(\varrho \boldsymbol{w}|\boldsymbol{C})}} \quad . \quad (9)$$

Within this framework, probabilities are computed with a generalized forward-backward algorithm which is very similar to the HMM forward-backward as we have special beginning probability matrices for the first g words. The most likely state sequence, which gives us the semantic segmentation for a word chain $\boldsymbol{w}$ and along with that the set of semantic attributes $\boldsymbol{C}$, results from a generalized Viterbi algorithm.

3 ANNOTATION SCHEME AND BASELINE EXPERIMENT

The data we use for our experiments are collected with the automatic dialog system EVAR. This system answers questions on train timetables for German Intercity connections and is hooked to a public telephone line (Tel. ++49/9131/16287). The EVAR system and the data are described in [2]. In this section we first describe the annotation scheme we use which is a quick and easy method to create the necessary training material. In the second part we report about the baseline experiment and the obtained recognition results.

3.1. Annotation Scheme

All dialogs performed by EVAR are recorded and information like the recognized word chain or the text of the synthesized answer is stored. The speech signals are then transcribed off-line, i.e. it is annotated what really was spoken. The semantic we annotate is derived from this spoken word chain. We defined a set of about 30 different semantic attributes to be distinguished in the users' utterances. We consider only those parameters as semantic attributes which influence the system's reaction. For example time and date expressions are important for the system to give the correct information. Also positive or negative feedback are regarded as semantic attributes because of their influence on the ongoing dialog. In contrast the caller's name or the reason why someone wants to travel by train is obviously unimportant and therefore neglected in the semantic annotation. The resulting semantic is independent of the dialog context. The same word sequence $\boldsymbol{w}$ has always the same semantic annotation regardless of the yet collected information or the actual system dialog step. For example if the system asks `'Where do you want to leave?'` often an answer with a city name e.g. `'Hamburg'` is observed. Then only `'city:hamburg'` is annotated as being the semantic content. Taking the dialog context into account it is obvious that the named city is the departure city (`'sourcecity:hamburg'`). Surely this information is used inside our dialog system but this interpretation step is carried out through the dialog manager who keeps the dialog context and does some context dependent interpretation. Here it is not the task of the semantic analysis to do this kind of interpretation. If we want to include the context in our probabilistic model we can do this e.g. by adjusting the a priori probabilities for different semantic attributes depending on the dialog state and the actual system dialog step. The semantic annotation for a spoken word chain $\boldsymbol{w}$ is the semantic described as the set of semantic attributes $\boldsymbol{C}$. As we do not need the explicit alignment of semantic attributes to words this annotation can be done without linguistic knowledge. The NIL attribute corresponds to an empty handmade annotation. This approach of describing the semantic is similar to the one described in [6] with semantic concepts. The main difference here again is the fact that for our models the annotation is much easier as in [6] every word has to be labeled with its corresponding attribute whereas we only need the overall annotation for the whole sentence and not an explicit alignment. For our experiments the 30 different semantic attributes we an-

notate are clustered together in the following twelve classes which then should be modeled by our probabilistic model:

(1)	**SOURCECITY**	departure city
	e.g.	*"from Berlin"*
(2)	**GOALCITY**	arrival city
	e.g.	*"to Bonn"*
(3)	**CITY**	cities not in 1 or 2
	e.g.	*"via Munich"*
(4)	**MARKER**	parts with influence on dialog
	e.g.	*"that's right"*
(5)	**RELDAY**	dates in relative manner
	e.g.	*"tomorrow"*
(6)	**SPECIAL**	legal holidays
	e.g.	*"on Christmas"*
(7)	**DATE**	dates not in 5 or 6
	e.g.	*"on November the 5th"*
(8)	**TRAINTYPE**	type of train
	e.g.	*"with the ICE"*
(9)	**POFDAY**	time as part of day
	e.g.	*"in the morning"*
(10)	**RELTIME**	time in relative manner
	e.g.	*"later"*
(11)	**TIME**	time not in 9 or 10
	e.g.	*"at 5 o'clock"*
(12)	**NIL**	parts semantically irrelevant
	e.g.	*"I need help"*

3.2. Baseline Experiment

For our Baseline Experiment we choose a training set of 10207 sentences from the EVAR corpus along with their semantic annotation. Due to the dialog strategy some of the sentences are very frequent. For example, the requested confirmation of parameters for the database query are very often answered with a `'yes'` or `'no'`. Therefore there are only 3977 different sentences in the training data. As test set we have another 4952 sentences collected with EVAR where 2539 are different, due to the same reasons. Some sentences in the test set are also included in the training set as we always take complete dialogs into the data sets but still 2421 sentences are unseen. In the training and test set there are 1682 different words which serve as observables for the probabilistic model, i.e. for each attribute we have a discrete probability distribution over these words.

Each attribute to be recognized is represented by one state in our probabilistic model. All states are connected with each other. If we set the statistical dependency to $g = 1$ we have an ergodic HMM. The output probabilities are discrete distributions over the words in the lexicon like said above. As the EM algorithm has only local optimality and a slow convergence we have to choose the initialization of the probabilities for our introduced method carefully. The transition probabilities representing the conditional probabilities for the assignment function are uniformly initialized for the baseline experiment. The output probabilities are simply relative frequencies of words' occurrences. We take all sentences from the training set where the modeled semantic attribute is observed and count the words therein. Afterwards we normalize the counts over all words for one attribute and use this as output probability distribution. We start to counting at 1 to avoid a probability of value 0.

Using this modeling technique and the described initialization methods we obtain the recognition results shown in Table 1. The percentages give the recognition accuracy, i.e. we count all sentences as wrong where either an attribute is deleted or inserted. Therefore the number of deletions plus the number of insertions is higher than the number of wrong sentences. If we have both, a deletion and an insertion in one word chain, we only have one erroneous sentence. What we do not count are insertions of the **NIL** attribute because in our annotation we only have **NIL** for meaningless sentences but the attribute can also be present in sentences which carry some meaning and then in the annotation **NIL** is omitted. Therefore we don't have the possibility to decide automatically whether there are meaningless parts in the utterance or not. For that task we would have to evaluate the semantic segmentation performance of the model, which is not done yet. Within this evaluation scheme the question arises if an insertion error should be judged as expensive as a deletion because in the next step after recognizing the semantic attributes we have to extract the corresponding values (e.g. the real spoken city name for the database query or the required time point). In this step an insertion error can be discovered and repaired but a deletion error not. At the moment we are only counting deletions and insertions to scale our model's performance.

	HMM	G2HMM
Accurate sentences	2843	2857
Wrong sentences	2109	2095
Accurate Detections	57 %	58 %
	Insertions	
Sentences	2068	2068
	Deletions	
Sentences	663	671

Table 1. Accuracy and number of errors of the baseline experiment with statistical dependency of order $g = 1$ (HMM) and order $g = 2$ (G2HMM) with twelve semantic attributes

The results show that our model is capable of learning the assignment of words to attributes and therefore to semantically analyze word chains. The main type of error we have are insertions and if we take a detailed look on the insertion errors we see that very often the semantic attribute **CITY** is inserted within a syntactical construct giving the destination or the starting point where the preposition (e.g. *"to ..."* or *"from ..."*) is assigned correctly but the city name is assigned to **CITY** and not to **GOAL** or respectively **SOURCE**. These errors are due to the initialization we choose for the first experiment. A detailed analysis of the deletions shows up that most often the attribute **NIL** is affected which is also comprehensible as we have as examples for the **NIL** attribute only those sentences which are completely meaningless for the task.

4 SUPPORTING METHODS AND RESULTS

The baseline experiment proves the capability of our approach to perform the task of semantic analysis in restricted domains. In this section we introduce and examine some methods which support our statistical model in semantic analysis. Employing a method on its own gives only small improvement in recognition rate. The impressive supporting power of these methods turns out when they are applied in combination.

4.1. Categorial System

The first method we use is well-known in language modeling and estimating word occurrence probabilities from small data sets. We use a system of categories which partitions the lexicon, i.e. each word belongs to one and only one category. These categories are built considering syntactic and semantic constraints. In our experiments we use two different categorial systems, the first one with 220 categories which serve as observables for the statistical model. For the second system we use only 13 different main categories taken from the first categorial system like `'CITYNAME'` or `'WEEKDAY'`. The words that do not belong to one of these categories build their own category. Altogether we have 606 words comprised in the 13 categories and additionally 1076 words respectively categories which results in an output alphabet consisting of 1089 observables. We still initialize the transition probabilities uniformly and the output probabilities as relative frequencies where we start the counting at 1, but we reduce the size of the lexicon. For the experiments with category systems the results are shown in the Tables 2 and 3.

	HMM	G2HMM
Accurate sentences	3111	3122
Wrong sentences	1841	1830
Accurate Detections	63 %	63 %
	Insertions	
Sentences	1814	1806
	Deletions	
Sentences	338	350

Table 2. Accuracy and Error-Rates for the experiments using a categorial system on the lexicon with 220 classes.

	HMM	G2HMM
Accurate sentences	3072	3093
Wrong sentences	1880	1859
Accurate Detections	62 %	62 %
	Insertions	
Sentences	1845	1836
	Deletions	
Sentences	350	323

Table 3. Accuracy and Error-Rates for the experiments using a categorial system on the lexicon with 1089 classes.

4.2. Initial Output Probabilities

In the next experiment we change the initialization method for the output probabilities slightly. As described in section 3 we use relative frequencies of words by counting the words that are observed besides a special attribute and we start to count at 1. In this approach we start to count at 0 because if a word w_j is never seen in combination with the attribute C_i we want to avoid explicitly the assignment of this attribute in the analysis phase. Giving the word w_j the output probability $P(w_j \mid C_i) = 0$ does this. The recognition results of this experiment using again the complete word lexicon of 1682 words without categories are shown in Table 4.

	HMM	G2HMM
Accurate sentences	2918	2913
Wrong sentences	2034	2039
Accurate Detections	59 %	59 %
	Insertions	
Sentences	1973	1956
	Deletions	
Sentences	778	825

Table 4. Accuracy and number of errors of the experiments starting the count for the relative frequencies for the initial output probabilities at 0.

4.3. Initial Transition Probabilities

When we take a brief look at the results obtained in the experiments described in sections 4.1. and 4.2. we see that a lot of insertion errors occur. This is partially due to the fact that our transition probabilities are uniformly initialized and the decision for the best assignment is only guided by the output probabilities. That fact also explains the problem described in section 3.2. with the insertion of the **CITY** attribute when city names are seen. If the probability for staying in the **GOAL** and **SOURCE** state is tuned to a higher value, the model wouldn't jump that much around and hopefully also assigns the city name to the **GOAL**/**SOURCE** segment. As the EVAR database proves that attributes have a minimum length in number of words that is needed to express the attribute we should use this information for our model. For example, it is impossible to express a departure city by only one word which then has to be the name of the city. There must be at least one additional word (e.g. `'from'`) indicating the special semantic meaning. Therefore, attributes have characteristic length distributions counted in number of words. We could introduce some simple length modeling in our statistical model by initializing transition probabilities not uniformly but using these distributions. The results of this initialization scheme are reported in Table 5. Obviously the improvement is due to the fact that the number of insertions is decreasing which was the desired effect.

Another modeling oppertunity which we have not examined yet is to scale dependencies between attribute by adjusting the transition probabilities. In the data we use there are e.g. attribute combinations that never occur as well as

	HMM	G2HMM
Accurate sentences	3164	3124
Wrong sentences	1788	1828
Accurate Detections	64 %	63 %
	Insertions	
Sentences	1299	1751
	Deletions	
Sentences	1347	910

Table 5. Accuracy and number of errors of the experiments with initial transition probabilities used for a crude length modeling of attributes.

some combinations that are seen very often, like the sequence of first giving the departure city followed by the destination. For our model we can use such information by decreasing the transition probability for seldom seen combinations and increasing the often seen.

4.4. Combinations

As we have seen in the previous sections the supporting methods are each on its own capable to improve the recognition accuracy of the semantic attribute detection task. We now want to combine those methods to see whether they work even better together than individually. The experiments we made showed the best accuracy rate for the following combination.

- start counting for relative frequencies at 0,
- use the categorial system with 220 classes,
- use initial transition probabilities for length modeling.

The results we obtain with these supporting methods are reported in Table 6.

	HMM	G2HMM
Accurate sentences	3652	3455
Wrong sentences	1300	1497
Accurate Detections	74 %	70 %
	Insertions	
Sentences	812	1374
	Deletions	
Sentences	890	835

Table 6. Accuracy and number of error of the experiments with the best recognition results.

5 CONCLUSION AND FUTURE WORK

We presented a probabilistic approach for the semantic analysis of word chains and several methods which support the statistical model and improve the performance of the analysis. The problem of finding the semantic of a word chain is formulated in a probabilistic manner. Our statistic model is based on an unknown assignment function for which the statistical parameters are estimated with the EM algorithm depending on the desired statistical dependency. For a dependency of order $g = 1$ the well-know HMM formalism results, increasing g gives generalized versions of HMMs.

We introduced our annotation scheme for the semantic content of a word chain with so called semantic attributes. For our purposes we use a system of 12 semantic attributes encoding the information slots we have to fill to start a database inquiry. One of these attributes is the **NIL** attribute which models parts having no meaning in the domain. In the baseline experiment we prove the feasibility of our statistical model. We achieve a semantic attribute detection accuracy of 57% for a statistical dependency of order 1 and 58% for order 2. The following section presented several methods for improving the performance of our model and show their usability. We tested categorial systems on the lexicon and several methods for initializing the output and transition probabilities. The methods give only small improvements when applied uniquely but their combination shows impressive results. We achieve the best recognition results of 74% for order $g = 1$ and 70% for order $g = 2$ with the categorial system with 220 classes, initial transition probabilities modeling the length of attributes and starting the count of relative word frequencies at 0.

In the future we are concentrating on the initial output probabilities and on ways of using the semantic annotation during training. As our statistical model is very sensitive about the initialization we are looking for better initialization methods to be used e.g. 'intelligent' smoothing techniques for the relative frequencies. We are also considering an information theoretical measure like the salience introduced in [4] for a smoothing operation on the output probabilities. Besides that we want to use the semantic annotation as additional knowledge during the training. At the moment we use the semantic annotation only for the initialization of the output probabilities. We think that by putting this information in the model we still can improve the performance of our probabilistic semantic analysis.

REFERENCES

[1] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38, 1977.

[2] W. Eckert, E. Nöth, H. Niemann, and E. Schukat-Talamazzini. Real Users Behave Weird — Experiences made collecting large Human–Machine–Dialog Corpora. In *Proc. of the ESCA Tutorial and Research Workshop on Spoken Dialogue Systems*, pages 193–196, Vigsø, Denmark, June 1995.

[3] M. Epstein, K. Papineni, S. Roukos, T. Ward, and S. Della Pietra. Statistical natural language understanding using hidden clumpings. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 176–179, Atlanta, 1996.

[4] A. Gorin. On automated language acquisition. *Journal of the Acoustic Society of America*, 97(6):3441–3461, 1995.

[5] J. Hornegger. *Statistische Modellierung, Klassifikation und Lokalisation von Objekten.* Phd–Thesis, Technical Faculty, University Erlangen-Nürnberg, 1996.

[6] R. Pieraccini and E. Levin. A learning approach to natural language understanding. In *NATO-ASI, New Advances & Trends in Speech Recognition and Coding*, volume 1, pages 261–279, Bubion (Granada), Spain, 1993.