Using Prior Domain Knowledge to Build Robust HMM-Based Semantic Tagger Trained on Completely Unannotated Data

Kinfe Tadesse Mengistu KINFE.TADESSE@OVGU.DE
Mirko Hannemann MIRKO.HANNEMANN@GMAIL.COM
Tobias Baum TOBIAS_BAUM@GMX.NET
Andreas Wendemuth ANDREAS.WENDEMUTH@OVGU.DE
Cognitive Systems Group, FEIT-IESK, Otto-von-Guericke University, 39106 Magdeburg, Germany

Abstract

In this paper, we propose a robust statistical semantic tagging model trained on completely unannotated data. The approach relies mainly on prior domain knowledge to counterbalance the lack of semantically annotated treebank data. The proposed method encodes longer contextual information by grouping strongly related semantic concepts together into cohesive units. The method is based on hidden Markov model (HMM) and offers high ambiguity resolution power, outputs semantically rich information, and requires relatively low human effort. The approach yields high-performance models that are evaluated on two different corpora in two application domains in English and German.

1. Introduction

A spoken dialog system with an ideal speech recognizer can barely serve any purpose without a spoken language understanding unit (SLU) that can infer the intention underlying a recognized utterance. Spoken language understanding can be easy for simple application domains where users are restricted in the choice of their formulation of a spoken request. However, the task gets more challenging when a dialog system allows human-to-human like conversation because the natural phenomena of spontaneous speech such as hesitations, false starts, filled pauses, etc. introduce undesirable noise into the input.

Spoken language understanding has been a topic of research since the 70s (Woods, 1983) and spontaneous spoken language understanding has been of particular interest since the early 90s when multiple research laboratories from both academia and industry participated in the DARPA-funded Air Travel Information System (ATIS) evaluations (Price, 1990). In general, the approaches in the domain of spoken language understanding can be grouped as data-driven, rule-based and a combination of the two. Data-driven approaches such as those implemented in CHRONUS of AT&T (Pieraccini & Levin, 1993), and Hidden Understanding Model of BBN (Miller et al., 1994) estimate model parameters from data by counting the frequencies of transitions between states, word observations while in each state and which states start a sentence. These statistical models are robust and perform well but require a large corpus of fully annotated training examples, which is often not practically available. Another popular statistical approach that uses HMMs in SLU is the Hidden Vector State model of Cambridge University (He & Young, 2005). In the Hidden Vector State Model, state transitions between two states are decomposed into separate stack operations that transform one state to the other. A remarkable feature of the HVS model is that it can be trained on "lightly" annotated data and it captures hierarchical structure. Rule-based systems, on the other hand, such as those implemented in TINA of MIT (Seneff, 1992), PHOENIX of CMU (Ward & Issar, 1994), and GEMINI of SRI (Dowding et al., 1994) use hand-crafted semantic rules to extract meaning from a spoken utterance. Rule-based systems do not require a large amount of semantically annotated data and they perform very well when the structure of the spoken utterance is covered by the grammar. However, rule-based systems, in general, are very expensive to build and maintain since they require extensive manual in-

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

volvement and expertise.

Different combinations of rule-based and statistical approaches have also been investigated. For instance, the generative HMM/CFG (context free grammar) model described in (Wang et al., 2005) integrates a knowledge-based approach into a statistical learning framework.

In this paper, we describe an approach towards spoken language understanding that makes extensive use of a priori domain knowledge in order to build domain-dependent semantic models with relatively less human intervention on completely unannotated data. We essentially add semantic information to the output of the speech recognizer of our dialog system so that a deterministic program can easily infer the intention of the user from the output of the semantic tagger. Assuming that an utterance consists of a sequence of concepts, the purpose of the required model is to determine the most likely sequence of semantic concepts that could have generated the observed sequence of words. The notably distinguishing ability of hidden Markov models (HMMs) to estimate the probability of hidden events from observed ones makes them a natural choice for this kind of task.

The remaining part of the paper is organized as follows. Section 2 briefly describes the architecture of our telephone-based spoken dialog system. The modeling approach is described in detail in Section 3. Section 4 describes the data used in the experiments that are described in Section 5. Finally, concluding remarks are presented in Section 6.

2. Architecture of the Dialog System

As can be seen in the high-level architecture of the system in Figure 1, our VoiceXML-based telephone dialog system consists of a telephony interface component to deliver calls into the system; an input component to accept, recognize and understand spoken requests from a caller; an output component to play prompts and responses back to the user; a back-end to serve dialog scripts and other resources; and at the core is a dialog manager that orchestrates the various components of the system.

The input component of the dialog system consists of an audio source component, a speech recognizer, a grammar (language model) component and a semantic analyzer. The recognition resources used by the recognizer; namely, the acoustic model, the language model, and the pronunciation lexicon are prepared offline using HTK (Young et al., 2006) and the real-time speech recognizer is built using ATK (Young, 2007).

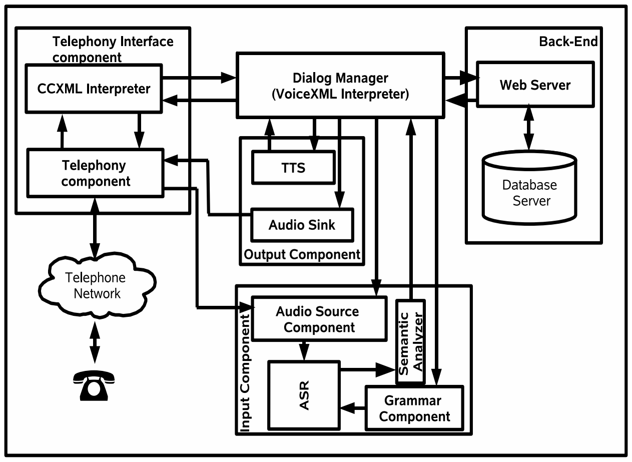

Figure 1. High-level Block Diagram of the Dialog System.

The output of the speech recognizer is sent to the semantic analyzer which we describe in this paper so that the text output of the recognized utterance is enriched with semantic information.

At the core of the system is ©OptimTalk [1] - a VoiceXML platform which consists of not only a VoiceXML interpreter but also a CCXML interpreter, and other abstract interfaces for the integration of our ASR engine, TTS engine, telephony interface, semantic interpreter, etc. The VoiceXML interpreter in OptimTalk serves as the dialog manager in that it executes the dialog by calling the appropriate methods of the various components of the dialog system as shown in Figure 1.

3. Modeling Approach

Understanding an application domain requires a precise identification of the activities, entities, events, attributes and relations within the domain of discourse. The ontologies of the two application domains of interest in this paper - namely, airline travel planning and train inquiries, are modeled in two stages. In the first stage, a detailed list of concepts that are relevant in each application domain are identified using prior domain knowledge and domain-specific example sentences from the training data. As a result, 68 semantic concepts in the domain of airline travel planning and 50 in that of train inquiries are identified. In the second stage, groups of attributes that describe a single semantic concept are grouped together to form cohesive units referred to as super-concepts. For instance, a super-concept DATE contains attributes such as

[1]http://www.optimsys.eu

DAY_OF_MONTH, DAY_OF_WEEK, MONTH, and YEAR. As can be imagined, the prior knowledge used to determine which attributes should belong together to form a super-concept is a commonplace knowledge. Moreover, in order to model multi-word city names and train stations such as "New York City" or "Berlin Friedrichstrasse", we model each with multiple states. The number of sub-concepts in a super-concept can vary; on average, a super-concept contains 3.6 sub-concepts in each application domain.

Accordingly, 14 super-concepts for airline travel planning and 9 for the domain of train inquires are identified. Figure 2 depicts examples of super-concepts along with their attributes (sub-concepts) in the domain of airline travel planning.

CITY (CITY_P1, CITY_P2, CITY_P3, SPELT_CITY),
DATE (DAY_OF_MONTH, DAY_OF_WEEK, MONTH, YEAR),
TIME (MINUTES, HOUR_OF_DAY, AMPM),
AIRLINE (AIRLINE_QUALIFIER, AIRLINE_NAME),
AIRPORT (AIRPORT_NAME, AIRPORT_TYPE,
AIRPORT_QUALIFIER, SPELT_AIRPORT),
CAR_INFO (RENTAL_COMPANY, CAR, CAR_TYPE),
FLIGHT_INFO (FLIGHT_CLASS, FLIGHT_NUMBER,
FLIGHT_TYPE, FLIGHT_QUALIFIER),
HOTEL_INFO (HOTEL_TYPE, HOTEL_QUALIFIER,
LOCATION),
USER (ID, ID_NUMBER, NAME_OF_USER),
PRICE (FARE, AMOUNT_OF_MONEY, FARE_CLASS),

Figure 2. Example super-concepts

Other single state concepts include COUNTRY, STATE, TO, FROM, AT, IN, ON, ARRIVAL, DEPARTURE, RETURN, COMMAND, YES, NO, DUMMY, ...

The rationale behind grouping related sub-concepts together is threefold. First, it improves the predictive power of the model since adjacent related concepts are well coupled. Second, the models produce outputs that are semantically rich and more informative since a phrase is more meaningful than a single word. For instance, phrases like "Saturday the sixteenth of August two thousand eight" or multiword location names such as "Washington D. C." etc. would be more informative if the phrases are labeled as "DATE" and "CITY", rather than tagging each piece with an atomic semantic label. Third, it offers high ambiguity resolution power. For instance, "twenty six" in "November twenty six" would not be confused with other semantic labels such as HOUR, MINUTES, QUANTITY, FLIGHT_NUMBER, ID_NUMBER, etc. as DATE is a super-concept whose attributes are well coupled.

The initial HMMs are defined to be fully connected networks such that any state or sub-network can follow any other single state concept or sub-network with equal probability. Self-loops are allowed in most of the semantic concepts to account for multiple observations from some concepts such as DUMMY, PERIOD_OF_DAY, DAY_OF_MONTH, MINUTES, etc. Every sub-network has its own non-emitting entry and exit states and the transitions between the states within a sub-network are initially ergodic. A one-step transition from the entry state to the exit state of a sub-network is explicitly prohibited to prevent non-emitting loops. Finally, two more non-emitting states INIT and FINAL are introduced to mark the beginning and end of the entire network. Figure 3 shows the partial structure of the HMM for the domain of airline travel planning.

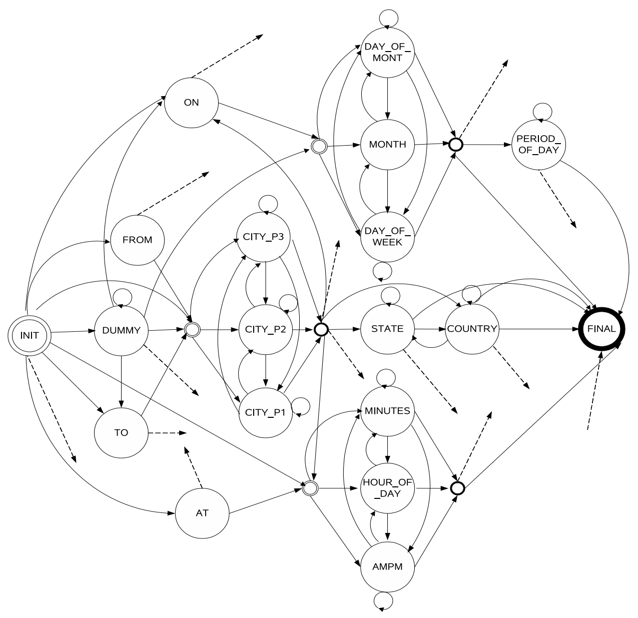

Figure 3. Structure of the HMM

The emission probabilities are initialized by classifying the words in the vocabulary of the application domains into the known set of lexical classes. All words belonging to a semantic label are set equiprobable.

Once we define the model, it may be necessary to bias the transition and emission probabilities of the HMM a bit since we do not have hand-labeled corpus (Elworthy, 1994). This can be done by performing preliminary tests on the training or development data and introducing necessary contraints. For instance, in order to disambiguate words belonging to multiple semantic classes, some unlikely transitions that could not be resolved by the model can be explictly prohibited.

To provide easy tuning and to keep the level of human effort to the minimum, we implemented a model compiler that generates the transition and emission probabilities of the model given the structure and constraints of the HMM in the form shown in Figure 4.

```
INIT
-> except{FINAL}
...
CITY
{
        CITY
        ->except{~CITY}
        CITY_P1
        ->all high{CITY_P2} low{~CITY}
        “city_p1.txt”
        CITY_P2
        ->all high{CITY_P3,~CITY} low{CITY_P1}
        “city_p2.txt”
        CITY_P3
        ->all high{CITY_P3, ~CITY}
        “city_p3.txt”
        SPELT_CITY
        ->all high{SPELT_CITY}
        “spelt_city.txt}
        ~CITY
        ->none
}
->except{INIT} high{AIRPORT, STATE, COUNTRY}
DATE
{
        DATE
        ->except{~DATE}
        MONTH
        ->all high{DAY_OF_MONTH}
        “month.txt”
        DAY_OF_MONTH
        ...
        ~DATE
        ->none
}
->except{INIT}
...
TO
->except{INIT} high{CITY,AIRPORT}
“to.txt”
...
FINAL
->none
```

Figure 4. Excerpt from the model definition

The initial model transition probabilities can be easily tuned as required using the keywords “all”, “high”, “low”, “except”, and “none”. The model compiler also allows us to try different modeling approaches at different levels of hierarchy with relatively less effort than would be required otherwise. The keywords are self-explanatory; for instance, “− >except{...}” means that all transitions out of a state (e.g. INIT) or sub-network (e.g. CITY) are equally likely except the state(s) specified in braces (e.g. FINAL). The keyword “high” sets a higher probability to the specified transitions than to the rest. For every sub-network the entry state is denoted by the name of the super-concept (e.g. CITY) and the exit state by a tilde followed by the name of the super-concept (e.g. `~CITY`).

Given “well-informed” initial models, the Expectation-Maximization (EM) algorithm can be used to estimate reliable model parameters. The algorithm starts with the carefully defined initial HMM described above and iteratively refines the model parameter values. Once we have a well-trained model, the Viterbi algorithm can be used to find the highest probability semantic label sequence which corresponds to the sequence of observed words.

4. Data Description

The semantic model for airline travel planning domain is trained and evaluated on the transcriptions of speech data from the 2001 DARPA Communicator Evaluation telephone speech corpus (Walker et al., 2003). Table 1 describes the training and test sets of the airline travel planning domain used in the experiments described in Section 5.

Table 1. Description of data for airline travel planning domain

Set	# of Utt.	# of Uniq. Words
Training	8000	915
Test	1000	581

The transcriptions of 8000 utterances were selected from the training data that we used to build the acoustic model for our speech recognizer by removing too many occurences of some very short utterances such as “yes”, and “no”. For testing purposes, 1000 distinct, relatively longer utterance transcriptions are selected from a 5000 utterance test-set. The average number of words per utterance are 4.04 and 8.98 in the training and test sets of the airline travel planning domain, respectively.

For the domain of train inquiries, we use the transcriptions of utterances from ©Erlanger Bahn Anfragen (ERBA) speech corpus in German. Table 2 describes the data used to build and evaluate the German semantic model.

Table 2. Description of data for train inquiries domain

Set	# of Utt.	# of Uniq. Words
Training	8000	921
Test	1000	830

The utterances in the domain of train inquiries are long, well-structured, and grammatically correct utterances. The average number of words per utterance are 12.26 and 11.76 in the training and test sets, respectively.

5. Experiments and Results

The performance of the systems is evaluated using precision, recall and F-measure where precision (P) is the number of correctly labeled concept chunks out of all tagged concepts, recall (R) is the number of correctly identified concepts from the ground truth annotation. F-measure is the harmonic mean of precision and recall defined by Equation 1

$$F = \frac{2PR}{P+R} \qquad (1)$$

5.1. The Effect of the Proposed Modeling Method

If we randomly assign to each token one of its possible tags, we achieve an average performance rate of 56.4% in F-measure. This can be considered as the minimum average baseline performance on the airline travel planning domain.

Table 3 summarizes the performance gain mainly due to the modeling approach we described in Section 3. “Flat” in Table 3 refers to a flat, ergodic HMM model, before grouping of related concepts where each state is one of the sub-concepts or single state concepts described in Section 3. “Grouped” in Table 3 refers to a model just after grouping related sub-concepts together. In both cases no tuning is performed.

Table 3. Flat vs. Grouped initial models on Communicator task

Model	P(%)	R(%)	F-Measure(%)
Flat	57.32	66.42	61.53
Grouped	85.67	77.98	81.64

As can be seen in Table 3, the performance was improved by 20.11% absolute in F-measure just after grouping related concepts together using our prior domain knowledge. This suggests the modeling approach we used is quite suitable for this kind of task.

The experiments that follow describe the performance of both airline travel planning and train inquiries systems stage by stage using the proposed modeling approach.

5.2. Performance of the Initial Models after Tuning

Table 4 depicts the performance of the initial models tuned as described in Section 3 before EM training.

Table 4. Performance of the tuned initial models

Data	P(%)	R(%)	F-Measure(%)
Communicator	96.15	84.46	89.92
ERBA	94.90	94.19	94.54

As can be observed, the performance of the initial model after tuning is improved significantly.

5.3. Performance of the Models after Training

The results after performing EM training are summarized in Tables 5.

Table 5. Performance of the models after training

Data	P(%)	R(%)	F-Measure(%)
Communicator	98.75	84.58	91.12
ERBA	96.94	96.19	96.56

The best performance was achieved after only one iteration of training on the airline travel planning and two on the train information inquiries domain, respectively. It can be noted in Table 5 that the recall, mainly for the airline travel planning domain, is quite low. This is due to unseen transitions and “out-of-vocabulary” words (OOVs) that resulted in some unparseable utterances. This is, in turn, attributed to the sparse data problem and the inevitability of OOVs. In order to combat this problem, we smoothed transition and emission probabilities. The recall in the train inquiries domain is high because the rate of OOVs in the German test-set is low.

5.4. Performance after Smoothing

In the case of transitions, we assigned a very small nonzero probability for all allowable transitions not seen in the training data. For words not found in the lexicon, we introduced a vocabulary item “oov” in those lexical classes where there is no exhaustive list of words; for instance, CITY, DUMMY, AIRLINE, etc. The probability of the “oov” word in a concept is set to the sum of the probabilities of all words belonging to that concept that occur only once in the training set; the rest of the probabilities are discounted accordingly so that all sum up to one. As a result, all the sentences could be parsed and 69% of the "oov" words out of 135 in the domain of airline travel planning were correctly labeled.

Accordingly, the performance of the models is significantly improved as can be seen in Table 6.

Table 6. Performance of the models after smoothing

Data	P(%)	R(%)	F-Measure(%)
Communicator	96.91	97.12	97.01
ERBA	96.82	97.41	97.11

5.5. Example Tagged Outputs

An example tagged output of an utterance in the domain of airline planning is:
(I want to fly) DUMMY (from) FROM (Los Angeles) CITY (to) TO (Osaka) CITY (Japan) COUNTRY (on) ON (September thirtieth) DATE (in the morning) PERIOD_OF_DAY
An example tagged output of an utterance in the domain of train inquiries in German[2] is:
(ich moechte) DUMMY (alle) MODIFIER (Abfahrts) DEPARTURE (und) CONNECTIVE (Ankunftszeiten) ARRIVAL (aller) MODIFIER (Zuege) TRAIN (nach) TO (Minden) LOCATION (an) ON (einem) DUMMY (Wochentag) DATE (zwischen neun und sechs Uhr) PERIOD_OF_DAY

6. Conclusions

In this paper, we described the use of prior domain knowledge to build an HMM-based semantic tagging model with four main virtues - namely, it is trained on completely unlabeled data, it offers high ambiguity resolution ability, it outputs naturally appealing and semantically rich information, and it requires relatively low human effort as the model itself takes care of many sources of ambiguity. Moreover, the model is robust in that it could parse unseen observations and could correctly label a significant amount of out-of-vocabulary words. The performance of the proposed model is 97.01% for airline travel planning and 97.11% for train-inquires system in F-measure on 1000-utterance test-sets. The success of this approach relies mainly on the use of a priori domain knowledge to build a reliable initial model while keeping the human effort to the minimum.

[2]Translation of the German utterance: I want all departure and arrival times of all trains to Minden on a weekday between nine and six o'clock

References

Dowding, J., Moore, R., Andry, F., & Moran, D. (1994). Interleaving syntax and semantics in an efficient bottom-up parser. *Proc. 32nd Annual Meeting of the Association for Computational Linguistics*, 110–116.

Elworthy, D. (1994). Does Baum-Welch Re-estimation Help Taggers? *4th Conference on Applied Natural Language Processing, pages 53-58. Association for Computational Linguistics*, 53–58.

He, Y., & Young, S. (2005). Semantic Processing using the Hidden Vector State Model. *Proceedings of Computer Speech and Language*, *19*, 85–106.

Miller, S., Bobrow, R., Ingria, R., & Schwartz, R. (1994). Hidden understanding models of natural language. *Proceedings of the Association of Computational Linguistics*, 25–32.

Pieraccini, R., & Levin, E. (1993). A learning approach to natural language understanding. *NATO-Advanced Study Institute: New Advances & Trends in Speech Recognition and Coding*, *1*, 261–279.

Price, P. (1990). Evaluation of Spoken Language System: The ATIS Domain. *DARPA Speech and Natural Language Workshop.*

Seneff, S. (1992). TINA: A natural language system for spoken language applications. *Comput. Linguistics*, *18*, 61–86.

Walker, M., Aberdeen, J., & Sanders, G. (2003). 2001 communicator evaluation.

Wang, Y.-Y., Deng, L., & Acero, A. (2005). Spoken language understanding: An introduction to statistical framework. *IEEE Signal Processing Magazine*, *19*, 85–106.

Ward, W., & Issar, S. (1994). Recent improvements in the CMU spoken language understanding system. *Proceedings of the ARPA Human Language Technology Workshop*, 213–216.

Woods, W. A. (1983). *Language Processing for Speech Understanding.* Englewood Cliffs, NJ: Computer Speech Processing, Prentice-Hall International.

Young, S. (2007). ATK:An Application Toolkit for HTK Version 1.6.1.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X. A., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., & Woodland, P. (2006). The HTK Book. Revised for HTK Version 3.4.