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

Kinfe Tadesse Mengistu, Mirko Hannemann,
Tobias Baum and Andreas Wendemuth

Cognitive Systems Group, Otto-von-Guericke University,
Magdeburg, Germany

09.07.2008

Introduction

- Spoken Language Understanding (SLU) unit in a dialog system aims at getting what is meant from what is said.
- Semantic tagging is an approach towards SLU where an utterance is conceived as a hidden sequence of semantic concepts expressed in words or phrases.
- The required model is one that determines the most likely sequence of the hidden semantic concepts that could have generated the observed sequence of words.
- Hidden Markov Model (HMM) is a natural choice for estimating the probability of hidden events from observed ones.

Introduction

In this paper, we describe:

- a robust HMM-based sematic tagger trained on completely unannotated data,
- the use of prior domain knowledge:
  - to model the ontology of a given application domain: identify the activities, entities, attributes and relations within a given domain of discourse.
  - to group semantically related concepts together so that adjacent, strongly related concepts are well-coupled and longer context can be captured,
  - to define well-informed initial HMM topology and determine good initial model parameter values so that E-M algorithm can be effectively used.
- a method to keep the amount of human intervention required to the minimum.

Architecture of the Dialog System

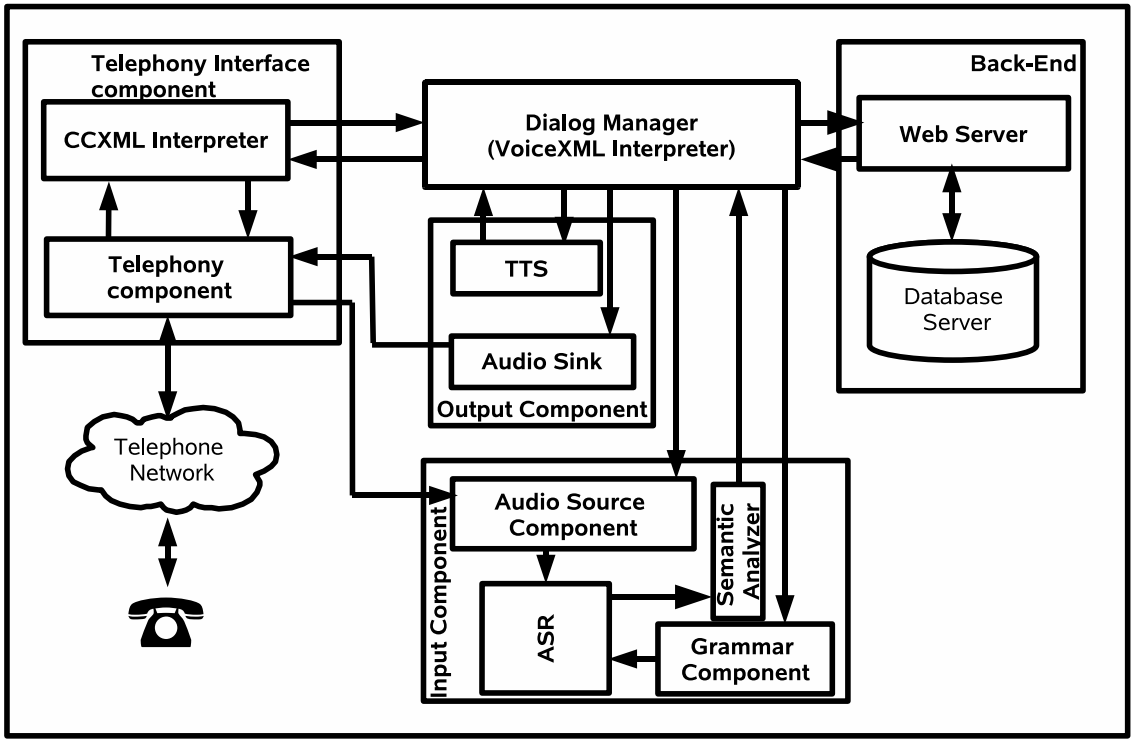


FIGURE *1: High-level Block Diagram of the Dialog System*

Modeling Approach

Two application domains are considered - namely:

- airline travel planning (in English), and
- train inquiries (in German)

Each application domain is modeled in two stages:

- a detailed list of semantic concepts in each application domain is identified using prior domain knowledge and training example sentences,
- groups of attributes (sub-concepts) that describe a single semantic concept are grouped together to form cohesive units referred to as super concepts.
  For instance, DATE, TIME, and multi-word city and train station names could be modeled as super-concepts (sub-networks).

Modeling Approach

The rationale for grouping of related concepts into super-concepts is threefold:

- Improve the predictive power of the model since adjacent related concepts are well-coupled,
- Produce semantically rich outputs (a phrase is more meaningful than a single word),
- Improve ambiguity resolution power.

Modeling Approach: Super-Concepts

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, DUMMY, YES, NO, etc.

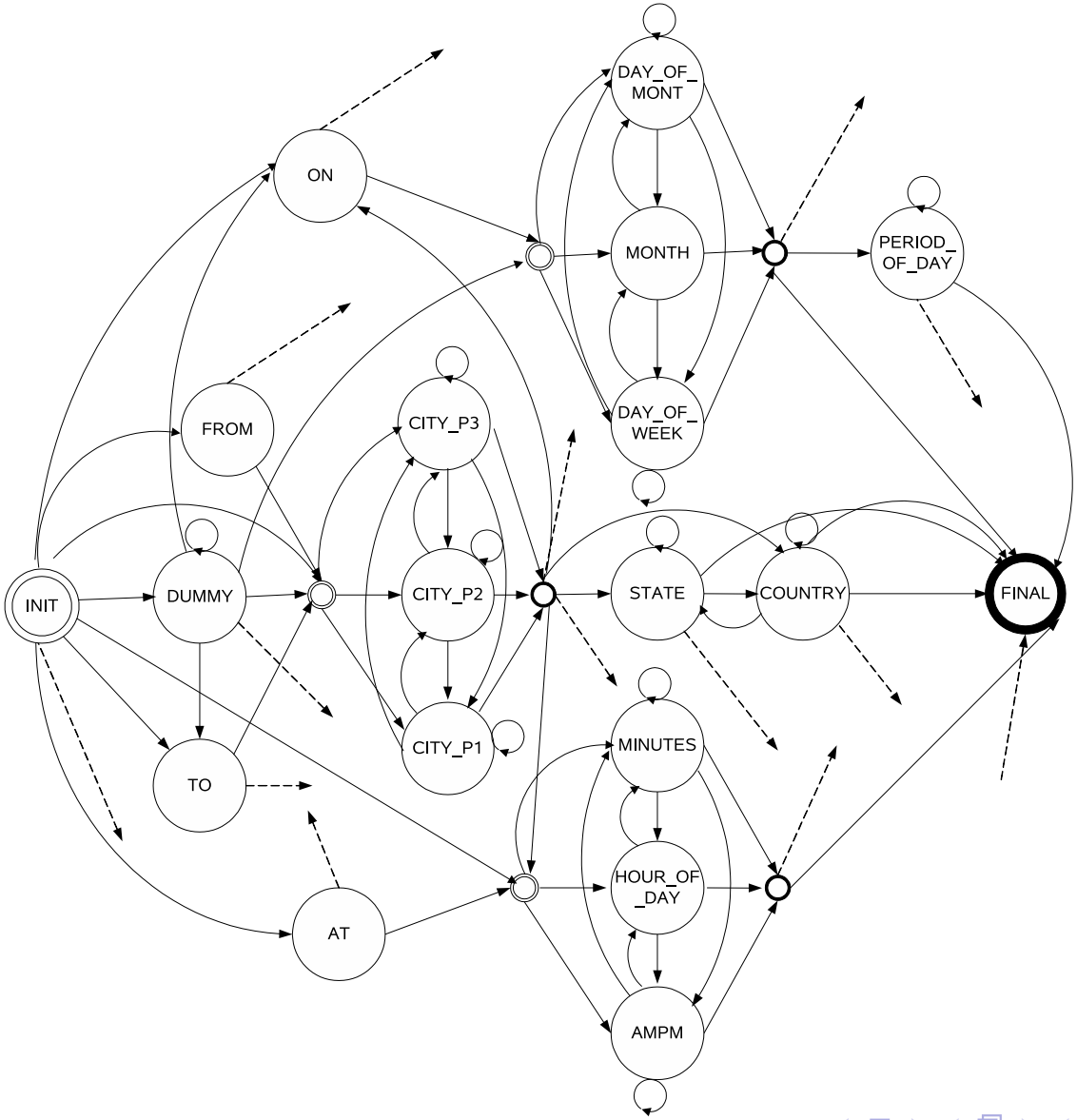
Modeling Approach: Model Definition

To provide easy tuning of model parameters and to keep the level of human effort to the minimum, we implemented a model compiler that generates the initial model parameter values given input of the following form:

Modeling Approach: Model Definition

```
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
```

Modeling Approach: Strurcture of the HMM



Data Description

TABLE *1: Description of data for airline travel planning domain*

SET	NO. OF UTT.	NO. OF UNIQ. WORDS
TRAINING	8000	915
TEST	1000	581

Average number of words per utterance is 4.04 in the training-set and 8.98 in the test-set.

TABLE *2: Description of data for train inquiries domain*

SET	NO. OF UTT.	NO. OF UNIQ. WORDS
TRAINING	8000	921
TEST	1000	830

The average number of words per utterance is 12.26 in the training set and 11.76 in the test-set.

Experiments and Results

The performance of the models is evaluated in terms of:

- Precision (P): the number of correctly labeled concept chunks out of all tagged concepts.
- Recall (R): the number of correctly identified concepts from the ground truth annotation.
- F-Measure: the harmonic mean of precison and recall.

Baseline: If we assign to each token one of its possible tags randomly, the average performance is 56.4% in F-measure.

Experiments and Results

Flat: refers to a flat, ergodic HMM model, before grouping of related concepts where each state is one of the sub-concepts.

Grouped: refers to a model after grouping of related sub-concepts.

TABLE *3: Flat vs. Grouped initial models on Communicator task*

MODEL	PRECISION(%)	RECALL(%)	F-MEASURE(%)
FLAT	57.32	66.42	61.53
GROUPED	85.67	77.98	81.64

Tuning: refers to biasing the initial transition probabilities based on prior domain knowledge and training examples.

TABLE *4: Performance of the tuned initial models*

DATA	PRECISION(%)	RECALL(%)	F-MEASURE(%)
COMMUNICATOR	96.15	84.46	89.92
ERBA	94.90	94.19	94.54

Experiments and Results

Training: a few iterations of E-M training algorithm.

TABLE *5: Performance of the models after training*

DATA	PRECISION(%)	RECALL(%)	F-MEASURE(%)
COMMUNICATOR	98.75	84.58	91.12
ERBA	96.94	96.19	96.56

Smoothing: to account for unseen transitions and out-of-vocabulary words.

TABLE *6: Performance of the models after smoothing*

DATA	PRECISION(%)	RECALL(%)	F-MEASURE(%)
COMMUNICATOR	96.91	97.12	97.01
ERBA	96.82	97.41	97.11

Example Tagged OUtput

TABLE *7: Example Tagged Output*

Phrase	Semantic Label
(I would like to fly)	DUMMY
(to)	TO
(Los Angeles)	CITY
(from)	FROM
(Dallas Fort Worth international airport)	AIRPORT
(on)	ON
(September thirtieth)	DATE
(in the morning)	PERIOD_OF_DAY

Summary

We described an HMM-based semantic tagging model with the following features. The model:

- is trained on completely unannotated data with the help of a priori domain knowledge,
- offers high ambiguity resolution power,
- outputs semantically rich information,
- requires relatively low human effort,
- correctly labels a significant amount of OOV words,
- gives very high performance; i.e., over 97% (in F-measure) in two different application domains and languages on two 1000-utterance test-sets.