Hidden Markov Models for Understanding in a Dialogue System^{*}

Fernando Blat, Sergio Grau, Emilio Sanchis, and María José Castro

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València
Camí de Vera s/n, 46022 València, Spain
{fblat,sgrau,esanchis,mcastro}@dsic.upv.es

Abstract. In this work, we present an approach to Automatic Speech Understanding based on stochastic models. In a first phase, the input sentence is transduced into a sequence of semantic units by using hidden Markov models. In a second phase, a semantic frame is obtained from this sequence of semantic units. We have studied some smoothing techniques in order to take into account the unseen events in the training corpus. We have also explored the possibility of using specific hidden Markov models, depending on the dialogue state. These techniques have been applied to the understanding module of a dialogue system of railway information in Spanish. Some experimental results with written and speech input are presented.

1 Introduction

Spoken dialogue systems are a natural human-machine interface within the framework of information systems. Due to advances in many areas, such as speech recognition, language modelling, speech understanding, or speech synthesis, it is possible to build prototypes of dialogue systems applied to restricted semantic domains, such as flight information (ATIS [1]), railway information (Basurde [2]), weather forecasts (Jupiter [3]), etc. . .

Dialogue systems are characterised by having a similar basic structure: an automatic speech recognition module, an understanding module, a dialogue module, an access module to the database and an answer generation module (usually, with speech synthesis). In this work, we will focus on the understanding module. There are two general ways to approach the understanding problem, one approach is based on rules and the other is based on stochastic models. *Rule-based understanding* is implemented through grammars defined from rules. In order to improve the behaviour of such systems, the human expert usually has to combine syntactic and semantic grammars to perform analyses at different levels. Examples of this approach are TINA [4], PHOENIX [5] and ARISE [6].

^{*} Thanks to the Spanish CICYT agency under contract TIC2002-04103-C03-03 for funding.

Stochastic models are inductively trained in order to transduce the input sentences into sequences of semantic units. The most commonly used stochastic models are n -grams, stochastic grammars and hidden Markov models [7–9].

Independently of the approach to the understanding problem, we have to define a semantic model to represent the meaning of the user turns so that the computer can store such information. Semantic *frames* and their *attributes* [10] are the most widely used for this. A frame is a template that represents a given concept, and the information is complemented with a series of related attributes. Therefore, understanding can be viewed as a transduction process from the input (speech in natural language) into one or more semantic frames with their corresponding attributes.

Hidden Markov Models (HMMs) are known to perform well in speech recognition tasks, but they can also be used in understanding processes [8, 11]. We will use a HMM as a transducer from the natural language into an intermediate semantic language. Afterwards, the sequence of intermediate semantic units is converted into one or more semantic frames. The possibility of learning HMMs from samples, which makes it easy to change the tasks or language, makes this approach attractive.

One interesting way to improve the performance of the understanding process is to take advantage of the structure of the dialogue. Turns of the dialogues can be classified in terms of one or more dialogue acts. We have defined a set of three-level dialogue act labels that represents the general dialogue behaviour (first level) and specific semantic characteristics (second and third level). In our work, we study the use of specific stochastic models, that is, different models that depend on the dialogue acts.

In this work, we explore two approaches to stochastic language understanding in a dialogue system: the first is to infer a *global HMM* to extract the semantic information by segmenting the user turn into semantic units, and the second is to use *specific HMMs*, that is, different models that depend on the class of user turn. The results of the experiments with the correct transcriptions of the user turns (text data) and with the sequences of words obtained from the recognition process (speech data) are presented for these approaches.

2 The Dialogue Task

The final objective of this dialogue system is to build a prototype for information retrieval by telephone for a Spanish nation-wide train system. Queries about timetables, prices and services for long distance trains can be made to the dialogue system. Four types of scenarios were defined (departure/arrival time for a one-way trip, prices, services, and one free scenario). A total of 215 dialogues were acquired using the Wizard of Oz technique.

2.1 Labelling the Turns

The definition of dialogue acts is an important issue because they represent the successive states of the dialogue. The labels must be specific enough to show

the different intentions of the turns in order to cover all the situations, and they must be general enough to be easily adapted to different tasks.

The main feature of the proposed labelling of our system is the division into three levels. The first level, called *speech act*, is general for all the possible tasks and it comprises the following labels: Opening, Closing, Undefined, Not_understood, Waiting, New_consult, Acceptance, Rejection, Question, Confirmation, Answer.

The second and third levels, called *frames* and *cases*, respectively, are specific to the working task and give the semantic representation. The frame labels defined for our task are: Affirmation, Departure_time, New_data, Price, Closing, Return_departure_time, Rejection, Arrival_time, Return_departure_time, Confirmation, Not_understood, Trip_length, Return_price, Return_train_type, Train_type. Each frame has a set of slots which have to be filled to make a query or which are filled by the retrieved data after the query. The specific data that fill in the slots are known as *cases*. Cases take into account the data given in the user turn to fill the slots. Examples of cases for this task are: origin_city, destination_city, departure_date, departure_time, train_type, price, ...

2.2 Specific Classes of User Turns

User turns can be classified in accordance with the dialogue state defined from the speech act label of the previous computer turn. Only four speech acts have enough sentences to train relevant HMMs (see Table 1). These four classes are¹:

Opening: user turns after the welcome message of the dialogue system.

Machine: *Welcome to the information system for nation-wide trains, what information would you like?*

User: *I would like to know the timetables of the Euromed train from Barcelona to Valencia.*

Confirmation: user turns after a confirmation question of the dialogue system.

Machine: *Do you wish to travel from Barcelona to Valencia?*

User: *Yes.*

New_consult: user turns after the question of the dialogue system “*Anything else?*”.

Machine: *Anything else?*

User: *How much is the one-way ticket?*

Question: the answers of the user after a question by the dialogue system.

Machine: *Which day do you want to travel?*

User: *Next Friday.*

2.3 Intermediate Semantic Language

The transduction of user turns into one or more semantic frames is not done directly, it is divided into two stages. In the first stage, the input sentence is

¹ An example of user turn from each class is given below (only the English translation).

Segmentation		Semantic frame
<i>Can you tell me</i>	query	(Departure_time): origin_city: valencia destination_city: barcelona departure_date: 2004/May/21
<i>the railway timetable</i>	<departure_time>	
<i>from</i>	origin_marker	
<i>Valencia</i>	origin_city	
<i>to</i>	destination_marker	
<i>Barcelona</i>	destination_city	
<i>for</i>	departure_marker	
<i>Friday?</i>	departure_weekday	

Fig. 1. An example of segmentation and final frame for the user turn “*Can you tell me the railway timetable from Valencia to Barcelona for Friday?*”. In the system, the user turn is first preprocessed using lemmas and categories. Only the English translation is given.

segmented and transduced into terms of a sentence from an intermediate semantic language using an HMM. In the second step, the result of the segmentation process will be the input of the translator which generates the semantic frame automatically. This step is quite simple, and it is performed by a rule-based approach. An example of segmentation and final frame is shown in Figure 1.

The intermediate semantic language we have defined for the task is composed by a set of semantic units which represents the meaning of words (or sequences of words) in the original sentences. For example, the semantic unit `query` can be associated to “*can you tell me*”, “*please tell me*”, “*what is*”, etc. In this way, an input sentence (sequence of words) has an associated semantic sentence (sequence of semantic units), and there is an inherent segmentation. A total of 64 semantic units have been defined in order to cover all the possible meanings of each user turn included in the corpus. Examples of semantic units are: `courtesy`, `closing`, `query`, `confirmation`, `<departure_time>`, etc.

3 Hidden Markov Models for Understanding

HMMs are known to perform well in speech recognition tasks, but they can also be used in understanding processes [8, 11]. We will use a HMM as a transducer from the natural language of the user turns into the intermediate semantic language described in section 2.3. This translation consists of a segmentation in which a semantic unit is assigned to one or more words.

Some advantages of these HMMs are the capability of automatic learning from training samples, the capability to represent the sequentiality of language in terms of sequence of states, and the representations of different observations (words) associated to each meaning (or semantic unit).

The transition probabilities between states are obtained in the learning process. These probabilities are the bigram probabilities of the semantic units in the training set. The observation probabilities of words in each state are obtained from the frequency of observations of each word in each state (see an example in Figure 2). Due to the lack of training samples, some observation probabilities

Fig. 2. Example of a fragment of a specific HMM of type New_consult.

Table 1. Characteristics of the Basurde corpus.

	Total	Opening	Confirmation	New_consult	Question
Total of turns	1,378	207	257	804	110
Total of words	16,219	5,034	1,371	9,228	586
Average of words by turn	11.7	24.3	5.3	11.5	5.3
Vocabulary size	635	410	214	483	163

are underestimated. Smoothing techniques can avoid this problem. In our case, we have implemented two different smoothing techniques: Laplace smoothing and back-off smoothing [12]. Once the model has been trained, the Viterbi algorithm [13] is used to determine the optimal segmentation of the user turns. Finally, a backtrace recovering algorithm determines the path followed by the model by extracting the maximum probability, that is, we obtain a segmentation of the sentence in terms of semantic units (see an example of segmentation from HMMs in Figure 1).

Afterwards, a simple translator based on manually defined rules converts the sequence of semantic units into one or more semantic frames with their attributes (see an example of the obtained frame in Figure 1).

4 The Basurde Corpus

For understanding purposes, we are only concerned with the semantics of the words present in the user turn of a dialogue. A total of 1,378 user turns were obtained from the 215 total dialogues of the task. Each of these turns can be composed of one or more sentences. The characteristics of the dataset are shown in Table 1, along with the distribution of each class of user turns².

4.1 Preprocessing

Problems of coverage and lack of training samples can occur when learning HMMs due to the fact that the number of instances of a word can be very low. We have to reduce the size of our lexicon by using lemmas and categories

² As a matter of fact, a total of 1,440 user turns were obtained from the dialogues, but some were discarded because their class frequency was very low.

[14]. Examples of lemmas are: substitution of any conjugated form of a verb with its corresponding infinitive form, plural words are substituted by singular words, etc. Examples of categories are: substitution of any instance of a city name with the category *CityName*, and the same for days, months, seasons, railway stations, etc.

In this way, we reduced the size of our lexicon from 635 to 386 different words. An example of a categorised and lemmatized sentence is:

Original user turn: *I would like to know the timetables of the Euromed train from Barcelona to Valencia.*

Preprocessed turn: want know timetable of TrainType train OriginMarker CityName DestinationMarker CityName

4.2 Speech Corpus

As mentioned above, 215 dialogues were acquired. The user turns were manually transcribed to have a correct written corpus. In addition, to emulate the real behaviour of the dialogue system, the user turns were processed using an automatic speech recognition system based on acoustic HMMs and bigrams as language model. The obtained Word Accuracy was 72%.

5 Experiments

Different measures were used to determine the correctness of the understanding models:

- **Correct semantic sequences (*css*):** percentage of semantic sentences that match up exactly with the reference sentences.
- **Correct semantic units (*csu*):** percentage of correct semantic units that match the reference sentences.
- **Correct frames (*cf*):** percentage of frames that match up exactly with the reference frames.
- **Correct frame units (*cfu*):** percentage of frame attributes that have the same value as the reference frames.

First of all, back-off and Laplace smoothing techniques were tested in order to decide which of the two smoothing techniques should be used. The results showed that back-off smoothing performed better. Thus, the rest of the tests were performed using back-off smoothing.

The categorised and lemmatized corpus was used for the experiments (some previous experiments were performed without lemmas or categories and the results were worse). First of all, a general HMM was trained using the whole training corpus. The final experiment involved five runs using the Leaving-One-Out scheme: training the HMM with the data of four partitions and testing only with one partition³. Therefore, the measures obtained for the test set reported

³ The corpus was homogeneously partitioned in five sets, using 80% of the corpus (4 partitions) for training and 20% of the corpus (1 partition) for testing.

Table 2. Performance of the general HMM and the specific HMMs.

General HMM										
	Opening		Confirmation		New_consult		Question		Total	
	<i>Text</i>	<i>Speech</i>	<i>Text</i>	<i>Speech</i>	<i>Text</i>	<i>Speech</i>	<i>Text</i>	<i>Speech</i>	<i>Text</i>	<i>Speech</i>
<i>css</i>	30.8	3.4	72.2	55.0	51.4	25.0	67.6	24.2	53.5	27.2
<i>csu</i>	74.2	46.0	74.3	58.9	67.0	47.7	71.3	28.8	69.8	48.0
<i>cf</i>	66.0	23.6	82.0	71.0	72.4	52.4	79.2	46.4	73.8	51.0
<i>cfu</i>	87.4	75.0	90.3	80.6	85.0	74.6	91.0	62.0	86.8	74.8

Specific HMMs										
	Opening		Confirmation		New_consult		Question		Total	
	<i>Text</i>	<i>Speech</i>	<i>Text</i>	<i>Speech</i>	<i>Text</i>	<i>Speech</i>	<i>Text</i>	<i>Speech</i>	<i>Text</i>	<i>Speech</i>
<i>css</i>	27.0	3.0	70.8	57.0	50.2	24.6	54.8	23.0	50.9	27.3
<i>csu</i>	71.5	43.0	63.9	52.5	65.7	46.0	45.5	17.3	64.6	44.4
<i>cf</i>	59.0	26.8	81.6	73.6	71.4	51.2	61.4	43.8	70.6	51.1
<i>cfu</i>	83.9	73.0	82.2	76.3	84.4	73.9	72.3	43.8	82.9	71.8

in Table 2 are the averaged results of the five runs of the experiment. Results are given for the entire test corpus (**Total** column of the table) and are also detailed for each class of user turn (**Opening**, **Confirmation**, **New_consult** and **Question**). Experiments were also performed with the correct transcriptions of the user turns (*Text* column in Table 2) and with the sequences of words obtained from the recognition process (*Speech* column).

Afterwards, we tested specific HMMs, that is, a different model for each class of the user turns. Experiments of this kind were performed to try to exploit the similarities of the user turns of each class in order to be able to better learn the structure of its class. The obtained results are also shown in Table 2 (as before, experiments with text and speech data were performed, and the averages of five runs of the experiments are given).

The results show that the understanding module is not capable of making a correct segmentation in many cases (*css* figures are very low for all cases, specially for speech input; only **Confirmation** turns can be segmented accurately). However, the number of correct semantic units (*csu*) and correct frames (*cf*) obtained are clearly much better than the correct segmentation sequences. This means that many segmentation errors are not semantically relevant and our model can recover from these errors.

If we compare the results obtained for the general HMM and the specific HMMs, it can be observed that general models are better than specific ones. One of the causes could be the lack of training samples for the classes of specific models, which makes it difficult to learn the structure of the classes they model.

6 Conclusions

We have presented the understanding process of a dialogue system as a transduction process from natural language into one or more semantic frames. In this process, we have seen the viability of using HMMs as transducers.

With this corpus we tested two types of HMMs: general and specific models for the different classes of user turns. The results show that the general model works better than the specific HMMs, because it is trained with more samples. It is necessary to use smoothing techniques in order to avoid the problem of underestimation and unseen words.

Other approaches to this same problem using grammatical inference techniques obtained similar results [15]. We can conclude that HMMs are a good approach as an understanding model in a dialogue system.

References

1. D. S. Pallet et al. The 1994 benchmark test for the ARPA spoken language program. In *Proceedings of ARPA Workshop on Spoken Language Technology*, 1995.
2. A. Bonafonte et al. Desarrollo de un sistema de diálogo oral en dominios restringidos. In *Primeras Jornadas de Tecnología del Habla*, Sevilla (Spain), 2000.
3. V. Zue et al. Jupiter: A telephone-based conversational interface for weather information. *IEEE Trans. on Speech and Audio Processing*, 8(1), 2000.
4. S. Seneff. TINA: A natural language system for spoken language applications. *Computational Linguistics*, 18(1):61–86, 1992.
5. W. Ward. Evaluation of the CMU ATIS system. In *Proc. of DARPA Speech and Natural Language Workshop*, pages 101–105, 1991.
6. L. Lamel et al. The LIMSI Arise system. *Speech Communication*, 31(4):339–354, 2000.
7. E. Levin and R. Pieraccini. Concept-Based Spontaneous Speech Understanding System. In *Proc. 4th Eurospeech'95*, pages 555–558, 1995.
8. H. Bonneau-Maynard and F. Lefèvre. Investigating stochastic speech understanding. In *Proc. of IEEE ASRU*, 2001.
9. Klaus Macherey, Franz Josef Och, and Hermann Ney. Natural Language Understanding Using Statistical Machine Translation. In *Proc. 7th Eurospeech'01*, pages 2205–2208, 2001.
10. C. J. Fillmore. The case for case. *Universals in Linguistic Theory*, pages 1–90, 1968.
11. Emilio Sanchis, Fernando García, Isabel Galiano, and Encarna Segarra. Applying dialogue constraints to the understanding process in a Dialogue System. In *Proc. of 5th TSD*, pages 389–395, 2002.
12. D. Vilar, M. J. Castro, and E. Sanchis. Connectionist classification and specific stochastic models in the understanding process of a dialogue system. In *Proc. Eurospeech'03*, pages 645–648, 2003.
13. Jr. G. D. Forney. The Viterbi algorithm. *Proc. IEEE*, pages 268–278, 1973.
14. E. Segarra and L. Hurtado. Construction of Language Models using Morfic Generator Grammatical Inference MGGI Methodology. In *Proc. Eurospeech'97*, pages 2695–2698, 1997.
15. E. Segarra et al. Extracting Semantic Information Through Automatic Learning Techniques. *IJPRAI*, 16(3):301–307, 2002.