

# Connectionist classification and specific stochastic models in the understanding process of a dialogue system

David Vilar, María José Castro, Emilio Sanchis

Departament de Sistemes Informàtics i Computació  
Universitat Politècnica de València

{dvilar, mcastro, esanchis}@dsic.upv.es

## Abstract

In this paper we present an approach to the application of specific models to the understanding process of a dialogue system. The previous classification process is done by means of Multi-layer Perceptrons, and Hidden Markov Models are used for the semantic modeling. The task consists of answering telephone queries about train timetables, prices and services for long distance trains in Spanish. A comparison between a global understanding model and the specific models is presented.

## 1. Introduction

The development of spoken dialogue systems is an important goal in the area of language technologies. Due to the advance in many areas, such as speech recognition, language modeling, speech understanding, or speech synthesis, it is possible to build prototypes of dialogue systems applied to restricted semantic domains [1, 2, 3].

Nevertheless, due to the inherent difficulty of the task, there are still many aspects to improve in this kind of systems. In particular, the difficulty of robust speech recognition and understanding and the effect of spontaneous speech, in the framework of a mixed initiative dialogue, are important drawbacks that produce errors in the initial phases of the process, which are difficult to detect and correct. Therefore, it is convenient to concentrate efforts in the recognition and understanding processes by means of adapting the usual techniques to the particularity of the context of the dialogue, or combining different methodologies that can be used to the same objective.

An approach to speech understanding based on stochastic models is presented in this work. We focus in the understanding module, that is, the process that gets the output of the speech recognizer (only one hypothesis) and supplies its output (a semantic message) to the dialogue manager. The task considered (BASURDE) consists of answering telephone queries about train timetables, prices and services for long distance trains in Spanish [4].

It is well known that the use of Hidden Markov Models (HMMs) gives very good results in speech recognition systems. They also can be used in understanding processes [5, 6]. The possibility of automatic learning 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 a dialogue and the limited types of utterances that appear in a dialogue. The turns of the dialogues can be classified in terms of one or more dialogue acts. We have defined a set of three-level dialogue acts that represents the general dialogue behaviour (first level) and specific semantic characteristics (sec-

ond and third level). In our work we study the use of specific stochastic models, that is, different models depending of the dialogue acts [7].

In order to detect the kind of dialogue act associated to the user utterance, a classification based on Neural Networks is done. Once the Neural Network supplies one or more hypothesis of dialogue acts, the specific HMM is applied to extract the semantic information, by segmenting the user turn into semantic units.

## 2. The dialogue task

The final objective of this dialogue system is to build a prototype for information retrieval by telephone for Spanish nationwide trains [4]. Queries are restricted to timetables, prices and services for long distance trains. Several other dialogue projects [2, 8] selected the same task.

A set of person-to-person dialogues corresponding to a real information system was recorded and analyzed. Then, four types of scenarios were defined (departure/arrival time for a one-way trip, departure/arrival time for a two-way trip, prices and services, and one free scenario). After that a total of 215 dialogues were acquired using the Wizard of Oz technique. From these dialogues, a total of 1,440 user turns (14,923 words with a lexicon of 637 words) were obtained. An example of two user turns from the dialogues is given in Figure 1 (see the *Original sentence*).

### 2.1. Labeling 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 labeling of our system is the division into three levels [7]. 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, 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. We focus on the second level labels (*frames*), which are specific to the task and represent the type of message supplied by the user. Table 1 shows the 16 frame classes, along with their frequencies.

The third level is also specific to the task and takes into account the data given in the sentence. Each frame has a set of slots which have to be filled to make a query or which are filled

Table 1: The 16 frame classes and their frequencies given as percentages of the total number of user turns in the overall corpus.

Frame class	%
Affirmation	26.75
Departure_time	18.27
New_data	13.16
Price	12.29
Closing	10.07
Return_departure_time	5.30
Rejection	4.34
Arrival_time	3.57
Train_type	3.37
Confirmation	1.73
Not_understood	0.63
Trip_length	0.24
Return_price	0.19
Return_train_type	0.05
Return_departure_time	0.05

by the retrieved data after the query. The specific data which fills the slots is known as *cases*. This level takes into account the slots which are filled by the specific data present in the segment, or the slots being used to generate the segment corresponding to an answer. To complete this level, it is necessary to analyze the words in the turn and to identify the case corresponding to each word. Examples of cases for this task are: Origin, Destination, Departure\_time, Train\_type, Price...

An example of the three-level labeling for some user turns is given in Figure 1. We will center our interest on the second level of the labeling, which is used to guide the understanding process. Note that each user turn can be labeled with more than one frame label (as in the second example of Figure 1), which allows a better specification of the meaning of the user turn, but it makes the classification and segmentation processes harder (see Sections 3 and 4).

## 2.2. Lexicon and codification of the user turns

For classification and understanding purposes, we are concerned with the semantics of the words present in the user turn of a dialogue, but not with the morphological forms of the words themselves. Thus, in order to reduce the size of the input lexicon, we decided to use categories and lemmas:

1. General categories: city names, cardinal and ordinal numbers, days of the week, months.
2. Task-specific categories: departure and arrival city names, train types.
3. Lemmas: verbs in infinitive, nouns in singular and without articles, adjectives in singular and without gender.

In this way, we reduced the size of the lexicon from 637 to 311 words.

## 3. Multiclass classification using Neural Networks

Our goal was to classify a user turn given in natural language in a specific class or classes of frames. We used a multiple a posteriori approach to classification by using Multilayer Perceptrons (MLPs) [9]. Other dialogue systems perform dialogue acts classification with other methods, such as  $n$ -gram models [10].

Table 2: Partition of the dataset.

	Total	Uniclass	Multiclass
Training	1,071 (80%)	692 (65%)	379 (35%)
Test	268 (20%)	175 (65%)	93 (35%)

### 3.1. Input and output for the MLP

We think that for this task the sequential structure of the sentence is not fundamental to classifying the type of frame.<sup>1</sup> For that reason, the words of a sentence were all encoded with a local coding: the input of the MLP is formed by 120 units, one for each word of the lexicon.<sup>2</sup> When the word appears in the sentence, its corresponding unit is set to 1, otherwise, its unit is set to 0.

Each user turn could be labeled with more than one label (as in the second example in Figure 1). Thus, the desired outputs for each training sample were set to 1 for those (one or more) classes that were correct and 0 for the remainder. We discarded the turns that were labeled with a class that had a frequency lower than five (a total of 1,338 user turns were selected), which comprised only the 10 most frequent classes shown in Table 1.

### 3.2. Multiclass classification rule

As we desire to test multiple outputs (a user turn can have more than one dialogue act label associated to it), after training the MLP with multiple desired classes, an input pattern can be classified in the classes  $I^*$  with a posteriori probability above a threshold  $\mathcal{T}$ :

$$I^* = \{i \in C \mid \Pr(i|x) \geq \mathcal{T}\} \approx \{i \in C \mid g_i(x, \omega) \geq \mathcal{T}\},$$

where  $g_i(x, \omega)$  is the  $i$ -th output of the MLP given the input pattern  $x$  and the set of parameters of the MLP  $\omega$ . The set of classes  $C$  are the 10 most frequent labels defined in Table 1. The threshold  $\mathcal{T}$  is also learnt in the training process.

### 3.3. Experiments

The dataset (1,338 user turns after discarding the less-frequent sentences) was randomly split into a training set (80% of the user turns) and a test set (20% of the user turns). The partition of the data, along with the relative frequency of uniclass and multiclass samples, are shown in Table 2.

The training of the MLPs was carried out using the neural net software package ‘‘SNNS: Stuttgart Neural Network Simulator’’ [11]. In order to successfully use Neural Networks as classifiers, a number of considerations had to be taken into account, such as the network topology, the training algorithm, and the selection of the parameters of the algorithm [9, 11, 12]. Proofs were conducted using different network topologies of increasing number of weights: a hidden layer with 2 units, two hidden layers of 2 units each, two hidden layers of 4 and 2 units, a hidden layer with 4 units, etc. Several learning algorithms were also tested: the incremental version of the back-propagation algorithm (with and without momentum term) and the quickprop algorithm. The influence of their parameters such

<sup>1</sup>Nevertheless, the sequential structure of the sentence is essential in order to *segment* the sentence into slots to have a real understanding of the sentence.

<sup>2</sup>After the processes explained in 2.2, we reduced the size of the lexicon from 637 to 311 words. Then, we discarded those words with a frequency lower than five, obtaining a lexicon of 120 words. Note that sentences which contained those words are not eliminated from the corpus, only those words from the sentence are deleted.

<b>Original sentence:</b>	Quería saber los horarios del Euromed Barcelona–Valencia. <i>I would like to know the timetables of the Euromed train from Barcelona to Valencia.</i>
<b>1st level (speech act):</b>	Question
<b>2nd level (frames):</b>	Departure_time
<b>3rd level (cases):</b>	Departure_time (Origin: barcelona, Destination: valencia, Train_type: euromed)
<hr/>	
<b>Original sentence:</b>	Hola, buenos días. Me gustaría saber el precio y los horarios que hay para un billete de tren de Barcelona a La Coruña el 22 de diciembre, por favor. <i>Hello, good morning. I would like to know the price and timetables of a train from Barcelona to La Coruña for the 22nd of December, please.</i>
<b>1st level (speech act):</b>	Question
<b>2nd level (frames):</b>	Price, Departure_time
<b>3rd level (cases):</b>	Price (Origin: barcelona, Destination: la_coruña, Departure_time: 12-22-2002) Departure_time (Origin: barcelona, Destination: la_coruña, Departure_time: 12-22-2002)

Figure 1: Example of the three-level labeling for two user turns. The Spanish original sentence and its English translation are given.

as learning rate or momentum term was also studied. Random presentation of the training samples was used in the training process. In every case, a validation criterion (20% of the training data was randomly selected for validation) was used to stop the learning process and to select the best configuration.

The best result on the validation data was obtained using the MLP of one hidden layer of 64 units trained with the standard backpropagation algorithm and a value of learning rate equal to 0.4 and a momentum term equal to 0.2.

Once we had selected the best combination of topology, learning algorithm and parameters for the MLP, according to the classification error rate of the validation data, we proved the trained MLP with the test data, obtaining a percentage of errors equal to 5.2%. If we analyze this result considering the type of sample (uniclass or multiclass user turn), we get an error percentage of 1.1% for the uniclass samples and an error percentage of 12.9% for the multiclass user turns.

#### 4. HMMs in the understanding process

Once we have classified the user utterances in one or more of the above defined frames, the next task in the dialogue system consists of extracting the relevant information in order to fill the cases associated with each user turn. In order to achieve this, we perform an additional step, finding an adequate segmentation of the user turn, according to the semantic function of each word or sequence of words. Figure 2 provides an example that will clarify this point. We defined a total of 53 semantic functions [6], such as *m\_origen* (“*departure\_mark*”), *clase\_billete* (“*ticket\_class*”) or *fecha\_actual* (“*current\_date*”). The objective of this analysis phase is to find a correct segmentation according to this newly defined units, which will simplify the extraction of the required information.

This task is accomplished using HMMs, where each state corresponds to a semantic unit. In order to reduce the size of the vocabulary and trying to avoid the problem of underestimation of parameters, we used the corpus obtained after the tokenization and lemmatization (see 3). In this case, however, no elimination of infrequent words was done, as the (whole) sequence of words is important to achieve a correct segmentation. We restrict ourselves to a closed vocabulary task, by eliminating of the test set the user turns with words not appearing in the training set, which slightly reduces the size of the test set. Future work will include smoothing techniques to handle these cases.

The goal in our experimentation is to compare the efficiency of a global model and a set of models for each of the 10 most frequent frame types defined in Table 1. The emission probabilities for each state in the HMMs will be shared between models, to avoid the problem of underestimation. The difference between the models will therefore lie in the transition probabilities between the different states conforming the model. Here we must again face the problem of the multiclass user turns, as we have to combine the output of two different models, that is, we have to find an adequate combination of two different segmentations, each from a different type of frame.

In the training process we replicate each multiclass turn and use it to train each of the corresponding models. In the test phase, for a multiclass sentence we concatenate the models we detect in the classification phase. With this strategy we try to achieve an automatic division of a multiclass sentence in each of its constituting parts, each belonging to a different frame. This is a natural approach for many turns (e.g. “*Yes, what type of train is it?*”) but it is not so clear if this approach will be adequate for other turns, where such a clear division between the frames can not be found (e.g. “*I want information about timetable and prices for traveling from Barcelona to Vigo.*”).

##### 4.1. Experiments

As a measure of the correctness of the understanding process we use the word error rate<sup>3</sup> (WER) between the output segmentation and the correct one. It is worth noting, that this is a pessimistic measure because the final goal of this phase will be to fill in the corresponding cases, so the exact segmentation of the user turn is not always necessary, if the relevant information (city names, type of information requested, etc.) of each turn is correctly detected.

The global model yields a WER of 14.6%. This can be decomposed into 16.3% for the uniclass turns and 12.6% for the multiclass ones. When using specific models with the *exact* classification we improve the WER to 13.2%, obtaining 12.5% for uniclass turns and 13.9% for multiclass. We can observe that the performance of the specific models is significantly better when dealing with the uniclass turns, as the models are more specialized. In contrast the multiclass are better handled with the global model. An important factor for this behaviour are the above mentioned sentences, where a correct division between the frames can not be found.

<sup>3</sup>In our case the ‘word’ is the semantic unit.

Original sentence			
Necesito saber los horarios de trenes de León a Córdoba para el tercer domingo de agosto.			
<i>I need to know the timetable of the trains from León to Córdoba for the third Sunday of August.</i>			
Segmentation			
necesito saber:	consulta	<i>I need to know:</i>	<i>question</i>
los horarios de trenes:	<hora_s>	<i>the timetable of the trains:</i>	<time_d>
de:	m_origen	<i>from:</i>	<i>departure_m</i>
le'on:	ciudad_origen	<i>le'on:</i>	<i>departure_city</i>
a:	m_destino	<i>to:</i>	<i>goal_m</i>
c'ordoba:	ciudad_destino	<i>c'ordoba:</i>	<i>city_goal</i>
para el tercer:	fecha_relativa_s	<i>for the third:</i>	<i>relative_date_d</i>
domingo:	dia_semana_s	<i>Sunday:</i>	<i>week_day_d</i>
de agosto:	mes_s	<i>of August:</i>	<i>month_d</i>

Figure 2: An example of the segmentation of an user turn. The Spanish original sentence and its English translation are given.

Table 3: WER of the segmentation with the different models.

Model	Classification	Test	Uniclass	Multiclass
Global	—	14.6%	16.3%	12.6%
Specific	100%	13.17%	12.5%	13.9%
Specific	94.8%	14.4%	14.3%	14.5%

As expected, the WER of the specific model grows when including the classification system, due to the errors coming from this previous phase. We obtain 14.4% WER, 14.3% in the uniclass turns and 14.5% in the multiclass ones. This slightly improvement over the global model is not significant. It must be considered however, that our task has only a very limited amount of training samples and we think that with a larger training set, the models will be better estimated and the robustness of the system will improve to errors from the classification phase. These results are summarized in Table 3.

## 5. Conclusions

We have shown that connectionist classification is a successful approach for classifying a user turn given in natural language into a specific class or classes of frames. It can also be observed that stochastic models are also a good approximation for the understanding task. The use of specific models outperforms the general model in the case of exact classification. When we use our previous connectionist classification process similar results are obtained.

For many classes the error falls below 5% while in others we obtain a higher error. It must be taken into account that, in some classes very few training samples are available, so the models are underestimated. It could be interesting to detect the best set of specific models and apply only this reduced set together with the global one.

We also hope to improve the performance of the system by a combination of specific and global models: If the user turn is classified with a high level of confidence, we could use the specific understanding model and if it is not, we choose the global understanding model. Moreover, a combination of both types of models could be used to improve the understanding process in the dialogue system.

This approach will be specially useful when dealing with speech data, as this model is more robust to the errors coming from the speech recognition module.

## 6. References

- [1] CMU Communicator Spoken Dialog Toolkit (CSDTK). <http://www.speech.cs.cmu.edu/communicator/>.
- [2] L. Lamel, S. Rosset, J. L. Gauvain, S. Bennacef, M. Garnier-Rizet, and B. Prouts. The LIMSI Arise system. *Speech Communication*, 31(4):339–354, 2000.
- [3] J. Glass and E. Weinstein. Speech builder: facilitating Spoken Dialogue System Development. In *Proc. of the 7th European Conference on Speech Communications and Technology (Eurospeech'01)*, pages 1335–1338, 2001.
- [4] A. Bonafonte et al. Desarrollo de un sistema de diálogo oral en dominios restringidos. In *Primeras Jornadas de Tecnología del Habla*, Sevilla (Spain), 2000.
- [5] H. Bonneau-Maynard and F. Lefèvre. Investigating stochastic speech understanding. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'01)*, 2001.
- [6] Emilio Sanchis, Fernando García, Isabel Galiano, and Encarna Segarra. Applying dialogue constraints to the understanding process in a Dialogue System. In *Proc. of 5th International Conference on Text, Speech and Dialogue (TSD'02)*, Brno (Czech Republic), 2002.
- [7] C. Martínez, E. Sanchis, F. García, and P. Aibar. A Labelling Proposal to Annotate Dialogues. In *Proc. of the Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1577–1582, Las Palmas de Gran Canaria (Spain), May 2002.
- [8] H. Aust, M. Oerder, F. Seide, and V. Steinbiss. The Philips automatic train timetable information system. *Speech Communication*, 17:249–262, 1995.
- [9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *PDP: Computational models of cognition and perception*, I, pages 319–362. MIT Press, 1986.
- [10] Andreas Stolcke et al. Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics*, 26(3):339–373, 2000.
- [11] A. Zell et al. *SNNS: Stuttgart Neural Network Simulator. User Manual, Version 4.2*. Institute for Parallel and Distributed High Performance Systems, University of Stuttgart, Germany, 1998.
- [12] C. M. Bishop. *Neural networks for pattern recognition*. Oxford University Press, 1995.