A Simple Connectionist Approach to Language Understanding in a Dialogue System

María José Castro and Emilio Sanchis

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

Abstract. A contribution to the understanding module of a domain-specific dialogue system is presented in this work. The task consists of answering telephone queries about train timetables, prices and services for long distance trains in Spanish. In this system, the representation of the meaning of the user utterances is made by means of *frames*, which determine the type of communication of the user turn, and by their associated *cases*, which supply the data of the utterance.

We focus on the classification of a user turn given in natural language in a specific class of frame. We used multilayer perceptrons to classify a user turn as belonging to a frame class. This classification can help in the posterior processes of understanding and dialogue management.

1 Introduction

The construction of dialogue systems applied to limited domain information systems is an important objective in the area of human language technologies. The advance in the design and analysis of the different knowledge sources involved in a spoken dialogue system, such as speech processing, language modeling, language understanding, or speech synthesis, has led to the development of dialogue system prototypes. Some characteristics of these systems are telephone access, limited semantic domains and mixed initiative [1–3].

A contribution to the understanding module of the BASURDE dialogue system [4] is presented. The task consists of answering telephone queries about timetables, prices and services for long distance trains in Spanish. In this system, the representation of the meaning of the user utterances is made through *frames*, which determine the type of communication of the user turn, and with *cases*, which supply the data of the utterance. The understanding module gets the output of the speech recognizer (sequences of words) as input and supplies its output to the dialogue manager. In this work, we are restricted to dealing with text data, that is, the correct transcription of each utterance. The semantic representation is strongly related to the dialogue management. In our approach, the dialogue behavior is represented by means of a stochastic network of *dialogue acts*. Each dialogue act has three levels of information: the first level represents the general purpose of the turn, the second level represents the type of semantic message (the frame or frames), and the third level takes into account the data supplied in the turn.

In this work, we focus on the process of classification the user turn in terms of the second level of the dialogue act, that is, the identification of the frame or frames given in the turn. This classification will help us to determine the specific data supplied in the sentence in a later process.

2 Artificial Neural Networks for Language Understanding

Language understanding tasks have usually been based on symbolic architectures, which use explicit rules that operate on symbols [5]. In contrast, machine learning techniques for inferring structural models have also been applied to this field. Specifically, hidden Markov models and stochastic regular grammars have been successfully used in the understanding module of dialogue systems [6, 7].

Recently, artificial neural networks have been used in language understanding, but most of the connectionist language models implemented until now have had severe limitations. Understanding models have been limited to simple sentences with small lexica (see [8] for a revision of understanding and production neural network models).

We used multilayer perceptrons (MLPs) for simple language understanding. The number of input units was fixed by the size of the input lexicon (natural language of a restricted-semantic task). There was one output unit corresponding to each class to classify the sentences by their meaning.

3 The Dialogue Task

The final objective of this dialogue system is to build a prototype for information retrieval by telephone for Spanish nation-wide trains [4, 9]. Queries are restricted to timetables, prices and services for long distance trains. Several other European dialogue projects [1, 10, 11] selected the same task.

A corpus of 200 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). A total of 215 dialogues were acquired using the Wizard of Oz technique. From these dialogues, a total of 1,460 user turns (14,902 words) were obtained. An example of two user turns is given in Figure 1 (see the *Original sentence*).

3.1 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. If the number of labels is too high the models will be underestimated because of the sparseness of the training samples. On the other hand, if we define a set of just a few labels only general purposes of the turn can be modeled.

The main feature of the proposed labeling is the division into three levels [12]. The first level, called *speech act*, is general for all the possible tasks. The second and third level, called *frames* and *cases*, respectively, are specific to the working task and give the semantic representation.the labeling is general enough to be applied to other tasks and specific enough to cover all the possible situations in the dialogue.

First level: *speech act*

The first level takes into account the intention of the segment (i.e., the dialogue behavior) and has a unique value. For this level, we define the following values, which are common to every task:

> *Opening, Closing, Undefined, Not understood, Waiting, Consult, Acceptance, Rejection, Question, Confirmation, Answer.*

Second level: *frames*

The second level is specific to each task and represents the type of message supplied by the user. This information is organized in *frames.* A total of 16 different classes of frames were defined for this task:

> *Departure_time, Return_departure_time, Arrival_time, Return_arrival_time, Price, Return_price, Length_of_trip, Train_type, Return_train_type, Services, Confirmation, Not_understood, Affirmation, Rejection, Closing, New_data.*

Third level: *cases*

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 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: *Origen, 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 (see the second example of Figure 1).

Original sentence:	Quería saber los horarios del Euromed Barcelona–Valencia. [I would like to know the timetables of the Euromed train from Barcelona to Valencia.]
1st level (speech act):	*Question*
2nd level (frames):	*Departure_time*
3rd level (cases):	*Departure_time* (*Origen*: barcelona, *Destination*: valencia, *Train_type*: euromed)

Original sentence:	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. [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.]
1st level (speech act):	*Question*
2nd level (frames):	*Price*, *Departure_time*
3rd level (cases):	*Price* (*Origen*: barcelona, *Destination*: la_coruña, *Departure_time*: 12-22-2002) *Departure_time* (*Origen*: barcelona, *Destination*: la_coruña, *Departure_time*: 12-22-2002)

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

3.2 Lexicon and codification of the sentences

For classification 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. Finally, we discarded those words with a frequency lower than five, obtaining a lexicon of 138 words. Note that sentences which contained those words are not eliminated from the corpus, only those words from the sentence are deleted. An example of the preprocessing of the original sentences is illustrated in Figure 2 (see *Original sentence* and *Preprocessed sentence*).

We think that for this task the sequential structure of the sentence is not fundamental to classifying the type of frame.[1] For that reason, the words of a sentence were all encoded with a local coding: the input of the MLP is formed by 138 units, one for each word of the lexicon. When the word appears in the sentence, its corresponding unit is set to 1, otherwise, its unit is set to 0. An example is given in Figure 2 (see from *Original sentence* to *Input local codification*).

3.3 Extended frames and multiple frames

A total of 16 different frames were defined for the task (see Section 3.1). Each user turn can be labeled with more than one frame label (as in the second example of Figures 1 and 2). We wanted to perform two types of classification experiments: a maximum a posteriori approach and a multiple a posteriori approach.

For the strict maximum a posteriori approach to classification, only one class was desired for each turn. To do so, we extended the frames classes, defining a new class for each different combination of classes: if a given turn is labeled with the classes "*Price*" and "*Departure_time*", a new class is defined as "*Price&Departure_time*". Finally, we discarded those turns labeled with a class with a frequency lower than five (a total of 1,338 user turns were selected), obtaining 28 frame classes (11 original frame classes and 17 extended frame classes). For each training sample, the corresponding output unit to the frame class is set to 1.

For the multiple a posteriori approach to classification, the desired outputs for each training sample are set to 1 for those (one or more) frame classes that are correct and 0 for the remainder. As we wanted to compare both approaches to classification (extended and multiple frames), the same data (1,338 user turns) which comprised only 11 original frame classes was used.

An example of codification of the frames as *extended frames* or *multiple frames* is illustrated in Figure 2.

4 The Classification Problem

In this work, we focus on the classification of a user turn given in natural language (categorized and leximized) in a specific class of frame. Multilayer perceptrons are the most common artificial neural networks used for classification. For this purpose, the number of output units is defined as the number of classes, C, and the input layer must hold the input patterns. Each unit in the (first) hidden layer forms a hyperplane in the pattern space; boundaries between classes can be approximated by hyperplanes. If a sigmoid activation function is used, MLPs can form smooth decision boundaries which are suitable to perform classification tasks [13]. The activation level of an output unit can be interpreted as an approximation of the a posteriori probability that the input pattern belongs to

[1] 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.

Original sentence:	Quería saber los horarios del Euromed Barcelona–Valencia. [I would like to know the timetables of the Euromed train from Barcelona to Valencia.]
Preprocessed sentence:	querer saber horario del tipo_tren nom_ciudad_origen nom_ciudad_destino [want know timetable of train_type from_city_name to_city_name]
▷*Input local codification:*	0000000000000000000000000010000000000000000000000100000000000000000000000000000000000110000000000000000000001000000100000000000000100000000000000 (7 active input units)
2nd level (frames):	*Departure_time*
Extended frame:	*Departure_time*
▷*Output codification:*	0000000000000001000000000000 (one of 28 classes)
Multiple frames:	*Departure_time*
▷*Output codification:*	00000100000 (one of 11 classes)

Original sentence:	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. [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.]
Preprocessed sentence:	hola bueno d'ia me gustar saber precio y horario que haber para billete de tren nom_ciudad_origen nom_ciudad_destino numero de nom_mes por_favor [hello good morning like know price and timetable of train from_city_name to_city_name for date of month_name please]
▷*Input local codification:*	00000000001100000000001010000000000000000011001010000000000000001000000000000001111000100000001001000100000001000000000000000001000000100 (20 active inputs units)
2nd level (frames):	*Price*, *Departure_time*
Extended frame:	*Price&Departure_time*
▷*Output codification:*	0000000000000000000000001000 (one of 28 classes)
Multiple frames:	*Price*, *Departure_time*
▷*Output codification:*	00000100010 (two of 11 classes)

Fig. 2. Example of the codification of two user turns (codification of the preprocessed sentence and codification of the type of frames) for both extended frames and multiple frames. The Spanish original sentence and its English translation are given.

the corresponding class. Therefore, an input pattern can be classified in the class $i^\star$ with maximum a posteriori probability:

$$i^\star = \operatorname*{argmax}_{i \in C} \Pr(i|x) \approx \operatorname*{argmax}_{i \in C} g_i(x, \omega)\,, \tag{1}$$

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, ω. The set of classes C are the 28 extended frames.

On the other hand, we desired to test multiple outputs (if only the original frames are used, a user turn can be labeled with more than one frame). To perform this type of experiment, after training the MLP with multiple desired classes, an input pattern can be classified in the classes $I^\star$ with a posteriori probability above a threshold $\mathcal{T}$:

$$I^\star = \{i \in C' \mid \Pr(i|x) \ \geq \mathcal{T}\} \approx \{i \in C' \mid g_i(x, \omega) \ \geq \mathcal{T}\}\,, \tag{2}$$

where, as before, $g_i(x, \omega)$ is the i-th output of the MLP given the input pattern, x, and the set of parameters of the MLP, ω. The set of classes C' are the 11 simple frame classes.

5 Experiments

We used multilayer perceptrons to classify a user turn (codified as explained in 3.2) as belonging to a unique frame class (*extended frames*) or as belonging to a set of classes (*multiple frames*). The number of input units was fixed by the size of the lexicon of the sentences (138 words). There was one output unit corresponding to each frame class (28 classes for the extended frame experiments and 11 classes for the multiple frame experiments).

The dataset (1,338 user turns) was randomly splitted into training (80%) and test (20%) sets.

5.1 Training the Artificial Neural Networks

The training of the MLPs was carried out using the neural net software package "SNNS: Sttutgart Neural Network Simulator" [14]. In order to successfully use neural networks, 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 [13–15]. Tests 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 backpropagation algorithm (with and without momentum term) and the quickprop algorithm. The influence of their parameters was also studied. Different combinations of learning rate (LR) and momentum term (MT), as well as different values of the maximum growth parameter (MG) for the quickprop algorithm, were proved. 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.

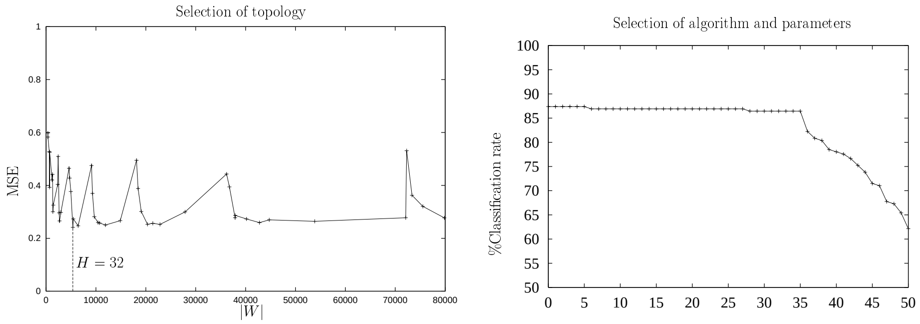

Fig. 3. Extended frame experiment. **a)** Mean square error (MSE) of the validation data with different MLPs of increasing number of weights. $|W|$ is the number of weights of each MLP. **b)** Percentage of validation user turns correctly classified with MLPs of one hidden layer of 32 units trained with different algorithms and parameters. Results are ordered from the best to the worst performance.

5.2 Selecting the best configuration of MLP

Extended frame experiments

We trained different MLPs of increasing number of weights using the standard backpropagation algorithm (with a sigmoid activation function and a learning rate equal to 0.2), selecting the best topology according to the mean square error (MSE) of the validation data (see Figure 3a). The minimum MSE of the validation data was achieved with an MLP of one hidden layer of 32 units.

We followed our experimentation with MLPs of this topology, training MLPs with several algorithms: the incremental version of the backpropagation algorithm (with and without momentum term) and the quickprop algorithm. Different combinations of learning rate (LR = 0.05, 0.1, 0.2, 0.3, 0.4, 0.5) and momentum term (MT = 0.1, 0.2, 0.3, 0.4, 0.5) as well as different values of the maximum growth parameter (MG = 1.75, 2) for the quickprop algorithm were proved. The performance on the validation data of each trained net is shown in Figure 3b. The best result on the validation data was obtained with the MLP trained with the standard backpropagation algorithm (LR = 0.3).[2]

Multiple frame experiments

The same scheme was followed to train MLPs with multiple outputs: different MLPs of increasing number of weights using the standard backpropagation algorithm (with a sigmoid activation function and a learning rate equal to 0.2) were trained and the best topology according to the MSE of the validation data was

[2] The same performance was achieved with six different configurations; we decided to select the configuration with the lowest MSE.

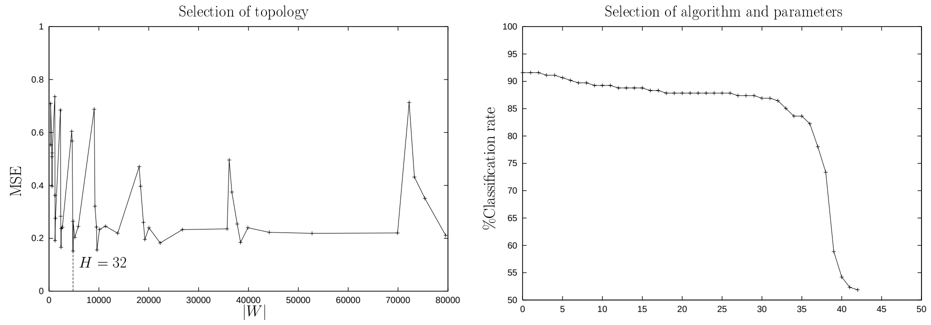

Fig. 4. Multiple frame experiment. **a)** Mean square error (MSE) of the validation data with different MLPs of increasing number of weights. $|W|$ is the number of weights of each MLP. **b)** Percentage of validation user turns correctly classified with MLPs of one hidden layer of 32 units trained with different algorithms and parameters. Results are ordered from the best to the worst performance.

selected (see Figure 4a). As in the previous experiment, the minimum MSE of the validation data was achieved with an MLP of one hidden layer of 32 units.

We followed our experimentation with MLPs of this topology, training MLPs with several algorithms (same proofs as before). The performance on the validation data of each trained MLP is shown in Figure 4b. The highest classification rate of the validation data was obtained with the MLP trained with the backpropagation with momentum (LR=0.4 and MT=0.3).[3] In this type of experiment, the threshold $\mathcal{T}$ was also fixed using the validation data.

5.3 Final experiment: Testing the best MLPs

Once we had selected the best combination of topology, learning algorithm and parameters for the MLPs of both types of experiments, according to the classification rate of the validation data, we proved the trained MLP with the test data, obtaining a percentage of classification equal to 83.96% for the extended frame approach and a percentage equal to 92.54% for the multiple frame approach. We think that the multiple frame experiment achieves better performance due to the fact that the number of training samples is very low and the data is better employed with this approach.

6 Conclusions

The results obtained show that, with the correct transcription of each utterance (text data is used for the experiments), using a connectionist approach to language understanding is effective for classifying the user turn according the type

[3] The same performance was achieved with three different configurations; we decided to select the configuration with the lowest MSE.

of frames. This automatic process will be helpful to the understanding module of the dialogue system: firstly, the user turn, in terms of natural language, is classified into a frame class or several frame classes; secondly, a specific understanding model for each type of frame is used to segment and fill the cases of each frame. This could be specially useful when we deal with speech data (with errors from the speech recognition module) instead of written data.

References

1. 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.
2. J. Glass and E. Weinstein. Speech builder: facilitating spoken dialogue system development. In *Proceedings of Eurospeech'01*, volume 1, pages 1335–1338, 2001.
3. CMU Communicator Spoken Dialog Toolkit (CSDTK). http://www.speech.cs.cmu.edu/communicator/.
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. S. K. Bennacef, H. Bonneau-Maynard, J. L. Gauvain, L. Lamel, and W. Minker. A Spoken Language System for Information Retrieval. In *Proceeedings of the 3th International Conference in Spoken Language Processing (ICSLP'94)*, pages 1271–1274, Yokohama (Japan), 1994.
6. H. Bonneau-Maynard and F. Lefèvre. Investigating stochastic speech understanding. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'01)*, 2001.
7. Emilio Sanchis, Fernando García, Isabel Galiano, and Encarna Segarra. Applying dialogue constraints to the understanding process in a Dialogue System. In *Proceedings of fifth International Conference on Text, Speech and Dialogue (TSD'02)*, Brno (Czech Republic), 2002.
8. Douglas L. T. Rohde. *A Connectionist Model of Sentence Comprehension and Production.* PhD thesis, Computer Science Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 2002.
9. A. Bonafonte et al. Desarrollo de un sistema de diálogo para habla espontánea en un dominio semántico restringido. Technical report, Spanish project from the Comisión Interministerial de Ciencia y Tecnología (TIC98-0423-CO6), 1998–2001. URL: `gps-tsc.upc.es/veu/basurde/`.
10. H. Aust, M. Oerder, F. Seide, and V. Steinbiss. The Philips automatic train timetable information system. *Speech Communication*, 17:249–262, 1995.
11. L. F. Lamel et al. The LIMSI RailTail System: Field trail of a telephone service for rail travel information. *Speech Communication*, 23:67–82, 1997.
12. C. Martínez, E. Sanchis, F. García, and P. Aibar. A labeling proposal to annotate dialogues. In *Proceedings of third International Conference on Language Resources and Evaluation (LREC'02)*, 2002.
13. 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.
14. 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.
15. C. M. Bishop. *Neural networks for pattern recognition.* Oxford University Press, 1995.