Uniclass and Multiclass Connectionist Classification of Dialogue Acts*

María José Castro[1], David Vilar[1], Emilio Sanchis[1], and Pablo Aibar[2]

[1] Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València. Camí de Vera s/n, 46022 València, Spain
[2] Departament de Llenguatges i Sistemes Informàtics
Universitat Jaume I de Castelló. E-12071 Castelló, Spain
{mcastro,dvilar,esanchis}@dsic.upv.es aibar@lsi.uji.es

Abstract. Classification problems are traditionally focused on uniclass samples, that is, each sample of the training and test sets has one unique label, which is the target of the classification. In many real life applications, however, this is only a rough simplification and one must consider some techniques for the more general multiclass classification problem, where each sample can have more than one label, as it happens in our task. In the understanding module of a domain-specific dialogue system for answering telephone queries about train information in Spanish which we are developing, a user turn can belong to more than one type of frame. In this paper, we discuss general approaches to the multiclass classification problem and show how these techniques can be applied by using connectionist classifiers. Experimentation with the data of the dialogue system shows the inherent difficulty of the problem and the effectiveness of the different methods are compared.

1 Introduction

In many real pattern recognition tasks, it is convenient to perform a previous classification of the objects in order to treat them in a specific way. For instance, if language models can be learnt for specific sub-domains of a task, better performance can be achieved in an automatic speech recognition/understanding system. The aim of this work is to propose some classification techniques in order to improve the understanding process of a dialogue system.

The task of our dialogue system consists of answering telephone queries about train timetables, prices and services for long distance trains in Spanish. The understanding module gets the output of the speech recognizer (sequences of words) as input and supplies its output to the dialogue manager. 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

* This work has been partially supported by the Spanish CICYT under contracts TIC2000-0664-C02-01 and TIC2002-04103-C03-03.

message (the frame or frames), and the third level takes into account the data supplied in the turn.

We focus our attention 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. This classification will help us to determine the data supplied in the sentence in a later process, where depending on the output of the classifier, one or more specific understanding models are applied. Our previous work on this same topic can be found in [1, 2].

Dealing with this frame detection problem, we encountered the problem of the multiclass classification as a natural issue in our system. A user can ask in the same utterance about timetables and prices of a train, for example, and these are two of the categories we have defined. This poses an interesting problem, as most of the classification problems and solutions up to now have focused exclusively on the uniclass classification problem, and few have dealt with this kind of generalization.

2 The Uniclass Classification Problem

Uniclass classification problems involve finding a definition for an unknown function $k^\star(\mathbf{x})$ whose range is a discrete set containing $|\mathcal{C}|$ values (i.e., $|\mathcal{C}|$ "classes" of the set of classes $\mathcal{C} = \{c^{(1)}, c^{(2)}, \ldots, c^{(|\mathcal{C}|)}\}$). The definition is acquired by studying collections of training samples of the form

$$\{(\mathbf{x}_n, c_n)\}_{n=1}^{N}\,, \quad c_n \in \mathcal{C}\,, \tag{1}$$

where $\mathbf{x}_n$ is the n-th sample and c_n is its corresponding class label.

For example, in handwritten digit recognition, the function $k^\star$ maps each handwritten digit to one of $|\mathcal{C}| = 10$ classes. The Bayes decision rule for minimizing the probability of error is to assign the class with maximum a posteriori probability to the sample $\mathbf{x}$:

$$k^\star(\mathbf{x}) = \operatorname*{argmax}_{k \in \mathcal{C}} \Pr(k|\mathbf{x})\,. \tag{2}$$

Uniclass Classification using Neural Networks Multilayer perceptrons (MLPs) 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, $|\mathcal{C}|$, and the input layer must hold the input samples. 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 [3].

For uniclass samples, the activation level of an output unit can be interpreted as an approximation of the a posteriori probability that the input sample belongs to the corresponding class. Therefore, given an input sample $\mathbf{x}$, the trained MLP computes $g_k(\mathbf{x}, \omega)$ (the k-th output of the MLP with parameters ω given the input sample $\mathbf{x}$) which is an approximation of the a posteriori probability

$\Pr(k|\mathbf{x})$. Thus, for MLP classifiers we can use the uniclass classification rule as in equation (2):

$$k^\star(\mathbf{x}) = \operatorname*{argmax}_{k\in\mathcal{C}} \Pr(k|\mathbf{x}) \approx \operatorname*{argmax}_{k\in\mathcal{C}} g_k(\mathbf{x},\omega)\,. \tag{3}$$

3 The Multiclass Classification Problem

In contrast to the uniclass classification problem, in other real-world learning tasks the unknown function $k^\star$ can take more than one value from the set of classes $\mathcal{C}$. For example, in many important document classification tasks, documents may each be associated with multiple class labels [4, 5]. A similar example is found in our classification problem of dialogue acts: a user turn can be labeled with more than one frame label. In this case, the training set is composed of pairs of the form[3]

$$\{(\mathbf{x}_n, C_n)\}_{n=1}^N\,, \quad C_n \subseteq \mathcal{C}\,. \tag{4}$$

There are two common approaches to this problem of classification of objects associated with multiple class labels.[4] The first is to use specialized solutions like the accumulated posterior probability approach described in the next section. The second is to build a binary classifier for each class as explained afterwards.

3.1 Accumulated Posterior Probability

In a traditional (uniclass) classification system, given an estimation of the a posteriori probabilities $\Pr(k|\mathbf{x})$, we can think of a classification as "better estimated" if the probability of the destination class is above some threshold (i.e., the classification of a sample $\mathbf{x}$ as belonging to class k is better estimated if $\Pr(k|\mathbf{x}) = 0.9$ than if it is only 0.4). A generalization of this principle can be applied to the multiclass approximation problem.

We can consider that we have correctly classified a sample only if the *sum* of the a posteriori probabilities of the assigned classes is above some threshold $\mathcal{T}$. Let us define this concept more formally. Suppose we have an ordering (permutation) $(k^{(1)}, k^{(2)}, \ldots, k^{(|\mathcal{C}|)}$ of the set $\mathcal{C}$ for a sample $\mathbf{x}$, such that

$$\Pr(k^{(i)}|\mathbf{x}) \geq \Pr(k^{(i+1)}|\mathbf{x}) \quad \forall 1 \leq i < |\mathcal{C}|\,. \tag{5}$$

We define the "accumulated posterior probability" for the sample $\mathbf{x}$ as

$$\Pr_{\mathbf{x}}(j) = \sum_{i=1}^{j} \Pr(k^{(i)}|\mathbf{x}) \qquad 1 \leq j \leq |\mathcal{C}|\,. \tag{6}$$

[3] The uniclass classification problem is a special case in which $|C_n| = 1$ for all samples.

[4] In certain practical situations, the amount of possible multiclass labels is limited due to the nature of the task. For instance, if we know that the only possible appearing multiple labels can be $\{c^{(i)}, c^{(j)}\}$ and $\{c^{(i)}, c^{(k)}\}$ we do not need to consider all the possible combinations of the initial labels. In such situations we can handle this task as an uniclass classification problem with the extended set of labels $\hat{\mathcal{C}}$ defined as a subset of $\mathcal{P}(\mathcal{C})$.

Using the above equation, we classify the sample $\mathbf{x}$ in n classes, being n the smallest number such that

$$\Pr_{\mathbf{x}}(n) \geq \mathcal{T}\,, \tag{7}$$

where the threshold $\mathcal{T}$ must also be learnt automatically in the training process. The set of classification labels for the sample $\mathbf{x}$ is simply

$$K^{\star}(\mathbf{x}) = \{k^{(1)}, \ldots, k^{(n)}\}\,. \tag{8}$$

Accumulated Probability using MLPs We can apply this approach using neural networks by modifying slightly equation (7). As the output of the output layer is an *approximation* of the a posteriori probabilities, it is possible that the sum exceeds the value of 1, so a more suitable estimation would be[5]

$$|1 - \Pr_{\mathbf{x}}(n)| \leq \mathcal{S}\,, \tag{9}$$

where the accumulated posterior probabilities $\Pr_{\mathbf{x}}(j)$ are computed as in equation (6) by approximating the posterior probabilities with an MLP of $|\mathcal{C}|$ outputs

$$\Pr_{\mathbf{x}}(j) = \sum_{i=1}^{j} \Pr(k^{(i)}|\mathbf{x}) \approx \sum_{i=1}^{j} g_i(\mathbf{x}, \omega) \qquad 1 \leq j \leq |\mathcal{C}|\,. \tag{10}$$

The outputs $g_i(\mathbf{x}, \omega)$ of the trained MLP are also ordered according (5). During the training phase, the desired outputs for the sample $\mathbf{x}$ are the “true” posterior probabilities of each class.[6]

3.2 Binary Classifiers

Another possibility is to treat each class as a separate binary classification problem (as in [6–8]). Each such problem answers the question, whether a sample should be assigned to a particular class or not.

For $C \subseteq \mathcal{C}$, let us define $C[c]$ for $c \in \mathcal{C}$ to be:

$$C[c] = \begin{cases} \textsf{true}, & \text{if } c \in C\,; \\ \textsf{false}, & \text{if } c \notin C\,. \end{cases} \tag{11}$$

A natural reduction of the multiclass classification problem is to map each multiclass sample $(\mathbf{x}, C)$ to $|\mathcal{C}|$ binary-labeled samples of the form $(\langle \mathbf{x}, c\rangle, C[c])$ for all $c \in \mathcal{C}$; that is, each sample is formally a pair, $\langle \mathbf{x}, c\rangle$, and the associated binary label, $C[c]$. In other words, we can think of each observed class set C as specifying $|\mathcal{C}|$ binary labels (depending on whether a class c is or not included in C), and we can then apply uniclass classification to this new problem. For

[5] Note the different interpretation of the threshold value in equations (7) and (9). In the first one, $\mathcal{T}$ represents the probability mass that we must have for correctly classifying a sample, whereas in the second one $\mathcal{S}$ is a measure of the distance to the “ideal” classification with a posteriori probability value of 1.

[6] Nevertheless, a simplification is assumed: as the true posterior probabilities usually cannot be known, we consider all the classes of a training sample equally probable.

instance, if a given training pair $(\mathbf{x}, C)$ is labeled with the classes $c^{(i)}$ and $c^{(j)}$, $(\mathbf{x}, \{c^{(i)}, c^{(j)}\})$, then $|\mathcal{C}|$ binary-labeled samples are defined as $(\langle \mathbf{x}, c^{(i)} \rangle, \mathsf{true})$, $(\langle \mathbf{x}, c^{(j)} \rangle, \mathsf{true})$ and $(\langle \mathbf{x}, c \rangle, \mathsf{false})$ for the rest of classes $c \in \mathcal{C}$.

Then a set of binary classifiers is trained, one for each class. The ith classifier is trained to discriminate between the ith class and the rest of the classes and the resulting classification rule is

$$K^{\star}(\mathbf{x}) = \{k \in \mathcal{C} \mid \Pr(k|\mathbf{x}) \geq \mathcal{T}\}\,, \tag{12}$$

being $\mathcal{T}$ a threshold which must also be learnt.

Binary Classification Using MLPs Let $(\omega_1, \ldots, \omega_{|\mathcal{C}|})$ be the MLP classifiers trained as in the uniclass case. Furthermore, let $g(\mathbf{x}, \omega_i)$ be the output of the ith MLP classifier when given an input sample $\mathbf{x}$. New samples are classified by setting the predicted class or classes to be the index of the classifiers attaining the highest posterior probability,

$$K^{\star}(\mathbf{x}) = \{k \in \mathcal{C} \mid \Pr(k|\mathbf{x}) \geq \mathcal{T}\} \approx \{k \in \mathcal{C} \mid g(\mathbf{x}, \omega_k) \geq \mathcal{T}\}\,. \tag{13}$$

An alternative approach is to assign a binary string of length $|\mathcal{C}|$ to each class $c \in \mathcal{C}$ or set of classes $C \subseteq \mathcal{C}$. During training for a pattern from classes $c^{(i)}$ and $c^{(j)}$, for example, the desired outputs of these binary functions are specified by the corresponding units for classes i and j. With MLPs, these binary functions can be implemented by the $|\mathcal{C}|$ output units of a single network.

In this case, the multiclass classification rule is redefined as: an input sample $\mathbf{x}$ can be classified in the classes $K^{\star}(\mathbf{x})$ with a posteriori probability above a threshold $\mathcal{T}$:

$$K^{\star}(\mathbf{x}) = \{k \in \mathcal{C} \mid \Pr(k|\mathbf{x}) \geq \mathcal{T}\} \approx \{k \in \mathcal{C} \mid g_k(\mathbf{x}, \omega) \geq \mathcal{T}\}\,, \tag{14}$$

being $g_k(\mathbf{x}, \omega)$ the k-th output of an MLP classifier with parameters ω given the input sample $\mathbf{x}$.

4 The Dialogue Task

The final objective of our dialogue system is to build a prototype for information retrieval by telephone for Spanish nation-wide trains [9]. Queries are restricted to timetables, prices and services for long distance trains. 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. The average length of a user turn is 10.27 words. All the utterances we used for our experiments were transcribed by humans from the actual spoken responses.

The turns of the dialogue were labelled in terms of three levels [10]. An example is given in Figure 1. We focus our attention on the most frequent second level labels, which are Affirmation, Departure_time, New_data, Price, Closing, Return_departure_time, Rejection, Arrival_time, Train_type, Confirmation. Note that each user turn can be labeled with more than one frame label[7] (as in the example).

[7] In related works of dialogue act classification [11], a hand-segmentation of the user turns was needed in order to have sentence-level units (utterances) which corre-

Original sentence	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 (Origin: barcelona, Destination: la_coruña, Departure_time: 12/22/2003) Departure_time (Origin: barcelona, Destination: la_coruña, Departure_time: 12/22/2003)

Fig. 1. Example of the three-level labeling for a multiclass user turn. Only the English translation of the original sentence is given.

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. In this way, 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.

We think that for this task the sequential structure of the sentence is not fundamental to classifying the type of frame.[8] For that reason, the words of the preprocessed sentence were all encoded with a local coding: a 120-dimensional bit-vector, one position 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.

4.1 Codification of the Frame Classes

For the uniclass problem we used the usual "1-of-$|\mathcal{C}|$" coding, the desired output for each training sample is set to 1 for the one frame class that is correct and 0 for the remainder. The codification in the multiclass problem is different for each approach:

Binary classification with $|\mathcal{C}|$ MLPs The target of the training sample is 1 if the sample belongs to the class of the MLP classifier, and 0 if not.

Binary classification with one MLP The target of the training sample is coded with a $|\mathcal{C}|$-dimensional vector: the desired outputs for each training sample (x_n, C_n) are set to 1 for those (one or more) frame classes that are correct and 0 for the remainder.

Accumulated posterior probability The target of the training sample is coded with a $|\mathcal{C}|$-dimensional vector: the desired outputs for each training sample (x_n, C_n) are set to $1/|C_n|$ for those (one or more) frame classes that are correct and 0 for the remainder.

sponded to a unique dialogue act. The relation between user turns and utterances was also not one-to-one: a single user turn can contain multiple utterances, and utternaces can span more than one turn. After the hand-segmentation process, each utterance unit was identified with a single dialogue act label.

[8] Nevertheless, the sequential structure of the sentence is essential in order to segment the user turn into slots to have a real understanding of it.

5 Experiments

The dataset is composed of 1 338 user turns after discarding the sentences labeled with the less-frequent frame classes. We have decided to split the corpus in two datasets, the first one containing only the uniclass turns (867 samples) and the complete one, which comprises uniclass and multiclass turns (1 338 samples). For each type of experiment, the dataset was randomly split (but we guarantee that each frame class is represented in the training and test set) so that about 80% of the user turns are used for training and the rest for testing.

5.1 Training the Neural Networks

With any neural network algorithm, several parameters must be chosen by the user. For the MLPs, we must select the network topology and their initialization, the training algorithm and their parameters and the stopping criteria [3, 12, 13]. We selected all the parameters to optimize performance on a validation set: the training set is subdivided into a subtraining set and a validation set (20% of the training data). While training on the subtraining set, we observed generalization performance on the validation set (measured as the mean square error) to determine the optimal setting of configuration and the best point at which to stop training. The thresholds $\mathcal{S}$ and $\mathcal{T}$ of the different multiclass classification rules were also learnt in the training process: we performed classification with the optimal configuration of MLP on the patterns of the validation set, proving several values of the thresholds and keeping the best one.

5.2 UC and MC Experiments

Table 1 shows the selected topology and the classification rate for each of the experiments. For the UC experiment, we used only the uniclass user turns (867 samples). For the MC experiments, we consider a sample as correctly classified if the set of the original frame classes is detected. That is, if a user turn is labeled with two frame classes, only and exactly those classes should be detected.

In the Accumulated Probability case, when applying classification rule (9) with a threshold $\mathcal{S}$ close to 0, that is, when the accumulated probability is close to 1, the error rate was very poor, misclassifying (nearly) all the multiclass samples. By analyzing the MLP outputs, we observed that when one or more classes are detected, each of the corresponding output values are close to one. Therefore, the MLP with a sigmoid activation function is unable to learn the true probability distribution across the whole set of classes. Due to this fact, we decided to apply the classification rule given in equation (14).

6 Discussion and Conclusions

This work is an attempt to show the differences between uniclass and multiclass classification problems applied to detecting dialogue acts in a dialogue system. We experimentally compare three connectionist approaches to this end: using accumulated posterior probability, binary multiple classifiers and one extended

Table 1. Classification error rates for the UC and MC experiments.

Experiment	Topology	Total	Uniclass	Multiclass
UC experiment	120-64-64-10	9.14	9.14	—
MC experiments				
Binary classifiers with $\lvert\mathcal{C}\rvert$ MLPs	120-8-1	17.91	13.71	25.80
Binary classifiers with 1 MLP	120-32-32-10	11.19	7.43	18.28
Accumulated Probability	120-32-16-10	14.55	8.57	25.81

binary classifier. The results clearly shows that: firstly, multiclass classification is much harder than uniclass classification and, secondly, the best performance is obtained using one extended binary classifier.

On the other hand, the results obtained for classifying dialogue acts also show that using a connectionist approach is effective for classifying the user turn according to the type 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.

References

1. M. J. Castro and E. Sanchis. A Simple Connectionist Approach to Language Understanding in a Dialogue System. In *Advances in Artificial Intelligence*, pages 664-673. Springer-Verlag, 2002.
2. E. Sanchis and M. J. Castro. Dialogue Act Connectionist Detection in a Spoken Dialogue System. In *Soft Computing Systems. Design, Management and Applications*, pages 644-651. IOS Press, 2002.
3. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *PDP: Computational models of cognition and perception, I*, pages 319-362. MIT Press, 1986.
4. A. K. McCallum. Multi-Label Text Classification with a Mixture Model Trained by EM. In *Proc. NIPS'99*, 1999.
5. R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135-168, 2000.
6. Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2):69-90, 1999.
7. T. Joachims. Text categorization with Support Vector Machines: Learning with many relevant features. In *Proc. ECML'98*, pages 137-142. Springer Verlag, 1998.
8. K. Nigam et al. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103-134, 2000.
9. A. Bonafonte et al. Desarrollo de un sistema de diálogo oral en dominios restringidos. In *Primeras Jornadas de Tecnología del Habla*, Sevilla (Spain), 2000.
10. C. Martínez et al. A Labelling Proposal to Annotate Dialogues. In *Proc. LREC 2002*, vol. V, pages 1577–1582, Las Palmas de Gran Canaria (Spain), 2002.
11. A. Stolcke et al. Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics*, 26(3):339-373, 2000.
12. A. Zell et al. *SNNS: Stuttgart Neural Network Simulator. User Manual, Version 4.2.* Univ. of Stuttgart, Germany, 1998.
13. C. M. Bishop. *Neural networks for pattern recognition.* Oxford Univ. Press, 1995.