Recurrent Neural Learning for Classifying Spoken Utterances *

Sheila Garfield and Stefan Wermter
Centre for Hybrid Intelligent Systems,
University of Sunderland, Sunderland, SR6 0DD, United Kingdom
e-mail: {stefan.wermter,sheila.garfield}@sunderland.ac.uk
http://www.his.sunderland.ac.uk/

Abstract

For telecommunications companies or banks, etc processing spontaneous lanaguage in helpdesk scenarios is important for automatic telephone interactions. However, the problem of understanding spontaneous spoken language is difficult. Learning techniques such as neural networks have the ability to learn in a robust manner. Recurrent networks have been used in neurocognitive or psycholinguistically oriented approaches of language processing. Here they are examined for their potential in a difficult spoken language classification task. This paper describes an approach to learning classification of recorded operator assistance telephone utterances. We explore simple recurrent networks using a large, unique telecommunication corpus of spontaneous spoken language. Performance of the network indicates that a simple recurrent network is quite useful for learning classification of spontaneous spoken language in a robust manner, which may lead to their use in helpdesk call routing.

1 Introduction

Language is not only extremely complex and powerful but also ambiguous and potentially ill-formed [1] and because of this and other factors such as acoustics, speaking style, parsing coverage or understanding gaps [9] errors can arise in the recognition of speech input. Additionally, spontaneous speech also includes artifacts such as filled pauses and partial words. Therefore, spoken dialogue systems must be able to deal with these as well as other discourse phenomena such as anaphora and ellipsis, and ungrammatical queries [9, 4].

Thus, a general research question arises whether neural network methods can be used effectively to classify telephone utterances. In this paper we describe an approach to the classification of recorded operator assistance telephone utterances. In particular, we explore simple recurrent networks and describe experiments using a large, unique corpus of spontaneous spoken language in a real-world scenario.

2 Description of the Helpdesk Corpus

Our task is to learn to classify incoming telephone utterances into a set of call classes. The corpus used in this task is from transcriptions of 8,441 recorded operator assistance telephone calls [5]. As shown by the following example utterances the callers use a wide range of language to express their problem, enquiry or to request assistance [6]. These can range from descriptive narrative requests for help to more simple direct requests for services:

*The authors thank Mark Farrell and David Attwater of BTexact Technologies for their helpful discussions.

1. "*can I book alarm call please*"
2. "*can I have a taxi number please*"
3. "*uh yeah hi um I'm in on my own and I've just got a phone call and I picked it up and it was just gone and it happened yesterday as well at the same time*"

2.1 Call Transcription

A corpus based on transcriptions of the first utterances of callers to the operator service is the focus of this investigation. The calls were recorded at all times of the day and night and therefore the calls provide a representative selection of call traffic to the operator service. Following analysis of the utterances a number of call classes shown in Figure 1, primary move types and request types [5] were identified. The *call class* and the *primary move* type differ in their focus. The *call class* is more concrete and is associated directly with the service the caller requires. The *primary move* is more abstract and reflects what the caller is trying to achieve or what the intention of the caller is believed to be. The *request* type identifies the range of language used by the caller to talk to the operator. Is the request for service expressed in explicit terms or is it an implicit request expressed by the use of free language [5].

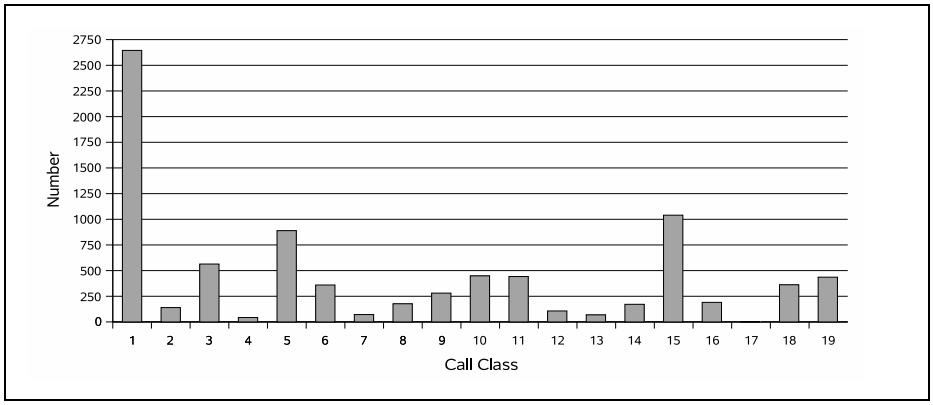

Figure 1: Number in each call class within the corpus

As shown by the example in Table 1, the *call class* is the classification of the utterance and is associated with the service that the caller requires. The *primary move* is a subset of the first utterance, it is like a dialogue act and gives an indication of which dialogue act is likely to follow the current utterance. The *request* type identifies whether the request is explicit or not [2].

Call	*Call Class*	*Primary Move*	*Request Type*
could I um the er international directory enquiries please	international directory enquiries	connect	explicit

Table 1: Example call from the corpus

The corpus is divided into nine separate call sets: eight sets contain 1,000 utterances each and one set contains 441 utterances. These sets were combined using a rotational approach to create the training and test sets. At each rotation one call set was excluded and used for the test set, for example, call sets 1-8 were used to create a training set and call set 9 was used for the test set. The average length of an utterance in both the training and test set is 22.12 words.

Depending on the call set rotation each call class is represented in the training and test set. An illustrative example is given in Table 2, however not all call classes are shown.

8,441 utterances Total of 8,000 utterances in Training set: Call sets 1-8 Total of 441 utterances in Test set: Call set 9							
Call Class:	class 1	class 2	class 3	class 4	class 5	class 6	class n
in train set:	2476	132	541	43	849	338	...
in test set:	169	8	23	0	40	22	...
Total	2645	140	564	43	889	360	...

Table 2: Breakdown of utterances in training and test sets from rotation set 1-8. Note: For illustration purposes not all classes are shown

The concept of entropy provides a way of measuring the complexity of a classification problem and is associated with uncertainty or information in relation to the task of selecting one or more items from a set of items. Thus entropy is a measure of how many bits of information we are trying to extract in a task and it gives an indication of how certain or uncertain we are about the outcome of the selection [3]. A low entropy value indicates diversity while a high value indicates more similarity and therefore greater difficulty in differentiating between items. In this task the number of call classes is 19 and the call sets under investigation have an entropy of 3.4.

$$entropy = \sum_{i=1}^{19} P(c_i) log_2(P(c_i)) \tag{1}$$

2.2 Semantic Vector Representation

The words in a lexicon are represented using a semantic vector [13]. Each element in the vector represents the frequency of a particular word occurring in a call class. The number of calls in a class can vary substantially. Therefore the frequency of a word w in class c_i is *normalized* according to the number of calls in c_i (2). A *value* $v(w, c_i)$ is computed for each element of the vector as the *normalized* frequency of occurrences of word w in semantic class c_i, divided by the *normalized* frequency of occurrences of word w in all classes (3). That is:

$$Normalized\ frequency\ of\ w\ in\ c_i = \frac{Frequency\ of\ w\ in\ c_i}{Number\ of\ calls\ in\ c_i} \tag{2}$$

where:

$$v(w, c_i) = \frac{Normalized\ frequency\ of\ w\ in\ c_i}{\sum_j Normalized\ frequency\ for\ w\ in\ c_j},\ j \in \{1, \cdots n\} \tag{3}$$

Each call class is represented in the semantic vector. An illustrative example is given in Table 3, however not all call classes are shown. As can be seen in the illustrative example, domain-independent words like 'to' and 'please' have fairly even distributions while domain-dependent words like 'book' and 'alarm' have more specific preferences.

3 Learning and Experiments

A simple recurrent network with input, output, hidden and context layers was used for the experiments. Supervised learning techniques were used for training [7, 12]. The input to a

Word	Call Class						
	class 1	class 2	class 3	class 4	class 5	class 6	class n
I'D	0.01	0.15	0.09	0.04	0.02	0.02	...
LIKE	0.04	0.10	0.06	0.07	0.03	0.01	...
TO	0.08	0.02	0.01	0.03	0.03	0.02	...
BOOK	0.01	0.00	**0.56**	0.24	0.01	0.02	...
AN	0.05	0.00	0.33	0.19	0.01	0.03	...
ALARM	0.00	0.00	**0.56**	0.40	0.00	0.00	...
CALL	0.01	0.18	0.25	0.19	0.00	0.01	...
PLEASE	0.03	0.11	0.11	0.04	0.06	0.06	...

Table 3: Example of semantic vectors. Note: For illustration purposes not all classes are shown.

hidden layer L_n is constrained by the underlying layer L_{n-1} as well as the incremental context layer C_n. The activation of a unit $L_{ni}(t)$ at time t is computed on the basis of the weighted activation of the units in the previous layer $L_{(n-1)i}(t)$ and the units in the current context of this layer $C_{ni}(t)$ limited by the logistic function f.

$$L_{ni}(t) = f(\sum_k w_{ki} L_{(n-1)i}(t) + \sum_l w_{li} C_{ni}(t)) \tag{4}$$

The result is a simple form of recurrence and a method of training networks to perform sequential tasks over time. This means that both the input and the state of the network at the previous time step contribute to the network output; retained events from the past can be used in current computations. Therefore in response to simple static input the network can produce complex time-varying outputs which is important when generating complex behaviour. Consequently, this simple form of recurrence can prove beneficial in terms of network performance and provide the facility for temporal processing.

3.1 Training Environment

All utterances from the training and test set are presented to the network in one epoch, or cycle of training through all training samples, and the weights are adjusted at the end of each utterance. Each call class has one input in the input layer. Utterances are presented to the network one word at a time as a sequence of word input and category output representations, one pair for each word, during training and test. Each input receives the value of $v(w, c_i)$, where c_i denotes the particular class associated with the input. In the output layer each unit corresponds to a particular call class. Prior to the presentation of each new sequence the context layers are cleared and initialised with 0 values. The target value of the output unit that represents the desired call class is set to 1 and the target values of all other output units are set to 0. At the end of the sequence if the activation value of the output unit for the required call class is higher than 0.5 the utterance is defined as being *classified* to that particular call class. This output classification is used to compute the recall and precision values for each call class as well as the overall rates for the training and test sets [8].

The network was trained for 135 epochs on the training transcribed utterances using a fixed momentum term and a changing learning rate. The initial learning rate was 0.01, this changed at 45 epochs to 0.006 and then again at 90 epochs to 0.001. The results for this series of experiments are shown in Table 4.

3.2 Recall, Precision and F-Score

The performance of the trained network in terms of recall, precision and F-score on each of the call sets is shown in Table 4. Recall and precision are common evaluation metrics [11]. The F-score [10] is a combination of the precision and recall rates and is a method for calculating a value without bias, that is, without favouring either recall or precision. There is a difference of 4.27% and 5.84% between the highest and the lowest test recall and precision rates respectively.

	Training Set			Test Set		
	Recall	Precision	F-Score	Recall	Precision	F-Score
Rotation 1	73.83%	88.40%	80.46	76.87%	89.21%	82.58
Rotation 2	73.74%	89.80%	80.98	73.70%	87.12%	79.85
Rotation 3	75.45%	91.15%	82.56	73.60%	90.64%	81.24
Rotation 4	75.50%	89.77%	82.02	75.90%	90.25%	82.46
Rotation 5	73.62%	91.68%	81.66	73.90%	92.96%	82.34
Rotation 6	73.85%	91.19%	81.61	73.70%	89.99%	81.03
Rotation 7	74.79%	90.24%	81.79	74.50%	89.98%	81.51
Rotation 8	74.63%	90.06%	81.62	72.60%	90.75%	81.62
Rotation 9	75.85%	90.45%	82.51	74.60%	88.70%	81.04

Table 4: Overall results for the simple recurrent network using semantic vectors

4 Analysis of Neural Network Performance

The focus of this work is the classification of utterances to call classes using a simple recurrent network. In general, the recall and precision rates for the simple recurrent network are quite high given the number of call classes available against which each utterance can be classified and the ill-formed input. The simple recurrent network achieved an average test recall performance of over 74% of all utterances. This result is calculated based on the overall performance figures for the simple recurrent network shown in Table 4.

In other related work on text classification [13] news titles were used to classify a news story as one of 8 categories. A news title contains on average about 8 words. As a comparison, the average length of the first caller utterance is 22.12 words and is subject to more ambiguity and noise. On the other hand, the size of the vocabulary used in the text classification task was larger than that used for our classification of call classes. The performance of the simple recurrent network is significant when this factor is taken into consideration because a larger vocabulary provides more opportunity for the network to learn and therefore generalise on unseen examples. While on an 8 category *text* classification task we reached about 90%, in this study presented in this paper here for a much more ill-formed *spoken language* classification task and 19 categories we reached above 72% (recall) and 87% (precision) for unseen examples.

5 Conclusions and Future Work

In conclusion the main aim of this research is to identify indicators about useful simple recurrent architectures that can be developed in the context of a larger hybrid symbolic/neural system for helpdesk automation. A description has been given of a recurrent neural architecture, the underlying principles and an initial evaluation of the approach for classifying the call classes of operator assistance telephone utterances. The main result from this work is that the performance of the simple recurrent network is noteworthy when factors such as noise in the utterance

and the number of classes are taken into consideration. This work makes a novel contribution to the field of robust learning classification using a large, unique corpus of spontaneous spoken language. From the perspective of connectionist networks it has been demonstrated that a connectionist network, in particular a simple recurrent network, can be used under *real-world* constraints for spoken language analysis.

It is possible that useful information is lost if only the best performing classifier is selected. One solution is to build a number of classifiers and combine them. Therefore to continue this research Support Vector Machines (SVM) have been investigated. The aim is to identify a non-neural component that could be combined with simple recurrent networks to form an ensemble architecture thereby taking advantage of information that is held in the classifiers that would not normally be used because their performance was not the best overall.

6 Acknowledgments

This research has been partially supported by the University of Sunderland and BTexact Technologies under agreement ML846657.

References

[1] Steven Abney. Statistical methods and linguistics. In Judith Klavans and Philip Resnik, editors, *The Balancing Act*. MIT Press, Cambridge, MA, 1996.

[2] D. Attwater. Oasis first utterance version 2.10 release note. *100CS:QUAL:REL:005 Version 1*, 2000.

[3] Eugene Charniak. *Statistical Language Learning*. MIT Press, Cambridge, MA, USA, 1993.

[4] H. Clark. Speaking in time. *Speech Communication*, 36:5–13, 2002.

[5] P.J. Durston and J.J.K. Kuo et al. OASIS natural language call steering trial. In *Proceedings of Eurospeech, Vol2*, pages 1323–1326, 2001.

[6] M. Edgington, D. Attwater, and P. Durston. OASIS - a framework for spoken language call steering. In *Proceedings of Eurospeech '99*, 1999.

[7] J. L. Elman, E. A. Bates, M. H. Johnson, A. Karmiloff-Smith, D. Parisi, and K. Plunkett. *Rethinking Innateness*. MIT Press, Cambridge, MA, 1996.

[8] S. Garfield and S. Wermter. Recurrent neural learning for helpdesk call routing. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 296–301, Madrid, Spain, 2002.

[9] J.R. Glass. Challenges for spoken dialogue systems. In *Proceedings of IEEE ASRU Workshop*, Keystone, CO, 1999.

[10] C.J. Van Rijsbergen. *Information Retrieval. 2nd edition.* Butterworths, London, 1979.

[11] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw Hill, New York, 1983.

[12] S. Wermter. *Hybrid Connectionist Natural Language Processing*. Chapman and Hall, Thomson International, London, UK, 1995.

[13] S. Wermter, C. Panchev, and G. Arevian. Hybrid neural plausibility networks for news agents. In *Proceedings of the National Conference on Artificial Intelligence*, pages 93–98, Orlando, USA, 1999.