SPEECH UTTERANCE CATEGORISATION GIVEN *ONE* TRAINING UTTERANCE PER CATEGORY

Amparo Albalate[1], **David Suendermann**[2]

[1] Department of Information Technology, University of Ulm, amparo.albalate@uni-ulm.de
[2] SpeechCycle, Inc., New York City, USA, david@speechcycle.com

Keywords: Automated Agents, Classification.

Abstract

In this paper, we address the categorisation of speech utterances within the scenario of technical support automated agents given only one labelled utterance per category . The categorisation algorithm maps input utterances into bag-of-word vectors and then applies feature extraction based on soft word clustering. We analyse two feature extraction schemes: pole-based overlapping clustering (PoBOC) and a combination of PoBOC with Fuzzy c-medoids. For the categorisation at the utterance level, we use the Nearest Neighbour (NN) approach. Finally, we evaluate the proposed methods on a test corpus with more than 3000 utterances recorded in a commercial dialog system.

1 Introduction

Automated agents for technical support are spoken language dialog systems (SLDSs) that automatically resolve customer care issues over the phone in a similar way as human agents do [1]. The principal motivation of applying automated solutions to technical support is to palliate some of the problems commonly related to technical support call centers, namely the long waiting time users might experience, and the cost of training and maintaining a large base of human agents.

In an SLDS, the interaction modality defines the way how a user is prompted for his input and, consequently, in which form the user input can be expected. The optimum interaction modality for a successful system design is strictly related to the application area where the SLDS is to be deployed. Given the intrinsic characteristics of the problem solving domain, in particular, the high diversity of call reasons to be handled, technical support applications demand a *natural language* interaction modality. Open prompts are presented to the users (e.g. *Please briefly describe the reason for your call*) who are thus allowed to describe the experienced problems in their own words. Consequently, one of the first system actions in the dialog process is to perform a diagnosis of the underlying problem, or symptom, provided the caller utterance. Once the problem is recognised, the automated agent carries out a sequence of troubleshooting steps to resolve the caller problem.

Natural language modality means, however, that we face an *open* range of possible user utterances reporting a single problem. The natural language understanding task is to identify which problem out of a given symptom set a new utterance belongs to. This task is referred to as *speech utterance categorisation* and is typically performed using a statistical classifier.

Standard classifiers are *supervised*–they are trained on a significant amount of utterances given their problem categories. These categories are determined by means of manual annotation. One of the major difficulties associated with the use of supervised classifiers is the extensive human effort of labelling large amounts of training data by hand. Moreover, human annotators are humans: They may annotate similar or even identical utterances with different symptoms–depending on their mood, time of the day, etc. Annotation instructions can never cover the variety of expressions callers use. This leads to gray areas–vague utterances potentially are put into many different places. Also, when several annotators work together, there is an intrinsic lack of correlation. E.g. if annotators label more than 90% of a common utterance set with identical labels, from the authors' experience, this is a very good result. This, however, means that there is uncertainty in 10% of the utterances!

These issues can result in suboptimal training sets which may affect utterance categorisation performance. Furthermore, the time factor involved in manual compilation of training data may considerably interfere with the system adaptability to possible changes in the application domain.

However, while the compilation of labelled corpora for training becomes very costly, the collection of unlabelled data is rather inexpensive. This fact motivates the development of classifiers, as the one proposed in this publication, which exploit the availability of large unlabelled data sets in order to reduce the usage of labelled samples. This is possible to the extent that appropriate analysis can be conducted on the unlabelled data, which derive sufficient additional information to compensate for the lack of labels.

In this paper, we trigger a basically unsupervised categorisation algorithm by means of a single labelled utterance per category providing suggestions on the number and very gross locations of the reference categories. Our strategy focuses on automatically expanding the semantic coverage of this minimally labelled set through a fuzzy clustering of words into semantic classes performed on the unlabelled utterance corpus.

In the remaining sections of this paper, we describe the modules of our categorisation algorithm and, in particular, a feature extraction method based on fuzzy word clustering.

Finally, we evaluate the algorithm on a manually labelled test corpus and discuss results and future directions of this research.

2 Utterance categorisation with three modules

Automated speech utterance categorisation was introduced about ten years ago to allow the caller to use unconstraint natural speech to express the call reason [5]. At the same time, speech utterance categorisation was capable of distinguishing many more reasons than directed dialogs, common at that time, could ever handle.

There is a number of approaches to statistical speech utterance categorisation (see [4]) which, however, are based on a significant amount of manually labelled training data. Being provided only a single training utterance per category requires special modifications of the categorisation procedure as discussed in the following.

Figure 1 provides an overview about the main components of our algorithm which consists of three modules: preprocessing, feature extraction and categorisation.

2.1 Preprocessing

The preprocessing module applies morphological analysis, stop word filtering, and bag-of-word representation.

First, a morphological analyser [7] is applied to reduce the surface word forms in utterances into their corresponding lemmas.

As a next step, stop words are eliminated from the lemmas, as they are judged irrelevant for the categorisation. Examples are the lemmas *a, the, be, for*. In this work, we used the SMART stop word list [2] with small modifications: in particular, we deleted confirmation terms (*yes* and *no*) from the list, whereas words typical for spontaneous speech (*eh, ehm, uh*) were treated as stop words.

The categoriser's vocabulary is then defined as the set of distinct lemmas in the preprocessed utterances: $W = (w_1, \ldots, w_D)$. In this work, the vocabulary dimension is $D = 1614$ lemmas.

Finally, the lemmas for each utterance are combined as a *bag-of-word*. I.e., each utterance is represented by a D-dimensional vector whose binary elements represent the presence/absence of the respective vocabulary element in the current utterance: $BW = (b_1, \ldots, b_D)$.

2.2 Feature extraction

To extract the set of salient features for the classification algorithm, we apply fuzzy classification of the D vocabulary words into a set of D' automatically inferred word senses. In contrast to hard clustering methods where each input pattern is unequivocally assigned to one cluster, in soft clustering, input patterns are associated to all output clusters through a membership matrix, M. Hard word clustering is useful to extract synonym terms, whereas, to deal also with the existence of polysemous words, fuzzy approaches are more appropriate. Details of the fuzzy feature extraction method are discussed in Section 3.

2.3 Categorisation

Given the previous mapping of the vocabulary terms into semantic classes, a new feature vector F reflects to what degree each semantic class is represented in the original utterance. This has been achieved through a matrix product of the bag-of-words vector by the membership matrix M calculated in the feature extraction step.

$$F = BW_{(1xD)} \cdot M_{(DxD')} \tag{1}$$

In order to categorise an utterance represented by a feature vector into one of the N symptom categories, we apply the Nearest Neighbour (NN) algorithm. This algorithm requires a codebook of prototypes, which in this work is composed of one labelled utterance per category. The utterances is then assigned to the category of the closest prototype. The closeness of an input utterances to the prototypes is calculated according to the cosine score $S_{\cos}$ between two feature vectors, F_a and F_b:

$$S_{\cos}(F_a, F_b) = \frac{F_a \cdot F_b'}{|F_a| \cdot |F_b|} \tag{2}$$

3 Fuzzy word clustering

The objective of the fuzzy word clustering used for feature extraction is a fuzzy mapping of words into semantic classes. In this section, we describe the algorithms used for extracting the membership matrix M that represents this association (cf. Section 2.2).

3.1 Term vector

A frequently reported problem to word clustering is the adequate representation of word lemmas in vector structures, so that mathematical (dis)similarity metrics applied to term vectors can reflect the terms' semantic relationships [8]. In the following, we also use *term* as a synonym for *lemma*. We applied a second-order term co-occurrence criterion [9] for detecting word-semantic proximities:

> Two words are similar to the degree that they co-occur with similar words.

Consequently, each vocabulary term w_i is represented in a D dimensional vector

$$W_i = (c_{i1}, \ldots, c_{iD}) \tag{3}$$

wherein the constituents c_{ij} denote the co-occurrence of the terms w_i and w_j, normalised with respect to the total sum of co-occurrences for the term w_i:

$$c_{ij} = \frac{nc_{ij}}{\sum\limits_{k \neq i} nc_{ik}}. \tag{4}$$

Here, nc_{ik} stands for the total number of times that w_i and w_k co-occur.

In the following, we analyse two term classification algorithms: pole-based overlapping clustering (PoBOC) and a combination of PoBOC with fuzzy C-medoids.

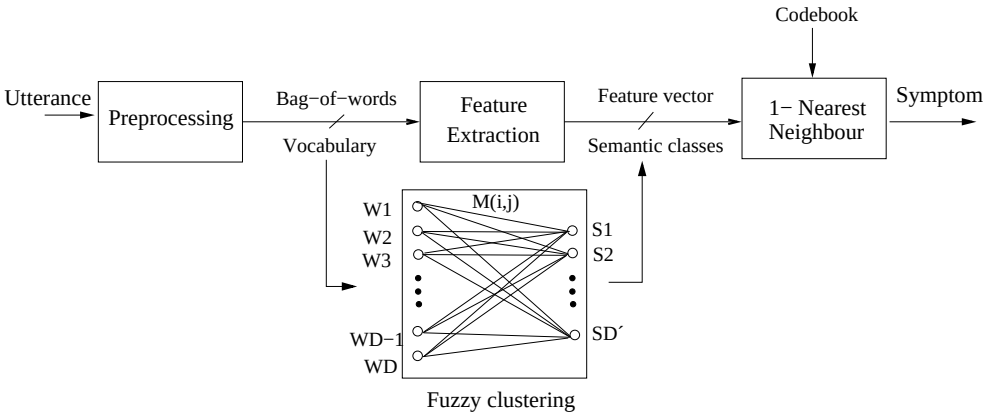

Figure 1: Utterance categorisation based on fuzzy term clustering.

3.2 Pole-based overlapping clustering

In the PoBOC algorithm [3], two kinds of patterns are differentiated: poles and residuals.

Poles are homogeneous clusters which are as far as possible from each other. In contrast, residuals are outlier patterns that fall into regions between two or more poles. The elements in the poles represent monosemous terms, whereas the residual patterns can be seen as terms with multiple related meanings (polysemous). The PoBOC algorithm is performed in two phases: (i) pole construction, and (ii) multiaffectation of outliers.

In the **pole construction** stage, the set of poles $\{P\} = \{P_1, \cdots, P_{D'}\}$ and outliers $\{R\}$ are identified and separated. Poles arise from certain terms with maximal separation inside a dissimilarity graph which are therefore known as the pole generators.

In the **multi-affectation** stage, the outliers' memberships to each pole in $\{P\}$ are computed. Finally, the term w_i is assigned a membership vector to each P_j pole as follows:

$$M_{ij} = \begin{cases} 1, & \text{if } w_i \in P_j \\ 1 - d_{av}(W_i, P_j)/d_{max} & \text{if } w_i \in \{R\} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where $d_{av}(W_i, P_j)$ denotes the average distance of the w_i word to all objects in P_j, and d_{max} refers to the maximum of the term dissimilarity matrix.

For computing the semantic dissimilarity of terms, experiments with both Euclidean and cosine distances[1] were carried out.

3.3 PoBOC with fuzzy C-medoids

The fuzzy C-medoids algorithm (FCMdd) [6] computes the fuzzy membership matrix M starting from an initial choice of cluster representatives or *medoids*. We initialise the algorithm with the D' pole generators ($C = D'$) obtained at the pole construction phase of the PoBOC scheme. The final solution for the membership matrix M is then reached through the iterative repetition of two steps: (i) (re)calculation of pattern memberships to the D' classes, and (ii) recomputation of the cluster medoids. The membership update of the term W_i to the j^{th} class is defined as:

$$M_{ij} = \frac{\left(\frac{1}{d(W_i, C_j)}\right)^{\frac{1}{m-1}}}{\sum_{k=1}^{C} \left(\frac{1}{d(W_i, C_k)}\right)^{\frac{1}{m-1}}} \quad (6)$$

denoting C_k, the k^{th} class medoid, $d(W_i, C_k)$, the dissimilarity between the term vector W_i and the medoid C_k, and m, a fuzzyfier factor, $m \in [1, \infty)$, denoting the smoothness of the clustering solution ($m = 2$ in this work). The procedure is iterated until either the updated cluster centroids remain the same, or a maximum number of iterations is reached.

4 Experiments

In order to evaluate the proposed soft word clustering methods for utterance classification, we compare the performance of an NN classifier directly applied to the bags-of-words vectors with that after performing feature extraction. As introduced in Section 1, this is done by comparing the output categories the proposed algorithm assigns to a number of test utterances with manually assigned categories thereof (the reference). If both categories coincide, the automatic categorisation is considered correct, otherwise it is counted as error. As overall accuracy, we define

$$\text{accuracy} = \frac{\text{\# correctly classified test utterances}}{\text{\# total utterances in test set}} \quad (7)$$

In the following, we describe the test corpus on which we evaluated the proposed algorithms. Then, we report on the experimental results and finally discuss the outcomes.

4.1 Corpus description

We used a corpus of 3,285 transcribed and annotated caller utterances gathered from user interactions of a commercial video troubleshooting agent. Example utteraces (categories) are:

- *Remote's not working* (Cable)
- *Internet was supposed to be scheduled at my home today* (Appointment)

[1] The cosine distance metric, D_{cos} is defined as the negative of the cosine score, $D_{cos} = 1 - S_{cos}$.

Table 1: Results of utterance categorisation experiments using several feature extraction techniques and distance measures

classifier	feature extraction	distance measure	accuracy
trivial	–	–	12.5%
NN	–	–	45%
NN	PoBOC	Euclidean	50%
NN	PoBOC	cosine	41%
NN	PoBOC + FCMdd	Euclidean	47%
NN	PoBOC + FCMdd	cosine	45.5%

- *I'm having Internet problems* (Internet)

The number of distinct categories in this corpus is $N = 28$. Most of the original utterances are composed of 1 to 10 words. After preprocessing, we have an average of 4.45 terms/utterance. The final vocabulary is composed of $D = 1614$ terms.

4.2 Results

Table 1 shows accuracies on the test set achieved by several configurations of the NN classifier: (i) no feature extraction (bag-of-word matching)[2], and (ii) feature extraction based on soft word clustering, using cosine and Euclidean distances between term vectors. As a standard of comparison, we also report the accuracy of a 'trivial' classifier which assigns the most frequent category to every utterance.

4.3 Discussion

Our experimental results show improvements of more than 10% relative using fuzzy word clustering for feature extraction (PoBOC with Euclidean distance) compared to the baseline which uses unmodified bag-of-word vectors as introduced in Section 2.1.
Applying the cosine distance produced consistently worse results than the Euclidean distance. This effect may be associated to the different numbers of extracted features generated in the pole construction phase of the PoBOC algorithm. With the Euclidean distance, a total number of $D' = 34$ semantic classes is inferred, whereas only $D' = 21$ clusters are detected with the cosine distance. This number is even lower than the number of symptom categories $N = 28$.
Although the best accuracy on the current scenario suggests that half of the utterances were misclassified, this number is higher than we would have expected being provided a single random example per category.
Human annotators, instead, are potentially given the whole world knowledge to categorise new utterances. Results with supervised utterance classification (Naïve Bayes algorithm) on the same type of data with more than 130,000 training utterances produced a performance of around 73% accuracy on a similar test set [4].

[2]The notion of bag-of-word matching can be also denoted as the inner product between bag-of-word vectors.

5 Conclusion

Being given only one sample utterance per category produces 50% correct classification results in our test scenario using the PoBOC feature extraction algorithm in conjunction with the Euclidean distance. Although this results may not sound very promising, considering the very high costs of producing training data for supervised utterance categorisation (our example referred to 130,000 transcribed and annotated utterances), it may be a practical solution for applications which require a rapid and cheap design.
In the future we aim at studying the use of confidence measures which allow for rejecting categorisations which most likely are wrong. This is to increase the ratio between correct and incorrect categorisations being one of the most important criteria in commercially deployed applications.

References

[1] K. Acomb, J. Bloom, K. Dayanidhi, P. Hunter, P. Krogh, E. Levin, and R. Pieraccini. Technical Support Dialog Systems: Issues, Problems, and Solutions. In *Proc. of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, Rochester, USA, 2007.

[2] C. Buckley. Implementation of the SMART information retrieval system. Technical report, Cornell University, Ithaca, USA, 1985.

[3] G. Cleuziou, L. Martin, and C. Vrain. PoBOC: An Overlapping Clustering Algorithm. Application to Rule-Based Classication and Textual Data. In *Proc. of the ECAI*, Valencia, Spain, 2004.

[4] K. Evanini, D. Suendermann, and R. Pieraccini. Call Classification for Automated Troubleshooting on Large Corpora. In *Proc. of the ASRU*, Kyoto, Japan, 2007.

[5] A. Gorin, G. Riccardi, and J. Wright. How May I Help You? *Speech Communication*, 23(1/2), 1997.

[6] R. Krishnapuram, A. Joshi, O. Nasraoui, , and L. Yi. Low-Complexity Fuzzy Relational Clustering Algorithms for Web Mining. *IEEE Trans. on Fuzzy Systems*, 9(4), 2001.

[7] G. Minnen, J. Carrol, and D. Pearce. Applied Morphological Processing of English. *Natural Language Engineering*, 7(3), 2001.

[8] C. A. Montgomery. A Vector Space Model for Automatic Indexing. *Communication of the ACM*, 18(11), 1975.

[9] J Picard. Finding Content-Bearing Terms using Term Similarities. In *Proc. of the EACL'99*, Bergen, Norway, 1999.