Unsupervised Categorisation Approaches for Technical Support Automated Agents

Amparo Albalate[1], *Dimitar Dimitrov*[1], *Roberto Pieraccini*[2]

[1]Institute of Information Technology, University of Ulm, Germany
[2]SpeechCycle, New York, USA
amparo.albalate@uni-ulm.de, dimitar.dimitrov@uni-ulm.de, roberto@speechcycle.com

Abstract

In this paper we describe an unsupervised approach for the automated categorisation of utterances into predefined categories of symptoms (or problems) within the framework of a technical support automated agent. The utterance classification is performed based on an iterative K-means clustering method. In order to improve the lower accuracy typical of unsupervised algorithms, we have analysed two different enhancements of the classification algorithm. The first method exploits the affinity among words by automatically extracting classes of semantically equivalent terms. The second approach consists of a disambiguation technique based on a new criterion to estimate the relevance of terms for the classification. An analysis of the results of an experimental evaluation performed on a corpus of 34848 utterances concludes the paper.
Index Terms: automated agents, automated categorisation, unsupervised learning, related words, term relevance.

1. Introduction

Technical support automated agents [1] are dialog applications specifically designed to resolve customer care issues over the phone in the same way as human agents do. These systems are nowadays adopted as an effective solution to alleviate common problems of technical support call centres, such as the long waiting time and poor service frequently experienced by the end user, the cost of training and maintaining a large base of human agents, and the scalability of the service.

Essentially, the function of automated agents for technical support applications (third generation automated dialog systems) consists in identifying the reason for the call (i.e. the symptom or problem) in order to guide the caller through a sequence of relevant troubleshooting steps, and eventually solve the problem. Given the high complexity of most technical support applications, in which the number of problems is large, *directed dialog* interaction designs (specific prompts with choice enumeration) become impractical for the call reason identification step. Instead, *natural language* designs are generally selected, which prompt the user in a more open manner. The system must then be able to interpret the user's description of the symptom to identify the corresponding problem. Based on the analysis of the user's utterances, an interactive dialogue is specifically directed to obtain additional information from the user which lead to the determination of the correct problem.

Statistical Spoken Language Understanding (SSLU) technology based on statistical classification algorithms is currently applied in automated agents to categorise users' utterances into one of a number of pre-defined problems. Statistical classifiers are generally based on *supervised* algorithms that can be trained on data sets previously annotated in order to automatically classify any further amount of unlabelled data with high accuracy.

However, the complexity of generating collections of manually annotated training data may be a limiting factor, especially for emerging services in which the list of predefined problems must be rapidly updated. This paper addresses the automated categorisation of utterances without, or with a minimal amount, of human supervision (labelling or annotation). To that goal, we propose a classification algorithm based on Unsupervised Learning (UL) models (e.g. data clustering). With a clustering algorithm, we aim at assigning individual utterances to a number of classes representing the possible symptoms, with no other input than the linguistic similarities among them. In particular, we propose two approaches to enhance the classification performance of unsupervised algorithms.

First, we analysed a feature extraction technique that relies on the semantic affinities among terms. In fact, because of the equivalence among different words (e.g synonymy), there are multiple ways of describing a problem or symptom. By identifying related terms, we assume that the clustering ability to extract semantics from utterances can be optimised. For instance, an unsupervised text categorisation method for the automatic creation of training sets has been already proposed in [2], which starts from a manually selected list of relevant keywords and synonyms for each topical category. In the proposed approach, we have implemented an algorithm that automatically identifies classes of semantically related words from a corpus of utterance transcripts. This method can provide certain advantages over manually constructed keyword thesaura, such as the possibility to capture equivalences of terms which are not apparently equivalent, but do show some statistical relation in a particular corpus.

Secondly, a new criterion to determine the discriminating power of words for the classification has been also proposed. It is essentially conceived for such cases in which the simultaneous co-occurrence of two or more relevant terms can lead to important classification ambiguities. Several measures of a word's "informativeness" have been generally applied in feature selection methods for the text categorisation task [3]. However, to our knowledge, the estimation of a word's relevance in preceeding work has always been done in a supervised manner, inasmuch the proposed formulations involve statistics of the terms distribution through categories. Rather, we have introduced a fully unsupervised criterion to estimate a word's informativeness which do not require previous assignments of utterances (and therefore terms) into categories.

The mentioned methods have been implemented based on a K-means clustering algorithm [4]. We have then evaluated their performance for unsupervised classification by comparing

the accuracy of the resulting classifier to the basic K-means.

The paper has been organised as follows: in Section 2 we introduce the proposed unsupervised learning approach for the classification of users' utterances. Section 3 describes a method to automatically extract semantic term classes. In Section 4 the disambiguation module is explained in detail. The evaluation methods and results are outlined in Section 5. Finally, we draw conclusions and state our future work on the topic in Section 6.

2. Unsupervised approach for symptom categorisation

The proposed unsupervised classification is based on a clustering of utterances according to their mutual linguistic similarity (Figure 1).

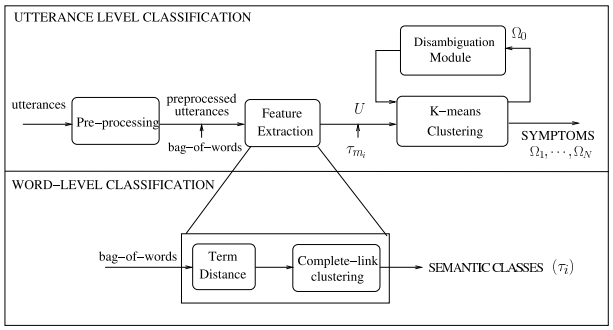

Figure 1: Unsupervised classification scheme

A *data pre-processing* stage has been applied to the utterance transcriptions, consisting of morphological filtering, stop-word removal, and word *bagging*. First, the morphological analyser for English developed within the PSET project [5] has been used in order to reduce the surface form of each word to its root form or lemma. Next, we have filtered out those root forms that belong to a list of stop words, similar to the standard SMART stop-word list [6]. Such list includes a set of ubiquitous terms which are therefore irrelevant for the classification purpose. Finally, each utterance is represented solely by the corresponding *bag-of-words*, defined as the set of unique root forms that result from the above described process (content words).

The *feature extraction* technique here implemented is based on an algorithm to identify classes of semantically equivalent terms from the bag-of-words corresponding to the utterance transcriptions. At this point, we can distinguish two levels of classification: The *word-level classification*, in which the terms in the bag-of-words are clustered into semantic classes τ, and the *utterance-level classification*, which maps individual utterances into problems.

For each word semantic class τ_i, we have selected the term with higher frequency of occurrence in the training corpus, τ_{m_i}. That term is used as the representative of the cluster.

After identifying similar words, we have replaced each content word by the corresponding word class representative, τ_{m_i}. This process enables us to express the original utterances as feature vectors $U = (u_1, \cdots, u_T)$, where T refers to the total number of salient features or term classes, and the components u_i are binary values denoting the presence or absence of each τ_{m_i} feature in the utterance.

A partitional *K-means* clustering algorithm has been subsequently used to classify utterance vectors into one problem among the predefined set of problem categories $\Omega = \{\Omega_1, \cdots, \Omega_M\}$, where M stands for the total number of problems. Therefore, a number of output classes $K = M$ has been specified. Further, the M initial cluster centres have been selected to coincide with exactly M utterances with different manual annotations. In this sense, the classification has been given a hint, i.e. one labelled utterance per category.

We have then applied a term-overlap-based similarity between pairs of utterances i and j, which can be formulated as the dot product of their feature vectors, U_i and U_j (Equation 1):

$$Sim(U_i, U_j) = U_i \cdot U_j = \sum_{k=1}^{T} u_{i_k} \cdot u_{j_k} \tag{1}$$

Each U_i vector is assigned to the cluster centres with maximum similarity. In the cases where several candidate centroids with maximum similarity are found, the utterance vectors are considered as *ambiguous* by the algorithm, and therefore rejected and put into a different cluster Ω_0. A *disambiguation module* has been especially devised for such cases, in order to perform a selection among candidate clusters and re-classify the Ω_0 patterns into one of the problem categories $(\Omega_i \neq \Omega_0)$.

3. Extraction of semantically equivalent terms

We describe here a method to automatically identify and group content terms that make reference to the same semantic concept. The basic assumption is that equivalent terms should be replaceable, and therefore co-occur with the same words. Thus the proposed proximity among words is based on the second-order term co-occurrence [7], i.e, the degree to which two different content words co-occur with similar terms.

A unigram model has been used for the construction of the word thesaurus. First, the co-occurrence among two words t_i and t_j is calculated as the total number of times that the terms appear in the same utterances, C_{ij}. The co-occurrence values are normalised $(\overline{C}_{ij})$ (Equation 2) in such a way that each t_i term vector $\overline{C}_i = (\overline{C}_{i1}, \cdots, \overline{C}_{iN})$ describes the probability density function of the i_{th} term co-occurrence with the N different terms in the bags-of-words.

$$\overline{C}_{ij} = \frac{C_{ij}}{\sum_{k=1}^{N} C_{ik}} \tag{2}$$

Next, the dissimilarity between two given terms t_i and t_j is estimated through the normalised Euclidean distance between the corresponding term vectors $\overline{C}_i$ and $\overline{C}_j$ (Equation 3).

$$d_{ij} = \sqrt{\sum_{k} \left(\frac{\overline{C}_{ik} - \overline{C}_{jk}}{min(C_{ik}, C_{jk})}\right)^2} \tag{3}$$

Finally, a hierarchical *complete link clustering* is used to group the terms into clusters according to a minimum-distance criterion. Under the assumption that equivalent words should belong to the same lexical categories, the algorithm has been further restricted so that the terms in each final cluster share identical Part-of-Speech (PoS) tags. To that end, the distance measure applied to the clustering algorithm has been redefined as shown in Equation 4.

$$d_{c_{ij}} = \begin{cases} d_{ij}, & if \quad L_i = L_j \\ \infty, & otherwise \end{cases} \tag{4}$$

Where L_i denotes the lexical category (PoS tag) of the t_i term. In particular, lexical categories of words have been extracted by parsing the initial utterances with the statistical parser developed by the Standford NLU group [8].

Table 1 shows some of the output semantic classes, sorted in increasing order according to the average distance among their member terms.

Table 1: *Some extracted semantic term classes, sorted in relevance order.*

Semantic term classes τ_i
speak talk
tech technical support customer service somebody someone person
operator human representative agent
security antivirus firewall virus protection suite software program c-d
webpage site website page web
code login password user name number
email mail outlook messenger
install hookup hook run purchase buy
schedule reschedule confirm cancel
remember forget
megabyte meg
cable converter t-v box channel screen picture sound signal line house installation
send receive reconnect

4. Disambiguation module

The i_{th} iteration of a K-means clustering algorithm is essentially performed in two steps: 1) recalculation of cluster centroids, and 2) reassignment of patterns into clusters by selecting the centroid with maximum similarity. In some cases, there are several candidate centroids and the clustering algorithm is not able to classify a given utterance. In particular, we found that a considerable proportion of the utterance vectors in our test corpus (approximately 20% of the total utterances) were eventually rejected by the classification algorithm and assigned to the Ω_0 cluster. A disambiguation module (Figure 2) has been therefore devised to resolve the mentioned ambiguities and enable assigning the patterns in Ω_0 to one of the problem categories $(\Omega_1, \cdots, \Omega_N)$.

First, the utterance vectors with more than one candidate cluster are selected. For each pattern, we have a list of pointers to all candidate centroids. Then the terms in each pattern that cause the ambiguity are identified and stored in a list of *competing terms*. As an example, let us consider the original utterance "*I want to get virus off my computer*" which corresponds, after the pre-processing and feature extraction stages, to the set of features (i.e. the bag-of-words) *get security off computer*. This feature vector has maximum similarity to the cluster centroids "*Computer freeze*" and "*Install security*", that were initially assigned, by human annotators, to the categories CRASHFROZENCOMPUTER and SECURITY, respectively. The competing terms that produce the ambiguity are in this case the words "*computer*" and "*security*". Therefore, the disambiguation among centroids (or clusters) is equivalent to a disambiguation among competing terms. To resolve the ambiguity, we have estimated the *informativeness* of a term t_i as shown in Equation 5:

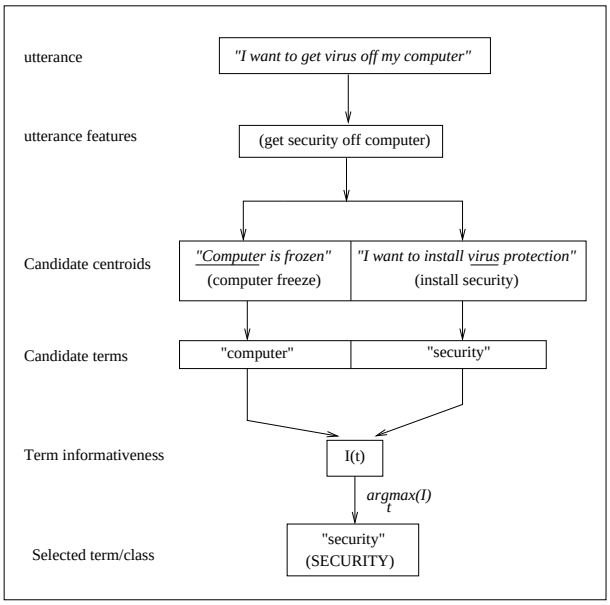

Figure 2: Scheme of the disambiguation module.

$$I(t_i) = -(log(Pr(t_i)) + \alpha \cdot log(\sum_{\substack{j \\ L_j = N}} \overline{C}_{ij} Pr(t_j))) \quad (5)$$

Where $Pr(t_i)$ denotes the maximum-likelihood estimation for the probability of the t_i term in the training corpus, and L_j makes reference to the lexical category of the t_j term (where N stands for the lexical category "noun").

As it can be inferred from equation 5, two main factors are taken into account in order to decide the relevance of a word for the disambiguation: a) the word probability (the information provided by one term decreases with the frequency of the term), and b) the term co-occurrence with the most frequent nouns in the corpus. The underlying assumption that justifies the second factor is that words representative of problem categories are mostly nouns and appear in the corpus with moderate frequencies. Thus a term that co-occurs with a large number of words with that characteristic may appear in utterances of different categories, and therefore, is less significant. The α parameter has been practically intended to place slight emphasis on this second factor. Although it performs reasonably well in a wide interval ($\alpha \in [1, 2]$), a particular value of $\alpha = 1.6$ has been selected.

Thus, the term with highest informativeness $I(t)$, computed as described above, is selected among the competitors and the ambiguous utterance vector is matched to the corresponding centroid or cluster.

The described approach is fully unsupervised to the extent that we do not need prior knowledge of the term distribution statistics within categories, which can only be estimated from manually annotated sets.

5. Evaluation and results

The approaches described above have been applied on a corpus of 34,848 utterances. 3,313 utterances (9.96%) have been

used as test, and the remaining 31,535 utterances (90.4 %) as training. Manual annotations of the utterances were also made available. The annotation was used only for testing and the for initialisation of the K-means clustering (i.e. we selected one labelled utterance from each category as the initial cluster centroids).

The classification performance of the proposed approach has been evaluated in terms of the classification accuracy rates. First, we have determined which one of the problem categories $\Omega_1, \cdots, \Omega_M$ is implicitely associated with each output cluster obtained by K-means algorithm. In this case, we tagged the clusters with the annotation of their, manually selected, initial centroids. We assumed that, being the K initial centroids adequately selected, the final clusters could shift away from the first assignments, but not sufficiently to represent different annotation categories.

For the accuracy test, we have compared the (manual) annotation label of each pattern with the tag assigned to the cluster to which it belongs. If they coincide, we can say that the given pattern has been correctly classified. We define the overall accuracy as:

$$Accuracy = \frac{\text{No. patterns correctly classified to } \Omega_1, \cdots, \Omega_M}{\text{Total No. patterns}} \quad (6)$$

We have also independently evaluated the disambiguation module. To that end, we have considered such cases when the ambiguity is resolvable, i.e, at least one of the candidate centroids has the same manual annotation that the ambiguous pattern. The performance of the disambiguation module, and hence the "correctness" of the proposed word-informativeness measure, has been calculated as the *Number of ambiguous cases correctly resolved/Total number of resolvable ambiguities*.

Table 2 shows the algorithm classification performance achieved for the following analysed cases: A) neither the feature extraction nor disambiguation are used (basic K-means); B) only the disambiguation module is applied on all terms in the bag-of-words as features; C) only the feature extraction technique based on semantic classes is applied; and D) the combination of both feature extraction and disambiguation techniques is used. In the experimental results reported here we used a number of symptom categories $M = 28$.

Table 2: *Performances achieved by the disambiguation module and the overall classification* .

	A	B	C	D
Disambiguation: *(Correctly resolved cases/ resolvable ambiguities)*	-	84%	-	71%
Classification: *(Accuracy rates)*	45%	57%	53%	**66.5%**

As it can be inferred from Table 2, the individual use of the feature extraction technique based on semantic classes improved the classification accuracy by up to 8%. Furthermore, the disambiguation module proved to be fairly effective by correctly assigning 84% of the ambiguous cases when applied to all content terms in the corpus. Its effectiveness seems to degrade when used in combination with the semantic classes. This fact can be attributed to a dependency of the disambiguation method with the term clustering performance. Finally, by combining all of the proposed methods (case D) the classification performance of the clustering algorithm improved by up to 21.5%. A total accuracy rate of 66.5% has been achieved, which is a reasonably good result, provided the unsupervised nature of the algorithm and the moderate number of problem categories ($M = 28$).

6. Conclusions and future directions

In this work we have studied methods to improve utterance classification performance of unsupervised learning algorithms within a technical support automated agent application. The first approach is a feature extraction method based on an automatic extraction of terms representing the same semantic concepts. Secondly, we have defined an unsupervised measure of the informativeness provided by a given word, which has been used to resolve the ambiguous cases at the utterance level classification. The potential of the proposed measure for disambiguation has been evidenced by the disambiguation performance results shown in Section 5.

The combination of the mentioned approaches has contributed to the overall classification performance with up to 21.5% of accuracy improvement, as shown from the comparison of the unsupervised classification results with the available manual categorisation. In consequence, we can presume that the proposed methods can aid the unsupervised text categorisation task for the cases where minimal categorisation examples are available. We showed that an overall accuracy of 66.5% can be achieved on the automatic classification of utterances into 28 distinct categories by providing only one example per category. We believe that this is a reasonably good result.

Currently, we are also investigating several alternatives to the extraction of semantic term classes. The first consists on the generation of fuzzy term classes to be used in combination with word-sense disambiguation methods in order to increase classification performance. Another possibility is to apply the calculated terms similarities directly into the text classification without intermediate term clustering.

7. References

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

[2] Ko, Y. and Seo, J., "Automatic Text Categorization by Unsupervised Learning", Proceedings of COLING-00, 18th International Conference on Computational linguistics, 2000.

[3] Yang, J and Pedersen, J.O , "A Comparative Study on Feature Selection in Text Categorization", Proceedings of ICML-97, 1997.

[4] Hartigan, J. and Wong, M., "A K-means Clustering Algorithm", Applied Statistics, (28):100-108.

[5] Minnen, G., Carrol, J. and Pearce, D., "Applied Morphological Processing of English", Natural Language Engineering, 7(3), 207-223, 2001.

[6] Salton, G., "The SMART Retrieval System", Prentice Hall, Inc., 1971.

[7] Picard, J., "Finding Content-bearing Terms using Term Similarities", Proceedings of EACL'99, 1999.

[8] Klein, D. and D. Manning, C., "Fast Exact Inference with a Factored Model for Natural Language Parsing", NIPS 2002. 2002.