AUTOMATIC CONCEPT IDENTIFICATION IN GOAL-ORIENTED CONVERSATIONS

Ananlada Chotimongkol and Alexander I. Rudnicky

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA
ananlada@cs.cmu.edu, air@cs.cmu.edu

ABSTRACT

We address the problem of identifying key domain concepts automatically from an unannotated corpus of goal-oriented human-human conversations. We examine two clustering algorithms, one based on mutual information and another one based on Kullback-Liebler distance. In order to compare the results from both techniques quantitatively, we evaluate the outcome clusters against reference concept labels using precision and recall metrics adopted from the evaluation of topic identification task. However, since our system allows more than one cluster to associate with each concept an additional metric, a *singularity score*, is added to better capture cluster quality. Based on the proposed quality metrics, the results show that Kullback-Liebler-based clustering outperforms mutual information-based clustering for both the optimal quality and the quality achieved using an automatic stopping criterion.

1. INTRODUCTION

Acquiring domain information is a resource intensive effort but is a necessary part of making language technologies useful. Automatic techniques have been described for language modeling [1,2] and as an adjunct to grammar writing [3,4,5] for spoken language systems. Parallel efforts, though with somewhat different goals and approaches, exist in text processing [6]. In this paper we focus on the problem of concept identification in goal-oriented human-human dialogs; our materials are different from those that have been previously studied in that they potentially allow us to take advantage of structure that is present across both sides of a conversation. Solutions to this problem can have a significant impact on the development of spoken language systems modeled on existing human models; concept identification is the first step towards the goal of automatically inferring domain ontologies suitable for use in automatic systems.

We observe that goal-oriented human-human conversation has a clear structure. When two persons engage in a conversation that has a specific goal such as finding information or negotiating a travel plan, they will organize their conversion accordingly and will make sure that the ideas are clearly communicated and understood. These characteristics will be reflected in the structure of the dialog and this structure can be used to automatically identify domain topics.

2. CONCEPT INDENTIFICATION

A number of techniques based on the analysis of distributional statistics of a corpus have been proposed for semantically clustering words or phrases. The Kullback-Liebler distance was used for automatic grammar induction process [3,4]. Kullback-Liebler distance, in conjunction with call-type probability has also been used to create grammar fragments for call-type classification task [5]. A clustering technique based on vector representation of words was described in [1].

For the most part, these techniques are seen as an aid to the authoring of grammar for a target domain or for adapting existing grammars to new domains [7].

For our purposes, we need substantially more accurate concept identification; that is, we need techniques that accurately and automatically group concepts together into "pure" clusters, even at the cost of missing some members or even entire concepts, and do so without the need for manual post-processing. (According to the evaluation metrics that we will introduce later, a pure cluster corresponds to the precision of one and an incomplete cluster corresponds to the recall of less than one.) Doing so successfully is necessary in order to eventually be able to generate consistent, if incomplete, behavior. Pure clusters will allow no mistake in the mapping from each concept member to a system action, which makes the system behaves consistently. However, missing some members will cause the system behavior to be incomplete. In this paper we examine two clustering algorithms, mutual information-based clustering and Kullback-Liebler-based clustering that express promising performance on words clustering.

In order to determine which clustering algorithm is best suited for our needs, we need to measure the quality of the outcome clusters. A straightforward approach is to have a human look at each set of clusters and decide which is better. However this approach is very subjective and also time-consuming. Some domain knowledge may also be needed since we would like to extract the concepts that are crucial to supporting dialog in the domain of interest. In other words, we care more about clustering together city names than verb inflections. In [4], an indirect evaluation of the clustering technique is used to evaluate the performance of Kullback-Liebler-based clustering. Since the clustering technique was used to semi-automatically produce the grammar, coverage and understanding accuracy of the grammar were used as quality measurements. In [2] when the resulting clusters were used to improve the language model, the quality of the clusters was indirectly evaluated through language model perplexity.

In the concept identification task, our ultimate goal is to obtain a set of clusters that precisely describe all the significant concepts in the domain of interest. Consequently, we need to evaluate the quality of output clusters themselves. Suppose that the correct set of concepts and their members is given. The quality of the output clusters can then be evaluated by comparing them against these reference concepts. This is quite similar in principle to the evaluation in the topic detection task. In topic detection, stories that discuss the same topic are grouped together and the groupings are evaluated against manually labeled stories [6,8]. We adopt two metrics, precision and recall, from topic identification system in our evaluation. However, in our work, more than one cluster is allowed to represent a single concept. Therefore, an additional metric, a *singularity score*, is added to better capture cluster quality.

3. CLUSTERING ALGORITHMS

The goal of automatic concept identification is to extract and group domain concept words together from an unannotated corpus. Since no concept notation is provided in the training data, we will focus only on unsupervised clustering techniques. To focus our attention to a clustering problem, we restrict each concept member to only a single word. Two word-clustering algorithms are investigated in this paper, one based on mutual information and one based on Kullback-Liebler distance. Both are hierarchical clustering techniques that iteratively merge words or clusters together in the order of their similarity; words are clustered into a treelike structure in which the cluster at a leaf corresponds to a word in the vocabulary. The intermediate nodes that are close to the leaves represent more specific word classes while the intermediate nodes that are close to the root represent more general concepts. Hierarchical clustering provides us with a more flexible way to understand and interpret the structure of the dialog since a concept may be broken into sub-concepts or grouped into a more general concept, as needed. In both algorithms the similarity between words or group of words is determined by their statistical similarity with surrounding words, but the definition of similarity differs.

3.1. Mutual information-based clustering

We use the mutual information-based clustering described in [9]. This approach defines the similarity between words or clusters based on their mutual information with adjacent words or clusters. The algorithm starts by assigning each word to its own cluster then iteratively merges clusters in a greedy way such that at each iteration the loss in average mutual information is minimized. The merging process continues until the desired number of clusters is reached. The average mutual information (AMI) is defined by the following equation.

$$AMI = \sum_{i,j} p(i,j) \log \frac{p(i,j)}{p(i)p(j)} \quad (1)$$

where $p(i,j)$ is the bigram probability of cluster$_i$ and cluster$_j$, i.e., the probability that a word in cluster i precedes a word in cluster j.

3.2. Kullback-Liebler-based clustering

In Kullback-Liebler-based clustering the similarity between words or clusters is determined by the Kullback-Liebler (KL) distance. The clustering technique that we use is similar to the ones described in [4,7]. We use a symmetric non-blow-up variant of the KL-distance, which is known as *Jensen-Shannon divergence* [10], to avoid the problem when one of the probabilities is equal to zero. The KL-distance between two probability functions p_a and p_b is given by the following equation.

$$J(p_a; p_b) = \frac{(D(p_a \parallel p_{\frac{a+b}{2}}) + D(p_b \parallel p_{\frac{a+b}{2}}))}{2} \quad (2)$$

where $D(p_a, p_b)$ is the conventional KL-distance.

$$D(p_a, p_b) = \sum_Y p_a(Y) \log \frac{p_a(Y)}{p_b(Y)} \quad (3)$$

A distance between cluster$_i$ and cluster$_j$, is the sum of the KL-distance between the left context probability, p^{left}, of the two clusters and the KL-distance between the right context probabilities, p^{right}, of the two clusters. More specifically,

$$Dist(i,j) = J(p_i^{left}, p_j^{left}) + J(p_i^{right}, p_j^{right}) \quad (4)$$

We calculate p^{left} and p^{right} from the bigram probability. $p_i^{left}(v_k)$ is the probability that word v_k is found to the left of words in cluster$_i$. Similarly, $p_i^{right}(v_k)$ is the probability that word v_k is found to the right of words in cluster$_i$. Specifically,

$$p_i^{left}(v_k) = p^{left}(v_k \mid cluster_i) = \frac{p(v_k, cluster_i)}{p(cluster_i)} \quad (5)$$

$$p_i^{right}(v_k) = p^{right}(v_k \mid cluster_i) = \frac{p(cluster_i, v_k)}{p(cluster_i)} \quad (6)$$

From the definition of p^{left} and p^{right}, the sum in equation 3 is the sum over all the context words v_k in the vocabulary.

The merging process is similar to that of mutual information-based clustering, except that the order of clusters that get merged are determined by KL-distance instead of AMI.

4. EVALUATION METRICS

The evaluation metrics that we adopt in our experiments are similar to the ones that used is topic detection task, but are modified to better suit our requirement. In the evaluation process, we first identify the reference concepts in the domain. Then the outcome clusters are matched against the reference concepts. Finally, two levels of metrics, concept-based metrics and overall metrics are calculated.

4.1. Defining concepts

We based our reference concepts on the language modeling classes used in travel planning task defined in the CMU Communicator system [11]. For simplicity, we restrict each word to belong to only one concept. Our reference set contains 15 concepts with 188 concept words. Some examples of domain concepts are *city*, *airline_company*, and *month*. Words that don't belong to any domain concepts are grouped into a single *general* concept.

To compare the output clusters with the reference concepts we first choose the concept that each cluster represents by identifying the concept that encompasses the greatest number of words; this is the *majority concept*. We allow more than one cluster to represent a concept. This is justified because it should lead to consistent behavior at a cost in the efficiency of representation that we deem acceptable.

4.2. Concept-based metrics

We evaluate the output clusters based on how well they match concepts in the reference set. To do this, we calculate the following metrics, *precision*, *recall* and *singularity score*, for each reference concept. Precision and recall are adopted directly from the topic detection task to measure the purity and completeness of the clusters. However, in the topic detection task only one cluster is associated with a topic, while in our work more than one clusters is allowed to map to a concept. Thus, precision and recall of each concept are calculated from average precision and average recall of the clusters that represents that concept. Even splitting the concept into two or more clusters is acceptable; one cluster per one concept is preferred. We define an additional quality metric, singularity score, to capture how well words that belong to the same concept are merged together. A penalty is assessed when a concept is split into more than one cluster. The singularity score is defined by the following equation

$$singularity\ score\ (SS)\ of\ concept_j = \frac{1}{m_j} \qquad (7)$$

where m_j is the number of clusters that represents $concept_j$.

4.3. Overall metrics

In order to perform an end-to-end comparison of the two clustering algorithms, we need a metric that indicates the overall cluster quality. There are two ways to combine concept-based metrics into a single number, *micro-average* (equal word weighting) and *macro-average* (equal concept weighting) [8]. Since the number of words in each concept is not uniformly distributed we choose a macro-average as a combination method. We then combine the macro-average concept-based metrics together using a harmonic mean. A *quality score*, an overall quality measurement of the output clusters, is thus denoted by a harmonic mean of the macro-average of the precision, recall and singularity scores.

5. EXPERIMENTS AND DISCUSSION

Our experiments make use of the CMU Travel Agent corpus, which contains 39 goal-oriented human-human dialogs in an air travel domain. There are 2,196 utterances in the corpus consisting of 1,108 utterances from a (single) travel agent and 1,088 utterances from multiple clients. In each dialog, the client attempts to arrange a travel plan including flight, hotel and car reservation. The total vocabulary size is 950 words.

We run both clustering algorithms until only one cluster is left and measure the quality of clusters at each merge step. We evaluate the quality of clusters based on two overall metrics, our proposed quality score (QS) and a conventional overall metric, *macro-average* F_1 (F-1), which is a harmonic mean of macro-average precision and macro-average recall [6]. The results are shown in Figure 1.

Comparing the two overall quality metrics, quality score and macro-average F_1, we found that the trends of their values are almost the same. However, at around the 300th iteration and the 450th iteration of KL-based clustering where values of F-1 are about the same, the QS of the 300th iteration is higher. This is because the resulting clusters at the 450th iteration contain more splitting which lowers the singularity score. As a result, a QS which includes a singularity score in its calculation reflects this penalty while an F-1 which doesn't include a singularity score can not capture the difference. We believe this demonstrates that the proposed quality score is a more precise evaluation metric.

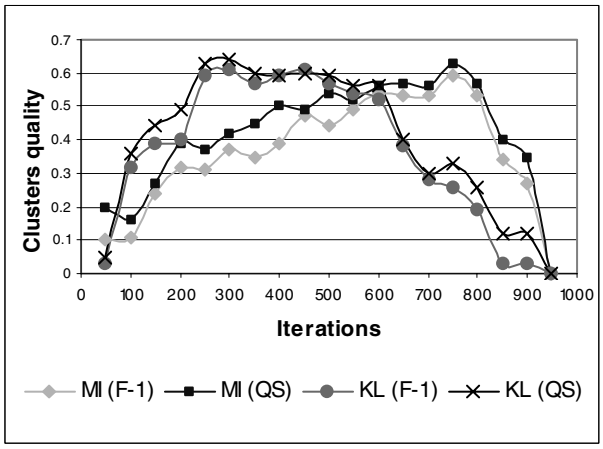

Figure 1: Quality of the clusters at each merging iteration of Kullback-Liebler-based clustering (KL) and mutual information-based clustering (MI).

The values of the quality metrics at the optimal points for both KL-based clustering and MI-based clustering are given in Table 1. From Figure 1, we can see that even though the optimal quality of both algorithms is about the same, the KL-based clustering converges to its optimal quality faster than the MI-based clustering. Moreover, in the KL-based clustering, the region that the QS is near the optimal value is larger than that of MI-based clustering.

	Iteration	Precision	Recall	SS	F-1	QS
MI (optimal)	762	0.77	0.49	0.72	0.60	0.64
MI (stop)	828	0.78	0.34	0.66	0.47	0.52
KL (optimal)	288	0.82	0.49	0.75	0.62	0.66
KL (stop)	485	0.74	0.49	0.67	0.59	0.62

Table 1: Optimal quality and the quality of the output clusters at the automatic stopping criteria.

When we examined the output clusters more closely, we found that KL-based clustering tends to merge concept words together first and it is more likely to group a single word into the exist clusters rather than create a new cluster. In contrast, MI-based clustering is more likely to merge two words into a new cluster and more of none-concept words get merged at the beginning. As a result, all the quality metrics, precision, recall and singularity score, of KL-based clustering are higher than those of MI-based clustering at the early iterations. This characteristic of KL-based clustering leads to a faster convergence to its optimal point. Toward the end of the

clustering process, the MI-based clustering begins to merge concept words and clusters into bigger clusters that make the values of all quality metrics increase. This leads to the late converged characteristic of MI-based clustering. The disadvantage of a slow converged clustering algorithm is that at it has more chance to merge irrelevant words into concepts and may also merge different concepts together at its optimal point. The results, shown in Table 1, tend to agree with this analysis. The optimal clusters of KL-based clustering have higher precision value. Even if the recall is the same, KL-based can recover more concepts than MI-based clustering.

However, it is not possible to stop the clustering process at the point where the quality of the cluster is optimal in the real situation since we naturally cannot obtain the reference concepts beforehand. Therefore, we need a stopping criterion that yields reasonable clusters quality. In Figure 2, we plot the difference in the average mutual information (AMIdelta) of successive iterations in log base, versus the clustering iteration. We can see from the graph that the log-AMIdelta forms a straight line for the most part but rises up at the end. It is reasonable to stop the clustering process when the AMIdelta increases significantly since we lose too much information from merging clusters together at that iteration. To obtain that iteration point, we draw a linear estimation of log-AMIdelta after removing the outliners. The intersection between the linear estimation and the actual AMIdelta is the stop point. For KL-based clustering, we simply use the median of the distances between clusters merged at each iteration as the cutoff value and stop the clustering process when the distance exceeds the cutoff value. This technique was also used in [1]. The quality of the resulting clusters at the stopping criteria are shown in Table 1 along with the optimal quality for both MI-based clustering and KL-based clustering

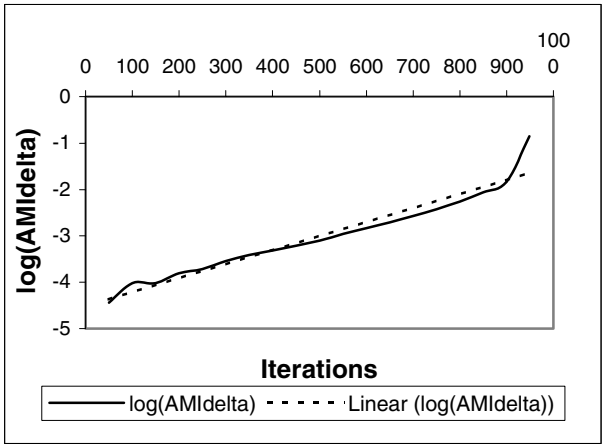

Figure 2: The growth of log(AMIdelta) along increasing iterations in the clustering process.

6. CONCLUSIONS

From our experiments we found that the KL-based clustering outperforms the MI-based clustering on both the optimal quality and the quality measure at the stopping criterion. We note that KL-based clustering also proposes a clear definition of distance between words and clusters that may be practical to use in conjunction with other clustering techniques. The experimental results also show that our proposed quality metric that incorporates a penalty for splitting a concept to more clusters is a suitable evaluation metric when more than one cluster is allowed to represent a concept.

In the future, we aim at improving the performance of clustering algorithm by incorporating additional information based on the turn structure of dialogs. For example, significant concepts introduced by one participant will be echoed by the other participant. We believe that this confirmation phenomenon and also the co-occurrence between words across the turns can augment the bigram co-occurrence currently used in the clustering algorithms we used.

7. ACKNOWLEDGEMENT

We would like to thank for Rose Hoberman for the MI-based clustering program and all her valuable suggestions. This research was sponsored in part by the Space and Naval Warfare Systems Center, San Diego, under Grant No. N66001-99-1-8905. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

8. REFERENCES

[1] Riccardi, G. and Bangalore, S., "Automatic Acquisition of Phrase Grammars for Stochastic Language Modeling", *Proc. of the 6th Workshop on Very Large Corpora*, Montreal, Canada, 1998.

[2] Fosler-Lussier, E. and Kuo, J., "Using Semantic Language Models within ASR Dialogue Systems", *Proc. of ICASSP-01*, Salt Lake City, UT, 2001.

[3] McCandless, M. and Glass, J., "Empirical Acquisition of Word and Phrases Classes in the ATIS Domain", *Proc. of EUROSPEECH-93*, Berlin, Germany, 1993.

[4] Siu, K. and Meng, H., "Semi-Automatic Acquisition of Domain-Specific Semantic Structures", *Proc. of EUROSPEECH-99*, Budapest, Hungary, 1999.

[5] Arai, K., Wright, J., Riccardi, G. and Gorin, A., "Grammar fragment acquisition using syntactic and semantic clustering", *Speech Communication*, vol. 27, no. 1, Jan. 1999.

[6] Allan, J., Carbonell, J., Doddington, G., Yamron, J. and Yang, Y, "Topic Detection and Tracking Pilot Study: Final Report", *Proc. Of DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, 1998.

[7] Pargellis A., Fosler-Lussier E., Potamianos A. and Lee C., "Metrics for Measuring Domain Independence of Semantic Classes", *Proc. of EUROSPEECH-01*, Aalborg, Denmark, 2001.

[8] National Institute of Standards and Technology (NIST). "The Topic Detection and Tracking Phase 2 (TDT2) Evaluation Plan Version 3.7", 1998.

[9] Brown, P., Della Pietra, V., deSouza, P., Lai, J. and Mercer, R., "Class-based n-gram models of natural language", *Computational Linguistics*, 18(4):467-479, 1992.

[10] Dagan, I., Lee, L., and Pereira, F., "Similarity-based models of word-cooccurence probabilities", *Machine Learning*, 34(1-3): 43-69, 1999.

[11] Rudnicky, A., Thayer, E., Constantinides, P., Tchou, C., Shern, R., Lenzo, K., Xu, W. and Oh, A., "Creating Natural Dialogs in the Carnegie Mellon University Communicator System", *Eurospeech 1999*, V.4: 1531-1534, 1999.