

Finding Structure in Noisy Text: Topic Classification and Unsupervised Clustering

R. Prasad, P. Natarajan, K. Subramanian, S. Saleem, R. Schwartz

BBN Technologies

Speech and Language Processing Department

50 Moulton Street, Cambridge, MA 02138

rprasad@bbn.com

Abstract

In this paper we present recent advances in automatic categorization of noisy, unstructured text messages posted on newsgroups. Newsgroup messages pose a significant challenge to automatic text categorization due to broad coverage of subject matter and the use of informal language. This paper addresses (a) spotting messages that are on topics of interest to the user, and (b) automatic organization of a large corpus of messages without any prior knowledge about topics of interest to the user. We present supervised classification results using our hidden Markov model based topic classification engine on messages from two different newsgroup corpora. Given that in an operational setting an overwhelming fraction of messages may not be related to topics of interest to the user, we describe techniques for reducing false alarms in such a scenario. For automatic organization of messages, we present a novel concept of unsupervised clustering of topics that enables human analysis of messages at multiple levels of granularity.

1 Introduction

Widespread use of the Internet has resulted in immense volume of unstructured text being generated within a short span. Even for a single user the number of documents that are sent to the user makes manual organization cumbersome. Typically, users categorize documents based on the topic or the primary theme of the document. Therefore, a system that automatically organizes documents based on topics will have a far reaching impact.

Topic based document classification has been applied to various domains including broadcast news [Schwartz *et al.*, 1997; Joachims, 1998] and Newsgroup messages [Baker *et al.*, 1998; Rennie *et al.*, 2003]. In [Schwartz *et al.*, 1997] our hidden Markov Model (HMM) based topic classifier was shown to outperform Naïve-Bayes (NB) classifiers and term frequency-inverse document frequency (tf-idf) classifiers on broadcast news articles consisting of thousands of topics. A significant difference in the approach in [Schwartz *et al.*, 1997] from other approaches is that instead of assuming that a document is always on a single topic, we postulate

that a document is often on multiple topics. Also, we do not require all words to be associated with a topic, i.e. some words are allowed to be relevant to the general language only.

In this paper, we describe our ongoing work on automatic text categorization for two different operational scenarios. In the first scenario, the topics of interest to the user are known *a priori*, such as in a topic spotting or an alerting application. In the second scenario, a large corpus of text documents is provided, and the system is required to organize the documents in a structure that enables effective human analysis.

Given our focus on messages posted on newsgroups, we first present supervised classification results obtained on two newsgroup corpora using our HMM based topic classification engine. Since in a real operation an overwhelming fraction of messages may be “off-topic”, i.e. not related to topics of interest, we also address the rejection of off-topic messages for maintaining low false alarms.

To enable effective navigation of a large corpus of unstructured text data we have developed an algorithm for performing unsupervised topic clustering (UTC) by extending our unsupervised topic discovery (UTD) [Sista *et al.*, 2002] capability. In UTC, we organize automatically discovered topic labels from the UTD system in a hierarchical tree structure. A document is assigned to one or multiple clusters based on the topic labels assigned to it. This approach overcomes the problem with traditional document clustering approaches that enforces a document to be assigned to a single cluster.

2 Corpora for Experimentation

We used the following data sources for development and testing at various stages of our research.

AFE Newsgroup Corpus: The Automated Front End (AFE) newsgroup corpus [Eick *et al.*, 2005] was collected by Washington University (WU) by harvesting messages from 12 Google newsgroups. There are 11,503 messages in this corpus. Since our intent as in [Eick *et al.*, 2005] was to use the newsgroup name as the topic for text message, the message headers, e-mail Ids, signatures, etc. were stripped to exclude newsgroup sensitive information. The 10,768 messages from “talk.origins” newsgroup were used as

“chaff” or off-topic messages in [Eick *et al.*, 2005] and we used them for the same purpose in our experiments.

20 NG Newsgroup Corpus: The 20 Newsgroup (20 NG) corpus [20 NG Corpus; Rennie *et al.*, 2003] consists of 18,820 messages from 20 newsgroups. These newsgroups cover six broad subject matters. As with the AFE corpus, we stripped message headers, e-mail IDs, and signatures. The average number of messages per newsgroup (i.e., per topic) is much higher for the 20 NG corpus as compared to the AFE newsgroup data.

Large Chaff Newsgroup Corpus: Although the messages designated as chaff from the talk.origins newsgroup in the AFE newsgroup corpus covers a broad range of topics, their number does not permit reliable measurement of false alarm rates lower than approximately 20 in 10,000. Therefore, we downloaded approximately 250,000 messages from 10 Yahoo! Groups to be used as off-topic messages.

PSM Corpus: The Primary Source Media (PSM) corpus consists of transcribed broadcast news articles from 1995-1996 collected from different news sources. This corpus consists of 45K articles from July 1995 through July 1996. Each article was annotated with up to 13 topic labels. In total there are 5,239 unique topic labels with an average of 4.5 topic labels per article. We used this corpus to compare classification performance on newsgroups with broadcast news text.

3 HMM based Topic Classification

Our topic classification engine uses an HMM to model multiple topics in documents explicitly. The model topology is shown in Figure 1. Each topic is represented by a 1-state HMM. In addition, there is an HMM for the General Language. A probability distribution for the words in the language is associated with each topic state. In the simplest case, this model is a unigram distribution $P(W_n|T_j)$ on words. However, the state could also contain a higher order n -gram language model for word sequences for that topic.

The parameters of the model shown in Figure 1 are estimated using the expectation maximization (EM) algorithm from a corpus of documents labeled (either automatically or by humans) with associated topics.

Classification of a test document D is performed in two stages. First we consider each topic independently using equation (1) to choose a small set of likely topics. $P(T_j|D)$ is the posterior probability of topic T_j given the document D , $P(T_j)$ is the *a priori* probability for T_j , $P(T_j | j \in Set)$ is the average percentage of words generated by topic state T_j given it is in the set of topics, and β is an exponential weight to counteract for the independence assumption. $\phi(x)$ is equal to x when x is positive and 0 when x is negative.

$$\log P(T_j | D) = \log P(T_j) + \sum_t \phi \left\{ \log \left[P(T_j | j \in Set)^\beta \frac{P(W_t | T_j)}{P(W_t)} \right] \right\} \quad (1)$$

Then, we rescore all subsets of the top-N topic labels assigned to a document [Schwartz *et al.*, 1997].

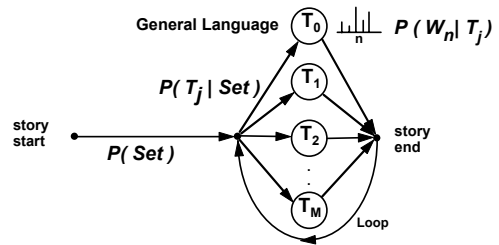


Figure 1: Generative model used in our HMM based topic classifier.

In all our experiments reported in this paper we present results from only the first stage of the classification, i.e. we have reduced the problem of finding the optimal set of topics for a document to finding the top-N topic labels.

4 Supervised Classification Experiments

In [Schwartz *et al.*, 1997], we had reported on the top-N precision obtained on the PSM corpus using the HMM based topic classifier, while classifying documents related to the topics of interest, i.e. in-topic documents. The top-1 accuracy, which is defined as the percentage of times the top-choice topic was the correct answer, was 75.7% on the held-out test set from the PSM corpus.

In this section, we report on classification accuracy obtained on the newsgroup data from AFE and 20 NG corpora. Instead of manual annotation all the messages in a newsgroup were automatically annotated with the name of that newsgroup. In this annotation scheme, each message is assumed (an inaccurate assumption) to be on a single topic. Although cost effective, the assumption that the name of the newsgroup is the only valid topic for the message often leads to inaccuracies in estimating system performance because all non-trivial messages consist of multiple topics.

4.1 Classification Accuracy on AFE Corpus

First, the entire corpus was randomly partitioned into training, development, and validation sets in the same proportion as described in [Eick *et al.*, 2005]. Next, we trained our topic classification engine with 241 messages available for training from 11 newsgroups in the AFE corpus. Chaff or off-topic messages from the talk.origins newsgroup of the AFE corpus were excluded from training and test, since in these experiments we are interested in evaluating the in-topic classification accuracy. Finally, we classified the 376 held-out test messages from the 11 newsgroups.

In Table 1, we summarize the top-choice accuracy obtained on test messages for each individual newsgroup as a function of the amount of training messages. The overall top-choice accuracy was 91.2%. Messages from newsgroups “Misc.consumers.frugal_living” and “Soc.libraries.talk” newsgroups seem to be most difficult to classify, primarily due to the lack of training data.

Although we used the same number of messages as in [Eick *et al.*, 2005] for training and test, we did not have access to the same set of messages. Therefore, the training and test sets differ in terms of the content. As a result, we are

Newsgroup	#Training Messages	%Top-1 accuracy
Misc.consumers.frugal_living	10	47.1
Soc.libraries.talk	10	58.8
Comp.ai.neural_nets	15	80.0
Rec.martial_arts.moderated	18	86.2
Humanities.musics.composers	19	100.0
Sci.logic	20	96.7
Alt.sports.baseball.stl.cardinals	21	100.0
Misc.writing.moderated	24	91.9
Rec.Equestrian	27	97.6
Comp.programming.threads	31	100.0
Sci.archaeology.moderated	46	95.7
Overall	241	91.2

Table 1: Top-1 accuracy measured on the 376 in-topic test messages from the AFE corpus as a function of the number of training messages.

unable make a direct comparison with the work reported in [Eick *et al.*, 2005].

4.2 Classification Accuracy on 20 NG Corpus

First, we split the entire 20 NG corpus into training, development, and validation sets using the following three partitioning methods. In each of the partitioning, 80% of the entire data was kept for training and the remaining 20% was divided equally among development and validation sets.

1. *Thread Partitioning*: The entire message thread was assigned to one of the three sets: training, development, or validation.
2. *Chronological Partitioning*: Each message in a thread was assigned to training, test, or validation based on the message date: the first 80% (chronologically) were assigned to training, and remaining to test and validation.
3. *Random Partitioning*: 80:20 split between training and test/validation, without regard to thread or chronology.

Chronological partitioning and thread partitioning are more likely to be representative of an actual operational scenario. We performed a random partitioning experiment in order to compare the classification accuracy with prior work reported in [Baker *et al.*, 1998; Rennie *et al.*, 2003].

We trained topic models for 20 topics (each newsgroup was treated as a topic) on the available training data for each of the partitioning. In Table 2, we list the results obtained for each of the partitioning methods. As one would expect, the classification accuracy is the best for the “random” partitioning, followed by “chronological” partitioning, and then “thread” partitioning.

The top-choice accuracy of 83.2% with random partitioning on the 20 NG corpus is similar to the state-of-the-art performance reported in [Rennie *et al.*, 2003]. However, direct comparison with [Baker *et al.*, 1998; Rennie *et al.*, 2003] was not possible as we did not have access to the same messages (with exactly the same processing for removing headers, signatures etc.) for training and test.

The classification accuracy on 20 NG corpus is significantly worse than the 91.2% top-choice accuracy we ob-

Experiment	%Top-1 accuracy		
	Thread	Chrono.	Random
Baseline Classification	76.0	79.6	83.2
+ Manual Clustering of groups	81.5	84.8	88.2
+ Manual Review	N.A.	88.0	N.A.

Table 2: Top-1 accuracy on the 20 NG corpus for different partitioning of the data.

tained on the AFE data set. One of the reasons for the inferior performance is the substantial overlap in the subject matter. The assumption that each newsgroup can be treated as a completely different topic is also not valid because most non-trivial messages contain multiple topics.

Ideally, we should annotate every message in the corpus with ALL relevant topic labels but such annotation for such a large data set is not feasible in a short amount of time. Therefore, we decided to read a few training messages from each newsgroup and manually organize newsgroups that have similar subject content into a single topic. Our manual organization resulted in 12 different “topics”, which is still conservative compared to the 6 clusters proposed in [20 NG Corpus]. We have recomputed the top-1 accuracy by using the manual organization of the topic clusters. As shown in Table 2, the classification accuracy increases by 5% absolute across the board. For the “chronological” partitioning condition, we further analyzed the classification errors for the 4 topics with the lowest accuracy. The analysis revealed that for a majority of the misclassified messages from these topics, the top-choice topic label assigned to the message by the classifier was indeed relevant to the message.

5 Rejection of Off-Topic Messages

In many runtime scenarios, it is highly likely that an overwhelming fraction of the data will be off-topic. Therefore, it is critical for the topic classification engine to have the ability to reject almost all the off-topic messages accurately, while still retaining a large fraction of the in-topic messages.

We have developed a rejection mechanism based on the assumption that the General Language (GL) state in the model shown in Figure 1 can serve as the alternate model, i.e., a composite model for all topics that are not of interest.

Accepting a message means asserting that the message contains the top-choice topic, while rejecting a message means asserting that the message does not contain the top-choice topic. As shown in equation (2), we accept a message if the ratio of the log posterior probability of the top-choice topic to the log posterior probability for the GL topic is greater than a pre-specified threshold, α , otherwise, we reject that message.

$$LR(T_j) = \frac{\log P(T_j | Message)}{\log P(GL | Message)} > \alpha \quad (2)$$

We used the ratio of the log-posterior instead of the log likelihood ratio because the ratio in equation (2) resulted in lower false alarm rates than the log-likelihood ratio in our preliminary experiments. The threshold α in equation (2) controls the trade-off between false acceptances (FA) and false rejections (FR). α can either be independent of the top-choice topic or be dependent on the top-choice topic. Given that each topic has a different FA/FR characteristics, topic-specific thresholds are likely to outperform topic-independent thresholds.

Since our goal is to evaluate FA/FR characteristics at very low false alarm rates (1% FA or lower), we constructed a new corpus for experimentation including data from all the 3 newsgroup sources described in Section 2. The distribution of the in-topic and off-topic messages across training, development, and validation sets is shown in Table 3. The in-topic messages are from 14 newsgroups from the 20 NG corpus. We excluded the messages from 6 of the 20 newsgroups from the 20 NG corpus from our experiment because there was significant subject matter overlap between these newsgroups and the large chaff corpus.

Message Type	Source	Distribution of Messages		
		Train	Dev.	Val.
In-topic	20 NG	5.6K	5.6K	2.8K
Off-topic	AFE Chaff and Large Chaff	9.6K	9.6K	76K

Table 3: Distribution of messages across training, test, and validation sets for rejection experiments.

In the following, we compare different threshold estimation techniques using the data set described in Table 3.

5.1 Topic-Independent Thresholds

First, we trained our topic classification engine with the messages from 14 newsgroups. The 9.6K chaff messages were used to train the General Language state of the model. Next, we classified the in-topic as well as off-topic messages from the development and validation set. Then for each message, we accepted or rejected the top-choice topic if the ratio in equation (3) was higher than a topic-independent threshold α decided *a priori*.

In Figure 2, we plot %FA versus %FR for different values of α (denoted as “topic-ind” in the figure) on the validation set. As shown in Table 4, the topic-independent thresholds result in 31.4% false rejections at an operating point of 1% false acceptances.

5.2 Parametric Topic-Specific Thresholds

In the parametric approach for estimating topic-specific thresholds, for each topic in our model we compute the empirical distribution (mean and variance) of the log-posterior ratio for all the chaff/off-topic messages in the development set that are mislabeled as that particular topic. At runtime, we first normalize the log-posterior ratio for each message according to the following equation:

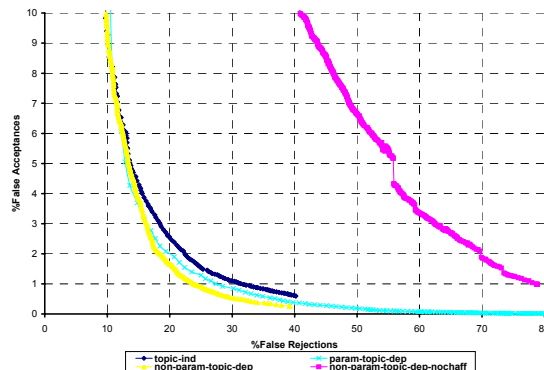


Figure 2: %False Acceptances vs. %False Rejections for different rejection techniques.

$$score_{normalized} = \frac{score - \mu_{off}(T_i)}{\sigma_{off}(T_i)} > \alpha \quad (3)$$

where, score is the log-posterior ratio of the message for the top-choice topic, T_i , that is being considered as a valid label for the message, $\mu_{off}(T_i)$ and $\sigma_{off}(T_i)$ are the empirical mean and variance computed from the log-posterior ratios of the off-topic messages for the topic T_i in the development set, and $score_{normalized}$ is the normalized log-posterior ratio. For accepting or rejecting a message we now compare the normalized score for the top-choice topic to a threshold α .

Rather than viewing the calculation in equation (3) as a normalization of the score, it can be viewed as a parametric topic-specific transformation of the threshold. The score normalization effectively results in a set of topic-specific thresholds that have been obtained through parametric transformations of a single threshold value.

In Figure 2, we plot the %FA vs. %FR for the parametric topic-specific thresholds on the validation set (denoted as “param-topic-dep” in the figure). As shown in Table 4, at 1% false acceptances, there is a 4% absolute reduction in the false rejections over the topic-independent thresholds.

5.3 Non-Parametric Topic-Specific Thresholds

The problem of estimating topic-specific thresholds $\alpha(T_i)$ is a constrained optimization problem, which can be formulated as finding thresholds $\alpha(T_i)$ that:

$$\min \sum_i f_i(x_i) \quad \text{subject to} \quad \sum_i x_i = k \quad (4)$$

In equation (4) x_i is the number of off-topic (chaff) messages accepted as topic T_i for a particular value of $\alpha(T_i)$, $f_i(x_i)$ is the number of messages related to topic T_i that were falsely rejected (represented as a function of number of false accept contribution from topic T_i), and k is the total number of false accepts.

We used a differentiable nonlinear optimization [DONLP2] algorithm on the development set to estimate topic-specific thresholds for different values of k . Since, we have a finite number of data points, the function $f_i(x_i)$ is a step function for each topic T_i , which is not differentiable.

Rejection Method	%False Rejections
Topic-independent thresholds	31.4
Topic-specific thresholds (parametric)	27.4
Topic-specific thresholds (non-param)	23.7

Table 4: Comparison of false rejections obtained with different rejection methods at %false acceptances = 1.0.

Therefore, we smooth our estimate for $f_i(x_i)$ using a Gaussian smoothing function.

In Figure 2, we plot the %FA vs. %FR for the validation set (denoted as “non-param-topic-dep”) using topic-specific thresholds estimated from the development set. As shown in Figure 2, the performance of the non-parametric topic-specific threshold is significantly better than the topic-independent and the parametric topic-specific scheme. Also at 1% false acceptances, there is a 3.7% absolute reduction in the false rejections over the parametric method for topic-specific thresholds.

In another experiment, we excluded the off-topic messages from training of the GL state in our model. Next, we estimated the non-parametric thresholds from the development set as before. In Figure 2, we denote the FA/FR curve for this experiment as “non-param-topic-dep-nochaff”. As one would expect, the FA/FR characteristics are significantly worse when off-topic messages were excluded from training of the GL state. Therefore, training the topic classifier on some amount of off-topic data is critical for maintaining low FA.

6 Unsupervised Topic Clustering

Supervised topic classification requires annotating documents with topic labels, which for a large number of topics of interest can be a significant cost. Moreover, when topics of interest to the user are not known then supervised classification is not a viable solution for categorizing documents.

Several approaches [Li *et al.*, 1998; Biswas *et al.*, 1998] for automatically extracting thematic or topic information from unstructured data have been proposed in the literature. These approaches are based on clustering documents using measures for inter-document similarity. A pitfall of these approaches is that they assign a document to a single cluster, thereby violating the assumption that a document is often relevant to multiple topics.

In [Sista *et al.*, 2002], we had presented a language and domain independent algorithm called Unsupervised Topic Discovery (UTD), which discovers topics from a collection of documents, provides a human understandable topic label and then assigns the topics to the documents. UTD results in a flat list of detailed topic labels, which can be used for document categorization, summarization, and story segmentation. Although, the flat list of topics can be used as search terms to navigate a large corpus of documents, we believe user experience can be significantly improved if these topics are arranged in a hierarchical tree. Therefore, in this section

we extend our UTD capability to cluster discovered topics in a hierarchical tree structure.

6.1 Algorithm for Unsupervised Topic Clustering

Our approach for extracting topics from a large corpus of documents and organizing them in a hierarchical structure consists of the following steps:

1. Perform UTD to automatically identify topic labels in the corpus and determine which documents/messages contain each of the topics.
2. Automatically cluster discovered topics in a hierarchical tree structure and assign documents to appropriate nodes in the tree based on topic labels.

The result of the above procedure is a hierarchical topic tree, where the leaves of the tree are the individual topics discovered from the UTD process and intermediate nodes are a collection of topics. Since the UTD process typically assigns multiple topics to a document, a document can belong to multiple clusters if the topics assigned to the document belong to different clusters. Thus, unsupervised topic clustering (UTC) overcomes the problem of single cluster assignment of traditional document clustering. In addition, UTC enables the user to navigate the large corpus of documents at multiple granularities by allowing the user to “zoom” into any cross-section of the topic tree.

We used the following agglomerative clustering procedure for clustering topics:

1. Each topic is initially assigned to its own individual cluster.
2. For every pair of clusters, we compute the distance between the two clusters in the pair using an appropriate distance metric.
3. The two clusters that are closest to each other are merged into a single cluster.
4. Steps 2 and 3 are repeated iteratively until the distance between the closest pair is higher than a threshold.

We enhanced the above algorithm by allowing clustering of more than 2 topics (or clusters) at any given iteration. In addition, we investigated several distance measures for measuring topic similarity. These measures can be broadly categorized into two the following groups.

Topic Co-occurrence Based Metrics: These metrics are based on co-occurrence counts of topics in documents labeled by the topic discovery system.

1. *Co-occurrence Probability:* The co-occurrence probability $P(T_1, T_2)$ for topics T_1 and T_2 was computed as a ratio of the number of documents in which the two topics co-occurred over the number of documents that were labeled as T_1 or T_2 .
2. *Mutual Information:* We used the topic co-occurrence probability $P(T_1, T_2)$ to compute the mutual information:

$$D_{MI}(T_1, T_2) = P(T_1, T_2) \log \left[\frac{P(T_1, T_2)}{P(T_1)P(T_2)} \right] \quad (5)$$

Support Word Based Metrics: These are metrics based on comparison of output observation probability distribution, $P(W|T)$ for two topics. We refer to these metrics as the support word based metrics, since the words that “support”

a topic are the ones that have a non-zero output observation probability for that topic.

1. *Support Word Overlap*: This metric measures the overlap between support words for two topics by computing the fraction of support words common to topics T_1 and T_2 with respect to the total number of unique words.
2. *J-Divergence*: We used the J-divergence [Johnson *et al.*, 2001], a variant of the KL divergence for comparing the support word distribution of two topics.

6.2 Experimental Results and Evaluation

We performed UTC experiments on the entire 20 NG corpus. We clustered 3,343 unique topics discovered using UTD from 19K messages using different distance metrics. First, we visually inspected the output of the agglomerative clustering with different topic similarity measures. Next, we defined a metric called the clustering rate that seemed to be correlated with our subjective evaluation. The clustering rate measures, at each iteration of the clustering algorithm, the rate of increase of clusters that contain more than one topic. A higher clustering rate generally implies that single-topics are being merged into a cluster whereas a lower rate implies that clusters with multiple topics are being merged.

We found that the mutual information based measure creates the most number of clusters with size ≥ 2 at any given iteration suggesting that its topic clusters are most “uniform”. The J-divergence seems to be the worst in terms of the same metric. We have also explored combining topic co-occurrence and support word based metrics. Subjective evaluation indicates that combining the J-divergence and topic co-occurrence metrics results in more uniform clusters than using either metric by itself.

Figure 3 illustrates a sample sub-tree from the 20 NG corpus. We also summarize some of the key statistics for the topic tree generated after executing 550 iterations of the clustering algorithm described in Section 6.1.

1. *Number of levels*: The depth of the resulting tree was 6, implying a maximum number of 6 clicks will be required to reach a topic at the leaf of the tree.
2. *Branching factor*: The average branching factor of the topic tree was 2.4. Since we allowed clustering a maximum of 4 topics or clusters at any given iteration the maximum number of branches at any node was 4.
3. *Cluster Size*: The maximum number of topics in any given cluster was 22 and on the average there were 2.7 topics per cluster.

Although these statistics are useful in evaluating the qual-

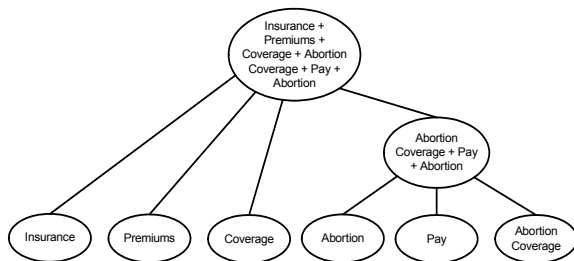


Figure 3: Sample topic sub-tree from 20 NG corpus.

ity of the topic tree, ultimately evaluation with real users is required to measure the effectiveness of the UTC tree.

7 Conclusions and Future Work

In this paper, we have demonstrated that HMM based topic classification delivers state-of-the-art performance while classifying newsgroup messages. We have shown that off-topic message rejection based on topic-specific thresholds outperforms topic-independent thresholds. For completeness, we plan to compare our rejection approach with support vector machines. For automatic categorization and navigation of unstructured text, we introduced a novel concept of unsupervised topic clustering. Preliminary clustering experiments were performed to compare different topic similarity measures. Recently, we have developed a demonstration prototype based on the UTC tree, which we will use to evaluate the impact of this technology on user experience.

8 References

- [20NG Corpus] The 20 Newsgroup Corpus, <http://www.people.csail.mit.edu/jrennie/20Newsgroups/>.
- [Baker *et al.*, 1998] L. D. Baker and A. McCallum. Distributional Clustering of Words for Text Classification. In *Proceedings of ACM SIGIR*, 1998.
- [Biswas *et al.*, 1998] G. Biswas, J. Weinberg, and D. Fisher. ITERATE: A Conceptual Clustering Algorithm for Data Mining. *IEEE Transactions on Systems, Man, and Cybernetics Reviews*, Vol. 28, No.2, May 1998.
- [DONLP2] Differentiable Nonlinear Optimization, <http://www.sai.msu.su/sal/B/3/DONLP2.html>.
- [Eick *et al.*, 2005] S. Eick, J. Lockwood, R. Loui, J. Moscola, C. Kastner, A. Levine, and D. Weishar. Transformation Algorithms for Data Streams. In *Proceedings of IEEE AAC*, March 2005.
- [Joachims, 1998] T. Joachims. Text Categorization with Support Vector Machines. In *Proceedings of ECML '98*.
- [Johnson *et al.*, 2002] D. H. Johnson and S. Sinanovic. Symmetrizing the Kullback Leibler Distance. Rice University Working Paper, 2001.
- [Li *et al.*, 1998] H. Li and N. Abe. Word Clustering and Disambiguation Based on Co-occurrence Data. In *Proceedings of COLING-ACL '98*, pages 749-755, 1998.
- [Rennie *et al.*, 2003] J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In *Proceeding of ICML 2003*, Washington, D.C., 2003.
- [Schwartz *et al.*, 1997] R. Schwartz, T. Imai, F. Kubala, L. Nguyen, and J. Makhoul. A Maximum Likelihood Model for Topic Classification of Broadcast News. In *Proceedings of Eurospeech*, Greece, 1997.
- [Sista *et al.*, 2002] S. Sista, R. Schwartz, T. Leek, and J. Makhoul. An Algorithm for Unsupervised Topic Discovery from Broadcast News Stories. In *Proceedings of ACM HLT*, San Diego, CA, 2002.