Topic Analysis Using a Finite Mixture Model

Hang Li and Kenji Yamanishi
NEC Corporation
{lihang,yamanisi}@ccm.cl.nec.co.jp

Abstract

We address the issue of 'topic analysis,' by which is determined a text's topic structure, which indicates what topics are included in a text, and how topics change within the text. We propose a novel approach to this issue, one based on statistical modeling and learning. We represent topics by means of word clusters, and employ a finite mixture model to represent a word distribution within a text. Our experimental results indicate that our method significantly outperforms a method that combines existing techniques.

1 Introduction

We consider here the issue of 'topic analysis,' by which is determined a text's *topic structure*, which indicates what topics are included in a text and how topics change within the text. Topic analysis consists of two main tasks: topic identification and text segmentation (based on topic changes).

Topic analysis is extremely useful in a variety of text processing applications. For example, it can be used in the automatic indexing of texts for purposes of information retrieval. With it, one can understand what the main topics and subtopics of a text are, and where those subtopics lie within the text.

To the best of our knowledge, however, no previous study has so far dealt with the topic analysis problem in the above sense. The most closely related are key word extraction and text segmentation. A keyword extraction method (e.g., that using tf-idf (Salton and Yang, 1973)) generally extracts from a text key words which represent topics within the text, but it does not conduct segmentation. A segmentation method (e.g., TextTiling (Hearst, 1997)) generally segments a text into blocks (paragraphs) in accord with topic changes within the text, but it does not identify (or label) by itself the topics discussed in each of the blocks.

The purpose of this paper is to provide a single framework for conducting topic analysis, i.e., performing both topic identification and text segmentation.

The key characteristics of our framework are 1) representing a topic by means of a *cluster* of words that are closely related to the topic, and 2) employing a stochastic model, called a *finite mixture model* (e.g., (Everitt and Hand, 1981)), to represent a word distribution within a text. The finite mixture model has a hierarchical structure of probability distributions. The first level is a probability distribution of topics (topic distribution). The second level consists of probability distributions of words included within topics (word distributions). These word distributions are linearly combined to represent a word distribution within a text, with the topic distribution being used as the coefficient vector. Hereafter we refer to a finite mixture model having this structure as a *stochastic topic model* (STM).

Before conducting topic analysis, we create word clusters (topics) on the basis of word cooccurrence in corpus data. We have developed a new method for word clustering using *stochastic complexity* (or the *MDL* principle) (Rissanen, 1996).

In topic analysis, we estimate a sequence of STMs that would have given rise to a given text, assuming that each block of a text is generated by an individual STM. We perform text segmentation by detecting significant differences between STMs and perform topic identification by means of estimation of STMs. With the results, we obtain the text's topic structure which consists of segmented blocks and their topics.

It is possible to perform topic analysis by combining an existing word extraction method (e.g., tf-idf) and an existing text seg-

mentation method (e.g., TextTiling). Specifically, one can extract key words from a text using tf-idf, view these extracted key words as topics, segment the text into blocks using TextTiling, and estimate the distribution of topics (key words) within each block. Experimental results indicate, however, that our method significantly outperforms such a combined method in topic identification and outperforms it in text segmentation, because it utilizes word cluster information and employs a well-defined probability framework.

Finite mixture models have been employed in a number of text processing applications, such as text classification (e.g., (Li and Yamanishi, 1997; Nigam et al., 2000)) and information retrieval (e.g., (Hofmann, 1999)). As will be discussed, however, our definition of a finite mixture model and the way we use it here differs significantly.

2 Stochastic Topic Model

2.1 Topic

While the term 'topic' is used in different ways in different linguistic theories, we simply view it here as a subject within a text. We represent a topic by means of a cluster of words that are closely related to the topic, assuming that a cluster has a seed word (or several seed words) which indicates a topic. Figure 1 shows an example topic with the word 'trade' being the seed word.

trade: trade export import tariff trader GATT protectionist

Figure 1: Example topic

2.2 Definition of STM

Let W denote a set of words, and K a set of topics. We first define a distribution of topics (clusters) $P(k) : \sum_{k \in K} P(k) = 1$. Then, for each topic $k \in K$, we define a probability distribution of words $P(w|k) : \sum_{w \in W} P(w|k) = 1$. Here the value of $P(w|k)$ will be zero if w is not included in k. We next define a Stochastic Topic Model (STM) as a finite mixture model, which is a linear combination of the word probability distributions $P(w|k)$, with the topic distribution $P(k)$ being used as the coefficient vector. The probability of word w in W is, then,

$$P(w) = \sum_{k \in K} P(k)P(w|k) \quad w \in W.$$

Figure 2 depicts an example STM.

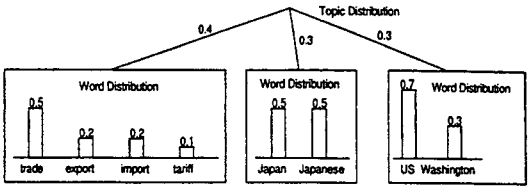

Figure 2: Example STM

For the purposes of statistical modeling, it is advantageous to conceive of a text (i.e., a word sequence) as having been generated by some 'true' STMs, which we then seek to estimate as closely as possible. A text may have a number of blocks, and each block is assumed to be generated by an individual STM. The STMs within a text are assumed to have the same set of topics, but have different parameter values.

From the linguistic viewpoint, a text generally focuses on a single main topic, but it may discuss different subtopics in different blocks. While a text is discussing any one topic, it will more frequently use words strongly related to that topic. Hence, STM is a natural representation of statistical word occurrence based on topics.

3 Word Clustering

Before conducting topic analysis, we create word clusters using a large data corpus. More precisely, we treat all words in a vocabulary as seed words, and for each seed word we collect from the data those words which frequently co-occur with it and group them into a cluster. As one example, the word-cluster in Figure 1 has been constructed with the word 'trade' as the seed word.

We have developed a new method for reliably collecting frequently co-occurring words on the basis of *stochastic complexity*, or the *MDL* principle. For a given data sequence $x^m = x_1 \ldots x_m$ and for a fixed probability model M,[1] the stochastic complexity of x^m relative to M, which we denote as $SC(x^m : M)$, is defined as the least code length required to encode x^m with M (Rissanen, 1996). $SC(x^m : M)$ can be interpreted as the amount information included in x^n relative to M. The

[1] Here, we use 'model' to refer to a probability distribution which has specified parameters but unspecified parameter values.

MDL (Minimum Description Length) principle is a model selection criterion which asserts that, for a given data sequence, the lower a model's SC value, the greater its likelihood of being a model which would have actually generated the data. MDL has many good properties as a criterion for model selection.[2]

For a fixed seed word s, we take a word w as a frequently co-occurring word if the presence of s is a statistically significant indicator of the presence of w.

Let a data sequence: $(s_1, w_1), (s_2, w_2), \cdots, (s_m, w_m)$ be given where (s_i, w_i) denotes the state of co-occurrence of words s and w in the i-th text in the corpus data. Here, $s_i \in \{1,0\}, w_i \in \{1,0\}, (i = 1, \cdots, m)$, 1 denotes the presence of a word, while 0 the absence of it. We further denote $s^m = s_1 \cdots s_m$, and $w^m = w_1 \cdots w_m$.

Then as in (Rissanen, 1996), the SC value of w^m relative to a model I in which the presence or absence of w is independent from those of s (i.e., a Bernoulli model), is calculated as

$$SC(w^m : I) = mH\left(\frac{m^+}{m}\right) + \frac{1}{2}\log\frac{m}{2\pi} + \log\pi,$$

where m^+ denotes the number of 1's in w^m. Here, log denotes the logarithm to the base 2, π the circular constant, and $H(z) \stackrel{def}{=} -z\log z - (1-z)\log(1-z)$, when $0 < z < 1$; $H(z) \stackrel{def}{=} 0$, when $z = 0$ or $z = 1$.

Let w^{m_s} be the sequence of all w_i's ($w_i \in w^m$) such that its corresponding s_i is 1, where m_s denotes the number of 1's in s^m. Let $w^{m_{\neg s}}$ be the sequence of all w_i's ($w_i \in w^m$) such that its corresponding s_i is 0, where $m_{\neg s}$ denotes the number 0's in s^m. The SC value of w^m relative to a model D in which the presence or absence of w is dependent on those of s is then calculated as

$$SC(w^m : D) = \left(m_s H\left(\frac{m_s^+}{m_s}\right) + \frac{1}{2}\log\frac{m_s}{2\pi} + \log\pi\right) + \left(m_{\neg s} H\left(\frac{m_{\neg s}^+}{m_{\neg s}}\right) + \frac{1}{2}\log\frac{m_{\neg s}}{2\pi} + \log\pi\right),$$

where m_s^+ denotes the number of 1's in w^{m_s}, and $w_{\neg s}^+$ the number of 1's in $w^{m_{\neg s}}$.

[2]For an introduction to MDL, see (Li, 1998).

We can then calculate

$$\delta SC = \frac{1}{m}\Big(SC(w^m : I) - SC(w^m : D)\Big) = \left[H\left(\frac{m^+}{m}\right) - \frac{m_w}{m}H\left(\frac{m_w^+}{m_w}\right) - \frac{m_{\neg w}}{m}H\left(\frac{m_{\neg w}^+}{m_{\neg w}}\right)\right] - \left\{\frac{1}{2m}\log\frac{m_w m_{\neg w}\pi}{2m}\right\}. \quad (1)$$

According to the MDL principle, the larger the δSC value, the more likely that the presence or absence of w is dependent on those of s.[3]

Actually, we may think of a word w for which the value of δSC is larger than a predetermined threshold γ and $P(w|s) > P(w)$ is satisfied as that which occurs significantly frequently with the seed word s.

Note that the word clustering process is independent of topic analysis. While one could employ other methods (e.g., (Hofmann, 1999)) here for word clustering, our clustering algorithm is more efficient than conventional ones. For example, Hofmann's is of order $O(|D||W|^2)$, while ours is only of $O(|D|+|W|^2)$, where $|D|$ denotes the number of texts and $|W|$ the number of words. That means that our method is more practical when a large amount of text data is available.

4 Topic Analysis

4.1 Input and Output

In topic analysis, we use STM to parse a given text and output a topic structure which consists of segmented blocks and their topics. Figure 3 shows an example topic structure as output with our method. The text has been segmented into five blocks, and to each block, a number of topics having high probability values have been assigned (topics are represented by their seed words). The topic structure clearly represents what topics are included in the text and how the topics change within the text.

4.2 Outline

Our topic analysis consists of three processes: a pre-process called 'topic spotting,' text segmentation, and topic identification. In topic

[3]Note that the quantity within $[\cdots]$ in (1) is *(empirical) mutual information*, which is an effective measure for word co-occurrence calculation (cf.,(Brown et al., 1992)). When the sample size is small, mutual information values tend to be undesirably large. The quantity within $\{\cdots\}$ in (1) can help avoid this undesirable tendency because its value will become large when data size is small.

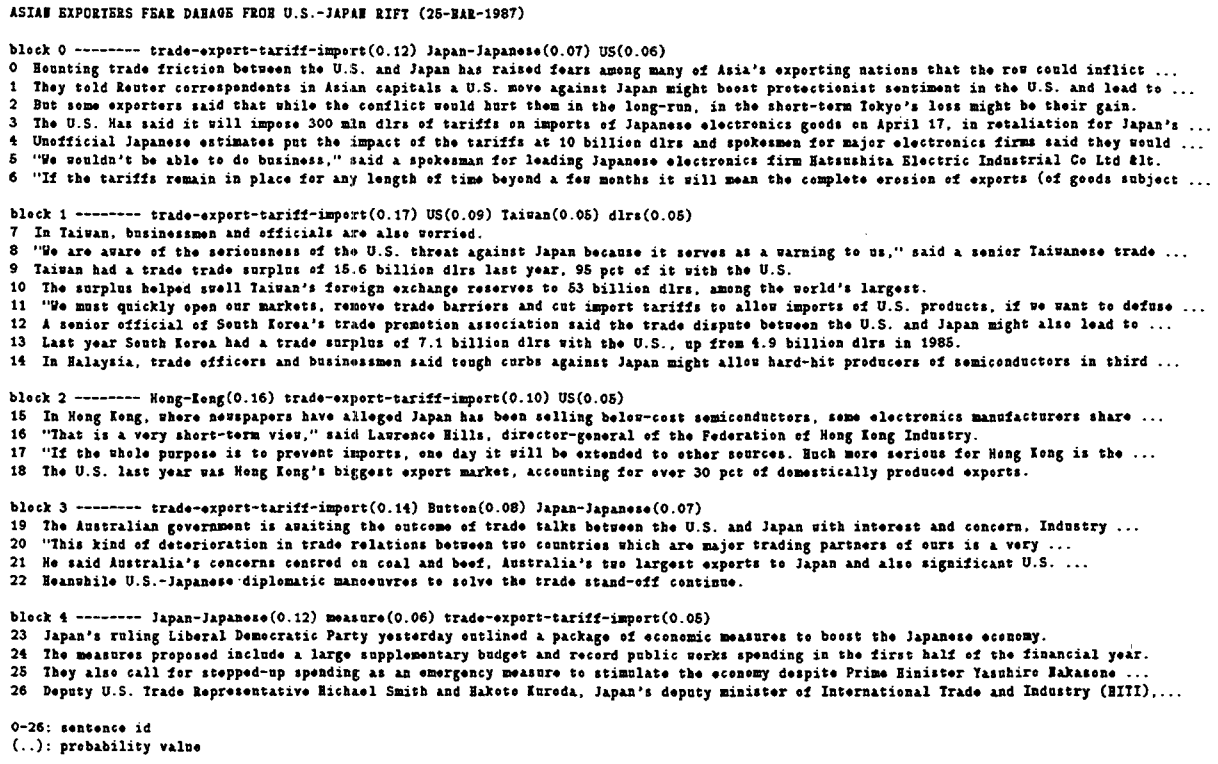

Figure 3: Topic structure of text

spotting, we select topics discussed in a given text. We can then construct STMs on the basis of the topics. In text segmentation, we segment the text on the basis of the STMs, assuming that each block is generated by an individual STM. In topic identification, we estimate the parameters of the STM for each segmented block and select topics with high probabilities for the block. In this way, we obtain a topic structure for the text.

4.3 Topic Spotting

In topic spotting, we first select key words from a given text. We calculate what we call the *Shannon information* of each word in the text. The Shannon information of word w in text t is defined as

$$I(w) = -N(w)\log P(w),$$

where $N(w)$ denotes the frequency of w in t, and $P(w)$ the probability of the occurrence of w as estimated from corpus data. $I(w)$ may be interpreted as the amount of information represented by w. We select as key words the top l words sorted in descending order of I.

While Shannon information is similar to the tf-idf widely used in information retrieval (e.g., (Salton and Yang, 1973)), the use of Shannon information can be justified on the basis of information theory, but that of tf-idf cannot. Our preliminary experimental results indicate that Shannon information performs better than or at least as well as tf-idf in key word extraction.[4]

From the results of word clustering, we next select any cluster (topic) whose seed word is included among the selected key words.

We next merge any two clusters if one of their seed words is included in the other's cluster. For example, when a cluster with seed word 'trade' contains the word 'import,' and a cluster with seed word 'import' contains the word 'trade,' we merge the two. After two such merges, we may obtain a relatively large cluster with, for example, 'trade-import-tariff-export' as its seed words, as is shown in Figure 3. Figure 4 shows the merging algorithm.

In this way, we obtain the most conspicuous and mutually independent topics discussed in a given text.

4.4 Text Segmentation

In segmentation, we first identify candidates for points of segmentation within the given text. When we assume a relatively short text

[4] We will discuss it in the full version of the paper.

$k_1, \ldots, k_n$: clusters,
$V = \{\{k_i\}, i = 1, 2, \cdots, n\}$.
For each cluster pair (k_i, k_j), if the seed word of k_i is included in k_j and the seed word of k_j is included in k_i, then push (k_i, k_j) into queue Q;
while $(Q \neq \emptyset)$ {
  Remove the first element (k_i, k_j) from Q;
  if (k_i and k_j belong to different sets W_1,W_2 in V)
    Replace W_1 and W_2 in V with $W_1 \cup W_2$;
}
For each element W of V, merge the clusters in it.

Figure 4: Algorithm: merge

for the purposes of our explanation here, all sentence-ending periods will be candidates. For each candidate, we create two pseudo-texts, one consisting of the h sentences preceding it, and the other of the h sentences following it (when fewer than h exist in any direction, we simply use those which do exist). We use the *EM* algorithm ((Dempster et al., 1977), cf., Figure 5) to separately estimate the parameters of an STM from each of the two pseudo texts. It is theoretically guaranteed that the EM algorithm converges to a local maximum of the likelihood. We next calculate the similarity (i.e., essentially the converse notion of distance[5]) between the STM based on the preceding pseudo-text, and the STM based on the following pseudo-text. These STMs are denoted, respectively, as $P_L(w)$ and $P_R(w)$. The similarity between $P_L(w)$ and $P_R(w)$ is defined as

$$S(L||R) = 1 - \frac{\sum_{w \in W} |P_L(w) - P_R(w)|}{2}.$$

The numerator is referred to in statistics as *variational distance* and has good properties as a distance between two probability distributions (cf., (Cover and Thomas, 1991), p.299).

Figure 7 shows a graph of calculated similarity values for each of the candidates in the

[5]We use similarity rather than distance here in order to simplify comparison between our method and TextTiling (Hearst, 1997).

s: predetermined number.
For the lth iteration ($l = 1, \cdots, s$), we calculate

$$P^{(l+1)}(k|w) = \frac{P^{(l)}(k)P^{(l)}(w|k)}{\sum_{k \in K} P^{(l)}(k)P^{(l)}(w|k)}$$

$$P^{(l+1)}(k) = \frac{N(w)P^{(l+1)}(k|w)}{N}$$

$$P^{(l+1)}(w|k) = \frac{N(w)P^{(l+1)}(k|w)}{\sum_{w \in W} N(w)P^{(l+1)}(k|w)}$$

$N(w)$ denotes the frequency of word w in the data; $N = \sum_{w \in W} N(w)$.

Figure 5: EM algorithm

n: number of segmentation candidates,
$S(i)$ $i(i = 0 \ldots n)$: similarity score.
for $(i = 1; i < n - 1; i++)$ {
  if $(S(i-1) > S(i)$ & $S(i+1) > S(i))$ {
    $j = i - 1$;
    while $(j > 0$ & $S(j-1) > S(j))$
      $j--$;
    $P1 = S(j)$;
    $j = i + 1$;
    while$(j < n$ & $S(j+1) > S(j))$
      $j++$;
    $P2 = S(j)$;
    if $(P1 - S(i) > \theta$ & $P2 - S(i) > \theta)$
      Conduct segmentation at i.
}}

Figure 6: Algorithm: segment

text shown in Figure 3. 'Valleys' (i.e., low-similarity values) in the graph suggest points for reasonable segmentations. In actual practice, segmentation is performed for each valley whose similarity values is lower to a predetermined degree θ than each of the values of its left 'peak' and right 'peak' (cf., Figure 6) For example, for the text in Figure 3, segmentation was performed at candidates (i.e., end of sentences) 6, 14, 18, and 22, with $\theta = 0.05$.

4.5 Topic Identification

After segmentation, we separately estimate the parameters of the STM for each block, again using the EM algorithm, and obtain a topic (cluster) probability distribution for each block. We then choose those topics (clusters) in each block having high probability values. In this way, we construct a topic struc-

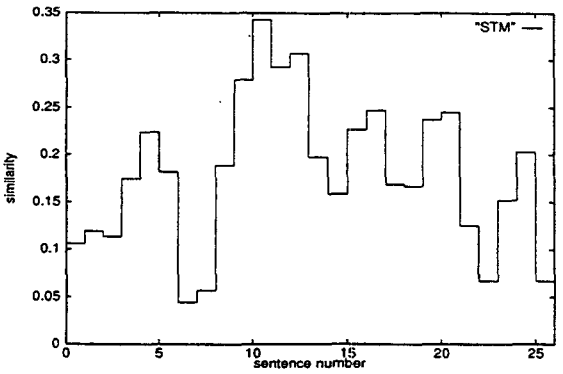

Figure 7: Similarity values for segmentation candidates

ture as in Figure 3 for the given text (topics are here represented by their seed words).

We can view topics appearing in all the blocks as main topics, and topics appearing only in individual blocks as subtopics. In the text in Figure 3, the topic represented by seed-words 'trade-export-tariff-import' is the main topic, and 'Japan-Japanese,' 'Hong Kong,' etc., are subtopics.

5 Applications

Our method can be used in a variety of text processing applications.

For example, given a collection of texts (e.g., home pages), we can automatically construct an index of the texts on the basis of the extracted topics. We can indicate which topic is from which text or even which block of a text. Furthermore, we can indicate which topics are main topics of texts and which topics are subtopics (e.g., by displaying main topics in boldface, etc). In this way, users can get a fair sense of the contents of the texts simply by looking through the index. For a specific text, users can get a rough sense of the content by looking at the topic structure as, for example, it is shown in Figure 3.

Our method can also be useful for text mining, text summarization, information extraction, and other text processing, which require one to first analyze the structure of a text.

6 Related Work

To the best of our knowledge, no previous study has so far dealt with topic identification and text segmentation within a single framework.

A widely used method for key word extraction calculates the tf-idf value of each word in a text and uses those words having the largest tf-idf values as key words for that text (e.g., (Salton and Yang, 1973)). One can view these extracted key words as the topics of the text. No keyword extraction method by itself, however, is able to conduct segmentation.

With respect to text segmentation, existing methods can be classified into two groups. One is to divide a text into blocks (e.g., TextTiling (Hearst, 1997)), the other to divide a stream of texts into its original texts (e.g.,(Allan et al., 1998; Yamron et al., 1998; Beeferman et al., 1999; Reynar, 1999)). The former group generally employs unsupervised learning, while the latter supervised one. No existing segmentation method, however, has attempted topic identification.

TextTiling creates for each segmentation candidate two pseudo-texts, one preceding it and the other following it, and calculates as similarity the cosine value between the word frequency vectors of the two pseudo texts. It then conducts segmentation at valley points in a similar way to that of our method. Since the problem setting of TextTiling (in general the former group) is most close to that of our study, we use TextTiling for comparison in our experiments.

Our method by its nature performs topic identification and segmentation within a single framework. While it is possible with a combination of existing methods to extract key words from a given text by using tf-idf, view the extracted key words as topics, segment the text into blocks by employing TextTiling, estimate distribution of topics in each block, and identify topics having high probabilities in each block. Our method outperforms such a combination (referred to hereafter as 'Com') for topic identification, because it utilizes word cluster information. It also performs better than Com in text segmentation because it is based on a well-defined probability framework. Most importantly is that our method is able to output an easily understandable topic structure, which has not been proposed so far.

Note that topic analysis is different from text classification (e.g., (Lewis et al., 1996; Li and Yamanishi, 1999; Joachims, 1998; Weiss et al., 1999; Nigam et al., 2000)). While text classification uses a number of pre-determined categories, topic analysis includes no notion of category. The output of topic analysis is a topic structure, while the output of text clas-

sification is a label representing a category. Furthermore, text classification is generally based on supervised learning, which uses labeled text data[6]. By way of contrast, topic analysis is based on unsupervised learning, which uses only *unlabeled* text data.

Finite mixture models have been used in a variety of applications in text processing (e.g., (Li and Yamanishi, 1997; Nigam et al., 2000; Hofmann, 1999)), indicating that they are essential to text processing. We should note, however, that their definitions and the ways they use them are different from those for STM in this paper. For example, Li and Yamanishi propose to employ in text classification a mixture model (Li and Yamanishi, 1997) *defined over categories*:

$$P(w|c) = \sum_{k \in K} P(k|c)P(w|k), w \in W, c \in C,$$

where W denotes a set of words, and C a set of categories. In their framework, a new text d is assigned into a category c^* such that $c^* = \arg\max_{c \in C} P(c|d)$ is satisfied. Hofmann proposes using in information retrieval a joint distribution which he calls 'an aspect model,' defined as (Hofmann, 1999)

$$\begin{aligned} P(w,d) &= P(d)P(w|d) \\ &= P(d)\sum_{k \in K} P(k|d)P(w|k), \\ & \quad w \in W, d \in D \end{aligned}$$

where D denotes a set of texts. Furthermore, he proposes extracting in retrieval those texts whose estimated word distributions $P(w|d)$ are similar to the word distribution of a query.

7 Experimental Results

We have evaluated the performance of our topic analysis method (STM) in terms of three aspects: topic structure adequacy, text segmentation accuracy, and topic identification accuracy.

7.1 Data Set

We know of no data available for the purpose of evaluation of topic analysis. We thus utilized Reuters news articles referred to as 'Reuters-21578,' which has been widely used in text classification[7]. We used a prepared split of the data 'Apte split,' which consists of 9603 texts for training and 3299 texts for test. All of the texts had already been classified into 90 categories by human subjects.

For each text, we used the Oxford Learner's Dictionary[8] to conduct stemming, and removed 'stop words' (e.g., 'the,' 'and') that we had included on a previously prepared list. The average length of a text was about 115 words. (We did not use phrases, however, which would further improve experimental results.)

7.2 Word Clustering

We conducted word clustering with 9603 training texts. 7340 individual words had a total frequency of more than 5, and we used them as seeds with which to collect frequently co-occurring words. The threshold for clustering γ was set at 0.005, and this yielded 970 word clusters having more than one word (i.e., not simply containing a seed word alone). Note that the category labels of the training texts need not be used in clustering.

We next conducted a topic analysis on all the 3299 texts. The thresholds of l, h, and θ were set at 20, 3, and 0.05, respectively, on the basis of preliminary experimental results.

7.3 Topic Structure

We looked at the topic structures of the 3299 texts obtained by our method to determine how well they conformed to human intuition.

For topic identification in this experiment, clusters in each block were sorted in descending order of their probabilities, and the top 7 seed words were extracted to represent the topics of the block.

Figure 3 show results for the text with ID 14826; they generally agree well with human intuition. The text has been segmented into 5 blocks and the topics of each block is represented by 7 seed words. The main topic is represented by the seed-words 'trade-export-tariff-import.' The subtopics are represented by 'Japan-Japanese,' 'Taiwan,' 'Hong Kong,' etc. There were, however, a small number of errors. For example, the text should also have been segmented after sentences 11 and 13, but, due to limited sentence content, it was not. Furthermore, assigning subtopic of 'Button' (from 'Mr. Button') into block 3 (due to the high Shannon information value of the word 'Button') was also undesirable.

[6] An exception is the method proposed in (McCallum and Nigam, 1999), which, instead of labeled texts, uses unlabeled texts, pre-determined categories, and keywords defined by humans for each category.

[7] Available at http://www.research.att.com/lewis/.

[8] Available at ftp://sable.ox.ac.uk.

Table 1: 10 categories and their identification words

category	identification words
earn	earning, share, profit, dividend
acq	acquisition, acquire, sell, buy
money-fx	currency, dollar, yen, stg
grain	grain, cereal, crop
crude	oil, crude, gas
trade	trade, export, import, tariff
interest	interest & rate
ship	ship, vessel, ferry, tanker
wheat	wheat
corn	corn, maize

7.4 Main Topic Identification

We conducted an evaluation to determine whether or not the main topics in the topic structures obtained for the 3299 test texts could be approximately matched with the labels (categories) assigned to the test texts.

Note that here labels are used only for evaluation, not for training. This is in contrast to the situation in most text classification experiments, in which labels are generally used both for training and for evaluation. It is not particularly meaningful, then, to compare the results for main topic identification obtained here with those for text classification.

With STM, clusters in each block were sorted in descending order of their probabilities, and the top k seed words were extracted to represent the topics of the block. Furthermore, a seed word appearing in all the blocks of the text was considered to represent a main topic. When a text had not been segmented (i.e., has only one block), all top k seed words were considered to represent main topics.

Table 1 lists the largest 10 categories in the Reuters data. On the basis of the definition of each of the 10 categories, we assigned based on our intuition to each of them the identification words that are listed in Table 1.

For the evaluation, when the seed words for main topics contained at least one of the identification words, we considered our method to have identified the corresponding main topic equivalent to a human-determined category.

We then evaluated these in terms of *precision* and *recall*. Here, precision is defined as the ratio of the number of decisions correctly made to the total number of decisions made. Recall is defined as the ratio of the number of decisions correctly made to the total number of decisions which should have been correctly made.

Table 2: Main topic identification results with respect to 7 top words

category	STM		Com	
	rec.	pre.	rec.	pre.
earn	0.790	0.971	0.526	0.976
acq	0.245	0.854	0.184	0.841
money-fx	0.436	0.456	0.285	0.421
grain	0.322	0.750	0.174	0.650
crude	0.487	0.676	0.407	0.664
trade	0.667	0.473	0.590	0.356
interest	0.107	0.700	0.084	0.733
ship	0.247	0.957	0.270	0.828
wheat	0.620	0.936	0.408	0.967
corn	0.429	0.960	0.446	1.00
micro-average	**0.515**	**0.824**	0.365	0.774

Table 3: Main topic identification results with respect to 5 top words

category	STM		Com	
	rec.	pre.	rec.	pre.
earn	0.742	0.971	0.348	0.977
acq	0.184	0.868	0.120	0.869
money-fx	0.413	0.503	0.268	0.471
grain	0.295	0.759	0.121	0.600
crude	0.471	0.718	0.333	0.656
trade	0.479	0.505	0.513	0.403
interest	0.053	0.700	0.069	0.818
ship	0.169	1.000	0.180	0.762
wheat	0.577	0.953	0.282	0.952
corn	0.357	0.952	0.321	1.000
micro-average	**0.461**	**0.850**	0.257	0.767

We also looked at the performance of Com (cf., Section 6). For Com, we extracted from a text the key words with the 20 largest Shannon information values, segmented the text using TextTiling, and extracted in each block the key words having the largest k probability values. Any key word extracted in all blocks was considered to represent a main topic. When the key words for main topics contained at least one of the identification words, we viewed that text as having the corresponding main topic.

Table 2 shows the results achieved with STM and Com in the case of $k = 7$.[9] Table 3

[9] For the definition of *micro-averaging*, see, for ex-

```
Title: EGYPT BUYS PL 480 WHEAT FLOUR - U.S. TRADERS

Body: Egypt bought 125,723 tonnes of U.S. wheat flour in its PL
480 tender yesterday, trade sources said. The purchase included
51,880 tonnes for May shipment and 73,843 tonnes for June shipment.
Price details were not available.

Content Words (Freq.): tonne(3) shipment(2) buy(1) detail(1)
Egypt(1) flour(1) include(1) June(1) PL(1) price(1) purchase(1)
source(1) trade(1) US(1) wheat(1)

Key Words (Shan. Inf.): tonne(17.3) shipment(15.3) PL(10.5) flour(9.8)
Egypt(9.3) detail(7.5) June(7.2) wheat(6.8) purchase(6.6) source(6.5)
US(6.1) buy(6.0) include(6.0) trade(5.3) price (5.1)

Com Topics (Prob.): tonne(0.17) shipment(0.11) price(0.06) June(0.06)
include(0.06) purchase(0.06) source(0.06)

STM Topics (Prob.) : flour-wheat(0.15) tonne(0.12) shipment(0.11)
purchase-buy(0.11) Egypt(0.06)

Cluster: (flour-wheat: wheat tonne flour)
         (purchase-buy: purchase buy)
```

Figure 8: Topic Identification Example

shows the results in the case of $k = 5$. The comparison may be considered fair in that it requires each of the two methods to provide the same number of words to represent topics. Results indicate that STM significantly outperforms Com, particularly in terms of recall.

The main reason for the higher performance achieved by STM is that it utilizes word cluster information. Figure 8 shows topic analysis results for the text with ID 15572 labeled with 'wheat.' The text contains only 15 content words (word types), thus all of the 15 words were extracted as key words and the text was not segmented by either method. Com was unable to identify the main topic 'wheat,' because the probability of each of the relevant key words 'wheat' and 'flour' was low. In contrast, STM successfully identified the topic because the relevant key words were classified into the same cluster, and its probability was relatively high.

7.5 Segmentation and Subtopic Identification

We collected the 50 longest test texts (referred to here as 'seed texts') from each of the 10 categories, and combined each with a test text randomly selected from other categories to produce 500 pseudo-texts. Placement of the seed text within its pseudo-text (i.e., before or after the other text) was determined randomly.

We used both STM and Com to segment each of the pseudo-texts into two blocks and identify subtopics. Table 4 shows the segmentation results for the two method evaluated

ample, (Lewis and Ringuette, 1994).

Table 5: Subtopic identification results

category of seed text	STM		Com	
	rec.	pre.	rec.	pre.
earn	0.430	0.945	0.324	0.973
acq	0.237	0.939	0.217	0.959
money-fx	0.585	0.950	0.533	0.961
grain	0.276	0.947	0.222	0.938
crude	0.572	0.979	0.557	0.990
trade	0.634	0.951	0.627	0.899
interest	0.211	0.937	0.136	1.000
ship	0.260	1.000	0.340	0.994
wheat	0.500	0.970	0.395	0.980
corn	0.317	1.000	0.441	0.882
Average	**0.402**	**0.962**	0.379	0.958

in terms of recall, precision, and error probability. Table 5 shows the results of subtopic identification as evaluated in terms of recall and precision. *Error probability* is a metric for evaluating segmentation results proposed in (Allan et al., 1998; Beeferman et al., 1999). It is defined here as the probability that a randomly chosen pair of sentences a distance of k sentence apart is incorrectly segmented.[10]

Experimental results indicate that STM outperforms Com in both segmentation and identification.[11]

8 Conclusions

We have proposed a new method of topic analysis that employs a finite mixture model, referred to here as a stochastic topic model (STM).

Topic analysis consists of two main tasks: text segmentation and topic identification. With topic analysis, one can obtain a topic structure for a text.

Our method addresses topic analysis within a single framework. It has the following novel features: 1) it represents topics by means of word clusters and employs a finite mixture model (STM) to represent a word distribution within a text; 2) it constructs topics on the basis of corpus data before conducting topic analysis; 3) it segments a text by detecting significant differences between STMs; and 4) it identifies topics by estimating parameters

[10] Here, k was set to 5 because the average length of a text was about 10 sentences.

[11] We will discuss the results in the full version of the paper.

Table 4: Text segmentation results

category of seed text	STM			Com		
	rec.	pre.	err.	rec.	pre.	err.
earn	0.660	0.660	0.167	0.640	0.640	0.171
acq	0.820	0.820	0.059	0.740	0.740	0.085
money-fx	0.700	0.700	0.087	0.660	0.660	0.121
grain	0.700	0.700	0.074	0.660	0.660	0.076
crude	0.860	0.860	0.051	0.820	0.820	0.066
trade	0.800	0.800	0.072	0.800	0.800	0.081
interest	0.760	0.760	0.119	0.820	0.820	0.084
ship	0.837	0.854	0.074	0.816	0.833	0.084
wheat	0.760	0.760	0.075	0.640	0.640	0.130
corn	0.625	0.625	0.147	0.650	0.650	0.105
Average	**0.752**	**0.754**	**0.092**	0.725	0.726	0.100

of STMs.

Experimental results indicate that our method outperforms a method that combines existing techniques. More specifically, it significantly outperforms the combined method in topic identification.

References

J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. 1998. Topic detection and tracking pilot study: Final report. *Proc. of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218.

D. Beeferman, A. Berger, and J. Lafferty. 1999. Statistical models for text segmentation. *Machi. Lrn.*, 34:177–210.

P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer. 1992. Class-based n-gram models of natural language. *Comp. Ling.*, 18(4):283–298.

T. M. Cover and J. A. Thomas. 1991. *Elements of Information Theory.* John Wiley & Sons Inc., New York.

A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journ. of Roy. Stat. Soci., Ser. B*, 39(1):1–38.

B. Everitt and D. Hand. 1981. *Finite Mixture Distributions.* Chapman and Hall.

M. Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Comp. Ling.*, 23(1):33–64.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. *Proc. of SIGIR'99*, pages 50–57.

T. Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. *Proc. of ECML'98.*

D. D. Lewis and M. Ringuette. 1994. A comparison of two learning algorithms for test categorization. *Proc. of 3rd Ann. Symp. on Doc. Ana. and Info. Retr.*, pages 81–93.

D. D. Lewis, R. E. Schapire, J. P. Callan, and R. Papka. 1996. Training algorithms for linear text classifiers. *Proc. of SIGIR'96.*

H. Li and K. Yamanishi. 1997. Document classification using a finite mixture model. *Proc. of ACL'97*, pages 39–47.

H. Li and K. Yamanishi. 1999. Text classification using ESC-based stochastic decision lists. *Proc. of ACM-CIKM'99*, pages 122–130.

H. Li. 1998. *A Probabilistic Approach to Lexical Semantic Knowledge Acquisition and Structural Disambiguation.* Ph.D. Thesis, Univ. of Tokyo.

A. K. McCallum and K. Nigam. 1999. Text classification by bootstrapping with keywords, em and shrinkage. *Proc. of ACL'99 Workshop Unsupervised Learning in NLP.*

K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. 2000. Text classification from labeled and unlabeled documents using em. *Machi. Lrn.*, 39:103–134.

J. C. Reynar. 1999. Statistical models for topic segmentation. *Proc. of ACL'99*, pages 357–364.

J. Rissanen. 1996. Fisher information and stochastic complexity. *IEEE Trans. on Info. Thry.*, 42(1):40–47.

G. Salton and C.S. Yang. 1973. On the specification of term values in automatic indexing. *Journ. of Doc.*, 29(4):351–372.

S. M. Weiss, C. Apte, F. Damerau, F. J. Oles, T. Goetz, and T. Hampp. 1999. Maximizing text-mining performance. *IEEE Intel. Sys.*, 14(4):63–69.

J.P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt. 1998. A Hidden Markov Model approach to text segmentation and event tracking. *Proc. of ICASSP'99*, pages 333–336.