

# Topic Segmentation with Shared Topic Detection and Alignment of Multiple Documents

Bingjun Sun\*, Prasenjit Mitra\*†, Hongyuan Zha‡, C. Lee Giles\*†, John Yen\*†

\*Department of Computer Science and Engineering

†College of Information Sciences and Technology

The Pennsylvania State University  
University Park, PA 16802

‡College of Computing

The Georgia Institute of Technology  
Atlanta, GA 30332

\*bsun@cse.psu.edu, †{pmitra,giles,jyen}@ist.psu.edu, ‡zha@cc.gatech.edu

## ABSTRACT

Topic detection and tracking [26] and topic segmentation [15] play an important role in capturing the local and sequential information of documents. Previous work in this area usually focuses on single documents, although similar multiple documents are available in many domains. In this paper, we introduce a novel unsupervised method for shared topic detection and topic segmentation of multiple similar documents based on mutual information (MI) and weighted mutual information (WMI) that is a combination of MI and term weights. The basic idea is that the optimal segmentation maximizes MI(or WMI). Our approach can detect shared topics among documents. It can find the optimal boundaries in a document, and align segments among documents at the same time. It also can handle single-document segmentation as a special case of the multi-document segmentation and alignment. Our methods can identify and strengthen cue terms that can be used for segmentation and partially remove stop words by using term weights based on entropy learned from multiple documents. Our experimental results show that our algorithm works well for the tasks of single-document segmentation, shared topic detection, and multi-document segmentation. Utilizing information from multiple documents can tremendously improve the performance of topic segmentation, and using WMI is even better than using MI for the multi-document segmentation.

## Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Clustering*; H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing—*Linguistic processing*; I.2.7 [Artificial Intelligence]: Natural Language Processing—*Text analysis*; I.5.3 [Pattern Recognition]: Clustering—*Algorithms; Similarity measures*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.  
Copyright 2007 ACM 978-1-59593-597-7/07/0007 ...\$5.00.

## General Terms

Algorithms, Design, Experimentation

## Keywords

Topic segmentation, shared topic detection, topic alignment, mutual information, multiple documents, term weight

## 1. INTRODUCTION

Many researchers have worked on topic detection and tracking (TDT) [26] and topic segmentation during the past decade. Topic segmentation intends to identify the boundaries in a document with the goal to capture the latent topical structure. Topic segmentation tasks usually fall into two categories [15]: *text stream segmentation* where topic transition is identified, and *coherent document segmentation* in which documents are split into sub-topics. The former category has applications in automatic speech recognition, while the latter one has more applications such as partial-text query of long documents in information retrieval, text summary, and quality measurement of multiple documents. Previous research in connection with TDT falls into the former category, targeted on topic tracking of broadcast speech data and newswire text, while the latter category has not been studied very well.

Traditional approaches perform topic segmentation on documents one at a time [15, 25, 6]. Most of them perform badly in subtle tasks like coherent document segmentation [15]. Often, end-users seek documents that have the similar content. Search engines, like, Google, provide links to obtain similar pages. At a finer granularity, users may actually be looking to obtain sections of a document similar to a particular section that presumably discusses a topic of the users interest. Thus, the extension of topic segmentation from single documents to identifying similar segments from multiple similar documents with the same topic is a natural and necessary direction, and multi-document topic segmentation is expected to have a better performance since more information is utilized.

Traditional approaches using similarity measurement based on term frequency generally have the same assumption that similar vocabulary tends to be in a coherent topic segment [15, 25, 6]. However, they usually suffer the issue of identifying stop words. For example, additional document-dependent stop words are removed together with the generic stop words in [15]. There are two reasons that we do not remove stop

words directly. First, identifying stop words is another issue [12] that requires estimation in each domain. Removing common stop words may result in the loss of useful information in a specific domain. Second, even though stop words can be identified, hard classification of stop words and non-stop words cannot represent the gradually changing amount of information content of each word. We employ a soft classification using term weights.

In this paper, we view the problem of topic segmentation as an optimization issue using *information theoretic techniques* to find the optimal boundaries of a document given the number of text segments so as to minimize the loss of mutual information (MI) (or a weighted mutual information (WMI)) after segmentation and alignment. This is equal to maximizing the MI (or WMI). The MI focuses on measuring the difference among segments whereas previous research focused on finding the similarity (e.g. cosine distance) of segments [15, 25, 6]. Topic alignment of multiple documents can be achieved by clustering sentences on the same topic into the same cluster. Single-document topic segmentation is just a special case of the multi-document topic segmentation and alignment problem. Terms can be co-clustered as in [10] at the same time, given the number of clusters, but our experimental results show that this method results in a worse segmentation (see Tables 1, 4, and 6). Usually, human readers can identify topic transition based on cue words, and can ignore stop words. Inspired by this, we give each term (or term cluster) a weight based on entropy among different documents and different segments of documents. Not only can this approach increase the contribution of *cue words*, but it can also decrease the effect of *common stop words*, *noisy word*, and *document-dependent stop words*. These words are common in a document. Many methods based on sentence similarity require that these words are removed before topic segmentation can be performed [15]. Our results in Figure 3 show that term weights are useful for multi-document topic segmentation and alignment.

The major contribution of this paper is that it introduces a novel method for topic segmentation using MI and shows that this method performs better than previously used criteria. Also, we have addressed the problem of topic segmentation and alignment across multiple documents, whereas most existing research focused on segmentation of single documents. Multi-document segmentation and alignment can utilize information from similar documents and improves the performance of topic segmentation greatly. Obviously, our approach can handle single documents as a special case when multiple documents are unavailable. It can detect shared topics among documents to judge if they are multiple documents on the same topic. We also introduce the new criterion of WMI based on term weights learned from multiple similar documents, which can improve performance of topic segmentation further. We propose an iterative greedy algorithm based on dynamic programming and show that it works well in practice. Some of our prior work is in [24].

The rest of this paper is organized as follows: In Section 2, we review related work. Section 3 contains a formulation of the problem of topic segmentation and alignment of multiple documents with term co-clustering, a review of the criterion of MI for clustering, and finally an introduction to WMI. In Section 4, we first propose the iterative greedy algorithm of topic segmentation and alignment with term co-clustering, and then describe how the algorithm can be optimized by us-

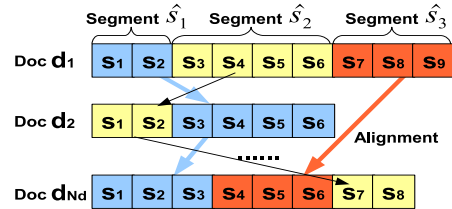


Figure 1: Illustration of multi-document segmentation and alignment.

ing dynamic programming. In Section 5, experiments about single-document segmentation, shared topic detection, and multi-document segmentation are described, and results are presented and discussed to evaluate the performance of our algorithm. Conclusions and some future directions of the research work are discussed in Section 6.

## 2. PREVIOUS WORK

Generally, the existing approaches to text segmentation fall into two categories: supervised learning [19, 17, 23] and unsupervised learning [3, 27, 5, 6, 15, 25, 21]. Supervised learning usually has good performance, since it learns functions from labelled training sets. However, often getting large training sets with manual labels on document sentences is prohibitively expensive, so unsupervised approaches are desired. Some models consider dependence between sentences and sections, such as *Hidden Markov Model* [3, 27], *Maximum Entropy Markov Model* [19], and *Conditional Random Fields* [17], while many other approaches are based on lexical cohesion or similarity of sentences [5, 6, 15, 25, 21]. Some approaches also focus on cue words as hints of topic transitions [11]. While some existing methods only consider information in single documents [6, 15], others utilize multiple documents [16, 14]. There are not many works in the latter category, even though the performance of segmentation is expected to be better with utilization of information from multiple documents. Previous research studied methods to find shared topics [16] and topic segmentation and summarization between just a *pair* of documents [14].

Text classification and clustering is a related research area which categorizes documents into groups using supervised or unsupervised methods. Topical classification or clustering is an important direction in this area, especially co-clustering of documents and terms, such as LSA [9], PLSA [13], and approaches based on distances and bipartite graph partitioning [28] or maximum MI [2, 10], or maximum entropy [1, 18]. Criteria of these approaches can be utilized in the issue of topic segmentation. Some of those methods have been extended into the area of topic segmentation, such as PLSA [5] and maximum entropy [7], but to our best knowledge, using MI for topic segmentation has not been studied.

## 3. PROBLEM FORMULATION

Our goal is to segment documents and align the segments across documents (Figure 1). Let  $T$  be the set of terms  $\{t_1, t_2, \dots, t_l\}$ , which appear in the unlabelled set of documents  $D = \{d_1, d_2, \dots, d_m\}$ . Let  $S_d$  be the set of sentences for document  $d \in D$ , i.e.  $\{s_1, s_2, \dots, s_{n_d}\}$ . We have a 3D matrix of term frequency, in which the three dimensions are random variables of  $D$ ,  $S_d$ , and  $T$ .  $S_d$  actually is a random

vector including a random variable for each  $d \in D$ . The term frequency can be used to estimate the joint probability distribution  $P(D, S_d, T)$ , which is  $p(t, d, s) = T(t, d, s)/N_D$ , where  $T(t, d, s)$  is the number of  $t$  in  $d$ 's sentence  $s$  and  $N_D$  is the total number of terms in  $D$ .  $\hat{S}$  represents the set of segments  $\{\hat{s}_1, \hat{s}_2, \dots, \hat{s}_p\}$  after segmentation and alignment among multiple documents, where the number of segments  $|\hat{S}| = p$ . A segment  $\hat{s}_i$  of document  $d$  is a sequence of adjacent sentences in  $d$ . Since for different documents  $s_i$  may discuss different sub-topics, our goal is to cluster adjacent sentences in each document into segments, and align similar segments among documents, so that for different documents  $\hat{s}_i$  is about the same sub-topic. The goal is to find the optimal topic segmentation and alignment mapping

$$Seg_d(s_i) : \{s_1, s_2, \dots, s_{n_d}\} \rightarrow \{\hat{s}'_1, \hat{s}'_2, \dots, \hat{s}'_p\}$$

and  $Ali_d(\hat{s}'_i) : \{\hat{s}'_1, \hat{s}'_2, \dots, \hat{s}'_p\} \rightarrow \{\hat{s}_1, \hat{s}_2, \dots, \hat{s}_p\}$ , for all  $d \in D$ , where  $\hat{s}_i$  is  $i^{th}$  segment with the constraint that only adjacent sentences can be mapped to the same segment, i.e. for  $d$ ,  $\{s_i, s_{i+1}, \dots, s_j\} \rightarrow \{\hat{s}'_q\}$ , where  $q \in \{1, \dots, p\}$ , where  $p$  is the segment number, and if  $i > j$ , then for  $d$ ,  $\hat{s}_q$  is missing. After segmentation and alignment, random vector  $S_d$  becomes an aligned random variable  $\hat{S}$ . Thus,  $P(D, S_d, T)$  becomes  $P(D, \hat{S}, T)$ .

Term co-clustering is a technique that has been employed [10] to improve the accuracy of document clustering. We evaluate the effect of it for topic segmentation. A term  $t$  is mapped to exactly one term cluster. Term co-clustering involves simultaneously finding the optimal term clustering mapping  $Clu(t) : \{t_1, t_2, \dots, t_l\} \rightarrow \{\hat{t}_1, \hat{t}_2, \dots, \hat{t}_k\}$ , where  $k \leq l$ ,  $l$  is the total number of words in all the documents, and  $k$  is the number of clusters.

## 4. METHODOLOGY

We now describe a novel algorithm which can handle single-document segmentation, shared topic detection, and multi-document segmentation and alignment based on MI or WMI.

### 4.1 Mutual Information

MI  $I(X; Y)$  is a quantity to measure the amount of information which is contained in two or more random variables [8, 10]. For the case of two random variables, we have

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}, \quad (1)$$

Obviously, when random variables  $X$  and  $Y$  are independent,  $I(X; Y) = 0$ . Thus, intuitively, the value of MI depends on how random variables are dependent on each other. The optimal co-clustering is the mapping  $Clu_x : X \rightarrow \hat{X}$  and  $Clu_y : Y \rightarrow \hat{Y}$  that minimizes the loss:  $I(X; Y) - I(\hat{X}; \hat{Y})$ , which is equal to maximizing  $I(\hat{X}; \hat{Y})$ . This is the criterion of MI for clustering.

In the case of topic segmentation, the two random variables are the term variable  $T$  and the segment variable  $S$ , and each sample is an occurrence of a term  $T = t$  in a particular segment  $S = s$ .  $I(T; S)$  is used to measure how dependent  $T$  and  $S$  are. However,  $I(T; S)$  cannot be computed for documents before segmentation, since we do not have a set of  $S$  due to the fact that sentences of Document  $d$ ,  $s_i \in S_d$ , is not aligned with other documents. Thus, instead of minimizing the loss of MI, we can maximize MI after topic

segmentation, computed as:

$$I(\hat{T}; \hat{S}) = \sum_{\hat{t} \in \hat{T}} \sum_{\hat{s} \in \hat{S}} p(\hat{t}, \hat{s}) \log \frac{p(\hat{t}, \hat{s})}{p(\hat{t})p(\hat{s})}, \quad (2)$$

where  $p(\hat{t}, \hat{s})$  are estimated by the term frequency  $tf$  of Term Cluster  $\hat{t}$  and Segment  $\hat{s}$  in the training set  $D$ . Note that here a segment  $\hat{s}$  includes sentences about the the same topic among all documents. The optimal solution is the mapping  $Clu_t : T \rightarrow \hat{T}$ ,  $Seg_d : S_d \rightarrow \hat{S}'$ , and  $Ali_d : \hat{S}' \rightarrow \hat{S}$ , which maximizes  $I(\hat{T}; \hat{S})$ .

### 4.2 Weighted Mutual Information

In topic segmentation and alignment of multiple documents, if  $P(D, \hat{S}, T)$  is known, based on the marginal distributions  $P(D|T)$  and  $P(\hat{S}|T)$  for each term  $t \in T$ , we can categorize terms into four types in the data set:

- *Common stop words* are common both along the dimensions of documents and segments.
- *Document-dependent stop words* that depends on the personal writing style are common only along the dimension of segments for some documents.
- *Cue words* are the most important elements for segmentation. They are common along the dimension of documents only for the same segment, and they are not common along the dimensions of segments.
- *Noisy words* are other words which are not common along both dimensions.

Entropy based on  $P(D|T)$  and  $P(\hat{S}|T)$  can be used to identify different types of terms. To reinforce the contribution of *cue words* in the MI computation, and simultaneously reduce the effect of the other three types of words, similar as the idea of the *tf-idf* weight [22], we use entropies of each term along the dimensions of document  $D$  and segment  $\hat{S}$ , i.e.  $E_D(\hat{t})$  and  $E_{\hat{S}}(\hat{t})$ , to compute the weight. A cue word usually has a large value of  $E_D(\hat{t})$  but a small value of  $E_{\hat{S}}(\hat{t})$ . We introduce term weights (or term cluster weights)

$$w_{\hat{t}} = \left( \frac{E_D(\hat{t})}{\max_{\hat{t}' \in \hat{T}} (E_D(\hat{t}'))} \right)^a \left( 1 - \frac{E_{\hat{S}}(\hat{t})}{\max_{\hat{t}' \in \hat{T}} (E_{\hat{S}}(\hat{t}'))} \right)^b, \quad (3)$$

where  $E_D(\hat{t}) = \sum_{d \in D} p(d|\hat{t}) \log_{|D|} \frac{1}{p(d|\hat{t})}$ ,  $E_{\hat{S}}(\hat{t}) = \sum_{\hat{s} \in \hat{S}} p(\hat{s}|\hat{t}) \log_{|\hat{S}|} \frac{1}{p(\hat{s}|\hat{t})}$ , and  $a > 0$  and  $b > 0$  are powers to adjust term weights. Usually  $a = 1$  and  $b = 1$  as default, and  $\max_{\hat{t}' \in \hat{T}} (E_D(\hat{t}'))$  and  $\max_{\hat{t}' \in \hat{T}} (E_{\hat{S}}(\hat{t}'))$  are used to normalize the entropy values. Term cluster weights are used to adjust  $p(\hat{t}, \hat{s})$ ,

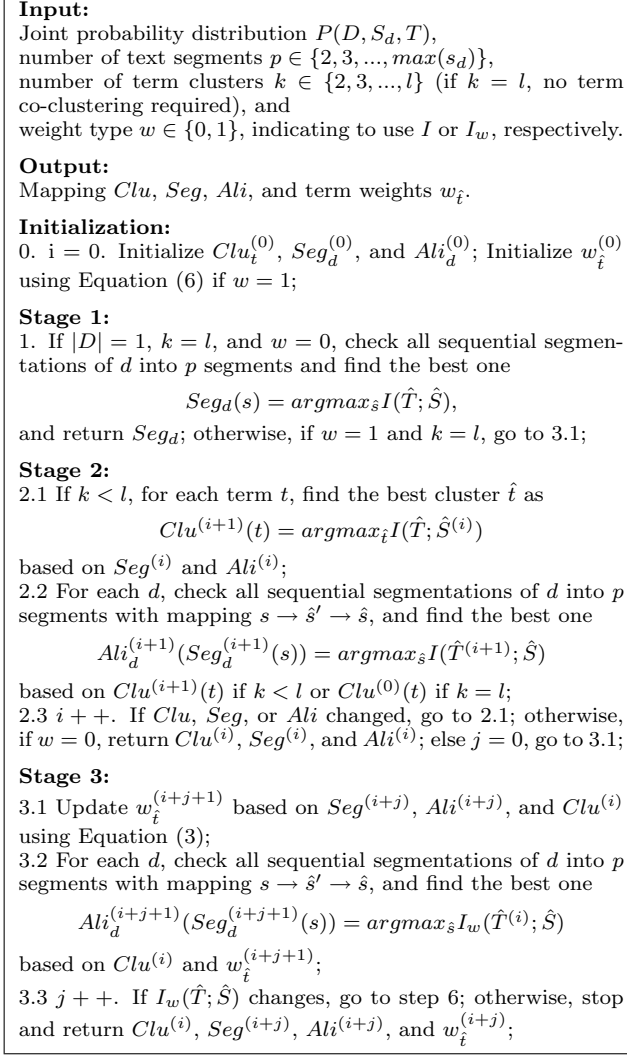
$$p_w(\hat{t}, \hat{s}) = \frac{w_{\hat{t}} p(\hat{t}, \hat{s})}{\sum_{\hat{t}' \in \hat{T}; \hat{s}' \in \hat{S}} w_{\hat{t}'} p(\hat{t}', \hat{s}')}, \quad (4)$$

and

$$I_w(\hat{T}; \hat{S}) = \sum_{\hat{t} \in \hat{T}} \sum_{\hat{s} \in \hat{S}} p_w(\hat{t}, \hat{s}) \log \frac{p_w(\hat{t}, \hat{s})}{p_w(\hat{t})p_w(\hat{s})}, \quad (5)$$

where  $p_w(\hat{t})$  and  $p_w(\hat{s})$  are marginal distributions of  $p_w(\hat{t}, \hat{s})$ .

However, since we do not know either the term weights or  $P(D, \hat{S}, T)$ , we need to estimate them, but  $w_{\hat{t}}$  depends on  $p(\hat{s}|\hat{t})$  and  $\hat{S}$ , while  $\hat{S}$  and  $p(\hat{s}|\hat{t})$  also depend on  $w_{\hat{t}}$  that is still unknown. Thus, an iterative algorithm is required to estimate term weights  $w_{\hat{t}}$  and find the best segmentation and alignment to optimize the objective function  $I_w$  concurrently. After a document is segmented into sentences



**Figure 2: Algorithm: Topic segmentation and alignment based on MI or WMI.**

and each sentence is segmented into words, each word is stemmed. Then the joint probability distribution  $P(D, S_d, T)$  can be estimated. Finally, this distribution can be used to compute MI in our algorithm.

### 4.3 Iterative Greedy Algorithm

Our goal is to maximize the objective function,  $I(\hat{T}; \hat{S})$  or  $I_w(\hat{T}; \hat{S})$ , which can measure the dependence of term occurrences in different segments. Generally, first we do not know the estimate term weights, which depend on the optimal topic segmentation and alignment, and term clusters. Moreover, this problem is NP-hard [10], even though if we know the term weights. Thus, an iterative greedy algorithm is desired to find the best solution, even though probably only local maxima are reached. We present the iterative greedy algorithm in Figure 2 to find a local maximum of  $I(\hat{T}; \hat{S})$  or  $I_w(\hat{T}; \hat{S})$  with simultaneous term weight estimation. This algorithm can be iterative and greedy for multi-document cases or single-document cases with term weight estimation and/or term co-clustering. Otherwise, since it is just a one

step algorithm to solve the task of single-document segmentation [6, 15, 25], the global maximum of MI is guaranteed. We will show later that term co-clustering reduces the accuracy of the results and is not necessary, and for single-document segmentation, term weights are also not required.

#### 4.3.1 Initialization

In Step 0, the initial term clustering  $Clu_{\hat{i}}$  and topic segmentation and alignment  $Seg_d$  and  $Ali_d$  are important to avoid local maxima and reduce the number of iterations. First, a good guess of term weights can be made by using the distributions of term frequency along sentences for each document and averaging them to get the initial values of  $w_{\hat{i}}$ :

$$w_t = \left( \frac{E_D(t)}{\max_{t' \in T} (E_D(t'))} \right) \left( 1 - \frac{E_S(t)}{\max_{t' \in T} (E_S(t'))} \right), \quad (6)$$

where

$$E_S(t) = \frac{1}{|D_t|} \sum_{d \in D_t} \left( 1 - \sum_{s \in S_d} p(s|t) \log_{|S_d|} \frac{1}{p(s|t)} \right),$$

where  $D_t$  is the set of documents which contain Term  $t$ . Then, for the initial segmentation  $Seg^{(0)}$ , we can simply segment documents equally by sentences. Or we can find the optimal segmentation just for each document  $d$  which maximizes the WMI,  $Seg_d^{(0)} = \operatorname{argmax}_{\hat{s}} I_w(T; \hat{S})$ , where  $w = w_{\hat{i}}^{(0)}$ . For the initial alignment  $Ali^{(0)}$ , we can first assume that the order of segments for each  $d$  is the same. For the initial term clustering  $Clu^{(0)}$ , first cluster labels can be set randomly, and after the first time of Step 3, a good initial term clustering is obtained.

#### 4.3.2 Different Cases

After initialization, there are three stages for different cases. Totally there are eight cases,  $|D| = 1$  or  $|D| > 1$ ,  $k = l$  or  $k < l$ ,  $w = 0$  or  $w = 1$ . Single document segmentation without term clustering and term weight estimation ( $|D| = 1, k = l, w = 0$ ) only requires Stage 1 (Step 1). If term clustering is required ( $k < l$ ), Stage 2 (Step 2.1, 2.2, and 2.3) is executed iteratively. If term weight estimation is required ( $w = 1$ ), Stage 3 (Step 3.1, 3.2, and 3.3) is executed iteratively. If both are required ( $k < l, w = 1$ ), Stage 2 and 3 run one after the other. For multi-document segmentation without term clustering and term weight estimation ( $|D| > 1, k = l, w = 0$ ), only iteration of Step 2.2 and 2.3 are required.

At Stage 1, the global maximum can be found based on  $I(\hat{T}; \hat{S})$  using dynamic programming in Section 4.4. Simultaneously finding a good term clustering and estimated term weights is impossible, since when moving a term to a new term cluster to maximize  $I_w(\hat{T}; \hat{S})$ , we do not know that the weight of this term should be the one of the new cluster or the old cluster. Thus, we first do term clustering at Stage 2, and then estimate term weights at Stage 3.

At Stage 2, Step 2.1 is to find the best term clustering and Step 2.2 is to find the best segmentation. This cycle is repeated to find a local maximum based on MI  $I$  until it converges. The two steps are: (1) based on current term clustering  $Clu_{\hat{i}}$ , for each document  $d$ , the algorithm segments all the sentences  $S_d$  into  $p$  segments sequentially (some segments may be empty), and put them into the  $p$  segments  $\hat{S}$  of the whole training set  $D$  (all possible cases of different segmentation  $Seg_d$  and alignment  $Ali_d$  are checked) to find the optimal case, and (2) based on the current segmentation

and alignment, for each term  $t$ , the algorithm finds the best term cluster of  $t$  based on the current segmentation  $Seg_d$  and alignment  $Ali_d$ . After finding a good term clustering, term weights are estimated if  $w = 1$ .

At Stage 3, similar as Stage 2, Step 3.1 is term weight re-estimation and Step 3.2 is to find a better segmentation. They are repeated to find a local maximum based on WMI  $I_w$  until it converges. However, if the term clustering in Stage 2 is not accurate, then the term weight estimation at Stage 3 may have a bad result. Finally, at Step 3.3, this algorithm converges and return the output. This algorithm can handle both single-document and multi-document segmentation. It also can detect shared topics among documents by checking the proportion of overlapped sentences on the same topics, as described in Sec 5.2.

#### 4.4 Algorithm Optimization

In many previous works on segmentation, dynamic programming is a technique used to maximize the objective function. Similarly, at Step 1, 2.2, and 3.2 of our algorithm, we can use dynamic programming. For Stage 1, using dynamic programming can still find the global optimum, but for Stage 2 and Stage 3, we can only find the optimum for each step of topic segmentation and alignment of a document. Here we only show the dynamic programming for Step 3.2 using WMI (Step 1 and 2.2 are similar but they can use either  $I$  or  $I_w$ ). There are two cases that are not shown in the algorithm in Figure 2: (a) single-document segmentation or multi-document segmentation with the same sequential order of segments, where alignment is not required, and (b) multi-document segmentation with different sequential orders of segments, where alignment is necessary. The alignment mapping function of the former case is simply just  $Ali_d(\hat{s}'_i) = \hat{s}_i$ , while for the latter one's alignment mapping function  $Ali_d(\hat{s}'_i) = \hat{s}_j$ ,  $i$  and  $j$  may be different. The computational steps for the two cases are listed below:

**Case 1** (no alignment):

For each document  $d$ :

(1) Compute  $p_w(\hat{t})$ , partial  $p_w(\hat{t}, \hat{s})$  and partial  $p_w(\hat{s})$  without counting sentences from  $d$ . Then put sentences from  $i$  to  $j$  into Part  $k$ , and compute partial WMI

$$PI_w(\hat{T}; \hat{s}_k(s_i, s_{i+1}, \dots, s_j)) \sum_{\hat{t} \in \hat{T}} p_w(\hat{t}, \hat{s}_k) \log \frac{p_w(\hat{t}, \hat{s}_k)}{p_w(\hat{t})p_w(\hat{s}_k)},$$

where  $Ali_d(s_i, s_{i+1}, \dots, s_j) = k$ ,  $k \in \{1, 2, \dots, p\}$ ,  $1 \leq i \leq j \leq n_d$ , and  $Seg_d(s_q) = \hat{s}_k$  for all  $i \leq q \leq j$ .

(2) Let  $M(s_m, 1) = PI_w(\hat{T}; \hat{s}_1(s_1, s_2, \dots, s_m))$ . Then

$$M(s_m, L) = \max_i [M(s_{i-1}, L-1) + PI_w(\hat{T}; \hat{s}_L(s_i, \dots, s_m))],$$

where  $0 \leq m \leq n_d$ ,  $1 < L < p$ ,  $1 \leq i \leq m+1$ , and when  $i > m$ , no sentences are put into  $\hat{s}_k$  when compute  $PI_w$  (note  $PI_w(\hat{T}; \hat{s}(s_i, \dots, s_m)) = 0$  for single-document segmentation).

(3) Finally  $M(s_{n_d}, p) = \max_i [M(s_{i-1}, p-1) +$

$PI_w(\hat{T}; \hat{s}_p(s_i, \dots, s_{n_d}))]$ , where  $1 \leq i \leq n_d+1$ . The optimal  $I_w$  is found and the corresponding segmentation is the best.

**Case 2** (alignment required):

For each document  $d$ :

(1) Compute  $p_w(\hat{t})$ , partial  $p_w(\hat{t}, \hat{s})$ , and partial  $p_w(\hat{s})$ , and  $PI_w(\hat{T}; \hat{s}_k(s_i, s_{i+1}, \dots, s_j))$  similarly as Case 1.

(2) Let  $M(s_m, 1, k) = PI_w(\hat{T}; \hat{s}_k(s_1, s_2, \dots, s_m))$ , where  $k \in \{1, 2, \dots, p\}$ . Then  $M(s_m, L, k_L) = \max_{i,j} [M(s_{i-1}, L -$

$$1, k_{L/j}) + PI_w(\hat{T}; \hat{s}_{Ali_d(\hat{s}'_L)=j}(s_i, s_{i+1}, \dots, s_m))],$$

where  $0 \leq m \leq n_d$ ,  $1 < L < p$ ,  $1 \leq i \leq m+1$ ,  $k_L \in Set(p, L)$ , which is the set of all  $\frac{p!}{L!(p-L)!}$  combinations of  $L$  segments chosen from all  $p$  segments,  $j \in k_L$ , the set of  $L$  segments chosen from all  $p$  segments, and  $k_{L/j}$  is the combination of  $L-1$  segments in  $k_L$  except Segment  $j$ .

(3) Finally,  $M(s_{n_d}, p, k_p) = \max_{i,j} [M(s_{i-1}, p-1, k_{p/j}) + PI_w(\hat{T}; \hat{s}_{Ali_d(\hat{s}'_L)=j}(s_i, s_{i+1}, \dots, s_{n_d}))]$ ,

where  $k_p$  is just the combination of all  $p$  segments and  $1 \leq i \leq n_d+1$ , which is the optimal  $I_w$  and the corresponding segmentation is the best.

The steps of Case 1 and 2 are similar, except in Case 2, alignment is considered in addition to segmentation. First, basic items of probability for computing  $I_w$  are computed excluding Doc  $d$ , and then partial WMI by putting every possible sequential segment (including empty segment) of  $d$  into every segment of the set. Second, the optimal sum of  $PI_w$  for  $L$  segments and the leftmost  $m$  sentences,  $M(s_m, L)$ , is found. Finally, the maximal WMI is found among different sums of  $M(s_m, p-1)$  and  $PI_w$  for Segment  $p$ .

## 5. EXPERIMENTS

In this section, single-document segmentation, shared topic detection, and multi-document segmentation will be tested. Different hyper parameters of our method are studied. For convenience, we refer to the method using  $I$  as  $MI_k$  if  $w = 0$ , and  $I_w$  as  $WMI_k$  if  $w = 2$  or as  $WMI'_k$  if  $w = 1$ , where  $k$  is the number of term clusters, and if  $k = l$ , where  $l$  is the total number of terms, then no term clustering is required, i.e.  $MI_l$  and  $WMI_l$ .

### 5.1 Single-document Segmentation

#### 5.1.1 Test Data and Evaluation

The first data set we tested is a synthetic one used in previous research [6, 15, 25] and many other papers. It has 700 samples. Each is a concatenation of ten segments. Each segment is the first  $n$  sentence selected randomly from the Brown corpus, which is supposed to have a different topic from each other. Currently, the best results on this data set is achieved by Ji et.al. [15]. To compare the performance of our methods, the criterion used widely in previous research is applied, instead of the unbiased criterion introduced in [20]. It chooses a pair of words randomly. If they are in different segments (*different*) for the real segmentation (*real*), but predicted (*pred*) as in the same segment, it is a *miss*. If they are in the same segment (*same*), but predicted as in different segments, it is a *false alarm*. Thus, the error rate is computed using the following equation:

$$p(err|real, pred) = p(miss|real, pred, diff)p(diff|real) + p(false\_alarm|real, pred, same)p(same|real).$$

#### 5.1.2 Experiment Results

We tested the case when the number of segments is known. Table 1 shows the results of our methods with different hyper parameter values and three previous approaches, C99[25], U00[6], and ADDP03[15], on this data set when the segment number is known. In  $WMI$  for single-document segmentation, the term weights are computed as follows:  $w_i = 1 - E_{\hat{s}}(\hat{t}) / \max_{\hat{t} \in \hat{T}}(E_{\hat{s}}(\hat{t}'))$ . For this case, our methods  $MI_l$  and  $WMI_l$  both outperform all the previous approaches. We compared our methods with ADDP03 using one-sample one-sided t-test and p-values are shown in Table 2. From the p-values, we can see that mostly the differences are very

**Table 1: Average Error Rates of Single-document Segmentation Given Segment Numbers Known**

Range of $n$	3-11	3-5	6-8	9-11
Sample size	400	100	100	100
C99	12%	11%	10%	9%
U00	10%	9%	7%	5%
ADDP03	6.0%	6.8%	5.2%	4.3%
$MI_l$	<b>4.68%</b>	<b>5.57%</b>	<b>2.59%</b>	<b>1.59%</b>
$WMI_l$	<b>4.94%</b>	<b>6.33%</b>	<b>2.76%</b>	<b>1.62%</b>
$MI_{100}$	9.62%	12.92%	8.66%	6.67%

**Table 2: Single-document Segmentation: P-values of T-test on Error Rates**

Range of $n$	3-11	3-5	6-8	9-11
ADDP03, $MI_l$	0.000	0.000	0.000	0.000
ADDP03, $WMI_l$	0.000	0.099	0.000	0.000
$MI_l$ , $WMI_l$	0.061	0.132	0.526	0.898

significant. We also compare the error rates between our two methods using two-sample two-sided t-test to check the hypothesis that they are equal. We cannot reject the hypothesis that they are equal, so the difference are not significant, even though all the error rates for  $MI_l$  are smaller than  $WMI_l$ . However, we can conclude that term weights contribute little in single-document segmentation. The results also show that  $MI$  using term co-clustering ( $k = 100$ ) decreases the performance. We tested different number of term clusters, and found that the performance becomes better when the cluster number increases to reach  $l$ .  $WMI_{k < l}$  has similar results that we did not show in the table.

As mentioned before, using MI may be inconsistent on optimal boundaries given different numbers of segments. This situation occurs especially when the similarities among segments are quite different, i.e. some transitions are very obvious, while others are not. This is because usually a document is a hierarchical structure instead of only a sequential structure. When the segments are not at the same level, this situation may occur. Thus, a hierarchical topic segmentation approach is desired, and the structure highly depends on the number of segments for each internal node and the stop criteria of splitting. For this data set of single-document segmentation, since it is just a synthetic set, which is just a concatenation of several segments about different topics, it is reasonable that approaches simply based on term frequency have a good performance. Usually for the tasks of segmenting coherent documents for sub-topics, the effectiveness decreases much.

## 5.2 Shared Topic Detection

### 5.2.1 Test Data and Evaluation

The second data set contains 80 news articles from Google News. There are eight topics and each has 10 articles. We randomly split the set into subsets with different document numbers and each subset has all eight topics. We compare our approach  $MI_l$  and  $WMI_l$  with LDA [4]. LDA treats a document in the data set as a bag of words, finds its distribution on topics, and its major topic.  $MI_l$  and  $WMI_l$  views each sentence as a bag of words and tag it with a topic label. Then for each pair of documents, LDA determines if they are on the same topic, while  $MI_l$  and

**Table 3: Shared Topic Detection: Average Error Rates for Different Numbers of Documents in Each Subset**

#Doc	10	20	40	80
LDA	8.89%	16.33%	<b>1.35%</b>	0.60%
$MI_l, \theta = 0.6$	<b>4.17%</b>	<b>1.71%</b>	1.47%	<b>0.0%</b>
$WMI_l, \theta = 0.8$	18.6%	3.16%	1.92%	<b>0.0%</b>

$WMI_l$  check whether the proportion overlapped sentences on the same topic is larger than the adjustable threshold  $\theta$ . That is, in  $MI_l$  and  $WMI_l$ , for a pair of documents  $d, d'$ , if  $[\sum_{s \in S_d, s' \in S_{d'}} 1_{(topic_s = topic_{s'})} / \min(|S_d|, |S_{d'}|)] > \theta$ , where  $S_d$  is the set of sentences of  $d$ , and  $|S_d|$  is the number of sentences of  $d$ , then  $d$  and  $d'$  have the shared topic.

For a pair of documents selected randomly, the error rate is computed using the following equation:

$$p(err|real, pred) = p(miss|real, pred, same)p(same|real) + p(false\_alarm|real, pred, diff)p(diff|real),$$

where a *miss* means if they have the same topic (*same*) for the real case (*real*), but predicted (*pred*) as on the same topic. If they are on different topics (*diff*), but predicted as on the same topic, it is a *false alarm*.

### 5.2.2 Experiment Results

The results are shown in Table 3. If most documents have different topics, in  $WMI_l$ , the estimation of term weights in Equation (3) is not correct. Thus,  $WMI_l$  is not expected to have a better performance than  $MI_l$ , when most documents have different topics. When there are fewer documents in a subset with the same number of topics, more documents have different topics, so  $WMI_l$  is more worse than  $MI_l$ . We can see that for most cases  $MI_l$  has a better (or at least similar) performance than LDA. After shared topic detection, multi-document segmentation of documents with the shared topics is able to be executed.

## 5.3 Multi-document Segmentation

### 5.3.1 Test Data and Evaluation

For multi-document segmentation and alignment, our goal is to identify these segments about the same topic among multiple similar documents with shared topics. Using  $I_w$  is expected to perform better than  $I$ , since without term weights the result is affected seriously by document-dependent stop words and noisy words which depends on the personal writing style. It is more likely to treat the same segments of different documents as different segments under the effect of document-dependent stop words and noisy words. Term weights can reduce the effect of document-dependent stop words and noisy words by giving cue terms more weights.

The data set for multi-document segmentation and alignment has 102 samples and 2264 sentences totally. Each is the introduction part of a lab report selected from the course of Biol 240W, Pennsylvania State University. Each sample has two segments, introduction of plant hormones and the content in the lab. The length range of samples is from two to 56 sentences. Some samples only have one part and some have a reverse order the these two segments. It is not hard to identify the boundary between two segments for human. We labelled each sentence manually for evaluation. The criterion of evaluation is just using the proportion of the number of sentences with wrong predicted segment labels in the total number of sentences in the whole training

**Table 4: Average Error Rates of Multi-document Segmentation Given Segment Numbers Known**

#Doc	$MI_l$	$WMI_l$	$k$	$MI_k$	$WMI_k$
102	3.14%	<b>2.78%</b>	300	4.68%	6.58%
51	4.17%	<b>3.63%</b>	300	17.83%	22.84%
34	5.06%	<b>4.12%</b>	300	18.75%	20.95%
20	7.08%	<b>5.42%</b>	250	20.40%	21.83%
10	10.38%	<b>7.89%</b>	250	21.42%	21.91%
5	15.77%	<b>11.64%</b>	250	21.89%	22.59%
2	25.90%	<b>23.18%</b>	50	25.44%	25.49%
1	<b>23.90%</b>	24.82%	25	25.75%	26.15%

**Table 5: Multi-document Segmentation: P-values of T-test on Error Rates for  $MI_l$  and  $WMI_l$**

#Doc	51	34	20	10	5	2
P-value	0.19	0.101	0.025	0.001	0.000	0.002

set as the error rate:

$$p(\text{error}|\text{predicted}, \text{real}) = \sum_{d \in D} \sum_{s \in S_d} 1_{(\text{predicted}_s \neq \text{real}_s)} / \sum_{d \in D} n_d.$$

In order to show the benefits of multi-document segmentation and alignment, we compared our method with different parameters on different partitions of the same training set. Except the cases that the number of documents is 102 and one (they are special cases of using the whole set and the pure single-document segmentation), we randomly divided the training set into  $m$  partitions, and each has 51, 34, 20, 10, 5, and 2 document samples. Then we applied our methods on each partition and calculated the error rate of the whole training set. Each case was repeated for 10 times for computing the average error rates. For different partitions of the training set, different  $k$  values are used, since the number of terms increases when the document number in each partition increases.

### 5.3.2 Experiment Results

From the experiment results in Table 4, we can see the following observations: (1) When the number of documents increases, all methods have better performances. Only from one to two documents,  $MI_l$  has decrease a little. We can observe this from Figure 3 at the point of *document number* = 2. Most curves even have the worst results at this point. There are two reasons. First, because samples vote for the best multi-document segmentation and alignment, but if only two documents are compared with each other, the one with missing segments or a totally different sequence will affect the correct segmentation and alignment of the other. Second, as noted at the beginning of this section, if two documents have more document-dependent stop words or noisy words than cue words, then the algorithm may view them as two different segments and the other segment is missing. Generally, we can only expect a better performance when the number of documents is larger than the number of segments. (2) Except single-document segmentation,  $WMI_l$  is always better than  $MI_l$ , and when the number of documents is reaching one or increases to a very large number, their performances become closer. Table 5 shows p-values of two-sample one-sided t-test between  $MI_l$  and  $WMI_l$ . We also can see this trend from p-values. When *document number* = 5, we reached the smallest p-value and the largest difference between error rates of  $MI_l$  and  $WMI_l$ . For single-document

**Table 6: Multi-document Segmentation: Average Error Rate for Document Number = 5 in Each Subset with Different Number of Term Clusters**

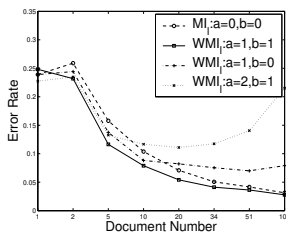
#Cluster	75	100	150	250	$l$
$MI_k$	24.67%	24.54%	23.91%	22.59%	15.77%

segmentation,  $WMI_l$  is even a little bit worse than  $MI_l$ , which is similar as the results of the single-document segmentation on the first data set. The reason is that for single-document segmentation, we cannot estimate term weights accurately, since multiple documents are unavailable. (3) Using term clustering usually gets worse results than  $MI_l$  and  $WMI_l$ . (4) Using term clustering in  $WMI_k$  is even worse than in  $MI_k$ , since in  $WMI_k$  term clusters are found first using  $I$  before using  $I_w$ . If the term clusters are not correct, then the term weights are estimated worse, which may mislead the algorithm to reach even worse results. From the results we also found that in multi-document segmentation and alignment, most documents with missing segments and a reverse order are identified correctly.

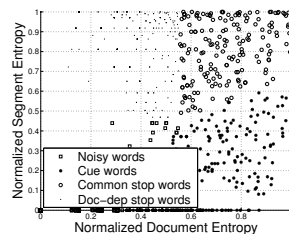
Table 6 illustrates the experiment results for the case of 20 partitions (each has five document samples) of the training set and topic segmentation and alignment using  $MI_k$  with different numbers of term clusters  $k$ . Notice that when the number of term clusters increases, the error rate becomes smaller. Without term clustering, we have the best result. We did not show results for  $WMI_k$  with term clustering, but the results are similar.

We also tested  $WMI_l$  with different hyper parameters of  $a$  and  $b$  to adjust term weights. The results are presented in Figure 3. It was shown that the default case  $WMI_l : a = 1, b = 1$  gave the best results for different partitions of the training set. We can see the trend that when the document number is very small or large, the difference between  $MI_l : a = 0, b = 0$  and  $WMI_l : a = 1, b = 1$  becomes quite small. When the document number is not large (about from 2 to 10), all the cases using term weights have better performances than  $MI_l : a = 0, b = 0$  without term weights, but when the document number becomes larger, the cases  $WMI_l : a = 1, b = 0$  and  $WMI_l : a = 2, b = 1$  become worse than  $MI_l : a = 0, b = 0$ . When the document number becomes very large, they are even worse than cases with small document numbers. This means that a proper way to estimate term weights for the criterion of WMI is very important. Figure 4 shows the term weights learned from the whole training set. Four types of words are categorized roughly even though the transition among them are subtle. Figure 5 illustrates the change in (weighted) mutual information for  $MI_l$  and  $WMI_l$ . As expected, mutual information for  $MI_l$  increases monotonically with the number of steps, while  $WMI_l$  does not. Finally,  $MI_l$  and  $WMI_l$  are scalable, with computational complexity shown in Figure 6.

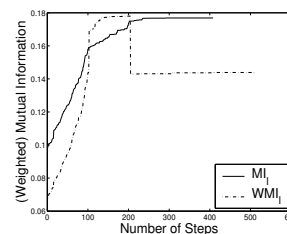
One advantage for our approach based on MI is that removing stop words is not required. Another important advantage is that there are no necessary hyper parameters to adjust. In single-document segmentation, the performance based on MI is even better for that based on WMI, so no extra hyper parameter is required. In multi-document segmentation, we show in the experiment,  $a = 1$  and  $b = 1$  is the best. Our method gives more weights to cue terms. However, usually cue terms or sentences appear at the beginning of a segment, while the end of the segment may be



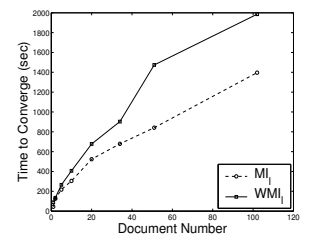
**Figure 3: Error rates for different hyper parameters of term weights.**



**Figure 4: Term weights learned from the whole training set.**



**Figure 5: Change in (weighted) MI for  $MI_l$  and  $WMI_l$ .**



**Figure 6: Time to converge for  $MI_l$  and  $WMI_l$ .**

much noisy. One possible solution is giving more weights to terms at the beginning of each segment. Moreover, when the length of segments are quite different, long segments have much higher term frequencies, so they may dominate the segmentation boundaries. Normalization of term frequencies versus the segment length may be useful.

## 6. CONCLUSIONS AND FUTURE WORK

We proposed a novel method for multi-document topic segmentation and alignment based on weighted mutual information, which can also handle single-document cases. We used dynamic programming to optimize our algorithm. Our approach outperforms all the previous methods on single-document cases. Moreover, we also showed that doing segmentation among multiple documents can improve the performance tremendously. Our results also illustrated that using weighted mutual information can utilize the information of multiple documents to reach a better performance.

We only tested our method on limited data sets. More data sets especially complicated ones should be tested. More previous methods should be compared with. Moreover, natural segmentations like paragraphs are hints that can be used to find the optimal boundaries. Supervised learning also can be considered.

## 7. ACKNOWLEDGMENTS

The authors want to thank Xiang Ji, and Prof. J. Scott Payne for their help.

## 8. REFERENCES

- [1] A. Banerjee, I. Ghillon, J. Ghosh, S. Merugu, and D. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *Proceedings of SIGKDD*, 2004.
- [2] R. Bekkerman, R. El-Yaniv, and A. McCallum. Multi-way distributional clustering via pairwise interactions. In *Proceedings of ICML*, 2005.
- [3] D. M. Blei and P. J. Moreno. Topic segmentation with an aspect hidden markov model. In *Proceedings of SIGIR*, 2001.
- [4] D. M. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- [5] T. Brants, F. Chen, and I. Tsochantaridis. Topic-based document segmentation with probabilistic latent semantic analysis. In *Proceedings of CIKM*, 2002.
- [6] F. Choi. Advances in domain independent linear text segmentation. In *Proceedings of the NAACL*, 2000.
- [7] H. Christensen, B. Kolluru, Y. Gotoh, and S. Renals. Maximum entropy segmentation of broadcast news. In *Proceedings of ICASSP*, 2005.
- [8] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley and Sons, New York, USA, 1991.
- [9] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Systems*, 1990.
- [10] I. Dhillon, S. Mallela, and D. Modha. Information-theoretic co-clustering. In *Proceedings of SIGKDD*, 2003.
- [11] M. Hajime, H. Takeo, and O. Manabu. Text segmentation with multiple surface linguistic cues. In *Proceedings of COLING-ACL*, 1998.
- [12] T. K. Ho. Stop word location and identification for adaptive text recognition. *International Journal of Document Analysis and Recognition*, 3(1), August 2000.
- [13] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the UAI'99*, 1999.
- [14] X. Ji and H. Zha. Correlating summarization of a pair of multilingual documents. In *Proceedings of RIDE*, 2003.
- [15] X. Ji and H. Zha. Domain-independent text segmentation using anisotropic diffusion and dynamic programming. In *Proceedings of SIGIR*, 2003.
- [16] X. Ji and H. Zha. Extracting shared topics of multiple documents. In *Proceedings of the 7th PAKDD*, 2003.
- [17] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, 2001.
- [18] T. Li, S. Ma, and M. Ogihara. Entropy-based criterion in categorical clustering. In *Proceedings of ICML*, 2004.
- [19] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of ICML*, 2000.
- [20] L. Pevzner and M. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistic*, 28(1):19–36, 2002.
- [21] J. C. Reynar. Statistical models for topic segmentation. In *Proceedings of ACL*, 1999.
- [22] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw Hill, 1983.
- [23] B. Sun, Q. Tan, P. Mitra, and C. L. Giles. Extraction and search of chemical formulae in text documents on the web. In *Proceedings of WWW*, 2007.
- [24] B. Sun, D. Zhou, H. Zha, and J. Yen. Multi-task text segmentation and alignment based on weighted mutual information. In *Proceedings of CIKM*, 2006.
- [25] M. Utiyama and H. Isahara. A statistical model for domain-independent text segmentation. In *Proceedings of the 39th ACL*, 1999.
- [26] C. Wayne. Multilingual topic detection and tracking: Successful research enabled by corpora and evaluation. In *Proceedings of LREC*, 2000.
- [27] J. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt. A hidden markov model approach to text segmentation and event tracking. In *Proceedings of ICASSP*, 1998.
- [28] H. Zha and X. Ji. Correlating multilingual documents via bipartite graph modeling. In *Proceedings of SIGIR*, 2002.