CUTS: CUrvature-Based Development Pattern Analysis and Segmentation for Blogs and other Text Streams *

Yan Qi †
Comp. Sci. and Eng. Dept.
Arizona State University
Tempe, AZ 85287
yan.qi@asu.edu

K. Selçuk Candan
Comp. Sci. and Eng. Dept.
Arizona State University
Tempe, AZ 85287
candan@asu.edu

ABSTRACT

Weblogs (blogs) are becoming prominent forms of information exchange in the Internet. A large number and variety of blogs, like personal journals or commentaries, are available for general consumption. However, effective indexes and navigation structures (like the table of content in a book) are not available for blogs. Therefore, it is generally not possible to navigate among entries in a given collection of blog entries in an informed manner. This paper focuses on the segmentation of entries in filter-type [9] blogs, with the aim of using this information for developing hypertext and navigational helps. In particular, we are interested in the analysis of topic development patterns that can provide information about not only the entries themselves, but how these entries develop and relate to each other. The proposed algorithm, CUTS, maps entries into a curve in a way that makes apparent a variety of topic development patterns. We then use curve analysis for automatic segmentation of topics. The resulting base topic segments are classified into different topic development patterns that can be visualized and indexed. Experimental results show that the proposed technique has very good performance in identifying boundaries in text streams, especially *filter* style blogs, versus existing schemes. Furthermore, compared with other topic segmentation methods, the proposed mechanism highlights not only topic boundaries, but also topic development patterns.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia—*Navigation, user issues*; H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Hypertext navigation and maps*; H.3.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Abstracting methods, indexing methods*

†Contact author.

*This work is supported by NSF ITR Grant, ITR-0326544.

HT'06, August 22–25, 2006, Odense, Denmark.

Keywords

Weblogs, topic segmentation, topic development patterns, curve segmentation

1. INTRODUCTION

Weblogs, also known as blogs, are frequently modified web pages, with entries usually organized in reverse chronological order [16]. The term weblog was first used in 1997. The so called *blogosphere* is expected to be doubling in size about once every 5 months and about 30-40,000 new weblogs are being created each day [1].

[9] places weblogs into three main categories: filters, personal journals, and notebooks. Filters reflect important developments, such as news about world events. Personal journals are similar to diaries. Notebooks, on the other hand, are hybrids of filters and personal journals, but often distinguished by longer focused essays. According to [16], personal journals and filters together take up over 80%.

Filters are usually maintained in archives, where entries are organized simply in reverse chronological order. A navigational structure depicting the topic development of a blog site is simply not available. Such a navigational structure for a blog should be able to catch topics in the blog entries and distinguish relationships of different topics. On the other hand, the first step in any such effort is the identification of boundaries between consecutive topics. Topic segmentation is the process by which boundaries between parts that bear on different topics (of a given text document, a collection of documents, or the Web [11]) are identified. Thus, topic segmentation is the first step to establish a navigational structure, usually even before the recognition of topics themselves.

1.1 Problem Statement

In most blog archives, dated entries are simply organized in a reverse chronological order: let us denote all dated entries in a blog with $S = (p_1, p_2, \ldots, p_N)$, where N is the number of entries in the blog and $\forall i \leq N$, p_i represents the ith entry in chronological sequence. Consecutive blog entries are generally related to each other in terms of their content (or *topic*). There are blog search indexes, such as [2], which enable keyword-based search on blogs, but these treat each blog entry atomically and do not consider topic development patterns. For personal journal blogs, where there is no development structure, more traditional, keyword-based search [2, 3] is generally sufficient. However, many weblogs (such as news commentaries and education

(a) Long vs. short segment (b) More vs. less change (c) High vs. low concentration

Figure 1: Visualizing topic development differences between consecutive segments; all three cases show two consecutive segments: in (a) the first segment is longer than the second one; in (b) the first segment diverges faster than the second; and in (c) the first segment is more concentrated than the second

sites) where there is a correlation between materials in blogs and real world events, it is important to understand the development of structure of the blog. Since blogs are not structured in a particular way but time, it is generally not possible to navigate among entries in a given collection of blog entries in an informed manner. Thus, our primary goal is to segment the sequence of entries in a filter-type blog on the basis of the topic development patterns.

EXAMPLE 1.1 (VISUALIZING TOPIC SEGMENTS OF A BLOG). *A major motivation of our work is to enable indexing as well as ease of navigation of the blogs. Therefore, the various topic development patterns need to be visualized in a convenient manner to the user. In Figure 1, we depict one way to achieve this in a compact and easy to use manner. In this scheme, a* gradient *is used to denote a topic segment. The length of the gradient visualizes how long the topic segment lasts; its height denotes how fast the topic segment changes, and the saturation of the gradient measures how concentrated the topic segment is.*

Figure 2 shows an example of the visualization scheme introduced in Figure 1. Figure 2 is a visualization of the topic development patterns for the weblog (Talking Point Memo [4]) from January, 2005 to October, 2005. There are three layers in the visualization. The top layer depicts the highest level topic segmentation. The segments on the lower layers are clustered under the higher level segments. A triangle shows the location and duration of an interrupt, i.e. a temporary, but significant change in content. The content for each topic segment can also be quickly explored though a mouse cursor.

Note that there are other possible visualization schemes as well. In this paper, we do not focus on any specific scheme, rather we concentrate on extraction of topic development patterns with the goal of enabling effectively visualization schemes.

Based on the above example, there are three *topic development* patterns that are of primary interest to us for indexing and navigation purposes:

- *Dominated:* there is one (or more) dominant topic(s) in a given sequence of entries.
- *Drifting:* the content of the segment is gradually transitioning from one topic to an other.
- *Interrupted:* a different, significant but temporary, topic emerges and disappears during the natural (concentrated or drifting) flow of the blog.

In addition to these patterns, how concentrated a given segment is in terms of its content and how fast it changes are important pieces of information needed both for visualization and indexing.

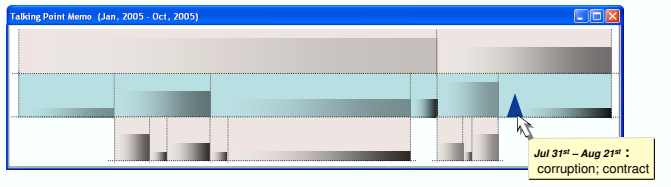


Figure 2: An example visualization scheme for a blog archive. The visualization interface not only presents a hierarchy of dominant topics, but also how fast topics develop and change and when significant interrupts occur. This interface is intended for experts who are interested in understanding and exploring the development patterns through blog analysis. Although we do not focus on a particular visualization or indexing scheme, CUTS is intended to enable development of this or similar tools

1.2 Related Work

One method often used in text segmentation is *text tiling*, which makes use of patterns of lexical co-occurrence to subdivide texts into multi-paragraph units that represent passages or subtopics [14]. In [20], Reynar provides an extensive discussion of algorithms for topic segmentation, including compression, optimization, and word frequency algorithms. All of these make use of topic shift indicators as cues for topic segmentation. Typical indicators include repetitions of character sequences, patterns of word and word n-gram repetition, word frequency, and the presence of certain key phrases. A probabilistic latent semantic analysis model is used in [11] for segmenting a document. Blei *et al.* [8] used partially unsupervised topic segmentation on unstructured text by means of a hidden Markov model. In [13], Fung *et al.* constructed a model with a hyper-geometric distribution to detect bursty events. The advantage of this model is the fact that it does not need tuning or estimating any of parameters.

A related research domain is topic detection and tracking (TDT) [5, 10, 27], which mainly focuses on detecting and tracking events in streaming news data. Most TDT systems compare a new document with the past documents and make a decision regarding the novelty of the story based on the content-based similarity values. In contrast, the naturally evolving nature of blogs and the need for fine-granularity segment boundary identification make the problem of topic segmentation in blogs significantly harder than the new-event detection problem addressed by the TDT technologies. In [15], Plaunt *et al.* exploit the idea of curve analysis to assist text segmentation. More recently, Andrews *et. al* also exploited a similar scheme [6]. The basic idea behind these approaches is as the following: A text is split into small pieces using a sliding window; a weighted keyword vector represents each piece. Then, the similarity between consecutive entries is measured using an appropriate similarity measure to obtain the sequence: $T = (s_{1,2}, \ldots, s_{i,i+1}, \ldots, s_{m-1,m})$, where $s_{i,i+1}$ is the similarity between the ith entry and the $(i+1)$th entry. In a sense, T tracks how drastically the content of the input stream of entries changes between consecutive entries (Figure 3). The main idea underlying these approaches is that topic shift occurs at local minima of the curve: low value of similarity

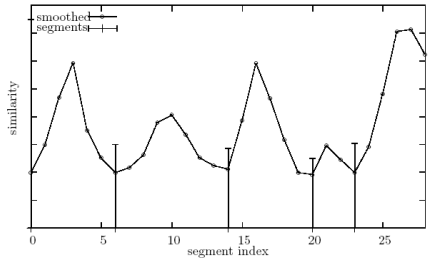


Figure 3: An existing text segmentation method which uses a *similarities-curve*[6]: the curve tracks how drastically the content of the input stream of entries changes between consecutive entries and the segment boundaries are chosen at the local minima of the curve

between entries means little relationship, and boundaries are likely to be located where consecutive entries are not related to each other. In the example shown in Figure 3, the local minima-based approach presented in [15, 6] leads to four boundaries.

This curvature-based approach is quite simple and effective, especially when the topic boundaries are indeed marked with drastic changes in the similarities of *consecutive* entries. However, as we will see, this usually is not the case. Furthermore, since these approaches ignore all other information except the relationship between consecutive entries, they do not capture richer content development patterns. In [25], Wong *et al.* combines techniques of both data mining and data visualization to explore sequential patterns in large text corpora. They focus on correlations between different specific items, not topic development patterns.

1.3 Proposed Approach and Contributions of this Paper

In this paper, we present a novel algorithm, CUTS(*CU*rvature-based development pattern analysis and segmentation for blogs and other *T*ext *S*treams) to topic segmentation of text streams and blogs. A key output of CUTS is the set of topic development patterns inherent in the sequence of entries. This knowledge helps in constructing properly annotated navigational helps, such as tables-of-contents, as well as proper segment indexes for context-aware search.

As in [15, 6], CUTS relies on curvature analysis for identifying boundaries. However, **unlike the previous approaches**, CUTS does not apply curvature analysis on the similarities curve (illustrated in Figure 3). Instead, it preprocesses the content in a way that brings out various topic development patterns as different curvature signatures. CUTS is composed of three phases:

(a) analysis of the content in the blog entries and generation of a representative signature (an appropriately weighted keyword vector) for each entry;

(b) mapping of the sequence of entries onto a curve which brings out various topic development patterns; and

(c) analysis of the resulting curve to identify topic segments and topic development patterns.

In the following sections, we discuss these three phases in detail. The structure of the paper is as follows. Section 2 briefly describes the data representation we use for individual blog entries. Section 3 discusses how CUTS maps blog entries onto a curve in a way that highlights topic development patterns. Section 4 provides an overview of the curve segmentation algorithms we rely on and discusses how to produce base topic segments and identify topic development patterns based on the output generated with these curve segmentation mechanisms. Section 5 experimentally verifies the performance of our approach.

2. REPRESENTATION OF THE BLOG ENTRIES

CUTS represents each blog entry using a conventional TF-IDF based scheme. For each entry, we construct a vector of weights, $P_i = \{w_{i,1}, w_{i,2}, \ldots, w_{i,n}\}$, where n is the number of keywords in all entries concerned and $i \in [1, N]$, N is the number of entries in the blog. When extracting keywords from the text, *stop words* are removed and a stemmer is employed. The TF-IDF weight, $w_{i,j}$, of a keyword j in entry i is then computed. Given these keyword weights, the similarity, $s_{i,j}$, between two entries, i and j, is defined as

$$s_{i,j} = \sum_{k=1}^{n} w_{i,k} \cdot w_{j,k}.$$

Finally a dissimilarity matrix D is generated for all of entries concerned in the blog. In D, each value represents the dissimilarity ($d_{i,j} = 1 - s_{i,j}$) between a given pair of entries.

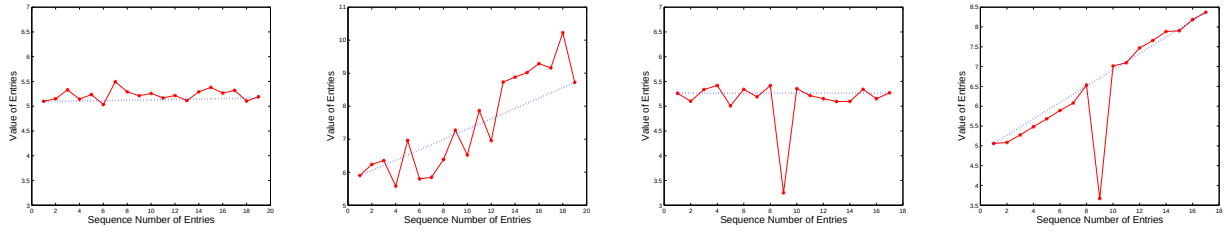
$$D = \begin{bmatrix} d_{1,1} & d_{1,2} & \ldots & d_{1,N} \\ d_{2,1} & d_{2,2} & \ldots & d_{2,N} \\ \vdots & \vdots & d_{i,j} & \vdots \\ d_{N,1} & d_{N,2} & \ldots & d_{N,N} \end{bmatrix}$$

Note that given the similarity values, it is possible to mine for independent topics in the collection using a *latent semantic indexing (LSI)*-based method [17]. LSI, however, would not consider the temporal correlation between entries, thus would miss an important dimension in topic development in blogs. Nevertheless, singular valued decomposition (which forms the basis of the LSI technique) can also be exploited in CUTS to reduce the number features to be considered for dissimilarity matrix construction.

We also note that the keyword weight ($w_{i,k}$) can also incorporate additional domain knowledge. For example, when there are multiple related blogs, the keyword weights can reflect importance of the words across related (in time or by reference) entries in different blogs. For simplicity and clarity, in this paper we do not focus on the co-segmentation of correlated blogs. However it is easy to extend CUTS to analysis of multiple correlated blogs and this will be part of our future work.

3. MATERIALIZING TOPIC DEVELOPMENT PATTERNS THROUGH A CUTS-CURVE

As discussed in the related work section, similarity curves are used in the literature for analyzing the content of text streams [15, 6]. Our main observation in this paper is that the similarity curves used in the literature for curvature analysis cannot capture anything but pairwise relationships of the neighbors. On the other hand, segments in a navigation structure need to capture the global structure of a large



(a) Dominated (b) Drifting (c) Interrupted (case 1) (d) Interrupted (case 2)

Figure 4: Topic development patterns highlighted by CUTS-curve

body of entries. Furthermore, identifying general development patterns (such as *dominated, drifting,* and *interrupted* introduced in Section 1.1) requires more knowledge than the relationship between immediate neighbors.

Thus, instead of using the similarity curves as in [15, 6], CUTS creates a curve which brings the underlying structure and development patterns into light. For this purpose, CUTS relies on multi-dimensional scaling (MDS) [24]. MDS is a class of search techniques for mapping data into a multi-dimensional space based on a given list of distance values between data elements. The mapping is such that the distances between points in the space best match the dissimilarities between the corresponding objects [12]. In this paper, we exploit this property of MDS. In particular,

(a) CUTS first maps all entries to a 1-dimensional space based on their dissimilarities (using the matrix D);

(b) CUTS then extends the space with the corresponding time dimension: i.e., CUTS maps the entries onto a 2-dimensional space, where the x-axis denotes the time (or the sequence number of the entries in the blog archive) and the y-axis denotes the MDS-computed dissimilarity dimension.

In the resulting two dimensional space, the consecutive entries form a curve (referred to as the CUTS-curve), on which various topic development patterns emerge:

- **Dominated**: In dominated segments, there is one or more stable topics in the given sequence of entries. In the CUTS-curve, such segments have the horizontal pattern shown in Figure 4(a): all points in this pattern are similar to each other (i.e., they form a clique where the distances between the entries are close to each other). Once this clique is mapped into a 1-dimensional space through MDS, the entries are naturally mapped close to each other and when they are plotted against time, a horizontal pattern emerges. In the example shown in Figure 5, such a dominated behavior is visible in the segment **a**, relatively horizontal, part of the curve.

- **Drifting**: In drifting segments, the content is in the way of transition from one topic to the other. When all entries are plotted in a CUTS-curve, we obtain the pattern shown in Figure 4(b). Although consecutive entries are similar to each other, MDS forces a placement, such that there is a significant difference between the entries in the beginning of the pattern and those entries in the end of the pattern (essentially, the end-points correspond to two different topics). Furthermore, the slope of the curve reflects the speed with which the topic evolves from one to the other. A drifting segment **b** is visible immediately after the first dominated segment **a**. The example shows in Figure 5.

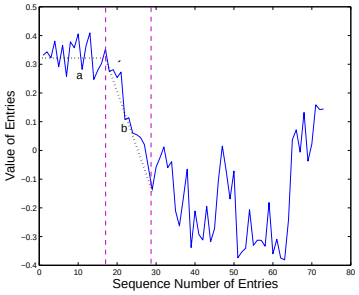
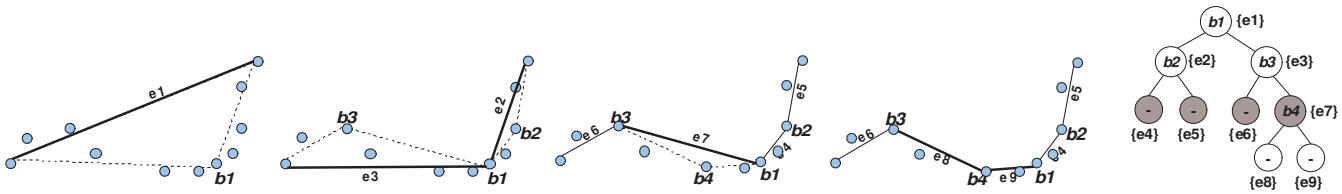


Figure 5: A CUTS-curve of a sequence of blog entires; the x-axis represents the sequence number of entries and the y-axis represents the position of each entry in the 1-dimensional space mapped by MDS

- **Interrupted**: In an interrupted segment, a significant, but temporary, topic emerges while one topic or a topic developing pattern is presented. When the entries are plotted in a CUTS-curve, we see two cases (Figure 4(c) and Figure 4(d)) of interruptions, corresponding to interruption of dominated or drifting patterns, respectively. The shapes of the interrupted patterns are usually of the form "$\vee$" or "$\wedge$". In fact, this pattern is usually composed of two opposite drifting patterns. However, unlike two independent drifting patterns, this class of segments usually have aligned (and similar) end-points; thus it is more reasonable to combine them.

Thus, given the dissimilarity matrix D as input, CUTS obtains a vector $\tilde{Y} = (y_{1,1}, y_{2,1}, \ldots, y_{n,1})$ through MDS. $\tilde{Y}$ represents the structural relationships of the entries in the blog. Finally, we extend the entries in $\tilde{Y}$ with time (the sequence numbers of the entries) to obtain a 2-dimensional CUTS-curve, C (as in Figure 5), where the x-axis denotes time and y-axis denotes the position of the entries in the 1-dimensional space mapped by MDS.

As discussed next, we then segment this CUTS-curve to identify the various segments and their temporal development patterns.



(a) Step 1 (b) Step 2 (c) Step 3 (d) Step 4 (e) Resulting tree

Figure 6: Adaptive curve segmentation by [19]

4. TOPIC SEGMENTATION AND PATTERN IDENTIFICATION BY CUTS

In the context of image processing, curve segmentation is a widely used technique for analyzing dominant visual features, such as shapes and outlines [18]. Curve segmentation helps detect critical boundary points (such as minima, maxima, zero-crossings, and discontinuities) which are important in understanding these features. In this work, we exploit curve segmentation for understanding the underlying patterns in the CUTS curve obtained as described in the previous section.

4.1 Adaptive Curve Segmentation [19]

Given a curve $C = \{x_i, y_i\}_{i=1}^{N}$, the goal of curve segmentation algorithms is to find the subset of dominant points $D = \{x_i, y_i\}_{i=1}^{M}$ (where $M \leq N$ and $D \subseteq C$), such that each resulting curve segment has uniform features. There are two major categories of approaches to curve segmentation. The first approach is segmentation through polygonal approximation. In this approach, a curve is approximated and then segmented into a series of simple straight lines. In [21], various of approaches of this category are discussed and assessed. The second (more general) approach is to treat the given curve as if it is made up of straight lines as well as arcs. Therefore, the goal becomes to find not only the subset of dominant points $D = \{x_i, y_i\}_{i=1}^{M}$, but also a set $S = \{\alpha_i\}_{i=1}^{M-1}$ where α_i represents the curvature between point i and point $i+1$ in the set D. In this paper, we rely on the first approach, where a curve is split into a series of straight lines. The use of the second approach for more descriptive development patterns is part of our future work.

Segmenting a curve into straight lines is not straightforward. For instance, a simple thresholding mechanism for determining boundaries is not appropriate, as the curve may behave differently at different parts and a single threshold may not be able to capture the temporal variations in the curve behavior. Thus, it is essential to use a scheme which can adapt to local changes in the curve behavior. In this paper, **we build on [19]**: in its first phase, [19] divides the points in the curve and constructs a binary tree; the second phase traverses the binary tree and decides the best representation of the curve with a series of straight lines.

For a detailed example, let us consider the curve in Figure 6. In the figure, the curve is approximated by a series of straight lines, through four iterations. First, the line $e1$ is used to approximate the entire curve. The associated *breakpoint* is identified as $b1$. The line $e1$ is placed as the root of the binary tree in Figure 6(e). Then the part of the curve after (or to the right of) $b1$ is considered. This curve gives the line $e2$ and the breakaway point $b2$. This process is repeated until the number of points in any given segment is at most a certain number, called $MinSpan$. At the end of the process, the resulting curve segment are organized as a binary tree where each node represents a straight line. Each level in the binary tree approximates the curve with a different granularity. Figure 6(e) shows the binary tree which represents the lines obtained from Figure 6(a) to Figure 6(d).

In order to get the optimal series of lines, the algorithm traverses the binary tree in a bottom-up manner. At each node, the significance of the line is compared with the significances of its children. If the significance of any of its two children is larger than that of the node, then the parent is removed from consideration; in a sense the children get *upgraded*, taking the place of their parent in the tree. The darker shaded nodes in Figure 6(e) illustrate the curve segments chosen through this process.

4.2 Elimination of Over-Segmentation and Identification of Topic Development Patterns

The curve segmentation scheme [19] described above has a number of desirable features in the context of topic segmentation:

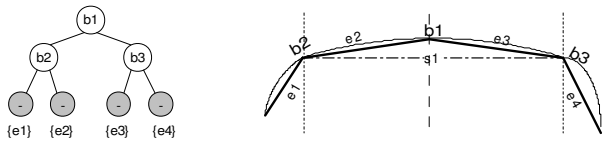
- Most parameters (except $MinSpan$) used in the algorithm do not require to be set by the user. There is no need for error tolerances or noise estimations.

- The measure of significance is based on a (pseudo-) psychological measure of perceptual significance: the longer the feature primitive, the greater the maximum deviation tolerated [22]. Thus, long lines will be kept for their stronger perceptual basis. This feature can reduce negative influence of local extremes when segmenting the curve.

  In topic segmentation, a line composed of a sequence of entries can represent a pattern. When longer patterns are preferred, this feature can help ignore or tolerate possible minor and very short term divergences in the development of a topic.

- The algorithm has scale invariance. That means, it will find the same structures without regards to the sizes of the curves. The feature makes it possible for us to effectively focus on a detailed analysis of parts of a curve, or study it at a higher level.

We on the other hand note that although this algorithm is adaptive and works well for curve segmentation tasks and has various advantages as stated above, it unfortunately does not immediately provide *topic* segments.



(a) Binary tree (b) Curve segments

Figure 7: An example for over-segmentation: $e2$ and $e3$ should be replaced by $s1$ because one topic is focused in this part rather than two

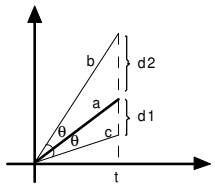


Figure 8: Angles between segments can not reflect the amount of drift correctly: b and c have the same difference of angles from a, but not same amount of drift.

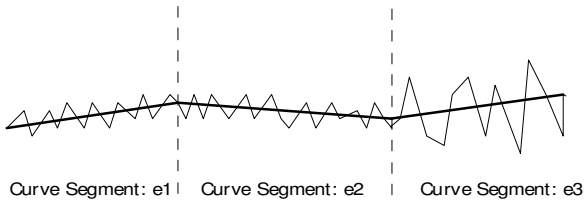


Figure 9: Combining curve segments: $e1$ and $e2$ can be combined because their slope (k) and concentration (σ) parameters are both relatively similar; however, although $e2$ and $e3$ have similar slopes, they can not be combined because of their very different concentrations

4.2.1 Reasons for Oversegmentation

The curve segmentation algorithm in [19], or any other curve segmentation algorithm which relies on curvature analysis is likely to oversegment topics, especially when there are frequent topic interrupts: sudden but temporary interrupts can also cause segmentation if the algorithm of adaptive curve segmentation in [19] is used naively.

Furthermore, the curve segmentation algorithm in [19] has a **greedy nature**, which may cause over-segmentation of a given CUTS-curve. This is illustrated in Figure 7. In this example, segments $e2$ and $e3$ are placed into two different subtrees by the curve segmentation algorithm (due to the greedy nature of the first split). However, slopes of these two segments are similar; thus from a topic development perspective they should belong to the same topic segment.

Therefore, curve segments obtained through an application of curve segmentation algorithms are not suitable to be used directly as topic segments, as this may reduce the overall precision. Therefore, to identify actual (potentially more complex) topic segments, we need to combine appropriate curve segments into larger topic segments, based on available contextual information.

4.2.2 Curve Segments to Base Topic Segments

Let us denote a 4-tuple $g_i = (k_i, \sigma_i, (x_{start}, y_{start})_i, (x_{end}, y_{end})_i)$ to represent each line segment in S:

- k_i is the slope of the corresponding line segment; this describes the *divergence speed* of the topics in the segment;
- σ_i is the average of distances from original points in this segment to the line approximating these points. This parameter measures the *concentration* of the entries around the dominant development pattern[1]; in other words, a large σ_i means that the curve is jagged (or saw-tooth shaped); i.e., there is a diverse set of topics being covered. For example, in Figure 9, the segment $e1$ and $e2$ are more concentrated than $e3$.
- the pairs, $(x_{start}, y_{start})_i$ and $(x_{end}, y_{end})_i$, denote the end points of a segment.

A *base topic segment* is composed of one or more consecutive curve segments, representing one topic or reflecting changes of topics. In particular, given two consecutive curve segments, g_i and g_{i+1} in S, these segments can be combined into a single base topic segment if the development of two curve segments is relatively homogeneous in terms of their topic evolution speeds and concentrations:

$$(|k_i - k_{i+1}| < \lambda_{drifting}) \wedge (|\sigma_i - \sigma_{i+1}| < (\sigma_i + \sigma_{i+1})/2)\,. \quad (1)$$

Here $\lambda_{drifting}$ is a parameter which determines the algorithms sensitivity to differences in topic evolution speeds; its value is computed as a percentage of the overall change in the data. As shown in Figure 8, angles between segments can not correctly capture the degrees of topic changes. Thus, $\lambda_{drifting}$ is described in terms of the difference of slopes of the curves as opposed to simple difference between their angles. When combining curve segments, we also require the difference of σs to be small. This process is illustrated in Figure 9.

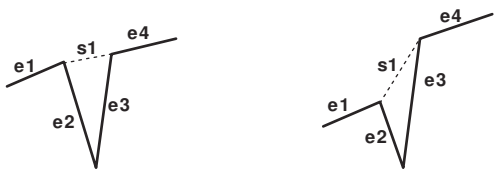
After two curve segments are combined, the k and σ of the new segment are recalculated. In particular, the slope is recomputed based on the end-points of the combined base topic segments. The new σ value is the average of distances from the points in both segments to the new line connecting end-points of the combined base topic segments.

Given a sequence $S = \{g_i\}_{i=1}^{M}$ of curve segments, we visit the pairs of segments in the chronological order of the underlying entries (i.e., in the increasing order of i) and combine the similar segments as described above. This gives us a new, reduced, sequence $S' = \{g'_i\}_{i=1}^{M'}$ of base topic segments, where a segment g'_i corresponds to one or more segments in S. The base topic segments, $B = \{b_h\}_{h=1}^{M'}$ are then the elements of the reduced segment set; i.e., $B = S'$.

4.2.3 Annotating Base Topic Segments with Patterns of Topic Development

Once the base topic segments, B, are identified, they need

[1]We use this average-based measure, rather than the maximum-deviation based *significance* value returned by the curve segmentation algorithm to identify the concentration of topics.



(a) interrupted pattern (b) not an interrupted pattern

Figure 10: Deciding interrupted patterns: the change of slopes between $e1$, $e4$ and $s1$ needs to be small for identification of an interrupted pattern

to be annotated with the topic development patterns: *dominated, drifting, or interrupted*. Foremost, a base topic segment, $b_h = (k_h, \sigma_h, (x_{start}, y_{start})_h, (x_{end}, y_{end})_h)$, is

- **dominated**, if $|k_i| < \lambda_{drifting}$, or
- **drifting**, if $|k_i| \geq \lambda_{drifting}$

where $\lambda_{drifting}$ is used to identify significant speeds of topic changes. However, this is not enough. As we mentioned earlier, when there are long-running dominant topics, which are interrupted with a short running topic, we need to differentiate these from simple topic drifts. Given two base topic segments, b_h and b_{h+1}, they can be combined into an

- **interrupted** segment if

$$\begin{array}{l}(|k_h| \geq \lambda_{drifting}) \wedge \\ (|k_{h+1}| \geq \lambda_{drifting}) \wedge \\ (k_h \times k_{h+1} < 0) \wedge \\ \left|k'_{h-1} - \tilde{k}_h\right| + \left|k'_{h+2} - \tilde{k}_h\right| < \lambda_{drifting}\end{array} \quad (2)$$

where k'_{h-1} and k'_{h+2} are the slopes of base topic segments just before b_h and after b_{h+1} respectively, and

$$\tilde{k}_h = \frac{|(y_{start})_h - (y_{end})_{h+1}|}{|(x_{start})_h - (x_{end})_{h+1}|}.$$

In other words, an *interrupted* pattern is composed of two (or more) consecutive *drifting* base topic segments which have slopes with opposite directions and which interrupt a *dominated* or *drifting* pattern. Based on the Equation 2, the context provided by the segments before and after the wedge shaped formation will decide whether a combination should happen. Examples are shown in Figure 10.

When an interrupted pattern is identified, the slope (k) and the concentration (σ) of the new base topic segment are recalculated. In order to keep the interrupted feature, we use the maximum of σ_i and σ_{i+1} as the σ for the new segment. Thus, interrupted pattern segments are (properly annotated and) replaced with the dominated or drifting base topic segments respectively for further processing.

Note that identification of an interrupt causes a number of base segments to be combined into a single (interrupted) segment. Therefore, the process of combining segments should be repeated until all segments (over-)split due to interrupts are identified and merged.

4.2.4 Elimination of Topic Oversegmentations

The previous steps of CUTS identify and classify topic segments and their development patterns. Note that, after the process ends, it is possible to have a number of consecutive dominated segments, each with a small drift but with significantly different concentrations, as shown in Figure 9. Since these segments are dominated (i.e., are concentrated around the same horizontal line), these segments are similar to each other in content; i.e., they have the same topic in focus. On the other hand, the different degrees of concentrations of the individual topic segments highlight that each segment has a different *degree of focus* around this common topic. We note that, for effective topic-based indexing and visualization, splitting consecutive segments concentrated around the same topic (because their degrees of concentrations around this common topic are different) may be unnecessary. Therefore, to enable proper labeling of a sequence of consecutive segments concentrated around the same common topic, CUTS merges such consecutive dominated patterns and places the segment boundaries only at the end-points of the merged segment.

4.2.5 Constructing A Hierarchy of Topic Segments

The above steps provide a sequence of segment boundaries and topic development patterns for each segment. However, in reality, these segments form the lowest layer of a hierarchy of segments. To discover the overall hierarchy, we reiterate the above steps with the boundaries that are already discovered:

(a) We first combine the entries in each segment into a single processing unit;

(b) We then apply the topic segmentation and development pattern analysis algorithm to the new sequence of units

These steps are repeated (bottom-up) until no new topic segments can be discovered. A hierarchical index or navigation structure can be constructed on the basis of these levels of segments.

5. EXPERIMENTS

In this section, we present experiments we carried out to evaluate the proposed segmentation and topic development analysis algorithms. The experimental evaluation is composed of two parts. In the first part, we use a blog as data source to verify our approach. In the second part, as a comparison for blog segmentation, a book (with a known table of contents) is taken as input. These experiments show the characteristic differences between the two text stream classes, blogs and book pages.

5.1 Topic Segmentation and Annotation of Blogs

Source Data. To evaluate the performance of the proposed approach, we used entries from popular political commentary weblog (Talking Point Memo [4]) as the data source. For the experiments presented here, we are using a 40-week sample (from January, 2005 to October, 2005) of entries from this source. For ease of analysis of the results, in this experiment, we unify the set of entries in one week into a single text unit. For the experiments presented here, we do not use any other supporting topic analysis method (such as LSI) to reduce the noise in the data. In other words, the difference matrix used for analysis is simply the keyword vectors of the weekly collections of blog entries from [4].

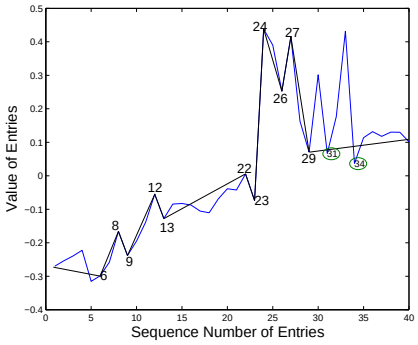


Figure 11: Topic segmentation for the blog entries. The numbers in the plot represent topic boundaries; the two circles denote the end-points of an interruption identified by the algorithm

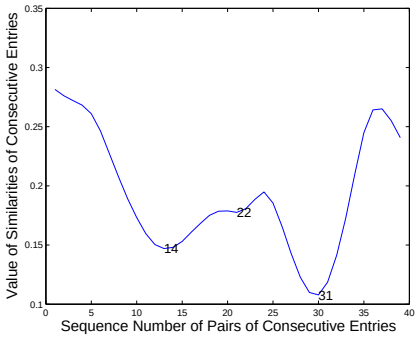


Figure 12: Similarities-curve based topic segmentation by [6]. The numbers denote the topic boundaries identified by that approach

Table 1: Base (lowest level) topic segments for the blog entries taken from [4], from January 02 to October 05

Seg#	Weeks	Var.(σ)	Slope(K)	Pattern	Length
1	1...3	0.0010	0.0158	**dominated**	3
2	4...5	0.0347	-0.0380	**dominated**	2
3	6...7	0.0096	0.0661	**drifting**	2
4	8	0.0000	-0.0719	**drifting**	1
5	9...11	0.0022	0.0610	**drifting**	3
6	12	0.0000	-0.0727	**drifting**	1
7	13...21	0.0047	0.0148	**dominated**	9
8	22	0.0000	-0.0797	**drifting**	1
9	23	0.0000	0.5122	**drifting**	1
10	24...25	0.0120	-0.0927	**drifting**	2
11	26	0.0000	0.1626	**drifting**	1
12	27...28	0.0116	-0.1723	**drifting**	2
13	29...40	0.2424	0.0028	**dom/interrupt.**	12

Comparison of Segmentation by CUTS versus [6]. When implementing CUTS, we used a *MinSpan* of 5% of the total number of units (i.e., about 2). Similarly, $\lambda_{drifting}$ is set to be 5% of the range of difference values in the curve. Table 1 contains the base segments obtained through Section 4.1. Figure 11 shows the corresponding topic segments obtained by processing these base segments. In this figure, the horizontal axis represents the sequence number of weeks in a chronological order; the vertical axis represents the MDS assigned values for each week of entries (Section 3). The plot contains both the original curve and the result of topic segmentation process: the line segments in the figure are the identified topic segments. The numbers on the curve denote the topic segment boundaries. According to Figure 11, there are eleven topic boundaries in this sequence of entries: $(6, 8, 9, 12, 13, 22, 23, 24, 26, 27, 29)$.

For comparison purposes, we also use [6] to get topic boundaries for the same data set. The result is shown in Figure 12. The horizontal axis depicts the sequence number of pairs of consecutive weeks; the vertical axis shows the similarity value between consecutive weeks. This approach identifies only three topic boundaries, $(14, 22, 31)$ using the local minima as shown in Figure 12.

Interpretation of the Results. The analysis of results from CUTS (and comparisons with the similarity curve based segmentation approach suggested in [6]) leads to the following observations supporting the proposed hierarchical topic development pattern identification mechanism:

(a) At the highest level, there is an almost constant-slope topic drifting of content in the blog entries depicted in Figure 11.

Such a pattern is not visible in Figure 12, which depicts the alternative similarities curve based method which relies on only the similarities between neighboring weeks.

(b) At the lowest level, the proposed segmentation approach provides 12 segments for the 40-week period; thus the average length of topic segments is about 3-4 weeks. However, there are two long segments (one 9 weeks long and the other 12 weeks long) where the content does not change much. Excluding these, the average segment length is around 2 weeks. In other words, at the lowest level, the proposed approach visualizes the short-term shifts in the content development.

[6], on the other hand, provides only four segments for the entire period of 40 weeks.

(c) There are two notable jumps in Figure 11. One is at week 24, and the other is at week 33. After both jumps, after a few weeks, the content development pattern reverts back to its normal constant-slope pattern. At the lowest-level (Table 1), the second jump is recognized as an interrupt (between weeks 31 and 34) in a 12-week long segment. Note that such topic development patterns are not visible with the alternative segmentation technique [6]:

EXAMPLE 5.1. *The analysis of the blog content shows that during the week of the first jump, news of a politician suspected of a corrupt transaction emerges and dominates the discussions for a few weeks. The second jump is also related to the news of the same political corruption case. Note that, the fact that these two jumps are related can easily be seen by analyzing the curve: the tops of both jumps are more or less at the same level (since MDS places them closer to each other, before the temporal dimension is considered). Such related clusters of jumps can easily be recognized by the proposed approach, by simply analyzing the y-axis of the curve.*

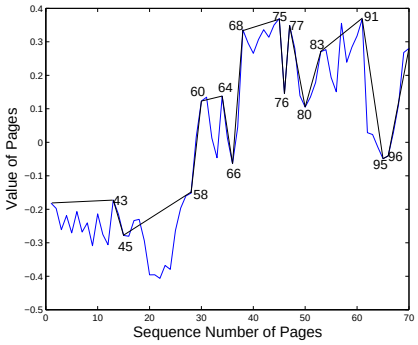


Figure 13: Topic segmentation results for the input *book* by CUTS.

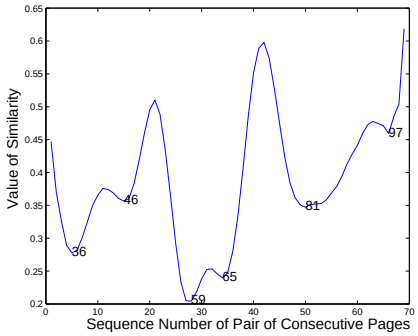


Figure 14: Similarities-curve based topic segmentation [6]

5.2 Topic Segmentation and Annotation of the Edited Material

Source Data. In order to **(a)** observe the differences between segmenting dynamically (and freely) growing weblog entries and edited material (such as books) and **(b)** revalidate the performance of the proposed segmentation algorithm against a material which already has a table of content (i.e., ground truth), we used a book [7] as the data source for segmentation and analysis.

In the experiments, we selected a number of sections as input (from page 31 to page 100) and treat each page as a separate entry. Moreover, we omitted the non-text material, such as figures, tables, etc. Naturally, since our aim is to measure the performance of segmentation and analysis mechanisms when there are no available table of contents, we ignored the tell-tale signs of segment boundaries, such as chapter, section, and subsection titles and headings.

Comparison of Segmentation by CUTS versus [6], and Ground Truth. Figure 13 shows the results of the curve segmentation. Once again, the plot contains both the original curve and the result of topic segmentation process: the line segments in the figure are the identified topic segments. Further analysis of the pages (Section 4.2) provides a total of 15 topic segment boundaries in the data:
$(43, 45, 58, 60, 64, 66, 68, 75, 76, 77, 80, 83, 91, 95, 96)$.

For comparison, we also segmented the book using the similarity curve based method method presented in [6] (and discussed in Section 1.2). The resulting similarity curve is shown in Figure 14. Using the local minima, this approach provides 6 segment boundaries: $(36, 46, 59, 65, 81, 97)$.

Table 2: Comparison of two approaches, [6] and CUTS, in terms of their edit distance from the ground truth provided by the table of content; we also present the edit distance values for random segmentation (RND) to show that edit distance measure correctly reflects quality of segmentation results

	Cost per Edit Oper.			Edit Distance		
No	**Ins**	**Del**	**Sub**	**RND**	**[6]**	**CUTS**
C1-1-1	1	1	1	31	17	14
C2-2-1	2	2	1	54	27	22
C4-2-1	4	2	1	58	38	28
C5-5-1	5	5	1	114	54	40

Finally, in order to measure the accuracy of the result, we used the table of content of the book as the ground truth, by identifying the lowest-level topic boundaries:

$$(43, 45, 49, 55, 57, 58, 61, 63, 69, 80, 84, 91, 94, 99).$$

Evaluation Measure. To quantify the precision of a segmentation approach, we computed the *edit distance* between the strings formed by the topic segment boundaries [23]. The edit operations include insertion of segment boundaries, deletion of boundaries, and substitution of one boundary with the other. Insertion makes up for the absence of boundaries missed by the segmentation algorithm. Deletion filters out boundaries resulting from over-segmentation. Substitution accounts for misplaced segmentation boundaries. Each operation has a cost; in particular, the cost of substitution measures the cost of the replacement (of one boundary with another) in terms of the page differences between the detected and correct boundary. Naturally, a result which has a small edit distance with the actual table of content is more desirable.

The edit distance results are shown in Table 2. For insertion or deletion, the value in the table represents the cost per operation. However, for substitution, the value is a multiplicative factor. For example, if we substitute a boundary at page 23 with a boundary at 32, the total substitution cost would be the product of the substitution factor in the table (i.e., 1) and the substitution page difference (i.e.,9). In a sense, the insertion and deletion costs in Table 2 measure the penalties associated with over- and under-segmentation, relative to mis-segmentation of the content by a single page.

Validation of the Evaluation Measure. Table 2 also contains edit distance values for random segmentations (averaged over 100 random iterations) to show that edit distance function we use in this paper properly measures the qualities of segmentation results. By comparing the edits distance values under the column RND against the columns titled [6] and CUTS, we can see that the edit distance values for random segmentation results are $\%100 - \%300$ times higher than the edit distance values for the segmentation results returned by [6] and our technique, CUTS. In other words, higher edit distance values reflect uninformed (more random) segmentations, while lower edit distance values reflect less-random (more informed) segmentation.

Interpretation of the Results. The analysis of the Figures 13 and 14 and Table 2, provides the following observations, supporting the proposed approach:

(a) At the highest level, Figure 13 shows the content concentration and divergences. There are two major topic

concentrations (pages 31-57 and 68-100) , which are connected by an intermediary segment (pages 58-67).

Such topic development patterns are not visible in the similarities curve based approach (Figure 14).

(b) The similarities curve based approach [6] significantly under-segments the material. When such under- segmentations are not penalized (C1-1-1) or lightly penalized (C2-2-1) the two approaches provide comparable results (Table 2), though the proposed scheme still performs better.

When on the other hand, the missing segment boundaries are probably penalized (C-4-2-1 and C-5-5-1), the advantage of the proposed approach relative to alternatives becomes clear.

Finally, note that, when compared with results of the blog, there are more dominated patterns in the the book. We omit the detailed table of the 15 segments due to space constraints, but the difference in topic development patterns is visible in Figures 11 and 13. The first has a constantly drifting shape (highlighted with the almost constant slope of the curve), but the second one has two more or less flat sections, with a clear jump from one to the other. This difference is due to the edited nature of the content in the book.

6. CONCLUSIONS

With the goal of constructing navigational tools and appropriate index structures for blogs, we developed a novel topic segmentation method, CUTS, which enables us to identify the topic development patterns in text streams. CUTS use multi-dimensional scaling to map entries in the blog into a curve which materializes the various patterns of interest. An adaptive, hierarchical curve segmentation technique is used for identifying topic segments and classifying them into different topic development patterns. Thus, unlike existing methods, CUTS has the advantage of not only better precision in segmentation, but also of providing topic development patterns, useful in annotating and indexing the returned segments. We experimentally showed that CUTS achieves a better performance than recently proposed methods for topic segmentation. As mentioned in Section 2, it's possible to make topic development analysis when multiple blogs with similar content structure is available. In the future work, we will explore benefiting from such knowledge.

7. REFERENCES

[1] David Sifry's Blog, http://www.sifry.com/alerts/.

[2] The search engine for blogs from Google: http://blogsearch.google.com/.

[3] A blog search engine supporting five language: http://www.bloogz.com/.

[4] The blog maintained by Joshua Micah Marshall, http://www.talkingpointsmemo.com/.

[5] James Allan, Courtney Wade, and Alvaro Bolivar. Retrieval and novelty detection at the sentence level. In *SIGIR*, pages 314–321, 2003.

[6] Pierre Andrews. Semantic topic extraction and segmentation for efficient document visualization. Master's thesis, School of Computer & Communication Sciences, Swiss Federal Institute of Technology, Lausanne, 2004.

[7] Grigoris Antoniou and Frank van Harmelen. *A Semantic Web Primer*. The MIT Press, 2004.

[8] David M. Blei and Pedro J. Moreno. Topic segmentation with an aspect hidden markov model. In *SIGIR*, pages 343–348, 2001.

[9] Rebecca Blood. *The Weblog Handbook: Practical Advice on Creating and Maintaining Your Blog*. Perseus Books Group, 2002.

[10] Thorsten Brants and Francine Chen. A system for new event detection. In *SIGIR*, pages 330–337, 2003.

[11] Thorsten Brants, Francine Chen, and Ioannis Tsochantaridis. Topic-based document segmentation with probabilistic latent semantic analysis. In *CIKM*, pages 211–218. ACM, 2002.

[12] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling, 2nd edition*. Monographs on Statistics and Applied Probability. Chapman & Hall, London, 2001.

[13] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Philip S. Yu & Hongjun Lu. Parameter free bursty events detection in text streams. In *VLDB*, pages 181–192, 2005.

[14] Marti A. Hearst. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

[15] Marti A. Hearst and Christian Plaunt. Subtopic structuring for full-length document access. In *SIGIR*, pages 59–68, 1993.

[16] Susan C. Herring, Lois Ann Scheidt, Sabrina Bonus, and Elijah Wright. Bridging the gap: A genre analysis of weblogs. In *HICSS*, 2004.

[17] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999.

[18] Nir Katzir, Michael Lindenbaum, & Moshe Porat. Curve segmentation under partial occlusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(5):513–519, 1994.

[19] David G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[20] Jeffrey C. Reynar. *Topic Segmentation: Algorithms and Applications*. PhD thesis, 1998.

[21] Paul L. Rosin. Techniques for assessing polygonal approximations of curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):659–666, 1997.

[22] Paul L. Rosin and Geoff A. W. West. Nonparametric segmentation of curves into various representations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(12):1140–1153, 1995.

[23] Steven S. Skiena. *The Algorithm Design Manual*. Springer-Verlag, 1998.

[24] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[25] Pak Chung Wong, Wendy Cowley, Harlan Foote, Elizabeth Jurrus, and Jim Thomas. Visualizing sequential patterns for text mining. In *INFOVIS*, pages 105–, 2000.

[26] Forrest W. Young. *Multidimensional Scaling, in Kotz & Johnson "Encyclopedia of Statistical Sciences"*, volume 5, pages 649–658. Wiley & Sons, 1985.

[27] Yi Zhang, James P. Callan, and Thomas P. Minka. Novelty and redundancy detection in adaptive filtering. In *SIGIR*, pages 81–88, 2002.