Trend-based Document Clustering for Sensitive and Stable Topic Detection [*]

Yoshihide Sato[a], Harumi Kawashima[b], Hidenori Okuda[b], and Masahiro Oku[b]

[a] NTT West Corporation [b] NTT Cyber Solutions Laboratories, NTT Corporation
1-1, Hikarino-oka, Yokosuka, Kanagawa, 239-0847 Japan
y.sato@west.east.ntt.co.jp, {kawashima.harumi, okuda.hidenori, oku.masahiro}@labs.ntt.co.jp

Abstract. The ability to detect new topics and track them is important given the huge amounts of documents. This paper introduces a trend-based document clustering algorithm for analyzing them. Its key characteristic is that it gives scores to words on the basis of the fluctuation in word frequency. The algorithm generates clusters in a practical time, with $O(n)$ processing cost due to preliminary calculation of document distances. The attribute allows the user to settle on the best level of granularity for identifying topics. Experiments prove that our algorithm can gather relevant documents with F measure of 63.0% on average from the beginning to the end of topic lifetime and it largely surpasses other algorithms.

Keywords: trend, clustering, gradient model, word frequency

1. Introduction

Due to the information explosion on the WWW, the cost of catching up with the latest trends has risen. Consumer Generated Media (CGM), such as weblogs and social networking service (SNS), are only accelerating this explosion. The best approach to recognizing trends from among the huge number of documents being created is to analyze the topics in them.

The goal of Topic Detection and Tracking (TDT) is to find state-of-the-art events in a stream of broadcast news stories (Allan et al., 1998). The study defines segmentation, new event detection, and event tracking as the major tasks. Segmentation proceeds by automatically dividing a text stream into topically homogeneous blocks. New event detection identifies stories in several continuous news streams that pertain to new or previously unidentified events. Event tracking identifies any and all subsequent stories describing the same event as sample instances of stories describing the event. Document clustering is an efficient approach to find topics in many documents.

In the tasks, new event detection is intimately related to clustering, and involves the functions of retrospective detection and on-line detection. The input to retrospective detection task is the entire corpus, and it is desired to divide them into event-specific groups. The input to on-line detection is a chronologically ordered document stream, and the change point of topics should be found.

On the WWW, where documents are numerous and increasing hourly, our goal is to provide an environment that supports users on finding and tracking the topics. In particular, sensitive detection of new topics is needed there. Then, our research is categorized as both on-line event detection and event tracking in TDT. As a matter of fact, both aspects are essential for adequately grasping the topics. This paper introduces a trend-based document clustering algorithm that enables the detection of topic occurrence at the earliest possible stage and the observation of topic transition.

The remainder of this paper is organized as follows: Section 2 describes related work; Section 3 describes our clustering algorithm; Section 4 describes our experiments and their results; and we conclude in Section 5.

2. Related Work

New event detection is the target of incremental clustering algorithms for on-line documents. In new event detection, conventionally, the similarity between new document and existing clsuters are calculated, and it is judged that which cluster is appropriate to include the document or any one is inappropriate. Developments of similarity measure (Dharanipragada et al., 1999) and term weighting (Brants et al., 2003) have proposed for better detection performance.

Many clustering algorithms have been applied for the task, such as single-pass based algorithm (Papka and Allan, 1998) and incremental k-means algorithm (Walls et al., 1999). They do not consider trends in on-line documents. However, it is required to focus attention on “time” for the sensitive topic detection.

The time-focused approach attempts to enhance detection performance by attenuating document similarities on the basis of time interval between documents (Yang et al., 1998). The strategy yielded measurable improvements in their on-line detection experiments. Word distribution in a corpus is used to choose core lexicon in the corpus (Zhang et al., 2004). The algorithm is applied to choose topical words in document if the documents are divided into some parts by their timestamps, though time temporal continuity is not considered.

Another incremental clustering algorithm F^{2}ICM (Ishikawa et al., 2001; Khy et al., 2006) is characterized by its ease in updating the statistics value used for calculating document similarities when new documents arrive. It defines the forgetting model as being exponential. It attenuates worth of documents exponentially as time passes, as if they are forgotten. In their model, recent documents are likely to be situated closer to each other, and older ones are likely to be more widely separated. The algorithm tends to generate clusters containing especially newer documents. On the other hand, persistent clusters are seldom generated. Thus, the algorithm is not the best way to observe topics continuously in terms of event tracking.

3. Trend-based Clustering Algorithm

What the prior studies lack in is the responsiveness to the current trends. Our approach to accomplish the goal is based on the trends in documents.

More and more documents describing the same event are created when people’s interest in the event arises. In on-line documents, a rapid increase in the frequency of a word indicates a trend toward one or more topics relevant to the word. Taking such trends into account, when clustering documents, yields the sensitive detection of new topics.

The most remarkable feature in our algorithm is that it senses current trends by word frequency fluctuation and gathers relevant documents based on the latest trends. Since its clustering process finishes in a short time after the classification granularity is indicated, it helps users to find adequate clustering results interactively that meet their intentions.

Word weights in our algorithm involve word appearance growth and its accumulative appearance. We declare the gradient model in the following part before describing word weights. The concept of gradient, essential idea in our algorithm, represents the growth of the two aspects. Word weighting algorithm is described in the second subsection, and the clustering algorithm is detailed in the third subsection.

3.1. Gradient Model

The impression of word appearance in a document declines over time. Suppose that the initial intensity of the impression is one, the intensity after time Δt can be defined as $e^{-\Delta t/T_l}$ following the forgetting model in F^{2}ICM (Ishikawa et al., 2001). T_l denotes the parameter deciding the rate of intensity attenuation.

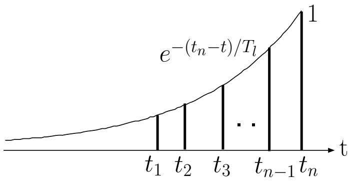

Figure 1: Impression Intensity.

Figure 1 shows the residual impression of each appearance of word *w*. Here, t_n is the time of the *n*th appearance. Given that the current time is t_n, the current appearance is assigned the maximum impression (the intensity is one), and the impression of *w* appearance at *t* remains $e^{-(t_n - t)/T_l}$. Then the total impressions score up to now, we define this as memory *M*, is described as follows.

$$M(w,t_n) = \sum\nolimits_{t=1}^{n} e^{-(t_n - t)/T_l}. \qquad (1)$$

In consecutive appearance of words, the memory *M* is efficiently updated if the previous result is stored. Updating follows Equation 2. If the word appears at t_{n+1}, time Δt after t_n, the new $M(w,t_{n+1})$ is represented as the sum of the previous result multiplied by the attenuation coefficient for the elapsed time Δt and the latest intensity.

$$\begin{aligned} M(w,t_{n+1}) &= \sum\nolimits_{i=1}^{n+1} e^{-(t_{n+1} - t_i)/T_l} \\ &= \sum\nolimits_{i=1}^{n} e^{-(t_n + \Delta t - t_i)/T_l} + 1 \\ &= e^{-\Delta t/T_l} \cdot \sum\nolimits_{i=1}^{n} e^{-(t_n - t_i)/T_l} + 1 \\ &= e^{-\Delta t/T_l} \cdot M(w,t_n) + 1. \qquad (2) \end{aligned}$$

On the other hand, the memory can be also regarded as the amount of word *w* appearances with time attenuation. With strong attenuation parameter T_s ($<T_l$), the memory is approximate recent frequency of the word; the recent frequency *F* is represented in Equation 3, as well as *M*.

$$F(w,t_n) = \sum\nolimits_{i=1}^{n} e^{-(t_n - t_i)/T_s}. \qquad (3)$$

Suppose that the recent frequency of a word is higher compared to the amount of permanent memory which the word has given up to now. Then the word is in the growth phase.

Here, we declare the concept of gradient as the difference of the memory from the recent frequency.

$$G(w,t_n) = \frac{\alpha F(w,t_n) - \beta M(w,t_n)}{M(w,t_n)}. \qquad (4)$$

The denominator is a normalizing element for eliminating the effects of general words with high frequency. α and β are coefficients.

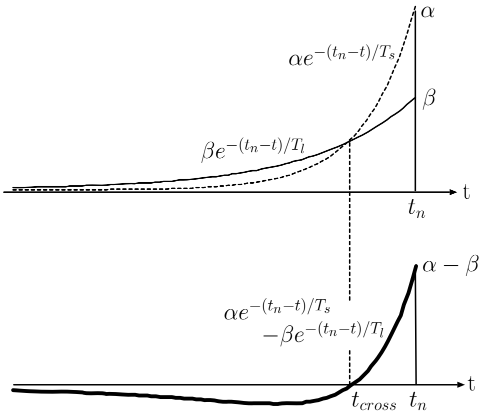

Figure 2: Differential Curve.

In the definition of gradient, the numerator, $\alpha F(w,t_n) - \beta M(w,t_n)$, is also explained by Figure 2. Two curves are drawn in the upper part. The solid line curve corresponds to the memory of *w*, with attenuation parameter T_l. The other dashed line corresponds to the recent frequency of the word, with strong attenuation parameter T_s. In fact, the curves do not directly plot memory or recent frequency rather the intensity attenuation used in calculating them. α and β in the definition of $G(w,t_n)$ correspond to intercepts respectively in the upper part of Figure 2. The bold line curve in the bottom part, we define this as differential curve, corresponds to the difference between the dashed and solid lines.

By the way, *M* and *F*, represented in summations, can be represented in integrations if a word appears continuously. Moreover, $G(w,t_n)$ score of a word with invariable frequency should be zero. Then the following equality is true.

$$\alpha\int_{-\infty}^{t_n} e^{-(t_n-t)/T_s}dt-\beta\int_{-\infty}^{t_n} e^{-(t_n-t)/T_l}dt=0. \qquad (5)$$

The first term is deformed as follows.

$$\begin{aligned}\alpha\int_{-\infty}^{t_n} e^{-(t_n-t)/T_s}dt &= \alpha\int_{-\infty}^{0} e^{\tau}T_s\,d\tau \\ &= \alpha T_s. \qquad (6)\end{aligned}$$

Deforming the second term in Equation 5 in the same way, the ratio of α to β is derived.

$$\frac{\alpha}{\beta}=\frac{T_l}{T_s}. \qquad (7)$$

For simplification, we regard the current (t=t_n) difference between two curves in Figure 2 as one.

$$\alpha-\beta=1. \qquad (8)$$

Incidentally, α and β are derived as follows, respectively, from Equation 7 and 8.

$$\alpha=\frac{T_l}{T_l-T_s},\quad \beta=\frac{T_s}{T_l-T_s}. \qquad (9)$$

$G(w,t_n)$ is interpreted as differential operator for frequency transition. $G(w,t_n)$ is zero if *w* appears with invariable frequency. The value is above zero if the frequency is increasing as time passes, and below zero if decreasing.

The time t_{cross} when differential curve crosses horizontal axis is also derived as follows.

$$t_{\text{cross}}=\frac{T_lT_s}{T_l-T_s}(\ln T_s-\ln T_l). \qquad (10)$$

The current growth rate is evaluated as an accumulation of past appearances by the differential curve. The appearance until t_{cross} affects negatively, and the appearance after t_{cross} is enhanced positively.

3.2. Trend Scores for Words and Document Expression

$G(w,t_n)$ quantifies the degree of growth. That is, the value does not accurately reflect the current trend. Word scores for trend-based clustering should reflect both growth and accumulative appearance.

Therefore, we define the trend score of word *w*, $TREND(w,t_n)$ in Equation 11, as the accumulation of its gradient scores with time attenuation. as well as recent frequency *F*.

$$TREND(w,t_n)=\sum_{i=1}^{n}G(w,t_i)\cdot e^{-(t_n-t_i)/T_s}. \qquad (11)$$

Trend score can be easily updated as well as memory or recent frequency by multiplying attenuation coefficient for elapsed time to the previous result, and then adding the latest gradient.

$$TREND(w,t_{n+1}) = e^{-\Delta t/T_s} \cdot TREND(w,t_n) + G(w,t_{n+1}). \qquad (12)$$

All the system has to do is to preserve the latest *M*, *F*, and *TREND* for each word, and the timestamp when they were last updated. When a new document arrives, *M* and *F* for the words in the document are updated as shown in Equation 2, and then the latest *TREND* is calculated as shown in Equation 12.

For trend-based document clustering, documents are expressed as vectors based on *TREND*. The vector of document d_n (arrived at t_n) is defined in the following equation.

$$\overrightarrow{v_n} = (W(w_1,t_n), W(w_2,t_n), W(w_3,t_n), \cdots, W(w_m,t_n)). \qquad (13)$$

Here, *m* denotes the number of unique words in d_n, and weight $W(w_x,t_n)$ is defined as follows.

$$W(w_x,t_n) = \begin{cases} TREND(w_x,t_n) & \text{if } TREND(w_x,t_n) > 0 \\ 0 & \text{else} \end{cases} . \qquad (14)$$

Equation 14 means that words with negative trend scores are eliminated from the document. This is because words with negative score can be considered to be completely irrelevant to current trends. However, the same word in another document is used for vector element if its trend score rises above zero due to the arrival of the second document.

3.3. Clustering based on Trend Scores

Many clustering algorithms have been invented, which are roughly classified into hierarchical or partitioning-optimization. The former generate a document tree called a dendrogram, whereas the latter give flat document clusters without any hierarchy.

For observing topic transition, the clustering algorithm should offer cluster reproducibility. There are two perspectives to reproducibility. First, the documents, already classified into clusters, should not be moved to another cluster when a new document arrives. If documents in a cluster move continuously, we cannot observe topic expansion and declination. Next, the documents should also remain stable when classification granularity is changed. If cluster coherency is preserved during the merging and partitioning processes, we can find the most effective level of granularity easily.

For the twin goals of reproducibilities, our approach is based on the single linkage clustering algorithm as a hierarchical scheme. Most hierarchical algorithms have processing cost of $O(n^2)$ where *n* is the number of documents. However, the single linkage algorithm has processing cost of $O(n)$ after a threshold is given and the nearest document for each document has been already identified.

Our clustering algorithm is composed of two steps. In the first step, distances between a new document and prior ones are calculated upon its arrival, and the nearest one is recorded in a nearest-neighbor table. In the second step, document clusters are generated based on the threshold given by the user. This admits of interactive analysis through the flexible threshold changes in a practical time by using the nearest-neighbor table as a cache.

The distance between two clusters, in the single linkage algorithm, is defined as the distance between the closest documents in the clusters. One of the problems in this algorithm is the chaining phenomenon, which is due to the definition of distances between clusters. Even if each two documents in a cluster are substantially relevant, the farthest documents in the cluster may cover decidedly irrelevant topics. The focus of the single linkage algorithm is the result of giving preference to both high speed clustering performance and reproducibility rather than trying to suppress the influence of chaining.

The two key steps in our clustering algorithm are detailed below.

Step1: Updating Nearest-Neighbor Table

The left of Figure 3 visually shows the structure of a nearest-neighbor table. The nodes denote documents, and d_n inside the circles denote document identifiers. The larger *n* is, the later the document arrived. Newer documents are placed right of others in the figure. The arrows denote the nearest links from newer document to older one, and the values beside the arrows denote the distance between two documents.

When the latest document d_6 arrives, the nearest-neighbor table is updated as follows.

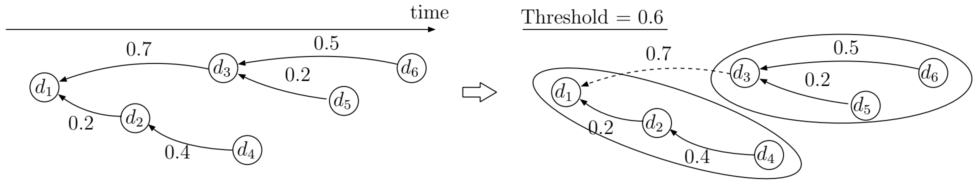

Figure 3: Nearest-Neighbor Table and Clusters

First, for the words in d_6, *M*, *F*, and *TREND* are updated based on their previous results and the elapsed time since they were last updated. In the definition of trend scores in Equation 11, the score for a word with the first appearance is one. The timestamp of d_6 is stored for the next update. Then, the current trend scores are assigned to d_6; the weighted words form the vector of the document. Though trend scores of words are updated whenever new documents arrive, $\overrightarrow{v_6}$ (the vector of d_6) is never updated even if the new documents contain same words in d_6. In other words, weighted words in a document reflect the trends at the time of its arrival. Therefore, scores in different documents may be different if the documents arrive at different times.

Second, $\overrightarrow{v_6}$ is compared to the vectors of older documents respectively, and the distances are calculated. The distance between two documents is defined by subtracting cosine similarity of respective vectors from one. The cosine similarity is normalized between zero and one.

$$\mathrm{dis}(d_m, d_n) = 1 - \mathrm{sim}(\overrightarrow{v_m}, \overrightarrow{v_n}). \qquad (15)$$

When the nearest document as d_6 is identified from among the older documents, the identifier of the nearest document and the corresponding distance are added to the nearest-neighbor table. In the Figure 3, the nearest neighbor as d_6 is d_3, and the distance is 0.5.

Step2: Clustering based on Nearest-Neighbor Table

The clustering process begins after the threshold is given. Since the nearest-neighbor table specifies the most similar older documents, document clusters can be generated in a short time.

The right of Figure 3 shows the composition of clusters after threshold 0.6 is given. Links shorter than 0.6 are valid, and only the link from d_3 to d_1 is regarded as invalid. As a result, two clusters, "d_1, d_2, d_4" and "d_3, d_5, d_6", are generated. Since it is not necessary to update distances between clusters during clustering processes, the algorithm can generate clusters with $O(n)$ processing cost. Even if the threshold is changed, the algorithm can process again with the same cost.

4. Evaluation

For the evaluation, we reviewed the clusters generated by our algorithm, and examined the sensitivity performance as regards new topics.

We used the Mainichi newspaper tagged corpus (in Japanese) in our experiments. All articles are tagged with issued date and category of the page on which they were placed, such as world topics, politics, economics, and sports. Several keywords are attached to each article.

We extracted 1,037 articles which are classified as world topics in January and February in 1994. The number of keywords in articles varies from 8 to 243, and the average article includes 54.2 keywords. We used the keywords as the elements of the document vectors. Though we stated that trend scores are updated each time a new document arrives, the scores are updated once a day in this experiment because the documents in the corpus had only day-based timestamps.

4.1. Reviewing Clusters

To comprehend the characteristics of our clustering algorithm, we reviewed the clusters formed with different thresholds.

In this experiment, we set T_l as 10(days) and T_s as 5(days) so as to set t_{cross} to 7(days) approximately. The appearance in the last seven days is thus emphasized in calculating trend scores. Our word weighting algorithm performs reasonably well only when the positive and negative regions of the differential curve are populated by existing documents; word scores calculated in the earlier period are not proper. Therefore, the articles prior to the last thirty days (about half of whole period) were not used for constructing the nearest-neighbor table, though all articles were processed to obtain trend scores during the last thirty days. Consequently, 567 articles (from the total of 1,037) were included in the nearest-neighbor table, and used for clustering.

Table 1 and Table 2 summarize the largest top 10 clusters yielded by our algorithm. They are the results at the end of the two months. In the results in Table 1, instead of giving distance threshold directly, we indicate the threshold that made the number of clusters half the number of articles in the nearest-neighbor table. In the results in Table 2, we set the threshold to make the number of clusters one quarter the number of articles. Clusters are sorted by size, the number of articles in it, in descending order. Span is the days from the timestamp of the latest article to that of the earliest one in each cluster. Clusters are manually summarized at the rightmost column.

Table 1: The Summary of Top 10 Clusters (clusters: 1/2).

No.	articles	span(days)	summary
1-1	84	22	Sarajevo, Bosnia-Herzegovina, Russia, etc
1-2	57	21	Bosnia, China, Vietnam, etc
1-3	13	3	Hebron random shooting
1-4	10	28	USA(North Korea, China), UN
1-5	6	5	Russian election
1-6	6	9	relationship between China and Taiwan
1-7	6	21	Austria, Ukraine, Russia
1-8	6	3	Myanmar(Aung San Suu Kyi)
1-9	6	5	USA(lifting of the economic sanctions for Vietnam)
1-10	6	23	North Korea(IAEA), China, Iran & Iraq

Table 2: The Summary of Top 10 Clusters (clusters: 1/4).

No.	articles	span(days)	summary
2-1	160	30	North Korea(IAEA), China, South Korea
2-2	87	22	Sarajevo, Bosnia-Herzegovina, PKO, etc
2-3	21	24	relationship between China and Taiwan, North Korea
2-4	14	8	Russian politics
2-5	13	3	Hebron random shooting
2-6	10	9	Italy, China, NATO
2-7	9	19	North Korea, Russia
2-8	9	14	Taiwan, Israel
2-9	7	21	NATO, Russia
2-10	6	3	Myanmar(Aung San Suu Kyi)

As for the clusters in Table 1, a review finds that No.1-3, 1-5, 1-6, 1-8, and 1-9 cover single topics; the other clusters are composed of several topics. The results are relative to cluster span. Clusters with short span gather relative articles quite precisely. Longer cluster spans indicate more topics. Larger clusters, such as No.1-1 and 1-2, cover especially wide various topics due to the chaining effect.

The clusters in Table 2 are larger. Though some larger clusters are created by general words such as country name, and other larger ones are affected by chaining, two of top ten clusters, No. 2-5 and 2-10, completely correspond to No.1-3 and 1-8 in Table 1. Both events described in these two clusters occurred at the end of February, the last one week in our dataset. Our algorithm sensitively gathered the events at the earliest possible stage just after their occurrences and separated them from the other events definitely.

4.2. Sensitivity to New Topics

The purpose of the next experiment was to evaluate the algorithm's sensitivity to new topics. Sensitivity is achieved when the word scores reflect current trends. We started by comparing the relationship between daily document frequency of a word and its scores. Next, we examined the performance of gathering documents related to new topics.

In the experiment, as benchmark word weighting algorithms, we prepared simple "IDF", "weighted-DF(W-DF)" using the summation of document frequency with time attenuation, and "*Gradient*" as the growth of word appearance, in addition to our word weighting algorithm "*Trend*". W-DF corresponds to $F(w,t_n)$, and *Gradient* corresponds to $G(w,t_n)$; they are obtained in the process of trend score calculation. In scoring words by IDF, the total number of documents and document frequency of words were updated each day by using articles up to the day because it is not feasible to obtain future given our assumption. Therefore, scores by IDF changed daily. We also regarded minus scores as zero in *Gradient*, as well as *Trend*.

Word Scores
Word scores yielded by the four algorithms are drawn in Figure 4.

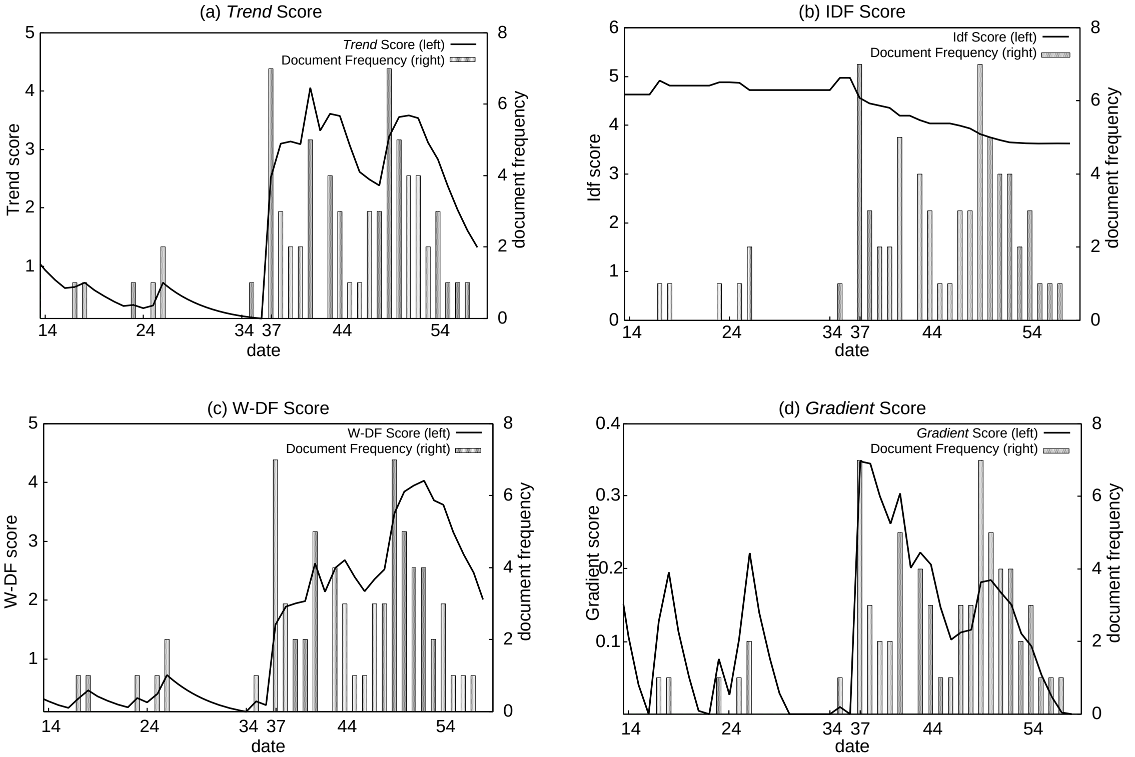

Figure 4: Document Frequency and Word Scores: "Sarajevo" (a)*Trend* (b)IDF (c)W-DF (d)*Gradient*.

They are the results for the word "Sarajevo", which was frequently used in news articles in the period. The bars denote daily document frequency where the word appears. The lines denote the scores output by each algorithm. *Trend*(a) surges at the first peak in document frequency at 37th day, and hovers at a relatively high level on following days. IDF(b) changes fluidly while it declines as document frequency increases. The specific difference between *Trend* and W-DF(c) is the sensitivity to dramatic increase of document frequency. W-DF rises gradually as time passes after the first high peak on the 37th day because it reflects the accumulation of frequency. *Gradient*(d) reflects the peak on the 37th day as well as *Trend*. However, it also overreacts to smaller peaks in the earlier period. It reflects the behavior seen when the scores are obtained from just the rate of frequency change. *Gradient* scores rise strongly at small peaks only if it was missing articles on prior days. On the other hand, due to the negative coefficient in the differential curve, it seldom reaches high scores after big peaks pass.

Eventually, *Trend* scores faithfully follow document frequency change, and they reflect the beginning and the end of trend lifetime. Moreover, they are unaffected by smaller peaks.

Clustering Performance

We evaluated clustering performance precisely.

For this experiment, we prepared a target article group with fifteen articles among the corpus. They are manually gathered so as to cover a single topic. For the four algorithms, nearest-neighbor tables were constructed for the last thirty days, as in the previous experiment. After constructing them, we generated clusters by adding articles day by day. The first results contain the articles in the first day of the thirty day period. The second ones contain the articles the first two days.

The previous experiments proved that using the large threshold, which yielded the cluster number of 25% of all documents, could gather articles related to new topics. Since reducing cluster number makes it easier to comprehend, we also used the large threshold in this experiment.

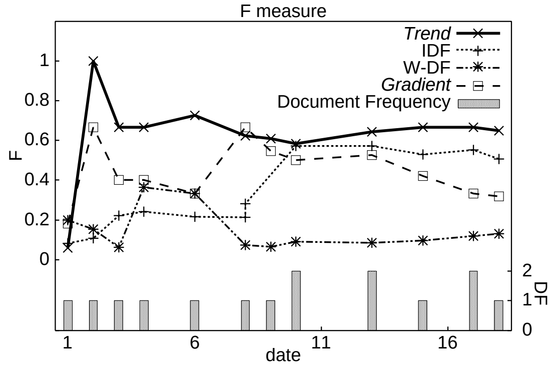

Figure 5: Target Group Detection Performance.

Table 3: Target Group Detection Performance.

date	*Trend*	IDF	W-DF	*Gradient*
1	0.059	0.083	0.20	0.18
2	**1.0**	0.11	0.15	**0.67**
3	0.67	0.22	0.062	0.40
4	0.67	0.24	**0.36**	0.40
6	0.73	0.22	0.33	0.33
8	0.63	0.21	0.073	**0.67**
9	0.61	0.28	0.065	0.55
10	0.58	**0.57**	0.091	0.50
13	0.64	**0.57**	0.086	0.53
15	0.67	0.53	0.097	0.42
17	0.67	0.55	0.12	0.33
18	0.65	0.51	0.13	0.32
Ave.	0.63	0.44	0.34	0.18

Figure 5 plots F measure day by day. The horizontal axis plots the days since the first article in the target group appeared. The bars are the number of articles in the target group. Data details are shown in Table 3; several dates are missing because the performance was estimated for the dates when the articles in the target group were issued. The bold values indicate the best performance achieved by each algorithm.

Trend performed well, especially for a few days after the first target article was issued, and recorded the highest performance throughout almost the entire period. The averaged performance was 63%. Though IDF demonstrated excellent performance after several articles were issued, it did not work well initially. The performance of W-DF was significantly below that of *Trend* while there was not so much of a difference between their word scores in Figure 4(a) and Figure 4(c). This is because the scores of general words tend to be overweighted by IDF. Though *Gradient* proved high performance compared to IDF or W-DF initially, it does not last so long, as also seen in word scores in Figure 4(d).

The experiment proved the performance of our algorithm to detect topics sensitively and follow them over time.

5. Conclusion

This paper introduced a trend-based clustering algorithm for detecting and tracking new topics in online documents. Our algorithm is marked by its word weighting algorithm based on the gradient model, that represents word appearance growth. The clustering process adopts the single linkage algorithm, that offers high speed clustering in a practical time.

The experiments proved that word weights in our algorithm reflect their frequency transitions and that the clustering algorithm can gather related news articles persistently as well as sensitively identify new topics. The performance meets the purpose of detecting new topics effectively and tracking them sustainably for documents increasing hourly.

Our next goal is to optimize the adequate parameters related to attenuation power for different types of documents.

References

Allan, James., Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. 1998. Topic Detection and Tracking Pilot Study Final Report, *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*.

Brants, Thorsten., Francine Chen, and Ayman Farahat. 2003. A System for New Event Detection, *Proceedings of SIGIR 2003, the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*

Dharanipragada, S., M. Franz, J.S. McCarley, S. Roukos, and T. Ward. 1999. Story Segmentation and Topic Detection in the Broadcast News Domain, *Proceedings of the DARPA Broadcast News Workshop*.

Ishikawa, Yoshiharu., Yibing Chen, and Hiroyuki Kitagawa. 2001. An On-Line Document Clustering Method Based on Forgetting Factors, *5th European Conference on Research and Advanced Technology for Digital Libraries*.

Khy, Sophoin., Yoshiharu Ishikawa, and Hiroyuki Kitagawa. 2006. Novelty-based Incremental Document Clustering for On-line Documents, *Proceedings of the 22nd International Conference on Data Engineering Workshops*.

Papka, Ron., and James Allan. 1998. On-Line New Event Detection using Single Pass Clustering, *UMASS Computer Science Technical Report 98-21*.

Walls, Frederick., Hubert Jin, Sreenivasa Sista, and Richard Schwartz. 1999. Topic Detection in Broadcast News, *Proceedings of the DARPA Broadcast News Workshop, 193-198*.

Yang, Yiming., Tom Pierce, and Jaime Carbonell. 1998. A Study on Retrospective and On-Line Event Detection, *Proceedings of the 21st Annual International ACM SIGIR conference on Research and development in information retrieval*.

Zhang, Huarui., Churen Huang, and Shiwen Yu. 2004. Distributional Consistency: As A General Method for Defining A Core Lexicon, *Proceedings of the 4th International Conference on Language Resources and Evaluation*.