Detecting Shifts in News Stories for Paragraph Extraction

Fumiyo Fukumoto **Yoshimi Suzuki**
Department of Computer Science and Media Engineering,
Yamanashi University
4-3-11, Takeda, Kofu, 400-8511, Japan
{fukumoto@skye.esb, ysuzuki@alps1.esi}.yamanashi.ac.jp

Abstract

For multi-document summarization where documents are collected over an extended period of time, the *subject* in a document changes over time. This paper focuses on subject shift and presents a method for extracting *key* paragraphs from documents that discuss the same *event*. Our extraction method uses the results of event tracking which starts from a few sample documents and finds all subsequent documents that discuss the same event. The method was tested on the TDT1 corpus, and the result shows the effectiveness of the method.

1 Introduction

Multi-document summarization of news stories differs from single document in that it is important to identify differences and similarities across documents. This can be interpreted as the question of how to identify an *event* and a *subject* in documents. According to the TDT project, an event is something that occurs at a specific place and time associated with some specific actions, and it becomes the *background* among documents. A subject, on the other hand, refers to a *theme* of the document itself. Another important factor, which is typical in a stream of news, is recognizing and handling subject shift. The extracted paragraphs based on an event and a subject may include the main points of each document and the background among documents. However, when they are strung together, the resulting summary still contains much overlapping information.

This paper focuses on subject shift and presents a method for extracting key paragraphs from documents that discuss the same event. We use the results of our tracking technique which automatically detects subject shift, and produces the optimal window size in the training data so as to include only the data which are sufficiently related to the current *subject*. The idea behind this is that, of two documents from the target event which are close in chronological order, the latter discusses (i) the same subject as an earlier one, or (ii) a new subject related to the target event. This is particularly well illustrated by the Kobe Japan quake event in the TDT1 data. The first document says that a severe earthquake shook the city of Kobe. It continues until the 5th document. The 6th through 17th documents report damage, location and nature of quake. The 18th document, on the other hand, states that the Osaka area suffered much less damage than Kobe. The subject of the document is different from the earlier ones, while all of these are related to the Kobe Japan quake event. We use the leave-one-out estimator of Support Vector Machines(SVMs)(Vapnik, 1995) to make a clear distinction between (i) and (ii) and thus estimate the optimal window size in the training data. For the results of tracking where documents are divided into several sets, each of which covers a different subject related to the same event, we apply SVMs again and induce classifiers. Using these classifiers, we extract key paragraphs.

The next section explains why we need to detect subject shift by providing notions of an *event*, a *subject class* and a *subject* which are properties that identify key paragraphs. After describing SVMs, we present our system. Finally, we report some experiments using the TDT1 and end with a very brief summary of existing techniques.

2 An Event, A Subject Class and A Subject

Our hypothesis about key paragraphs in multiple documents related to the target event is that they include words related to the *subject* of a document, a *subject class* among documents, and the target *event*. We call these words subject, subject class and event words. The notion of a subject word refers to the *theme* of the document itself, i.e., something a writer wishes to express, and it appears across paragraphs, but does not appear in other documents(Luhn, 1958). A subject class word differentiates it from a specific subject, i.e. it is a broader class of subjects, but narrower than an event. It appears across documents, and these documents discuss related subjects. An event word, on the other hand, is something that occurs at a specific place and time associated with some specific actions, and

it appears across documents about the target event. Let us take a look at the following three documents concerning the Kobe Japan quake from the TDT1.

1. Emergency work continues after earthquake in Japan
1-1. Casualties are mounting in [Japan], where a strong [earthquake] eight hours ago struck [Kobe]. Up to 400 {people} related {deaths} are confirmed, thousands of {injuries}, and rescue crews are searching
. . . . . .

2. Quake Collapses Buildings in Central Japan
2-1. At least two {people} died and dozens {injuries} when a powerful [earthquake] rolled through central [Japan] Tuesday morning, collapsing buildings and setting off fires in the cities of [Kobe] and Osaka.
2-2. The [Japan] Meteorological Agency said the [earthquake], which measured 7.2 on the open-ended Richter scale, rumbled across Honshu Island from the Pacific Ocean to the [Japan] Sea.
2-3. The worst hit areas were the port city of [Kobe] and the nearby island of Awajishima where in both places dozens of fires broke out and up to 50 buildings, including several apartment blocks,
. . . . . .

3. US forces to fly blankets to Japan quake survivors
3-1. United States forces based in [Kobe] [Japan] will take blankets to help [earthquake] survivors Thursday, in the U.S. military's first disaster relief operation in [Japan] since it set up bases here.
3-2. A military transporter was scheduled to take off in the afternoon from Yokota air base on the outskirts of Tokyo and fly to Osaka with 37,000 blankets.
3-3. Following the [earthquake] Tuesday, President Clinton offered the assistance of U.S. military forces in [Japan], and Washington provided the Japanese
. . . . . .

Figure 1: Documents from the TDT1

The underlined words in Figure 1 denote a subject word in each document. Words marked with '{}' and '[]' refer to a subject class word and an event word, respectively. Words such as 'Kobe' and 'Japan' are associated with an event, since all of these documents concern the Kobe Japan quake. The first document says that emergency work continues after the earthquake in Japan. Underlined words such as 'rescue' and 'crews' denote the subject of the document. The second document states that the quake collapsed buildings in central Japan. These two documents mention the same thing: A powerful earthquake rolled through central Japan, and many people were injured. Therefore, words such as 'people' and 'injuries' which appear in both documents are subject class words, and these documents are classified into the same set. If we can determine that these documents discuss related subjects, we can eliminate redundancy between them. The third document, on the other hand, states that the US military will fly blankets to Japan quake survivors. The subject of the document is different from the earlier ones, i.e., the subject has shifted.

Though it is hard to make a *clear* distinction between a subject and a subject class, it is easier to find properties to determine whether the later document discusses the same subject as an earlier one or not. Our method exploits this feature of documents.

3 SVMs

We use a supervised learning technique, SVMs(Vapnik, 1995), in the tracking and paragraph extraction task. SVMs are defined over a vector space where the problem is to find a decision surface that 'best' separates a set of positive examples from a set of negative examples by introducing the *maximum* 'margin' between two sets. Figure 2 illustrates a simple problem that is linearly separable.

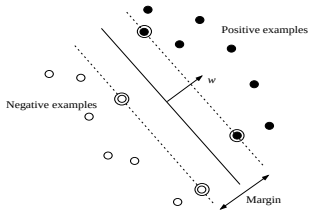

Figure 2: The decision surface of a linear SVM

Solid line denotes a decision surface, and two dashed lines refer to the boundaries. The extra circles are called support vectors, and their removal would change the decision surface. Precisely, the decision surface for linearly separable space is a hyperplane which can be written as $\mathbf{w}\cdot\mathbf{x}+b=0$, where $\mathbf{x}$ is an arbitrary data point($\mathbf{x}\in R^n$) and $\mathbf{w}$ and b are learned from a training set. In the linearly separable case maximizing the margin can be expressed as an optimization problem:

$$Minimize: \quad -\sum_{i=1}^{l}\alpha_i + \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j \mathbf{x}_i\cdot\mathbf{x}_j \quad (1)$$

$$\text{s.t}: \quad \sum_{i=1}^{l}\alpha_i y_i = 0 \quad \forall i:\ \alpha_i \geq 0$$

$$\mathbf{w} = \sum_{i=1}^{l}\alpha_i y_i \mathbf{x}_i \quad (2)$$

where $\mathbf{x}_i = (x_{i1},\cdots,x_{in})$ is the i-th training example and y_i is a label corresponding the i-th training example. In formula (2), each element of $\mathbf{w}$, w_k ($1 \leq k \leq n$) corresponds to each word in the training examples, and the larger value of $w_k = \sum_i^l \alpha_i y_i x_{ik}$ is, the more the word w_k features positive examples.

We use an upper bound value, E'_{loo} of the leave-one-out error of SVMs to estimate the optimal window size in the training data. E'_{loo} can estimate the performance of a classifier. It is based on the idea of leave-one-out technique: The first example is removed from l training examples. The resulting example is used for training, and a classifier is induced. The classifier is tested on the held out example. The process is repeated for all training examples. The number of errors divided by l, E_{loo}, is the leave-one-out estimate of the generalization error. E'_{loo}

uses an upper bound on E_{loo} instead of calculating them, which is computationally very expensive. Recall that the removal of support vectors change the decision surface. Thus the worst happens when every support vector will become an error. Let l be the number of training examples of a set S, and m be the number of support vectors. $E'_{loo}(S)$ is defined as follows:

$$E_{loo}(S) \leq E'_{loo}(S) = \frac{m}{l} \quad (3)$$

4 System Design

4.1 Tracking by Window Adjustment

Like much previous research, our hypotheses regarding event tracking is that exploiting time will lead to improved data adjustment because documents closer together in the stream are more likely to discuss related subject than documents further apart. Let $\vec{x}_1$, $\cdots$, $\vec{x}_p$ be positive training documents, i.e., being the target event, which are in chronological order. Let also $\vec{y}_1, \cdots, \vec{y}_q$ be negative training documents. The algorithm can be summarized as follows:

1. Scoring negative training documents

In the TDT tracking task, the number of labelled positive training documents is small (at most 16 documents) compared to the negative training documents. Therefore, the choice of *good* training data is an important issue to produce optimal results. We first represent each document as a vector in an n dimensional space, where n is the number of words in the collection. The cosine of the angle between two vectors, $\vec{x}_i$ and $\vec{y}_j$ is shown in (4).

$$cos(\vec{x}_i, \vec{y}_j) = \frac{\sum_{k=1}^{n} x_{ik} \cdot y_{jk}}{\sqrt{\sum_{k=1}^{n} x_{ik}^2} \cdot \sqrt{\sum_{k=1}^{n} y_{jk}^2}} \quad (4)$$

where x_{ik} and y_{jk} are the term frequency of word k in the document $\vec{x}_i$ and $\vec{y}_j$, respectively. We compute a relevance score for each negative training document by the cosine of the angle between a vector of the center of gravity on positive training documents and a vector of the negative training document, i.e., $cos(\vec{g}, \vec{y}_j)$ $(1 \leq j \leq q)$, where $\vec{y}_j$ is the j-th negative training document, and $\vec{g}$ is defined as follows:

$$\vec{g} = (g_1, \cdots, g_n) = (\frac{1}{p}\sum_{i=1}^{p} x_{i1}, \cdots, \frac{1}{p}\sum_{i=1}^{p} x_{in}) \quad (5)$$

x_{ij} $(1 \leq j \leq n)$ is the term frequency of word j in the positive document $\vec{x}_i$. The negative training documents are sorted in the descending order of their relevance scores: $\vec{y}_1, \cdots, \vec{y}_{q-1}$ and $\vec{y}_q$.

2. Adjusting window size

We estimate that the most recent positive training document, $\vec{x}_p$ discusses either (i) the same subject as the previous positive one, or (ii) a new subject. To do this, we use the value of E'_{loo}. Let $\vec{y}_1, \cdots, \vec{y}_r$ be negative training documents whose cosine similarity values are the top r among q negative training documents. Let also Set_1 be a set consisting of $\vec{x}_1$, $\vec{x}_p$, $\vec{y}_1, \cdots, \vec{y}_r$, and Set_2 be a set which consists of $\vec{x}_{p-1}$, $\vec{x}_p$, $\vec{y}_1, \cdots, \vec{y}_r$. We compute E'_{loo} on sets Set_1 and Set_2. If the value of E'_{loo} on Set_2 is smaller than that of Set_1, this means that $\vec{x}_p$ has the same subject as the previous document $\vec{x}_{p-1}$, since a classifier which is induced by training Set_2 is estimated to generate a smaller error rate than that of Set_1. In this case, we need to find the optimal window size so as to include only the positive documents which are sufficiently related to the subject. The flow of the algorithm is shown in Figure 3.

```
begin
  for k = 1 to p-3
      num = φ,
      Let Set_a = {x_1, x_p, ..., x_{p-k}, y_1, ..., y_{r-1}, y_r}.
      Set_b = {x_1, x_{(p-1)}, ..., x_{(p-1)-k}, y_1, ..., y_{r-1}, y_r}
          if E'_loo(Set_a) < E'_loo(Set_b)
              then num = k + 2 exit loop
          end_if
  end_for
  if num = φ
      then num = p
  end_if
end
```

Figure 3: Flow of window adjustment

On the other hand, if the value of E'_{loo} of Set_2 is larger than that of Set_1, $\vec{x}_p$ is regarded to discuss a new subject. We use all previously seen positive documents for training as a default strategy.

3. Tracking

Let num be the number of adjusted positive training documents. The top num negative documents are extracted from q negative documents and merged into num positive documents. The new set is trained by SVMs, and a classifier is induced. Recall that E'_{loo} is computationally less expensive. However, they are sometimes too tight for the small size of training data. This causes a high F/A rate which is signaled by the ratio of the documents that were judged as negative but were evaluated as positive. We then use a simple measure for the test document which is determined to be positive by a classifier. For each training document, we compute the cosine between the test and the training document vectors. If the cosine between the test and the negative documents is largest, the test document is judged to be negative. Otherwise, it is truly positive and tracking is terminated. The procedure **1**, **2** and **3** is repeated until the last test document is judged.

4.2 Paragraph Extraction

Our window adjustment algorithm is applied each time the document discusses the target event.

Therefore, some documents are assigned to more than one set of documents. We thus eliminate some sets which *completely* overlap each other, and apply paragraph extraction to the result. Our hypothesis about key paragraphs is that they include subject, subject class, and event words. Let x_p be a paragraph in the document x and $x^{\backslash 1}$ be the resulting document with x_p removed. Let also l be the total number of documents in a set where each document discusses subjects related to x. If x_p includes subject words, x_p is related to $x^{\backslash 1}$ rather than the other l-1 documents, since subject words appear across paragraphs in $x^{\backslash 1}$ rather than the other $l-1$ documents. We apply SVMs to the training data, which consists of l documents, and induce a classifier $sbj(x_p)$, which identifies whether x_p is related to $x^{\backslash 1}$ or not.

$$sbj(x_p) = \begin{cases} 1 & \text{if } x_p \text{ is assigned to } x^{\backslash 1} \\ 0 & \text{else} \end{cases}$$

We note that SVMs are basically introduced for solving binary classification, while our paragraph extraction is a multi-class classification problem, i.e., l classes. We use the *pairwise* technique for using SVMs with multi-class data(Weston and C.Watkins, 1998), and assign x_p to one of the l documents. In a similar way, we apply SVMs to the other two training data and induce classifiers: $sbj\_class(x_p)$ and $event(x_p)$.

$$sbj\_class(x_p) = \begin{cases} 1 & \text{if } x_p \text{ is assigned to } sbj\_class_{x^{\backslash 1}} \\ 0 & \text{else} \end{cases}$$

$$event(x_p) = \begin{cases} 1 & \text{if } x_p \text{ is assigned to } event_{x^{\backslash 1}} \\ 0 & \text{else} \end{cases}$$

$sbj\_class(x_p)$ refers to a classifier which identifies whether or not x_p is assigned to the set $sbj\_class_{x^{\backslash 1}}$ including $x^{\backslash 1}$. It is induced by training data which consists of m different sets including the set $sbj\_class_{x^{\backslash 1}}$, each of which covers a different subject related to the target event. The classifier $event(x_p)$ is induced by training data which consists of two different sets: one is a set of all documents including $x^{\backslash 1}$, and concerning the target event. The other is a set of documents which are not the target event. We extract paragraphs for which (6) holds.

$$sbj(x_p) = 1 \ \& \ sbj\_class(x_p) = 1 \ \& \ event(x_p) = 1 \qquad (6)$$

5 Experiments

We used the TDT1 corpus which comprises a set of different sources, Reuters(7,965 documents) and CNN(7,898 documents)(Allan et al., 1998a). A set of 25 target events were defined. Each document is labeled according to whether or not the document discusses the target event. All 15,863 documents were tagged by a part-of-speech tagger(Brill, 1992) and stemmed using WordNet information(Fellbaum, 1998). We extracted all nouns in the documents.

5.1 Tracking Task

Table 1 summarizes the results which were obtained using the standard TDT evaluation measure[1].

Table 1: Tracking results

N_t	Miss	F/A	Prec	F1
1	31%	0.16%	70%	0.68
2	27%	0.16%	79%	0.78
4	24%	0.09%	87%	0.78
8	23%	0.09%	87%	0.79
16	22%	0.09%	86%	0.79

'N_t' denotes the number of initial positive training documents where N_t takes on values 1, 2, 4, 8 and 16. When N_t takes on value 1, we use the document d and one negative training document $\vec{y}_1$ for training. Here, $\vec{y}_1$ is a vector whose cosine value of d and $\vec{y}_1$ is the largest among the other negative documents. The test set is always the collection minus the $N_t = 16$ documents. 'Miss' denotes Miss rate, which is the ratio of the documents that were judged as Yes but were not evaluated as Yes. 'F/A' shows false alarm rate, which is the ratio of the documents judged as No but were evaluated as Yes. 'Prec' stands for precision, which is the ratio of correct assignments by the system divided by the total number of the system's assignments. 'F1' is a measure that balances recall and precision, where recall denotes the ratio of correct assignments by the system divided by the total number of correct assignments. Table 1 shows that there is no significant difference among N_t values except for 1, since F1 ranges from 0.78 to 0.79. This shows that the method works well even for a small number of initial positive training documents. Furthermore, the results are comparable to the existing event tracking techniques, since the F1, Miss and F/A score by CMU were 0.66, 29 and 0.40, and those of UMass were 0.62, 39 and 0.27, respectively, when N_t is 4(Allan et al., 1998b).

The contribution of the adaptive window algorithm is best explained by looking at the window sizes it estimates. Table 2 illustrates the sample result of tracking for 'Kobe Japan Quake' event on the $N_t = 16$. This event has many documents, each of these discusses a new subject related to the target event. The result shows the first 10 documents in chronological order which are evaluated as positive. Columns 1-3 in Table 2 denote id number, dates, and title of the document, respectively. 'id=1', for example, denotes the first document which is evaluated as positive. Columns 4 and 5 stand for the result of our method, and the majority of three human judges, respectively. They take on three values: 'Yes' denotes that the document discusses the same subject as an earlier one, 'New' indicates that the document discusses a new subject, and 'No', that the document is not a positive document. We can

[1] http://www.nist.gov/speech/tests/tdt/index.htm

Table 2: The adaptive window size in Event 15, 'Kobe Japan Quake'

id	date	title	shifts		adjusted window size		
			system	actual	*recall*	*precision*	F1
1	01/17/95	Kobe Residents Unable to Commence Rescue Operations	New	New	100%	100%	1.00
2	01/17/95	Emergency Efforts Continue After Quake in Japan	Yes	Yes	100%	100%	1.00
3	01/17/95	Japan Helpline Worker Discusses Emergency Efforts	Yes	New	100%	5%	0.10
4	01/17/95	U.S. Businessman Describes Japan Earthquake	Yes	Yes	100%	80%	0.89
5	01/17/95	Osaka, Japan, Withstands Earthquake Better Than Others	**Yes**	**New**	100%	5%	0.09
6	01/17/95	President Clinton Drums Up Support in Humanitarian Trip	**No**	**New**	100%	5%	0.09
7	01/17/95	Engineer Examines Causes of Damage in Japan Quake	**Yes**	**New**	100%	50%	0.67
8	01/18/95	Mike Chinoy Updates Japan's Earthquake Recovery Efforts	Yes	Yes	100%	100%	1.00
9	01/18/95	Smoke Hangs in a Pall Over Quake-, Fire-Ravaged Kobe	New	New	100%	4%	0.08
10	01/18/95	Japanese Wonder If Their Cities Are Really 'Quakeproof'	New	New	100%	4%	0.07

see that the method correctly recognizes a test document as discussing an earlier subject or a new one, since the result of our method('system') and human judges('actual') coincide except for 'id=5, 6 and 7'.

Columns 6-8 stand for the accuracy of the adjusted window size. *Recall* denotes the number of documents selected by both the system and human judges divided by the total number of documents selected by human judges, and *precision* shows the number of documents selected by both the system and human judges divided by the total number of documents selected by the system. When the method correctly recognizes a test document as discussing an earlier subject('system = actual = Yes'), our algorithm selects documents which are sufficiently related to the current subject, since the total average of F1 was 0.82. We note that the ratio of precision in 'system = New' is low. This is because we use a default strategy, i.e., we use all previously seen positive documents for training when the most recent training document is judged to discuss a new subject.

5.2 Paragraph Extraction

We used 15 out of 25 events which have more than 16 positive documents in the experiment. Table 3 denotes the number of documents and paragraphs in each event. 'Avg.' in 'doc' shows the average number of documents per event, and 'Avg.' in 'para' denotes the average number of paragraphs per document. The maximum number of paragraphs per document was 100.

Table 4 shows the result of paragraph extraction. 'CNN' refers to the results using the CNN corpus as both training and test data. 'Reuters' denotes the results using the Reuters corpus. 'Total' stands for the results using both corpora. 'Tracking result' refers to the F1 score obtained by using tracking results. 'Perfect analysis' stands for the F1 achieved using the perfect (post-edited) output of the tracking method, i.e., the errors by both tracking and detecting shifts were corrected. Precisely, the documents judged as Yes but were not evaluated as Yes were eliminated, and the documents judged as No but were evaluated as Yes were added. Further, the documents were divided by a human into several sets, each of which covers a different subject related to the same event. The evaluation is made by three humans. The classification is determined to be correct if the majority of three human judges agrees. Table 4 shows that the average F1 of 'Tracking results'(0.68) in 'Total' was 0.06 lower than that of 'Perfect analysis'(0.74). Overall, the result using 'CNN' was better than that of 'Reuters'. One reason behind this lies in the difference between the two corpora: CNN consists of a larger number of words per paragraph than Reuters. This causes a high recall rate, since a paragraph which consists of a large number of words is more likely to include event, subject-class, and subject words than a paragraph containing a small number of words.

Table 3: Data

Event	CNN		Reuters	
	doc	para	doc	para
3(Carter in Bosnia)	26	314	8	37
5(Clinic Murders (Salvi))	36	416	5	34
6(Comet into Jupiter)	41	539	4	23
8(Death of Kim Jong Il)	28	337	39	353
9(DNA in OJ trial)	108	1,407	6	75
11(Hall's copter (N. Korea))	77	875	22	170
12(Humble, TX, flooding)	22	243	0	0
15(Kobe Japan quake)	72	782	12	64
16(Lost in Iraq)	34	395	10	78
17(NYC Subway bombing)	22	374	2	2
18(OK-City bombing)	214	3,209	59	439
21(Serbians down F-16)	50	572	15	135
22(Serbs violate Bihac)	56	669	35	349
24(USAir 427 crash)	32	435	7	98
25(WTC Bombing trial)	18	132	4	54
Avg.	55.4	12.7	15.2	9.7

Recall that in SVMs each value of word w_k is calculated using formula (2), and the larger value of w_k is, the more the word w_k features positive examples. Table 5 illustrates sample words which

Table 4: Performance of paragraph extraction

N_t	Tracking results			Perfect analysis		
	CNN	Reuters	Total	CNN	Reuters	Total
1	0.70	0.56	0.62	0.78	0.62	0.74
2	0.75	0.60	0.67			
4	0.76	0.61	0.70			
8	0.76	0.62	0.70			
16	0.77	0.62	0.72			
Avg.	0.85	0.60	0.68			

have the highest weighted value calculated using formula (2). Each classifier, $sbj(x_p)$, $sbj\_class(x_p)$, and $event(x_p)$ is the result using both corpora. The event is the Kobe Japan quake, and the document which includes x_p states that the death toll has risen to over 800 in the Kobe-Osaka earthquake, and officials are concentrating on getting people out. 'Words' denote words which have the highest weighted value in each classifier and they are used to determine whether x_p is a key paragraph or not. We assume these words are subject, subject class and event words, while some words such as 'earthquake' and 'activity' appear in more than one classifier.

Table 5: Sample words in the Kobe Japan quake

classifier	words
$sbj(x_p)$	earthquake activity Japan seismologist news conference living prime minister Murayama crew Bill Dorman
$sbj\_class(x_p)$	city something floor quake Tokyo aftershock activity street injury fire seismologist police people building cry
$event(x_p)$	Kobe magnitude survivor earthquake collapse death fire damage aftershock Kyoto toll quake magnitude emergency Osaka-Kobe Japan Osaka

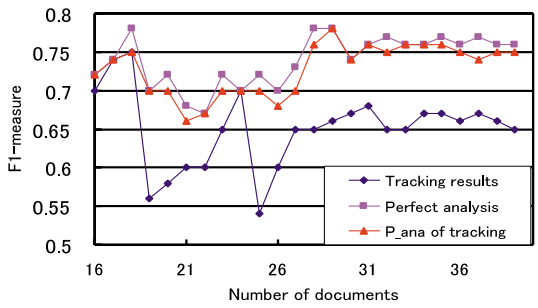

Figure 4: F1 v.s. the number of documents

Figure 4 illustrates how the number of documents influences extraction accuracy. The event is the US-Air 427 crash, and F1 is 0.68, which is lower than the average F1 of all events(0.79). The result is when N_t is 16. 'P_ana of tracking' refers to the result using the post-edited output of the tracking, i.e., only the errors of tracking were corrected, while 'Perfect analysis' refers to the result using the output: the errors by both tracking and detecting shifts were corrected. Figure 4 shows that our method does not depend on the number of documents, since the performance does not monotonically decrease when the number of documents increases. Figure 4 also shows that there is no significant difference between 'P_ana of tracking' and 'Perfect analysis' compared to the difference between 'Tracking results' and 'Perfect analysis'. This indicates that (i) subject shifts are correctly detected, and (ii) the performance of our paragraph extraction explicitly depends on the tracking results.

We note the contribution of detecting shifts for paragraph extraction. Figures 5 and 6 illustrate the recall and precision with two methods: with and without detecting shift. In the method without detecting shift, we use the 'full memory' approach for tracking, i.e., SVMs generate its classification model from all previously seen documents. For the result of tracking, we extract paragraphs for which $sbj(x_p) = 1$ and $sbj\_class(x_p) = 1$ hold. We can see from both Figure 5 and Figure 6 that the method that detects shifts outperformed the method without detecting shifts in all N_t values. More surprisingly, Figure 6 shows that the precision scores in all N_t values using the tracking results with detecting shift were higher than that of 'P_ana' without detecting shift. Further, the difference in precision between two methods is larger than that of recall. This demonstrates that it is necessary to detect subject shifts and thus to identify subject class words for paragraph extraction, since the system without detecting shift extracts many documents, which yields redundancy.

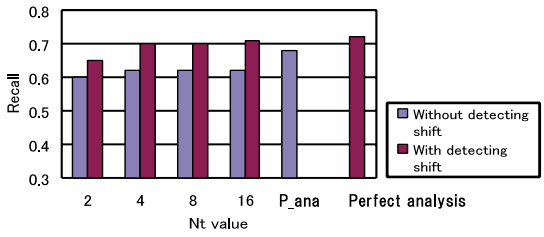

Figure 5: Recall with and without detecting shift

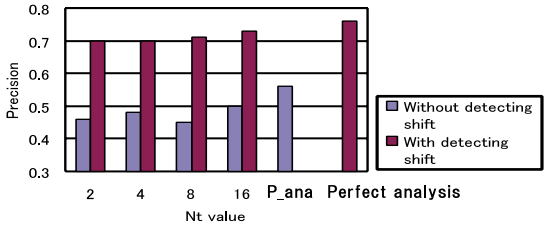

Figure 6: Precision with and without detecting shift

6 Related Work

Most of the work on summarization task by paragraph or sentence extraction has applied statistical techniques based on word distribution to the target document(Kupiec et al., 1995). More recently, other

approaches have investigated the use of machine learning to find patterns in documents(Strzalkowski et al., 1998) and the utility of parameterized modules so as to deal with different genres or corpora(Goldstein et al., 2000). Some of these approaches to single document summarization have been extended to deal with multi-document summarization(Mani and E.Bloedorn, 1997), (Barzilay et al., 1999), (McKeown et al., 1999).

Our work differs from the earlier work in several important respects. First, our method focuses on subject shift of the documents from the target event rather than the sets of documents from different events(Radev et al., 2000). Detecting subject shift from the documents in the target event, however, presents special difficulties, since these documents are collected from a very restricted domain. We thus present a window adjustment algorithm which automatically adjusts the optimal window in the training documents, so as to include only the data which are sufficiently related to the current subject. Second, our approach works in a *living* way, while many approaches are *stable* ones, i.e., they use documents which are prepared in advance and apply a variety of techniques to create summaries. We are interested in a substantially smaller number of initial training documents, which are then utilized to extract paragraphs from documents relevant to the initial documents. Because the small number of documents which are used for initial training is easy to collect, and costly human intervention can be avoided. To do this, we use a tracking technique. The small size of the training corpus, however, requires sophisticated parameters tuning for learning techniques, since we can not make one or more *validation sets* of documents from the initial training documents which are required for optimal results. Instead we use E'_{loo} of SVMs to cope with this problem. Further, our method does not use specific features for training such as 'Presence and type of agent' and 'Presence of citation', which makes it possible to be extendable to other domains(Teufel, 2001).

7 Conclusion

This paper studied the effectiveness of detecting subject shifts in paragraph extraction. Future work includes (i) incorporating Named Entity extraction into the method, (ii) applying the method to the TDT2 and TDT3 corpora for quantitative evaluation, and (iii) extending the method to on-line paragraph extraction for real-world applications, which will extract key paragraphs each time the document discusses the target event.

Acknowledgments

We would like to thank Prof. Virginia Teller of Hunter College CUNY for her valuable comments and the anonymous reviewers for their helpful suggestions.

References

J. Allan, J.Carbonell, G.Doddington, J.Yamron, and Y.Yang. 1998a. Topic Detection and Tracking pilot study final report. In *Proc. of DARPA Workshop*.

J. Allan, R.Papka, and V.Lavrenko. 1998b. Online new event detection and tracking. In *Proc. of ACM SIGIR'98*, pages 37–45.

R. Barzilay, K. R. McKeown, and M. Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proc. of ACL'99*, pages 550–557.

E. Brill. 1992. A simple rule-based part of speech tagger. In *Proc. of ANLP'92*, pages 152–155.

C. Fellbaum, editor. 1998. *Nouns in WordNet, An Electronic Lexical Database*. MIT.

J. Goldstein, V.Mittal, J.Carbonell, and M.Kantrowitz. 2000. Multi-document summarization by sentence extraction. In *Proc. of the ANLP/NAACL-2000 Workshop on Automatic Summarization*, pages 40–48.

J. Kupiec, J.Pedersen, and F.Chen. 1995. A trainable document summarizer. In *Proc. of ACM SIGIR'95*, pages 68–73.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM journal*, 2(1):159–165.

I. Mani and E.Bloedorn. 1997. Multi-document summarization by graph search and merging. In *Proc. of AAAI-97*, pages 622–628.

K. McKeown, J.Klavans, V.Hatzivassiloglou, R.Barzilay, and E.Eskin. 1999. Towards multidocument summarization by reformulation: Progress and prospects. In *Proc. of the 16th National Conference on AI*, pages 18–22.

D. Radev, H.Jing, and M.Budzikowska. 2000. Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies. In *Proc. of the ANLP/NAACL-2000 Workshop on Automatic Summarization*, pages 21–30.

T. Strzalkowski, J.Wang, and B.Wise. 1998. A robust practical text summarization system. In *Proc. of AAAI Intelligent Text summarization Workshop*, pages 26–30.

S. Teufel. 2001. Task-based evaluation of summary quality: Describing relationships between scientific papers. In *Proc. of NAACL 2001 Workshop on Automatic Summarization*, pages 12–21.

V. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.

J. Weston and C.Watkins. 1998. Multi-class Support Vector Machines. In *Technical Report CSD-TR-98-04*.