Topic Tracking Based on Linguistic Features

Fumiyo Fukumoto and Yusuke Yamaji

Interdisciplinary Graduate School of Medicine and Engineering,
Univ. of Yamanashi, 4-3-11, Takeda, Kofu, 400-8511, Japan
fukumoto@yamanashi.ac.jp, g03mk031@ccn.yamanashi.ac.jp

Abstract. This paper explores two linguistically motivated restrictions on the set of words used for topic tracking on newspaper articles: named entities and headline words. We assume that named entities is one of the linguistic features for topic tracking, since both topic and event are related to a specific *place* and *time* in a story. The basic idea to use headline words for the tracking task is that headline is a compact representation of the original story, which helps people to quickly understand the most important information contained in a story. Headline words are automatically generated using headline generation technique. The method was tested on the Mainichi Shimbun Newspaper in Japanese, and the results of topic tracking show that the system works well even for a small number of positive training data.

1 Introduction

With the exponential growth of information on the Internet, it is becoming increasingly difficult to find and organize *relevant* materials. Tracking task, i.e. starts from a few sample stories and finds all subsequent stories that discuss the target topic, is a new line of research to attack the problem. One of the major problems in the tracking task is how to make a clear distinction between a *topic* and an *event* in the story. Here, an event refers to the subject of a story itself, i.e. a writer wants to express, in other words, notions of who, what, where, when, why and how in the story. On the other hand, a topic is some unique thing that occurs at a specific place and time associated with some specific actions [1]. It becomes *background* among stories. Therefore, an event drifts, but a topic does not. For example, in the stories of 'Kobe Japan quake' from the TDT1 corpus, the event includes early reports of damage, location and nature of quake, rescue efforts, consequences of the quake, and on-site reports, while the topic is Kobe Japan quake.

A wide range of statistical and machine learning techniques have been applied to topic tracking, including k-Nearest Neighbor classification, Decision Tree induction [3], relevance feedback method of IR [12,13], hierarchical and non-hierarchical clustering algorithms [20], and a variety of Language Modeling [15,5,10,17]. The main task of these techniques is to tune the parameters or the threshold for binary decisions to produce optimal results. In the TDT context, however, parameter tuning is a tricky issue for tracking. Because only the small number of labeled positive stories is available for training. Moreover, the well-known past experience from IR that notions of who, what, where, when, why, and how may not make a great contribution to the topic tracking task [1] causes this fact, i.e. a topic and an event are different from each other.

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 10–21, 2005.

This paper explores two linguistically motivated restrictions on the set of words used for topic tracking on newspaper articles: named entities and headline words. A topic is related to a specific $place$ and $time$, and an event refers to notions of who($person$), where($place$), when($time$) including what, why and how in a story. Therefore, we can assume that named entities is one of the linguistic features for topic tracking. Another linguistic feature is a set of headline words. The basic idea to use headline words for topic tracking is that headline is a compact representation of the original story, which helps people to quickly understand the most important information contained in a story, and therefore, it may include words to understand what the story is about, what is characteristic of this story with respect to other stories, and hopefully include words related to both topic and event in the story. A set of headline words is automatically generated. To do this, we use a technique proposed by Banko [2]. It produces coherent summaries by building statistical models for content selection and surface realization. Another purpose of this work is to create Japanese corpus for topic tracking task. We used Mainichi Shimbun Japanese Newspaper corpus from Oct. to Dec. of 1998 which corresponds to the TDT3 corpus. We annotated these articles against the 60 topics which are defined by the TDT3.

The rest of the paper is organized as follows. The next section provides an overview of existing topic tracking techniques. We then describe a brief explanation of a headline generation technique proposed by Banko et al. [2]. Next, we present our method for topic tracking, and finally, we report some experiments using the Japanese newspaper articles with a discussion of evaluation.

2 Related Work

The approach that relies mainly on corpus statistics is widely studied in the topic tracking task, and an increasing number of machine learning techniques have been applied to the task. CMU proposed two methods: a k-Nearest Neighbor (kNN) classifier and a Decision-Tree Induction (dtree) classifier [1,20,3]. Dragon Systems proposed two tracking systems; one is based on standard language modeling technique, i.e. unigram statistics to measure story similarity [18] and another is based on a Beta-Binomial model [10]. UMass viewed the tracking problem as an instance of on-line document classification, i.e. it classifies documents into categories or classes [4,8,19,9,14]. They proposed a method including query expansion with multi-word features and weight-learning steps for building linear text classifiers for the tracking task [13]. These approaches, described above, seem to be robust and have shown satisfactory performance in stories from different corpora, i.e. TDT1 and TDT2. However, Carbonell claims that something more is needed if the system is intended for recognizing topic drift [3]. Yang et al. addressed the issue of difference between early and later stories related to the target event in the TDT tracking task. They adapted several machine learning techniques, including k-Nearest Neighbor(kNN) algorithm and Rocchio approach [21]. Their method combines the output of a diverse set of classifiers and tuning parameters for the combined system on a retrospective corpus. The idea comes from the well-known practice in information retrieval and speech recognition of combining the output of a large number of systems to yield a better result than the individual system's output. They reported that the new variants of kNN reduced up to 71% in weighted error rates on the TDT3-dryrun corpus.

GE R&D proposed a method for topic tracking by using summarization technique, i.e. using *content compression* rather than on corpus statistics to detect relevance and assess topicality of the source material [16]. Their system operates by first creating a topic tracking query out of the available training stories. Subsequently, it accepts incoming stories, summarizes them topically, scores the summaries(passages) for content, then assesses content relevance to the tracking query. They reported stories whose compressed content summaries clear the empirically established threshold are classified as being 'on topic'. Unlike most previous work on summarization which focused on extractive summarization: selecting text spans - either complete sentences or paragraphs - from the original story, this approach solves a problem for extractive summarization, i.e. in many cases, the most important information in the story is scattered across multiple sentences. However, their approach uses frequency-based term weighting. Therefore, it is not clear if the method can identify the most important information contained in a story.

These methods, described above, show that it is crucial to develop a method for extracting words related to both topic and event in a story. Like other approaches, our method is based on corpus statistics. However, our method uses two linguistically motivated restrictions on the set of words: named entities and headline words. We assume that named entities is one of the linguistic features for topic tracking, since both topic and event are related to a specific $place$ and $time$ in a story. Another linguistic feature is a set of headline words. The basic idea to use headline words is that headline is a compact representation of the original story, and therefore, it may include words to understand what the story is about, and hopefully include words related to both topic and event in the story.

3 Generating Headline

Banko et al. proposed an approach to summarization capable of generating summaries shorter than a sentence. It produces by building statistical models for *content selection* and *surface realization*. We used their method to extract headline words. Content selection requires that the system learns a model of the relationship between the appearance of words in a story and the appearance of corresponding words in the headline. The probability of a candidate headline, H, consisting of words (w_1,w_2,$\cdots$,w_n), can be computed:

$$P(w_1, \cdots, w_n) = \prod_{i=1}^{n} P(w_i \in H \mid w_i \in D) \cdot P(len(H) = n) \cdot \prod_{i=2}^{n} P(w_i \mid w_1, \cdots, w_{i-1}) \quad (1)$$

In formula (1), the first term denotes the words selected for the headline, and can be computed:

$$P(w_i \in H \mid w_i \in D) = \frac{P(w_i \in D \mid w_i \in H) \cdot P(w_i \in H)}{P(w_i \in D)} \quad (2)$$

where H and D represent the bags of words that the headline and the story contain. Formula (2) shows the conditional probability of a word occurring in the headline given

that the word appeared in the story. It has been estimated from a suitable story/headline corpus. The second term in formula (1) shows the length of the resulting headline, and can also be learned from the source story. The third term shows the most likely sequencing of the words in the content set. Banko et al. assumed that the likelihood of a word in the story is independent of other words in the headline. Surface realization is to estimate the probability of any particular surface ordering as a headline candidate. It can be computed by modeling the probability of word sequences. Banko et al. used a bigram language model. When they estimate probabilities for sequences that have not been seen in the training data, they used back-off weights [6].

Headline generation can be obtained as a weighted combination of the content and structure model log probabilities which is shown in formula (3).

$$arg\ max_H(\alpha \cdot \sum_{i=1}^{n} \log(P(w_i \in H \mid w_i \in D)) + \beta \cdot \log(P(len(H) = n)) + \gamma \cdot \sum_{i=2}^{n} \log(P(w_i \mid w_{i-1}))) \quad (3)$$

To generate a headline, it is necessary to find a sequence of words that maximizes the probability, under the content selection and surface realization models, that it was generated from the story to be summarized. In formula (3), cross-validation is used to learn weights, α, β and γ for a particular story genre.

4 Extracting Linguistic Features and Tracking

We explore two linguistically motivated restrictions on the set of words used for tracking: named entities and headline words.

4.1 Extracting Named Entities and Generating Headline Words

For identifying named entities, we use CaboCha [7] for Japanese Mainichi Shimbun corpus, and extracted Person Name, Organization, Place, and Proper Name.

Headline generation can be obtained as a weighted combination of the content and structure model log probabilities shown in formula (3). The system was trained on the 3 months Mainichi Shimbun articles((27,133 articles from Jan. to Mar. 1999) for Japanese corpus. We estimate α, β and γ in formula (3) using 5 cross-validation[1]. Fig. 1 illustrates sample output using Mainichi Shimbun corpus. Numbers to the right are log probabilities of the word sequence.

4.2 Tracking by Hierarchical Classification

In the TDT tracking task, the number of labeled positive training stories is small (at most 16 stories) compared to the negative training stories. Therefore, the choice of *good* negative stories from a large number of training data is an important issue to detect subject shifts for a binary classifier such as a machine learning technique, Support Vector Machines(SVMs) [22]. We apply hierarchical classification technique to the training data.

[1] In the experiment, we set α, β, γ to 1.0, 1.0, 0.8, respectively.

<Headline> パキスタン (Pakistan) カシミール問題解決へ (Kashimir issue)
第 3 パーティ仲裁国 (third party mediation) と会合 (meeting) </Headline>
イスラマバード, パキスタンは 2 週間以内に, 隣国インドについて話合いを持つ, すなわちパキスタンは第 3 国の仲裁国に対してインドとパキスタンで起きた過去の戦争に関するカシミール問題に対して調査するよう強く迫った...
(ISLAMABAD, Pakistan, Less than two weeks ahead of fresh talks with its uneasy neighbor India, Pakistan pressed on Saturday for international mediation in the thorny Kashmir issue, the flashpoint of two previous wars between the two countries...)

[Generated title words]
2: カシミール (Kashimir) 問題 (issue) -6.83
3: 第 3 (third) パーティ (party) 仲裁 (mediation) -11.97
4: カシミール (Kashimir) インド (India) Islamabad 再開 (resume) -23.84
5: カシミール (Kashimir) インド (India) 再開 (resume) イスラマバード (Islamabad) カシミール (Kashimir) -33.36
6: カシミール (Kashimir) インド (India) 再開 (resume) イスラマバード (Islamabad) カシミール (Kashimir) イスラム教 (Muslim) -38.32

Fig. 1. Simple story with original headline and generated output

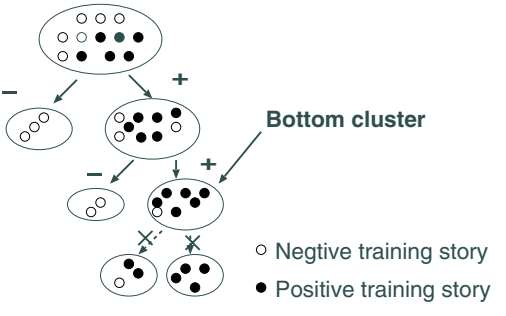

Fig. 2. Graphical representation of hierarchical classification

A hierarchical decomposition of a classification problem can be used to set the negative set for discriminative training. We use partitioning clustering algorithm, k-means ($k = 2$) which partitions a training data into clusters where similar stories are found in the same cluster and separated from dissimilar stories. Fig. 2 illustrates hierarchical classification of training data with k-means. Each level in Fig. 2 denotes the result obtained by a simple k-means (k=2) algorithm, and consists of two clusters: one is a cluster which includes positive and negative stories. Another is a cluster with only negative stories, each of these are dissimilar with the positive stories. The algorithm involves iterating through the data that the system is permitted to classify during each iteration. More specifically:

1. In the training data which includes all the initial positive training stories, select two initial seeds $\mathbf{g}$ and $\bar{\mathbf{s}}_i$, where $\mathbf{g}$ is a vector of the center of gravity on positive training stories, and $\bar{\mathbf{s}}_i$ is a vector of the negative training story which has the smallest value(as measured by cosine similarity) between $\bar{\mathbf{s}}_i$ and $\mathbf{g}$. The center of gravity $\mathbf{g}$ is defined as:

$$\mathbf{g} = (g_1, \cdots, g_n) \ = \ (\frac{1}{p}\sum_{i=1}^{p} s_{i1}, \cdots, \frac{1}{p}\sum_{i=1}^{p} s_{in}) \tag{4}$$

where s_{ij} $(1 \leq j \leq n)$ is the TF$*$IDF value of word j in the positive story $\mathbf{s}_i$.

2. Apply k-means (k=2) to the training data.
3. For the cluster which includes positive stories, iterate step 1 and 2 until positive training stories are divided into two clusters[2].

Tracking involves a training phase and a testing phase. During the training phase, we employ the hierarchy which is shown in Fig. 2 by learning separate classifiers trained by SVMs. '$\pm$1' in Fig. 2 denotes binary classification for stories at each level of the hierarchy. Each test story is judged to be negative or positive by using these classifiers to greedily select sub-branches until a leaf is reached. Once, the test story is judged to be negative, tracking is terminated. When the test story is judged to be positive by using a classifier of the bottom cluster, a cluster is divided into two: positive and negative stories. For each training data in the bottom cluster and test stories, we extract named entities and headline words. The result of training data is used to train SVMs and a classifier is induced. Each test story which also consists of a set of words produced by named entities and generating headline word procedures is judged to be negative or positive by using the classifier. This procedure, tracking, is repeated until the last test story is judged.

5 Experiments

5.1 Experiments Set Up

We chose the TDT3 corpus covering October 1, 1998 to December 31, 1998 as our gold standard corpus for creating Japanese corpus. The TDT3 corpus, developed at LDC, is a larger and richer collection, consisting of 34,600 stories with 60 manually identified topics. The stories were collected from 2 newswire, 3 radio programs and 4 television programs. We then create a Japanese corpus, i.e. we annotate Mainichi Shimbun Japanese Newspaper stories from October 1, 1998 to December 31, 1998 against the 60 topics. Not all the topics could have seen over the 3 months Japanese Newspaper stories. Table 1 shows 20 topics which are included in the Japanese Newspaper corpus.

'Topic ID' in Table 1 denotes ID number defined by the TDT3. The evaluation for annotation is made by three humans. The classification is determined to be correct if the majority of three human judges agrees. The Japanese corpus consists of 27,133 stories. We used it in the experiment. We obtained a vocabulary of 52,065 unique words after tagging by a morphological analysis, Chasen [11].

5.2 Basic Results

Table 2 summarizes the results using all words for each sequence that maximizes the probability, i.e. 14 sequences in all. The results were obtained using the standard TDT

[2] When the number of positive training stories(N_t) is 1, iterate step 1 and 2 until the depth of the tree in the hierarchy is identical to that of N_t=2.

Table 1. Topic Name

Topic ID	Topic name	Topic ID	Topic name
30001	Cambodian government coalition	30003	Pinochet trial
30006	NBA labor disputes	30014	Nigerian gas line fire
30017	North Korean food shortages	30018	Tony Blair visits China in Oct.
30022	Chinese dissidents sentenced	30030	Taipei Mayoral elections
30031	Shuttle Endeavour mission for space station	30033	Euro Introduced
30034	Indonesia-East Timor conflict	30038	Olympic bribery scandal
30042	PanAm lockerbie bombing trial	30047	Space station module Zarya launched
30048	IMF bailout of Brazil	30049	North Korean nuclear facility?
30050	U.S. Mid-term elections	30053	Clinton's Gaza trip
30055	D'Alema's new Italian government	30057	India train derailment

Table 2. The results

N_t	Prec.	Rec.	F	Miss	F/A	N_t	Prec.	Rec.	F	Miss	F/A
1	.000	.000	.000	1.000	.0000	8	.858	.432	.575	.568	.0001
2	.846	.040	.077	.960	.0000	16	.788	.520	.626	.480	.0004
4	.905	.142	.245	.858	.0000	Avg.	.679	.227	.305	.663	.0001

evaluation measure. 'N_t' denotes the number of positive training stories where N_t takes on values 1, 2, 4, 8 and 16. The test set is always the collection minus the N_t = 16 stories. 'Miss' denotes Miss rate, which is the ratio of the stories that were judged as YES but were not evaluated as YES for the run in question. 'F/A' shows false alarm rate, which is the ratio of the stories judged as NO but were evaluated as YES. 'Prec.' is the ratio of correct assignments by the system divided by the total number of system's assignments. 'F'(pooled avg) is a measure that balances recall(Rec.) and precision, where recall denotes the ratio of correct assignments by the system divided by the total number of correct assignments. We recall that a generated headline is a sequence of words that maximizes the probability. We set the maximum number of word sequence by calculating the average number of the original titles, and obtained the number of 15 words. The minimum number of words in a sequence is two. Fig. 3 illustrates the extracted headline for each sequence. Box in Fig. 3 shows a word, and 'arg max P(x)' denotes the maximum probability of a candidate headline. For example, the extracted sequence

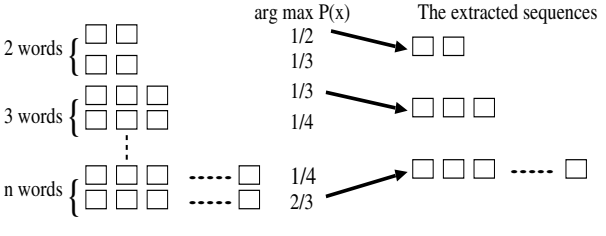

Fig. 3. The extracted headline for each sequence

of two words is the sequence whose maximum probability is $\frac{1}{2}$. Table 2 shows that our method is more likely to be effective for higher values of N_t, while F-score was 0 when $N_t = 1$.

5.3 Title Words

Our approach using the headline generation is to find a sequence of words that maximizes the probability. It can be produced for an arbitrary number of words. We recall that Table 2 shows the result using each sequence that maximizes the probability. However, when $N_t = 1$, the result was not good, as the F-score was zero. We thus conducted the following two experiments to examine the effect of the number of words in a sequence: (1) the tracking task using all words, each of which is the element of only one sequence that maximizes the probability(Fig. 4) and (2) the tracking using various number of word sequences(Fig. 5). In (2), we tested different number of words in a sequence, and we chose six words that optimized the global F-score. The results are shown in Tables 3 and 4.

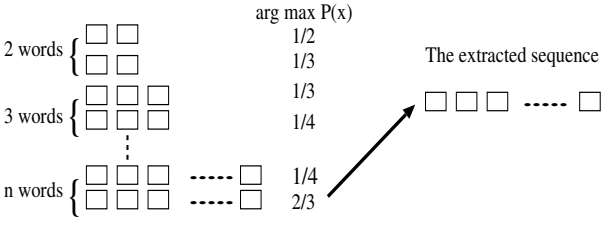

Fig. 4. The extracted headline for maximizing the probability

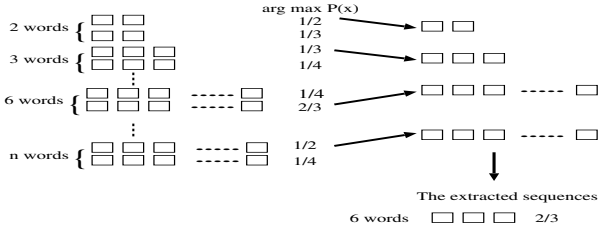

Fig. 5. The extracted headline for various sequences

Table 3 shows the tracking result using only one sequence of words that maximizes the probability, and Table 4 shows the result of six words. In Table 3, the average number of words which maximizes the probability for all the training data is 4.4, and the result is similar to that of Table 4. We can see from both Tables 3 and 4 that when the number of words in a sequence is small, the result has no effect with the number of positive training data, since the range of F-score in Table 3 is $0.415 \sim 0.478$, and that in Table 4

Table 3. The result using title words with high probabilities

N_t	Prec.	Rec.	F	Miss	F/A	N_t	Prec.	Rec.	F	Miss	F/A
1	.466	.375	.415	.626	.0005	8	.702	.372	.487	.628	.0003
2	.591	.402	.478	.599	.0003	16	.604	.393	.476	.607	.0007
4	.674	.340	.452	.660	.0003	Avg.	.607	.376	.462	.624	.0004

Table 4. The result using 3 title words

N_t	Prec.	Rec.	F	Miss	F/A	N_t	Prec.	Rec.	F	Miss	F/A
1	.608	.378	.465	.622	.0003	8	.687	.334	.453	.662	.0003
2	.652	.365	.466	.635	.0002	16	.734	.397	.516	.603	.0004
4	.709	.336	.456	.664	.0002	Avg.	.678	.362	.471	.637	.0003

is $0.453 \sim 0.516$. On the other hand, as we can see from Table 2, when the number of title words is large, the smaller the number of positive training data is, the worse the result is. To summarize the evaluation, the best result is when we use a sequence which consists of a small number of words, six words.

5.4 Named Entities

We assume that named entities is effective for topic tracking, since both topic and event are related to a specific $place$ and $time$ in a story. We conducted an experiment using various types of named entities. The results are shown in Table 5.

Table 5 shows the tracking result using six words which is the output of the headline generation with some named entities. In Table 5, 'Org', 'Per', 'Loc', 'Proper' denotes organization, person, location, and proper name, respectively. 'None' denotes the baseline, i.e. we use only the output of the headline generation, six words. Table 5 shows that the best result was when we use 'Org', 'Person', and 'Proper' with N_t = 16, and the F-score is 0.717. When N_t is larger than 8 positive training stories, the method which uses six title words with named entities consistently outperforms the baseline. When N_t

Table 5. Combination of Named Entities

Named entities	N_t [F-measure]					Avg.	Named entities	N_t [F-measure]					Avg.
	1	2	4	8	16			1	2	4	8	16	
Org Per Loc Proper	.138	.302	.377	.589	.673	.416	Per Loc	.237	.379	.453	.565	.647	.456
Org Per Loc	.138	.307	.391	.586	.668	.418	Per Proper	.437	**.474**	.542	.580	.671	.541
Org Per Loc	.118	.187	.296	.590	**.717**	.382	Loc Proper	.440	.461	.496	.647	.633	.535
Org Loc Proper	.159	.342	.350	.607	.667	.471	Org	.143	.205	.270	.561	.606	.357
Per Loc Proper	.239	.397	.458	.574	.652	.464	Per	**.498**	**.497**	.517	.543	.629	.537
Org Per	.112	.178	.288	.579	.704	.372	Loc	.439	.459	.485	.561	.612	.511
Org Loc	.165	.350	.342	.594	.657	.422	Proper	**.486**	**.473**	.470	.453	.557	.488
Org Proper	.143	.229	.235	.548	.638	.359	None	.465	.466	.456	.453	.516	.471

Table 6. The result with v.s. without hierarchical classification

With hierarchy						Without hierarchy					
N_t	Prec.	Rec.	F	Miss	F/A	N_t	Prec.	Rec.	F	Miss	F/A
1	.695	.422	.525	.578	.0002	1	.669	.396	.498	.604	.0002
2	.707	.475	.568	.526	.0002	2	.671	.394	.497	.606	.0002
4	.835	.414	.554	.586	.0001	4	.747	.396	.517	.605	.0002
8	.823	.523	.639	.477	.0002	8	.709	.440	.543	.560	.0003
16	.819	.573	.674	.428	.0003	16	.818	.511	.629	.489	.0003
Avg.	.776	.481	.592	.519	.0001	Avg.	.723	.427	.537	.573	.0002

Table 7. The result with a hierarchy was worse than that of without a hierarchy

		With hierarchy			Without hierarchy		
Topic	N_t	**F/A**	Prec.	F	**F/A**	Prec.	F
Pinochet trial	16	.0003	.828	.870	.0002	.837	.875
Taipei Mayoral elections	4	.0004	.333	.400	.0002	1.000	.667
Taipei Mayoral elections	8	.0003	.333	.500	.0002	1.000	.667
North Korean food shortages	16	.0002	.700	.298	.0001	.700	.304

is smaller than 4 positive training stories, the result was improved when we add 'Per' and 'Proper' to the baseline. This indicates that these two named entities are especially effective for topic tracking.

5.5 Hierarchical Classification

We recall that we used partitioning clustering algorithm, k-means ($k = 2$) to balance the amount of positive and negative training stories used per estimate. To examine the effect of hierarchical classification using k-means, we compare the result with and without a hierarchy. Table 6 shows the results using the same data, i.e. we use the output of headline generation, six words, and named entities, Person name, and Proper name.

Overall, the result of 'with hierarchy' was better than that of 'without hierarchy' in all N_t values. On the other hand, there are four topics/N_t patterns whose results with hierarchical classification were worse than those of without a hierarchy. Table 7 shows the result. The F/A for all results with a hierarchy were lower than those without a hierarchy. One reason behind this lies iteration of a hierarchical classification, i.e. our algorithm involves iterating through the data that the system is permitted to classify during each iteration. As a result, there are a few negative training data in the bottom cluster, and the test stories were judged as NO but were evaluated as YES. We need to explore a method for determining the depth of the tree in the hierarchical classification, and this is a rich space for further investigation.

5.6 Comparative Experiments

The contribution of two linguistically motivated restrictions on the set of words is best explained by looking at other features. We thus compared our method with two baselines:

Table 8. Comparative experiment

Method	Prec.	Rec.	F	Miss	F/A	Method	Prec.	Rec.	F	Miss	F/A
Stories	.875	.026	.057	.974	.0000	Headlines and NE	.835	.414	.554	.586	.0001
Original headlines	.911	.190	.315	.810	.0000						

Table 9. N_t and F-measure

Method	N_t					Method	N_t				
	1	2	4	8	16		1	2	4	8	16
Stories	-5%	-5%	-	+45%	+61%	Headlines and NE	-2%	-2%	-	+2%	+11%
Original headlines	-26%	-16%	-	+22%	+34%						

(1) all words in the stories as features, and (2) the original headlines in the stories as features[3]. Table 8 shows each result, when N_t = 4. 'Stories' shows the result using all words in the stories and 'Original headlines' shows the result using the original headlines in the stories. 'Headlines and NE' denotes the best result obtained by our method, i.e. the output of headline generation, six words, and named entities, Person and Proper name. Table 8 shows that our method outperformed the other two methods, especially attained a better balance between recall and precision. Table 9 illustrates changes in pooled F1 measure as N_t varies, with N_t = 4 as the baseline. Table 9 shows that our method is the most stable all N_t training instances before N_t = 16, especially our method is effective even for a small number of positive training instances for per-source training: it learns a good topic representation and gains almost nothing in effectiveness beyond N_t = 16.

6 Conclusion

We have reported an approach for topic tracking on newspaper articles based on the two linguistic features, named entities and headlines. The result was 0.776 average precision and 0.481 recall, especially our method is effective even for a small number of positive training instances for per-source training in the tracking task. Future work includes (i) optimal decision of seed points for k-means clustering algorithm, (ii) exploring a method to determine the depth of the tree in the hierarchical classification, and (iii) applying the method to the TDT3 corpus.

References

1. J.Allan and J.Carbonell and G.Doddington and J.Yamron and Y.Yang: Topic Detection and Tracking Pilot Study Final Report. Proc. of the DARPA Workshop. (1997)
2. M.Banko and V.Mittal and M.Witbrock: Headline Generation Based on Statistical Translation. Proc. of ACL-2000. (2000) 318–325

[3] In both cases, we used hierarchical classification to make our results comparable with these two results.

3. J.Carbonell and Y.Yang and J.Lafferty and R.D.Brown and T.Pierce and X.Liu: CMU Report on TDT-2: Segmentation, Detection and Tracking, Proc. of the DARPA Workshop, (1999)
4. D.R.Cutting, D.R.Karger and L.O.Pedersen and J.W.Tukey: Scatter/Gather: a Cluster-based Approach to Browsing Large Document Collections . Proc. of ACM SIGIR-1992. (1992) 318–329
5. H.Jin and R.Schwartz and S.Sista and F.Walls: Topic Tracking for Radio, TV Broadcast, and Newswire. Proc. of the DARPA Broadcast News Transcription and Understanding Workshop. (1999)
6. S.Katz: Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. IEEE Transactions on Acoustics, Speech and Signal Processing. **24** (1987)
7. T.Kudo and Y.Matsumoto: Fast Methods for Kernel-Based Text Analysis. Proc. of the ACL-2003. (2003) 24–31
8. D.D.Lewis: An Evaluation of Phrasal and Clustered Representations on a Text Categorization Task. Proc. of the ACM SIGIR-1994. (1994) 37–50
9. D.D.Lewis and R.E.Schapire and J.P.Callan and R.Papka: Training Algorithms for Linear Text Classifiers. Proc. of the ACM SIGIR-1996. (1996) 298–306
10. S.A.Lowe: The Beta-binomial Mixture Model and its Application to TDT Tracking and Detection. Proc. of the DARPA Workshop. (1999)
11. Y.Matsumoto and A.Kitauchi and T.Yamashita and Y.Haruno and O.Imaichi and T.Imamura: Japanese Morphological Analysis System Chasen Mannual. NAIST Technical Report NAIST-IS-TR97007. (1997)
12. D.W.Oard: Topic Tracking with the PRISE Information Retrieval System. Proc. of the DARPA Workshop. (1999)
13. R.Papka and J.Allan: UMASS Approaches to Detection and Tracking at TDT2. Proc. of the DARPA Workshop. (1999)
14. R.E.Schapire: BoosTexter: A Boosting-based System for Text Categorization. Journal of Machine Learning. (1999)
15. R.Schwartz and T.Imai and L.Nguyen and J.Makhoul: A Maximum Likelihood Model for Topic Classification of Broadcast News. Proc. of Eurospeech. (1996) 270–278
16. T.Strzalkowski and G.C.Stein and G.B.Wise: GE.Tracker: A Robust, Lightweight Topic Tracking System. Proc. of the DARPA Workshop. (1999)
17. Yamron and Carp: Topic Tracking in a News Stream. Proc. of the DARPA Broadcast News Transcription and Understanding Workshop. (1999)
18. J.P.Yamron and I.Carp and L.Gillick and S.Lowe and P.V.Mulbregt: Topic Tracking in a News Stream. Proc. of the DARPA Workshop. (1999)
19. Y. Yang: Expert Network: Effective and Efficient Learning from Human Decisions in Text Categorization and Retrieval. Proc. of the ACM SIGIR-1994. (1994) 13–22
20. Y.Yang and T.Pierce and J.Carbonell: A Study on Retrospective and On-Line Event Detection. Proc. of the ACM SIGIR-1998. (1998) 28–36
21. Y.Yang and T.Ault and T.Pierce and C.W.Lattimer: Improving Text Categorization Methods for Event Tracking. Proc. of the ACM SIGIR-2000. (2000) 65–72
22. V.Vapnik: The Nature of Statistical Learning Theory. Springer. (1995)