

Identification of Relevant and Novel Sentences Using Reference Corpus

Hsin-Hsi Chen, Ming-Feng Tsai, and Ming-Hung Hsu

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan

hh_chen@csie.ntu.edu.tw, {mftsai,mfhsu}@nlg.csie.ntu.edu.tw

Abstract. The major challenging issue to determine the relevance and the novelty of sentences is the amount of information used in similarity computation among sentences. An information retrieval (IR) with reference corpus approach is proposed. A sentence is considered as a query to a reference corpus, and similarity is measured in terms of the weighting vectors of document lists ranked by IR systems. Two sentences are regarded as similar if they are related to the similar document lists returned by IR systems. A dynamic threshold setting method is presented. Besides IR with reference corpus, we also use IR systems to retrieve sentences from given sentences. The corpus-based approach with dynamic thresholds outperforms direct retrieval approach. The average F-measure of relevance and novelty detection using Okapi system was 0.212 and 0.207, 57.14% and 58.64% of human performance, respectively.

1 Introduction

How to obtain relevant information from a considerable amount of data collection has become increasingly important. Current information retrieval (IR) systems only return documents satisfying users' information needs, but they do not precisely locate the relevant sentences. Therefore, users have to go through the whole documents to find the relevant information. Moreover, traditional IR systems do not identify which sentences contain new information. Filtering redundant information out and locating novel information is indispensable for some emerging applications like summarization and question-answering [4].

There are some sorts of relevance and novelty detection on document level in Topic Detection and Tracking (TDT) [2]. Link detection relates news stories on the same topic [3] and first story detection tries to identify the first article with a new event. Novelty track in TREC 2002 [5] is the first attempt to locate relevant and new sentences instead of the whole documents containing duplicate and extraneous information. Similarity computation is a fundamental operation for relevance and novelty judgment on both sentence and document levels. However, the amount of information of a sentence that can be used in similarity computation is much fewer than that of a document. That forms the major challenging issue.

In the past, word matching and thesaurus expansion were adopted to recognize if two sentences touched on the same theme in multi-document summarization [4]. Such an approach has been employed to detect the relevance between a topic description and a sentence [8]. The similarity computation can also be performed by an information retrieval system. Zhang *et al.* [11] employed an Okapi system to retrieve relevant sentences with a topic description, and a fixed heuristic threshold was adopted. Larkey *et al.* [6] studied how many sentences were relevant in different size of documents. Allan *et al.* [1] focused on the novelty detection algorithms and showed how the performance of relevant detection affects that of novelty detection. Instead of using an IR system to select relevant sentences directly, an external corpus can be consulted [8]. Both a topic description and a sentence are considered as queries to the reference corpus through an IR system. Two sentences are relevant if similar sets of relevant documents are retrieved.

This paper shows how to extract relevant sentences from several known relevant documents, and how to determine new sentences from the extracted relevant sentences. The decision about what information is new depends on the order of the occurrence of the information. In other words, “a novel sentence” means that all of the relevant information in this sentence is never covered by the relevant sentences delivered previously. Section 2 presents a concept matching approach, a test set and evaluation metrics. Section 3 uses reference corpus and IR systems. The effects of different issues, including with/without reference corpus, static/dynamic settings of thresholds, and various IR systems, are compared. Section 4 extracts novel sentences from relevant sentences. Section 5 concludes the remarks.

2 A Concept Matching Approach

The problem of novelty task is defined as follows:

Given a topic description and a sequence of sentences, a novelty detection system should identify which sentences are relevant to the topic description, and which sentences are novel relative to the other sentences under a specific topic.

The original sequence of sentences is called *given sentences*, and the resulting two lists are called *relevant sentences* and *novel sentences*. The given sentences came from some relevant documents. A novelty task is composed of two major components, i.e., a relevance detector and a novelty detector. The relevance detector receives a sequence of sentences from known relevant documents, and determines which sentences are on topic. Those relevant sentences will be delivered to the novelty detector and the redundant sentences will be filtered out. The remaining sentences are *novel* and *relevant*. Relevant detector is very important because its performance will affect the performance of a novelty detector.

A relevance detector attempts to identify those sentences containing the relevant information from the known relevant documents. The key issue behind relevance detection is how to measure the similarity of a topic description and the given sentences. Because the basic unit of similarity measure is a sentence instead of the whole document, we have to deal with the problem of the lack of information within a sentence during distinguishing relevant and irrelevant sentences. A concept matching approach is proposed for relevance detection. A predicate and its surrounding

arguments form a kernel skeleton in a sentence, so that verbs and nouns are important features for similarity computation. In this way, all the given sentences are tagged by using a part-of-speech tagger. After tagging, nouns and verbs are extracted. Then WordNet is applied to find the synonymous terms for concept matching. Noun and verb taxonomies with hyponymy/hypernymy relations are consulted. The similarity of two sentences is in terms of noun-similarity and verb-similarity as follows.

$$\textit{noun_sim}(s_1, s_2) = \frac{m}{\sqrt{ab}} \quad (1)$$

$$\textit{verb_sim}(s_1, s_2) = \frac{n}{\sqrt{cd}} \quad (2)$$

$$\textit{sim}(s_1, s_2) = \textit{noun_sim}(s_1, s_2) + \textit{verb_sim}(s_1, s_2) \quad (3)$$

where s_1 and s_2 denote two sentences, respectively; m and n denote the number of concept matching for nouns and verbs, respectively; a and b are the total number of nouns in s_1 and s_2 , respectively; and c and d are the total number of verbs in s_1 and s_2 , respectively.

Total 49 topics and 49 sets of given sentences in TREC 2002 Novelty track [5] are applied to evaluate the performance of relevance detector. Precision, recall and F-measure shown as follows are employed.

$$\text{Recall (R)} = \# \text{RELEVANT matched} / \# \text{RELEVANT} \quad (4)$$

$$\text{Precision (P)} = \# \text{RELEVANT matched} / \# \text{sentences submitted} \quad (5)$$

$$\text{F-measure (F)} = 2 \text{ Recall} * \text{Precision} / (\text{Recall} + \text{Precision}) \quad (6)$$

$$\text{Average F-measure} = \sum \text{F-measure} / \# \text{TOPIC} \quad (7)$$

When the threshold is set to 0.4, the average F-measure of the concept matching approach is 0.125. Besides, a baseline model that randomly selects sentences from the given sentences is also adopted for comparison. The average F-measure of the baseline model was 0.040, and the average F-measure of human judge was 0.371 [5]. The experiments show that the performance of the concept matcher is better than that of the baseline model, but is still far less than that of human being. The outside resource, i.e., WordNet, seems not to be enough to measure the similarity in these experiments. In the following we will consult another resource – say, a reference corpus.

3 Relevance Detection Using IR Approach

3.1 IR with Reference Corpus

To use a similarity function to measure if a sentence is on topic is similar to the function of an IR system. We use a reference corpus, and regard a topic and a

sentence as queries to the reference corpus. An IR system retrieves documents from the reference corpus for these two queries. Each retrieved document is assigned a relevant weight by the IR system. In this way, a topic and a sentence can be in terms of two weighting vectors. Cosine function measures their similarity, and the sentence with similarity score larger than a threshold is selected. The issues behind the IR with reference corpus approach include the reference corpus, the performance of an IR system, the number of documents consulted, the similarity threshold, and the number of relevant sentences extracted.

The reference corpus should be large enough to cover different themes for references. In the experiments, the document sets used in TREC-6 text collection [10] were considered as a reference corpus. It consists of 556,077 documents. Two IR systems, i.e., Smart [9] and Okapi [7], were adopted to measure the effects of the performance of an IR system. In the initial experiments, Smart system with the basic setting (i.e., tf^*idf scheme without relevance feedback) was employed. It had average precision 0.1459 on the TREC topics 301-350. Okapi was in the option of bm25, and had average precision 0.2181 on the same document set.

3.2 How Many Documents Reported

How many documents should be reported by an IR system is an important issue for similarity measurement between a topic and a given sentence. Both relevant and irrelevant documents may be reported in the result list. That depends on the IR performance. The effects of the sizes of resulting document lists were investigated. Table 1 summarizes the results of using Smart and Okapi when the threshold is set to 0.1. The first column shows that different number of documents, i.e., 50, 100, 150, 200, 250, 300, 350, 400, 450, and 500 documents, are returned by Smart and Okapi, respectively.

It shows that smaller result list (e.g., 50 documents) is better than larger result list when Smart system is adopted. This is because the relevant document set is comparatively much smaller than the irrelevant document set for a query, and the irrelevant documents in the two result lists tend to be different. Smaller result list decreases the possibility to incorporate different irrelevant documents, but also decreases the possibility to find out the same relevant documents. Enlarging the result list means the number of the same relevant documents may be increased, but different irrelevant documents are also added. In contrast, the performance of Okapi-based system is increased from reporting 50 documents till 250 documents. After that, the performance starts to decrease. This is because Okapi outperforms Smart. Larger result list (within 250 documents) covers more relevant documents. In the experiments, the best average F-measures, 0.170 and 0.176, were achieved when the sizes of result list were 50 and 250 documents by using Smart and Okapi, respectively.

3.3 Threshold Setting

We also made experiments with different thresholds (between 0 and 0.3), and smaller number of returned documents. Figures 1 and 2 show the experimental results. The best F-measures of using Smart and Okapi are 0.175 and 0.182, respectively. Because

we did not employ the distribution of similarity scores, the thresholds were “guessed”, and the thresholds were fixed in different topics. That is unfair in some cases.

Table 1. Effects of Size of Returned Documents

Number of consulted documents	Smart-based			Okapi-based		
	Avg. P	Avg. R	Avg. F	Avg. P	Avg. R	Avg. F
50	0.13	0.4	0.170	0.15	0.49	0.169
100	0.13	0.43	0.154	0.15	0.48	0.174
150	0.12	0.46	0.144	0.13	0.49	0.174
200	0.11	0.48	0.137	0.14	0.48	0.176
250	0.11	0.50	0.137	0.13	0.49	0.176
300	0.10	0.51	0.135	0.13	0.49	0.173
350	0.10	0.52	0.130	0.12	0.49	0.170
400	0.10	0.54	0.127	0.12	0.50	0.171
450	0.09	0.54	0.124	0.12	0.50	0.170
500	0.09	0.55	0.120	0.11	0.51	0.169

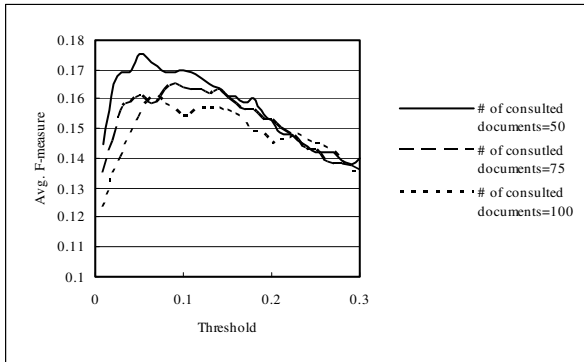


Fig. 1. Effects of Fixed Thresholds Using Smart

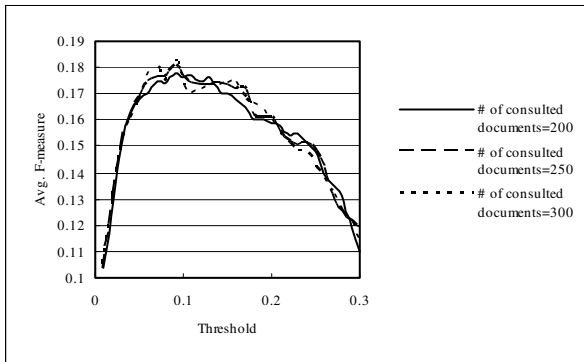


Fig. 2. Effects of Fixed Thresholds Using Okapi

A threshold setting model is proposed as follows to deal with this problem. Assume normal distribution with mean μ and standard deviation σ is adopted to specify the similarity distribution of the given sentences with a topic. We compute the cosine of a topic vector T and a given sentence vector S_i ($1 \leq i \leq m$) as below, where m denotes total number of the given sentences. The percentage n denotes that top n percentages of the given sentences will be reported. Similarity thresholds ($TH_{\text{relevance}}$) are determined by these percentages.

$$\mu = \frac{\sum_{i=1}^m \cos(T, S_i)}{m} \tag{8}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^m (\cos(T, S_i) - \mu)^2}{m}} \tag{9}$$

$$TH_{\text{relevance}} = \mu + z\sigma \tag{10}$$

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-y^2/2} dy = 1 - n \tag{11}$$

Figure 3 shows that total n (%) of given sentences fall in the gray area are considered as relevant. z is equal to 1.282, 0.84, 0.524, 0.253 and 0 when n is 10%, 20%, 30%, 40%, and 50%, respectively.

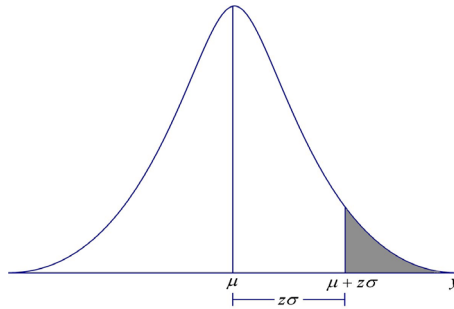


Fig. 3. Normal Distribution with Mean μ and Standard Deviation σ

Various settings of n (percentage) were experimented, and the results using Smart and Okapi are listed in Figures 4 and 5, respectively. Smart-based relevance detector achieves better performance when larger percentage of sentences is selected. On the contrary, the larger the percentage is, the worse the performance is, when some critical point is reached using Okapi. The major reason is: Okapi gets better retrieval performance than Smart, so that it pulls the relevant sentences in the front of normal distribution. The best F-measures are 0.190 and 0.206. Using n (%) to determine the thresholds is a dynamic approach, which is better than static threshold approach.

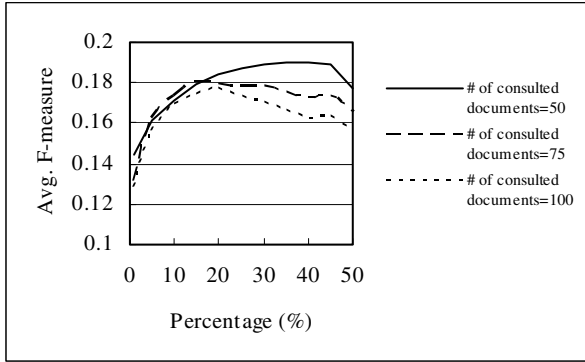


Fig. 4. Effects of Fixed Percentages Using Smart

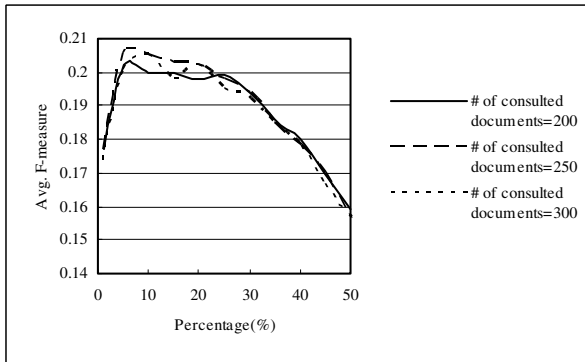


Fig. 5. Effects of Fixed Percentages Using Okapi

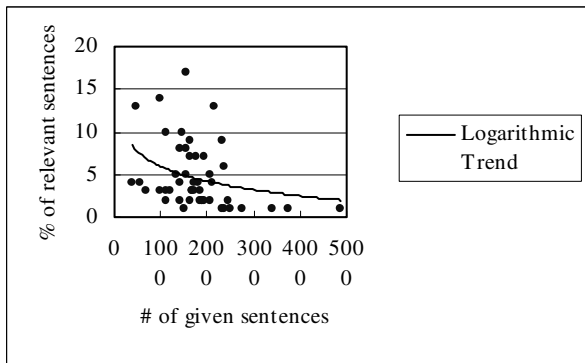


Fig. 6. An illustration of Logarithmic Trend

Table 2. Effects of Dynamic Percentage

Number of consulted documents	Smart-based			Okapi-based		
	50	75	100	200	250	300
Ln-1	0.164	0.167	0.163	0.203	0.208	0.212
Ln-2	0.177	0.180	0.176	0.204	0.207	0.205
Ln-3	0.185	0.178	0.176	0.204	0.205	0.205
Ln-4	0.189	0.179	0.174	0.200	0.201	0.198
Ln-5	0.191	0.181	0.172	0.194	0.191	0.191

Even though the above dynamic approach has better performance, it is still “fixed percentage” for all topics. We consider further how to select “good” percentages for individual topics. Larkey *et al.* [6] showed that only 5% of the sentences contained relevant materials for average topic. From their collection statistics [6], we used logarithmic regression as follows to simulate the relationship between total number of the given sentences and number of the relevant sentences. Figure 6 illustrates the trend.

$$n = -2.4938Ln(x) + 23.157 \quad (12)$$

where x is total number of given sentences, and n is the suggested percentage.

After computing n using Formula (12), we derived z using Formula (11) and finally $TH_{\text{relevance}}$ using Formula (10).

Table 2 summarizes the F-measures of using dynamic percentage by Smart and Okapi, respectively. Dynamic percentage is better than fixed percentage. The best performance of dynamic percentage using Smart is 0.191 when the size of consulted documents is set to 50 and logarithmic metric is multiplied by 5, which gets about 1% improvement to the fixed percentage. The best F-measure of dynamic percentage using Okapi is 0.212, when the size of consulted documents is set to 300 and the original logarithmic metric is employed. It gets about 3% increases to the fixed percentage experiments.

The best performance among these experiments is 0.212, i.e., 57.14% of human performance (0.371). Figure 7 lists the performance of each topic when the number of consulted documents is 300 using Okapi system. Two dotted lines, i.e., one is human performance (0.371) and the other one is baseline performance (0.040), are provided for reference. Performance of our system in 6 topics (358, 364, 365, 368, 397, and 449) is competitive to that of human judge. In contrast, performance in 3 topics (377, 420, and 432) is lower than that of random selection. The average F-measure of the remaining 40 topics are below human performance, but better than that of baseline model.

3.4 IR without Reference Corpus

Even using an IR system, we have two alternatives to select the relevant sentences, i.e., with and without a reference corpus. In the corpus-free approach, the given sentences form a database itself, and a topic is submitted to an IR system to retrieve the similar sentences directly. The resulting sentences ranked and reported by the IR system are called *candidate sentences*. A dynamic percentage of candidates with higher scores will be reported as relevant. The percentage and the relevant thresholds

are determined in the similar way as the corpus-based approach. The best F-measures of IR approach without reference corpus are 0.113 and 0.165, respectively. Smart-based and Okapi-based systems without reference corpus decrease 10% and 9% performance, respectively.

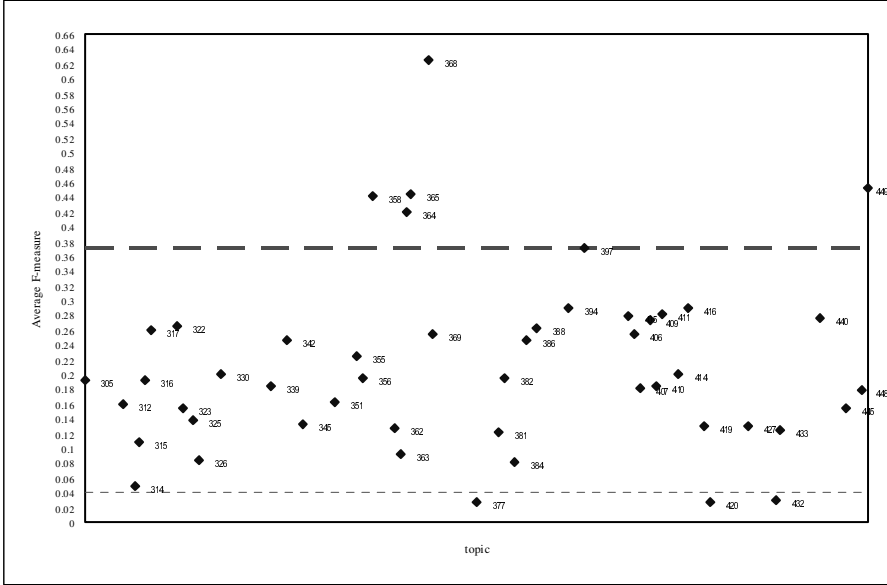


Fig. 7. Average F-measure of Relevance Detection for Each Topic

4 Novelty Detection Using Reference Corpus

Novelty detector identifies new information among the sentences extracted by the relevance detector. In other words, novelty detector will filter out the redundant sentences among the relevant sentences. The key issue on the detection of new information is how to differentiate the meaning of sentences accurately. Sentences may contain too less information to distinguish their differences, so that certain information expansion method is required.

We extend the idea in Section 3, i.e., employing a reference corpus to select the relevant information, to find the relationship among relevant sentences. Similarly, we use the same reference corpus and regard each relevant sentence as a query to this corpus. Documents in the corpus are ranked by an IR system, and the documents with higher scores are reported for each relevant sentence. Each retrieved document is assigned a weight, in such a way that a sentence is still represented as a vector. Cosine function measures the similarity of any two sentences. Two sentences are regarded as similar if they are related to the similar document lists.

On the one hand, the cosine value of two sentences indicates that how similar they are. On the other hand, the higher value indicates one sentence is somewhat redundant relative to the other sentence. A threshold of novelty decision, $TH_{novelty}$, determines the

degree of redundancy. If the similarity score of two sentences is larger than $TH_{novelty}$, then one of them has to be filtered out depending to their temporal order. In this way, the redundant sentences are filtered out and only the novel sentences are kept. The remaining sentences are the result of the novelty detector.

Two algorithms are proposed as follows to deal with the novelty detection problem. Assume there are r relevant sentences, s_1, s_2, \dots, s_r for topic t .

- (1) Static threshold approach
 Let T be a set containing novel sentences found up to know. Initially, $T=\{s_1\}$. For each relevant sentence s_i ($2 \leq i \leq r$), if there exists a sentence in T whose similarity with s_i is larger than a predefined threshold, then s_i is not a novel sentence and is removed; otherwise, s_i is kept in T .
- (2) Dynamic threshold approach
 Assume s_1 is a novel sentence. Compute the similarities between s_1 and s_i ($2 \leq i \leq r$). Determine the novelty threshold, $TH_{novelty}$, in the same way as $TH_{relevance}$. Filter out the top $n\%$ of sentences with the higher similarities with s_1 . Let R be the remaining sentences. If the number of sentences in R is less than 30¹, then regard these sentences as novel sentences and stop. Otherwise, select the first sentence in R , regard it as a novel sentence and repeat the same filtering task.

We chose the results from the best relevance detectors mentioned in last section, i.e., Smart-based and Okapi-based systems with average F-measure 0.191 and 0.212, to test these two approaches. The performance of static threshold approach is shown in Figure 8. Okapi-based novelty detector still outperforms Smart-based novelty detector. Besides, it also indicates that more sentences are filtered out when $TH_{novelty}$ is lower. The performance increased as $TH_{novelty}$ increased. Using higher novelty threshold, two sentences should have much higher similarity to pass the threshold if they are similar. The lower the probability two sentences pass the threshold, the higher the probability both sentences are novel. Figure 9 illustrates the results of dynamic threshold approach. When more percentages of sentences are filtered out, the performance of both Smart-based and Okapi-based novelty detectors are decreased.

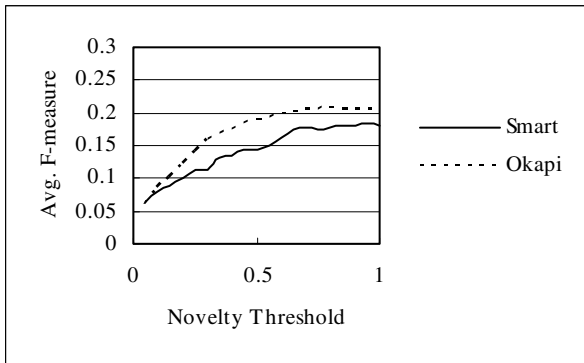


Fig. 8. Results of Static Novelty Threshold

¹ A sample size of at least 30 has been found to be adequate for normal distribution.

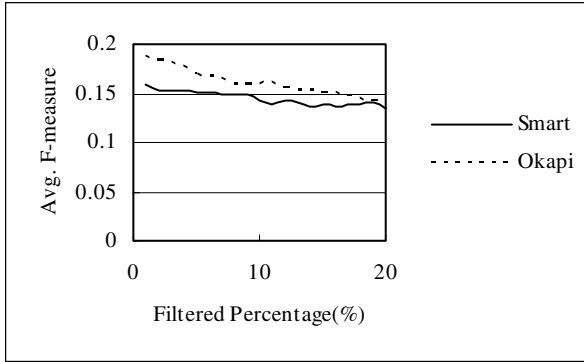


Fig. 9. Results of Dynamic Novelty Threshold

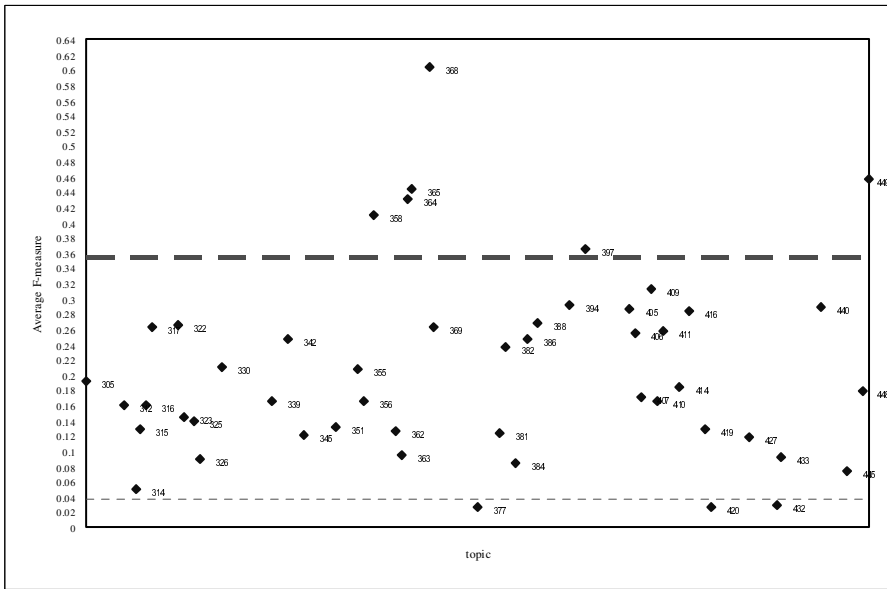


Fig. 10. Further Examination of the Best Novelty Detection

The best performance among these experiments is 0.207, when the novelty threshold is set to 0.8 statically, and total 300 documents reported by Okapi are consulted. Figure 10 examines the performance of each topic furthermore. Two dotted lines, one for human performance (0.353) and the other one for baseline performance (0.036), are provided for reference. Performance of our approach in 6 topics (i.e., 358, 364, 365, 368, 397, and 449) is competitive to that of human judge. In contrast, performance in 3 topics (377, 420, and 432) is lower than that of the baseline model. The average F-measure of the remaining 40 topics are below human performance, but better than that of baseline model.

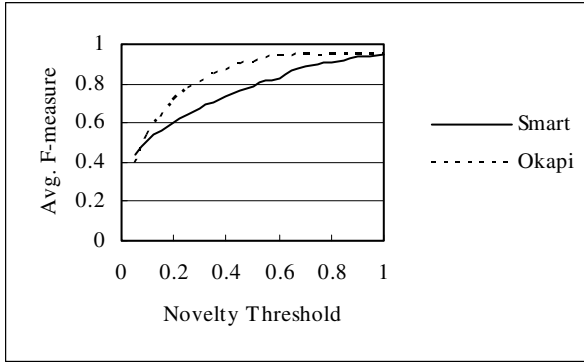


Fig. 11. Ideal Performance of Static Novelty Threshold Approach

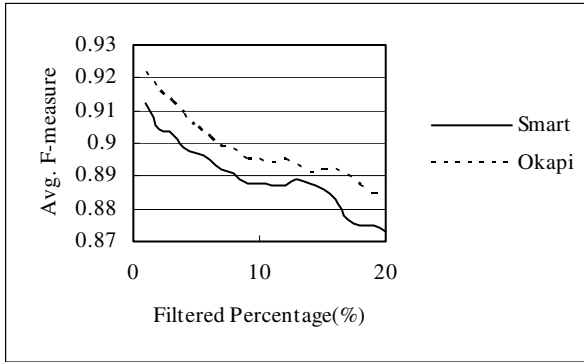


Figure 12. Ideal Performance of Dynamic Novelty Threshold Approach

In general, the average F-measure of the novelty detector is better than that of the baseline model (i.e., 0.036). However, the performance is still not comparable to the human assessors (i.e., 0.353). It only achieves 58.64% of human performance. The major reason is that the result of relevance detector contains irrelevant sentences, so that novelty detector false identifies that those irrelevant sentences contain new information. As mentioned before, the relevance part is more difficult to be overcome in this task.

We also conducted another set of experiments to evaluate the ideal performance of locating novel sentences. These experiments take correct relevant information as input to novelty detector, so that there are no propagation errors from relevant detectors. Figure 11 shows the results of static novelty threshold approach. The performance was increased when $TH_{novelty}$ was increased. This is because more sentences are filtered out when $TH_{novelty}$ is lower. The ideal performance of Okapi-based novelty detector is 0.945 and the performance is above 0.912 when threshold is larger than 0.5. Figure 12 shows the results of dynamic novelty threshold approach. The ideal performance of the Okapi-based system is 0.922. The average F-measure dropped quickly, when more percentage of sentences are filtered out.

5 Conclusions and Future Work

This paper proposed concept matching and IR approaches to identify sentences that are novel and redundant as well as relevant and irrelevant. Although the method of matching noun and verb keywords and the related expansion achieved average F-measure 0.125, which is better than the baseline performance (i.e., 0.040), words in sentences are still not enough for the relevance detection. We presented an information expansion using a reference corpus to deal with this problem. We postulated that if two sentences have the similar meaning, then their behavior on information retrieval to the reference corpus is similar. Logarithmic regression approximates how many percentages of sentences are relevant for each topic. This value determines an offset from mean in normal distribution and thus the similarity threshold. That forms a rigid procedure instead of heuristics to determine the needed parameters. The experiment results show that Okapi-based relevant detector with dynamic threshold setting, which depend on topics and given sentences, are better than the other approaches. The best average F-measure of relevance detector is 0.212, which is 57.14% of human performance (0.373). When the idea was extended to novelty detector, the average F-measure is 0.207, which is 58.64% of human performance (0.353). The effects of the IR systems, e.g., query construction and relevance feedback, will be investigated. Besides, the deep syntactic and semantic analysis of sentences to distinguish relevant and novel sentences will be explored.

References

1. Allan, J., Wade, C., and Bolivar, A.: Retrieval and Novelty Detection at the Sentence Level. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Toronto, Canada, July 28–August 01, 2003. ACM (2003) 314-321
2. Allan, J., Carbonnell, J., and Yamron, J.: Topic Detection and Tracking: Event-Based Information Organization. Kluwer (2002)
3. Chen, H.H., and Ku, L.W.: An NLP & IR Approach to Topic Detection. In Topic Detection and Tracking: Event-Based Information Organization, James Allan, Jaime Carbonnell, Jonathan Yamron (Editors). Kluwer (2002) 243-264
4. Chen, H.H., Kuo, J.J., Huang, S.J., Lin, C.J., and Wung, H.-C.: A Summarization System for Chinese News from Multiple Sources. In Journal of American Society for Information Science and Technology. (2003)
5. Harman, D.: Overview of the TREC 2002 Novelty Trec. In Proceedings of the Eleventh Text REtrieval Conference. NIST Special Publication: SP 500-251, Gaithersburg, Maryland, November 19-22, 2002. TREC (2002)
6. Larkey, L. S. et al.: UMass at TREC2002: Cross Language and Novelty Tracks. In Proceedings of the Eleventh Text REtrieval Conference. Gaithersburg, NIST Special Publication: SP 500-251, Gaithersburg, Maryland, November 19-22, 2002. TREC (2002)
7. Robertson, S.E., Walker, S., and Beaulieu, M.: Okapi at TREC-7: Automatic ad hoc, Filtering, VLC and Interactive. In Proceedings of the Seventh Text REtrieval Conference, Gaithersburg, NIST Special Publication: SP 500-242, Gaithersburg, Maryland, November 9-11, 1998. TREC 7 253-264.
8. Tsai, M.F., and Chen, H.H.: Some Similarity Computation Methods in Novelty Detection. In Proceedings of the Eleventh Text REtrieval Conference. Gaithersburg, NIST Special Publication: SP 500-251, Gaithersburg, Maryland, November 19-22, 2002. TREC (2002)

9. Salton, G., and Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. In *Information Processing and Management*. Vol. 5, No. 24, pp. 513-523.
10. Voorhees, E.M., Harman, D.K. (Eds.) *Proceedings of the Sixth Text Retrieval Conference*. NIST Special Publication: SP 500-240, Gaithersburg, Maryland, November 19-21, (1997)
11. Zhang, M. et al.: THU at TREC2002: Novelty, Web and Filtering. In *Proceedings of the Eleventh Text REtrieval Conference*, NIST Special Publication: SP 500-251, Gaithersburg, Maryland, November 19-22, 2002. TREC (2002).