Some Similarity Computation Methods in Novelty Detection

Ming-Feng Tsai and Hsin-Hsi Chen
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
E-mail: mftsai@nlg.csie.ntu.edu.tw; hh_chen@csie.ntu.edu.tw

Abstract

In the novelty task, the amount of information of a sentence that can be used in similarity computation is the major challenging issue. Some sort of information expansion methods was introduced to tackle this problem. Our approach to relevance identification was to expand the information of a sentence with the context of this sentence using a sliding window method. The similarity was measured by the number of words of a topic description that match the sentences within a window. Besides, WordNet was employed to relax word match operation to inexact match. In the novelty detection part, we first applied a coherent text segmentation algorithm to partition the sentences extracted from the relevance identification part into several coherent segments denoting sub-topics. Then we compute the similarity of each sentence with each segment. A sentence was in terms of a sentence-segment similarity vector. Two sentences are regarded as similar if they are related to the same sub-topics. In this way, the redundant sentences were filtered out.

1 Introduction

Information explosion is one of challenging problems in the new information era. How to obtain relevant information from a large amount of data collection has become important. Current information retrieval (IR) systems only return documents satisfying users' information needs, but they do not locate the relevant sentences. Users have to go through the whole documents to find the relevant information. Moreover, traditional IR systems do not tell out which sentences contribute new information. To filter the redundant information and locate the novel information becomes more and more important for many emerging applications like summarization and question-answering. Novelty track, the new task of TREC, aims to locate relevant and new sentences (within context) rather than the whole documents containing duplicate and extraneous information.

Only few attempts have been made so far on novelty detection problem, because

there is little agreement to the definition of *novelty* and the lack of evaluation data. In Topic Detection and Tracking (TDT) project (Allan, Carbonnell and Yamron, 2002), link detection task relates news stories on the same topic (Chen and Ku, 2002) and first story detection tries to find out the first article of a new event. It is some sort of novelty detection on document level. In novelty track of TREC, the basic unit that we confront with is a sentence. The amount of information of a sentence that can be used in similarity computation is the major challenging issue. In multi-document summarization (Chen and Huang, 1999; Chen and Lin, 2000), we faced the similar problem. We had to compute the similarity of meaningful units, which contain less information than passages and documents. Word matching and thesaurus expansion were adopted to tell out if two meaningful units touch on the same theme.

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 the architecture of our system. It uses sliding window to exact relevant sentences and uses relevant segments to exact novel sentences. Section 3 shows the performance of this system and makes some discussions. Section 4 concludes the remarks.

2 Architecture

Figure 1 shows the architecture of our novelty system. It 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 sentence is on topic. Those relevant sentences will be delivered to the novelty detector and the redundant sentences will be filtered out. The remaining sentences are *new (novel)* and *relevant*.

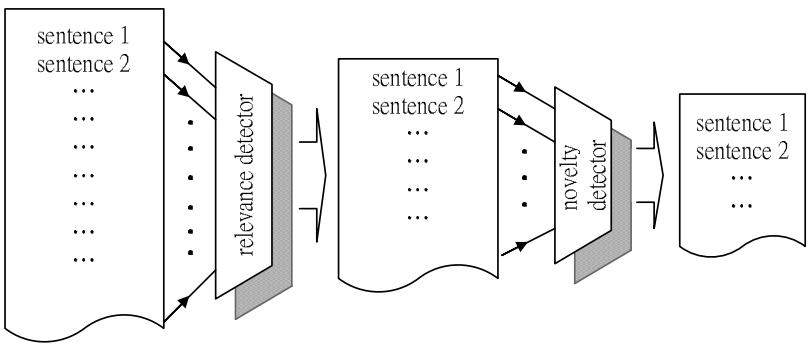

Figure 1: Architecture of Our Novelty System

The basic idea in our study is to measure the similarity of the sentences in the relevant documents. The following subsections will deal with the similarity model, relevance detector and novelty detector in sequence.

2.1 Similarity Model

Because the basic unit of similarity measure is a sentence instead of the whole document, we have to deal with the problem of less information in a sentence during distinguishing relevant and irrelevant sentences. Predicate-argument structure forms the kernel of a sentence, thus verbs and nouns are important features for similarity measures. All the sentences were parsed using Eric Brill's part-of-speech tagger. After tagging, nouns and verbs were extracted. Then we utilized WordNet to find the synonymous terms for inexact matching. Noun and verb taxonomies with hyponymy/hypernymy relations were consulted. The shortest path of each sense of word w_1 to each sense of word w_2, denoted $dist(w_1, w_2)$, was computed. Figure 2 demonstrates an example. Each note represents a synset in WordNet. In this example, the distance between *universe* and *sky* is 4.

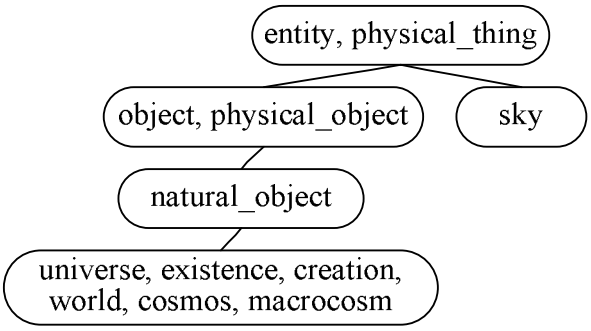

Figure 2: An Example of Distance Measurement

A threshold is employed to decide whether two words are similar or not. If their distance is less than the threshold, then 0.5 is added in the matching score. In summary, our similarity model is shown as follows:

- Nouns in one sentence are matched to nouns in another sentence, so are verbs. The value of 1 is added to the matching score for each exact matching.
- In inexact matching, we set word distance threshold to 4. In other words, if the $dist(w_1, w_2)$ is less than the threshold, the value of 0.5 is added to the matching score.
- Each term is matched only once.

The similarity of two sentences is in terms of noun-similarity and verb-similarity:

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

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

$$sim(s_1, s_2) = noun\_sim(s_1, s_2) + verb\_sim(s_1, s_2) \quad (3)$$

where s_1 and s_2 denote two sentences, respectively;
m and *n* denote the number of matching 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.

2.2 Relevance Detector

The relevance detector aims to identify those sentences containing the relevant information from the known relevant documents. The approach to determine if a sentence is on topic is to use the above similarity function to measure the similarity of a sentence and the given topic. Its function is similar to traditional information retrieval system. The main difference is that the relevance detector extracts relevant information from sentences. The major problem of calculating similarity of a sentence and a topic is that sentence contains less information for comparison.

In Section 2.1, we try to augment a sentence with the synonymous terms retrieved from WordNet. We call it *within-sentence expansion*. Here one more expansion, called *between-sentence expansion* later, is considered. The context of a sentence is also a cue to determine relevance. In one extreme case, all the sentences surrounding the specific sentence form a context. But the context may be so large that noise may be introduced. In another extreme case, only the specific sentence is considered without adding any other sentence. In other words, it employs the information coming from itself. Trading the two extreme cases off, a sliding window controls how large a context is. Figure 2 shows a sliding window of size 2.

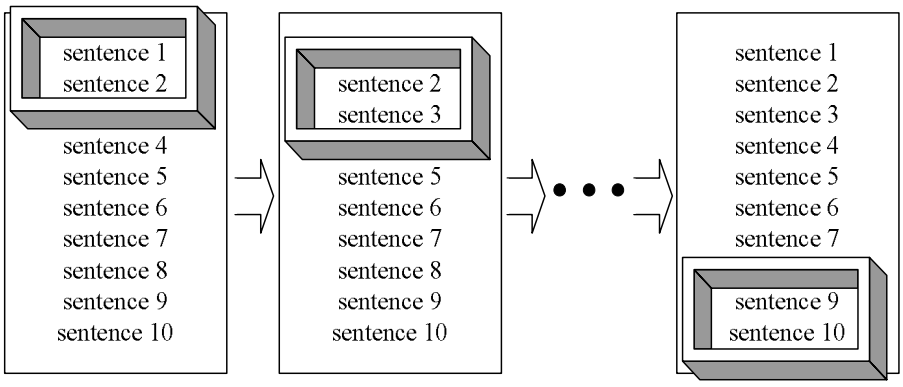

Figure 3: A Sliding Window (window size = 2)

A predefined relevance threshold, $TH_{relevance}$, is employed to determine whether sentences within a window are on topic or not. The sentences within a window are on topic if the similarity is larger than the predefined threshold. That is, if the sentences within a window are on topic, then those sentences within a window are identified as relevant sentences and sent to the next component, i.e., novelty detector. The window size and the relevance threshold, $TH_{relevance}$, are trained from the pre-released sample data.

2.3 Novelty Detector

The next step is to detect new information among the sentences extracted by the relevance detector. It would be better to say that we plan to filter 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 postulate that the relevant sentences may touch on several particular sub-topics. Under this postulation, a text segmentation algorithm developed by Utiyama, *et al.* (2001) was employed to partition the relevant sentences into several segments. Each segment corresponds to a sub-topic. This algorithm finds the maximum-probability coherent segmentation of a given text. The similarity between each sentence and each segment is calculated, and then each sentence is represented as a sentence-segment similarity vector. Two sentences are regarded as similar if they are related to the same sub-topics. In this way, the redundant sentences are filtered out and only the novel sentences are kept. Figure 4 sketches this idea.

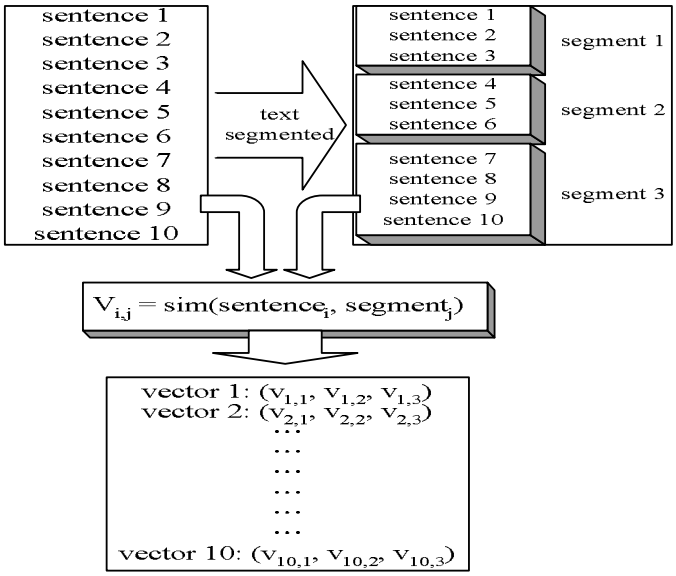

Figure 4: An Illustration of Novelty Detector

Assume a sentence s_i is represented as a vector ($v_{i,1}$, $v_{i,2}$, …, $v_{i,n}$), where n is the number of segments. Cosine function shown in formula (4) measures the similarities of two vectors.

$$\cos(s_i, s_j) = \frac{\sum_{k=1}^{n} v_{i,k} \times v_{j,k}}{\|s_i\| \cdot \|s_j\|} \quad (4)$$

This value indicates that how similar sentences s_i and s_j are. From the other point of view, the higher value indicates sentence s_i is somewhat redundant relative to sentence s_j. A threshold of novelty decision, $TH_{novelty}$, determines the degree of redundancy. If the similarity score of sentences s_i and s_j is larger than $TH_{novelty}$, then one of the two sentences has to be filtered out depending to their temporal order. The remaining sentences are the result of the novelty detector. The novelty threshold, $TH_{novelty}$, was trained from the pre-released sample data set.

In the above approach, we employed the relevant data itself to select the new information. Alternatively, we may use a reference corpus and regard each relevant sentence as a query to this corpus. An IR system may retrieve top n documents from the reference corpus for each relevant sentence. Each retrieved document is assigned a weight 1/r, where r is a rank of a retrieved document. In this way, a sentence is still represented as a vector. Cosine function measures the similarity of any two sentences, and the novel sentence is selected.

3 Experimental Results

Traditional precision and recall is counted to measure the performance of our novelty system and the product of precision and recall is also calculated for TREC measure. In the relevance part, we used description 1, description 2 as well as narrative part of the topic to retrieve relevant sentences. WordNet 1.7.1 was employed.

Tables 1 and 2 show our official runs at TREC 2002 Novelty Track. Our result of novelty part is not so good in this experiment, because the threshold, $TH_{novelty}$, is set to 0.97. This setting is according to the observation in pre-released sample set. The novelty sentence is ten percent of relevant sentences, thus we applied high novelty threshold to filter more sentences. After evaluation results were returned, we found that assessor also considered the sentence is novel if the sentence is relevant. Therefore, we applied higher novelty threshold in the latter unofficial experiments. In this way, 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.

Table 1. Performance of Official Relevance Detection

	Relevance Part		
	Precision (P)	Recall (R)	P*R
ntu1	0.07	0.47	0.037
ntu2	0.07	0.47	0.033
ntu3	0.08	0.40	0.037

Table 2. Performance of Official Novelty Detection

	Novelty Part		
	Precision (P)	Recall (R)	P*R
ntu1	0.07	0.07	0.009
ntu2	0.06	0.07	0.008
ntu3	0.09	0.06	0.010

Our unofficial results are shown as follows. The set of sentences randomly selected from the target documents is regarded as a baseline model, its P*R score is 0.006. Table 3 lists the performance of relevance detector. The threshold for relevance detector is set to 0.4. Performance of the system (i.e., the P*R value) is improved as window size is increased from 1 to 4. When the window size is increased a little larger after the critical point, the performance starts to decline. The results show that larger window size may incorporate useful context information, but it may also select more irrelevant sentences.

Table 3. Performance of Relevance Detector ($TH_{relevance}$ = 0.4)

Window size	Precision (P)	Recall (R)	P*R
1	0.137	0.211	0.029
2	0.094	0.393	0.037
3	0.080	0.474	0.038
4	0.077	0.532	**0.041**
5	0.069	0.565	0.039

We chose the best performance of relevance part to experiment with the next component, Novelty detector. The experimental result is shown in Table 4. In this experiment, the novelty thresholds are set to 0.98 and 0.99. Table 4 indicates that more sentences are filtered as $TH_{novlety}$ is lower. The experimental result shows that the performance of revised novelty detector is two times better than that of the original one in the formal run. However, the performance is still not comparable to the human assessors. The major reason is that the result of relevance detector

contains irrelevant sentences, so novelty detector false identifies that those irrelevant sentences contain new information. As we mention before, the relevance part is the major difficulty to overcome in this task.

Table 4. Performance of Novelty Detector

Novelty Threshold	Precision (P)	Recall (R)	P*R
0.98	0.123	0.132	0.016
0.99	0.099	0.221	**0.022**

4 Conclusions and Future Work

In this paper, we proposed an approach to identify sentences that are novel and redundant as well as relevant and irrelevant. The method of matching keywords and related words in sentences may not be appropriate to the relevance part. We presented an information expansion approach to deal with this problem. We postulated that if two sentences have the similar meaning, then their behavior on information retrieval to a reference corpus (relevant sentence segments or an independent corpus) is similar. The current estimators for our approach should be improved, even though they sometimes work well on some topics. The syntactic and semantic analysis of sentences may help distinguish relevant sentence from target corpus.

To use a similarity function to measure if a sentence is on topic is similar to the function of an IR system. We may use a reference corpus, and regard a topic and a sentence as queries to this corpus. An IR system may retrieve top *n* 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 this approach include the reference corpus, the IR system, the number of documents reported, the similarity threshold, and the number of relevant sentences extracted.

The reference corpus consulted should be large enough to cover different themes for references. In the first experiments, the document sets used in TREC-6 text collection were considered as a reference corpus. It consists of 556,077 documents. 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.

We compute the Cosine of a topic vector T and a given sentence vector S_i ($1 \le i \le m$), where m denotes total number of the given sentences. Assume normal distribution

with mean μ and standard deviation σ is adopted to specify the similarity distribution of the given sentences with a topic.

$$\mu = \frac{\sum_{i=1}^{m} \cos(T, S_i)}{m} \quad (5)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{m} (\cos(T, S_i) - \mu)^2}{m}} \quad (6)$$

$$\cos(s_i, s_j) = \frac{\sum_{k=1}^{n} v_{i,k} \times v_{j,k}}{\|s_i\| \cdot \|s_j\|} \quad (7)$$

The percentage *n* denotes that top *n* percentages of the given sentences will be reported. Similarity thresholds ($TH_{relevance}$) shown as follows are determined by these percentages.

$$TH_{relevance} = \mu + z\sigma \quad (8)$$

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-y^2/2} dy = 1 - n \quad (9)$$

Even though the above dynamic approach has better performance, it is still "fixed percentage" for every topic. We consider further how to select "good" percentages for individual topics. Larkey *et al*. (2002) showed that only 5% of the sentences contained relevant materials for average topic. From their collection statistics (Larkey *et al*., 2002), we used linear regression as follows to capture the relationship between total number of the given sentences and number of the relevant sentences.

$$n = 47.903 - 0.006x \quad (10)$$

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

After computing *n* using Formula (10), we derived *z* using Formula (9) and finally $TH_{relevance}$ using Formula (8). Table 5 summarizes experimental results. For different size of ranked document lists, the performance is more stable (i.e., between 0.71 and 0.81). The best average P×R is 0.081, i.e., 42.41% of human performance.

Table 5. Performance of Relevance Detection with Dynamic Percentages

doc-size	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
P	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.13	0.12
R	0.46	0.48	0.49	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.51	0.51	0.51	0.50
P×R	**.072**	**.077**	**.079**	**.080**	**.081**	**.078**	**.078**	**.076**	**.075**	**.075**	**.074**	**.074**	**.074**	**.074**	**.074**	**.071**

Figure 5 lists the performance of each topic when 45 documents were returned by IR system. Two dotted lines, i.e., one is human performance (0.191) and the other one

is baseline performance (0.006), are provided for reference. Performance of our system in 8 topics (358, 364, 365, 368, 397, 414, 433 and 449) is competitive to that of human judge. In contrast, performance in 6 topics (305, 312, 315, 419, 420, and 432) is lower than that of random selection. The average P×R of the remaining 36 topics are below human performance, but better than that of baseline model.

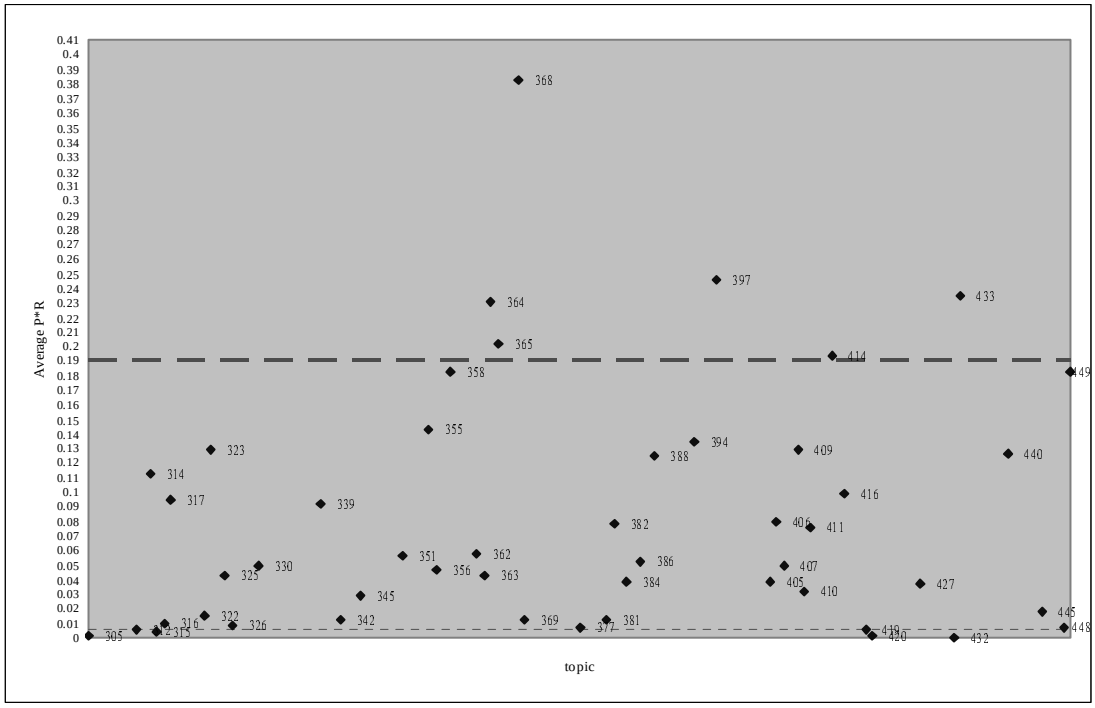

Figure 5. Average P×R of Relevance Identification for Each Topic

References

Allan, James; Carbonnell, Jaime; and Yamron, Jonathan (2002) *Topic Detection and Tracking: Event-Based Information Organization*, Kluwer, 2002.

Chen, Hsin-Hsi and Huang, Sheng-Jie (1999) "A Summarization System for Chinese News from Multiple Sources," *Proceedings of the 4th International Workshop on Information Retrieval with Asian Languages*, 1999, Taipei, Taiwan, pp. 1-7.

Chen, Hsin-Hsi and Ku, Lun-Wei (2002) "An NLP & IR Approach to Topic Detection," *Topic Detection and Tracking: Event-Based Information Organization*, James Allan, Jaime Carbonnell, Jonathan Yamron (Editors), Kluwer, pp. 243-264.

Chen, Hsin-Hsi and Lin, Chuan-Jie (2000) "A Multilingual News Summarizer," *Proceedings of 18th International Conference on Computational Linguistics*, 2000, University of Saarlandes, pp. 159-165.

Utiyama, M. and Isahara, H. (2001) "A statistical Model for Domain-Independent Text Segmentation," *Proceedings of ACL/EACL*, 2001, pp. 491-498.