Meiji University Web and Novelty Track Experiments at TREC 2003

Ryosuke Ohgaya, Akiyoshi Shimmura and Tomohiro Takagi
Department of Computer Science, Meiji University
{ohgaya, sinmura, takagi}@cs.meiji.ac.jp

Akiko Aizawa
National Institute of Informatics
akiko@nii.ac.jp

1. Introduction

This year we participated in TREC for the first time. We submitted runs for Novelty track and the topic distillation task of Web track.

2. Conceptual Fuzzy Sets

To represent the meaning of a word, we proposed conceptual fuzzy sets (CFS) [1][2]. In CFS, the meaning of a word is represented by the distribution of the activation of other words and dynamically changes reflecting context. The image of CFS is shown in Figure 1.

We used two different implementation of CFS in each track.

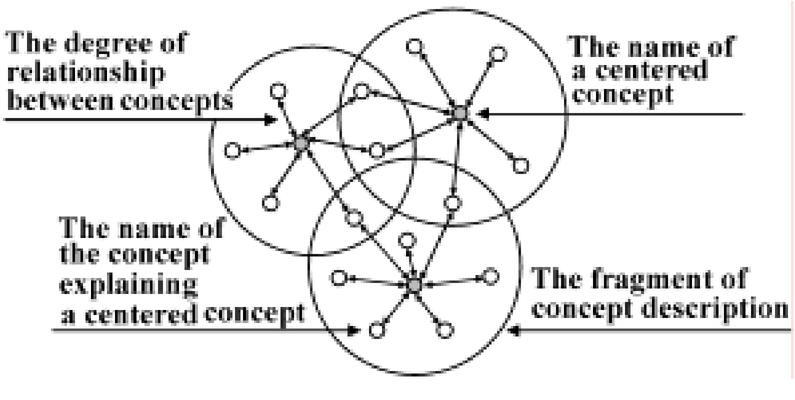

The degree of relationship between concepts
The name of a centered concept
The name of the concept explaining a centered concept
The fragment of concept description

Figure 1. Image of CFS

In Figure 1, white surrounding concepts explain the centered gray concept. The strength of the links between concepts reflects their degrees of relationship. The centered concept and its connected concepts constitute a fragment of concept description. A CFS is generated by overlapping the fragments of the activated concept description. A CFS expresses the meaning of a concept by the activation values of other concepts in these fragments.

3. Web Track

We submitted five runs for the topic distillation task. Our system is based on vector space model with tf-idf weighting. To create a document vector, we used the contents of a target page and those of its neighboring pages in the run *meijihil3*, *meijihil4* and *meijihil5*.

Searching procedure is:

1. Expand query using conceptual fuzzy sets (in *meijihil2*, *meijihil4* and *meijihil5*)
2. Calculate similarities
3. Rerank search results based on out-degree (in *meijihil5*)
4. Aggregate pages from the same server into one

Table 1 shows the description of each run.

Table 1. Evaluation results of submitted runs

Run	Query expansion	Inlinks & Outlinks	Reranking	R-Prec	P@10
meijihil1				0.0918	0.0920
meijihil2	O			0.0614	0.0700
meijihil3		O		0.0902	0.1060
meijihil4	O	O		0.0687	0.0700
meijihil5	O	O	O	0.0523	0.0620

3.1. Using the contents of inlinks and outlinks

In the World Wide Web, a web page and its neighboring pages are likely to be on the same topic. We evaluated whether incorporating the contents of neighboring pages in that of a target page improve search accuracy.

We create the document vector of a target page as follows:

1. Create the word vector of each page using only its contents with tf-idf weighting.
2. Aggregate the word vector of the target page and those of its neighboring pages:

$$V_i = \alpha V_{d_i} + \beta \sum_j V_{d_j} + \gamma \sum_k V_{d_k}$$

where V_{d_i} is the word vector of the target page, V_{d_j} is the word vector of a page that is linking to the target page and V_{d_k} is the word vector of a page that is linked to by the target page.

In the run *meijihilw3*, *meijihilw4* and *meijihilw5*, we set α, β and γ to be 1.0.

3.2. Query expansion using conceptual fuzzy sets

We used CFS to expand queries. To construct CFS, we need a dictionary in which the meanings of words are represented by other words and their degree of relationship.

3.2.1. Dictionary for conceptual fuzzy sets

To create the dictionary, we used a method proposed in [3] in which overlapping clusters of terms are generated based on co-occurrence (Actually, documents and other related information are also clustered simultaneously with terms, but we used only term clusters for the dictionary). A term cluster is composed of a representative word and related words with their degrees of relationship and is considered as a word vector that represents the concept of the word. We refer to this word vector as concept vector.

3.2.2. Expansion procedure

The similarities between the input vector and each concept vector are calculated using cosine measure:

$$S_i = \frac{V_q \cdot V_{C_i}}{|V_q||V_{C_i}|}$$

where V_q is the input vector and V_{C_i} is the i th concept vector.

The expanded query vector is the weighted sum of the concept vectors:

$$V_q' = \sum_i S_i \cdot V_{C_i}$$

3.3. Similarity calculation

We used cosine measure to calculate the similarity between input vector and each document vector. Document structure and proximity of query terms are also used: a document gets an additional score if the query terms appeared in title (<title>) or headings (<h*n*>) field or if the query terms appeared closely in the document.

3.4. Out-degree reranking

A key resource is expected to have links to many relevant pages. Thus we reranked initial search results based on out-degree as follows:

$$Sim'_{d_i} = Sim_{d_i} + \alpha \frac{1}{n} \sum_{j}^{n} Sim_{d_j}$$

where Sim_{d_i} is the initial score of the document d_i, Sim_{d_j} is the initial score of the document d_j that is pointed to by d_i and n is the number of outlinks in d_i. This technique is used in the run *meijihilw5* and α is set to be 1.0.

3.5. Site aggregation

Initial search results often give higher rank to pages from the same server. We simply merged them into one that has the shortest URL.

3.6. Results

Results are shown in Table 1. Query expansion and reranking failed to improve R-precision and P@10. Incorporating the contents of neighboring pages on the other hand showed some improvements.

4. Novelty Track

In Novelty Track, our main challenge is conceptual expansion of profiles and sentences. Expanding them using CFS can calculate similarities more correctly than only using word frequency.

We regarded sentences as very short documents, and converted them to word vectors. In the conversion phase, we removed stop words, stemmed words using Porter's algorithm and assigned weights to them using tf-idf.

4.1. Conceptual Expansion

We constructed the network shown in Figure 2 to implement CFS.

Concept vector C_i (fragment of the concept description) is created by clustering documents in Reuters corpus. The weights between concept layer and output layer are also trained using Reuters corpus.

An input is expanded as follows:

1. Calculate similarities between input vector X and each concept vector C_i:

$$S_i = \frac{X \cdot C_i}{|X||C_i|}$$

2. Expanded vector Y is calculated by propagating the similarities:

$$Y_j = \sum_i (a_{ij} \times S_i)$$

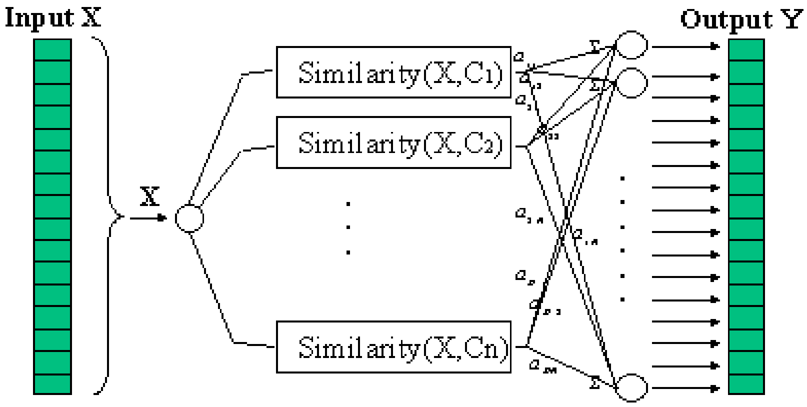


Figure 2. Network structure for CFS

4.2. Relevant Sentence

4.2.1. Relevancy Detection System Description

To identify relevant sentences, we used an information-filtering-based approach. Initial profiles, which are made with the topic descriptions, are expanded conceptually. If the cosine similarity between an

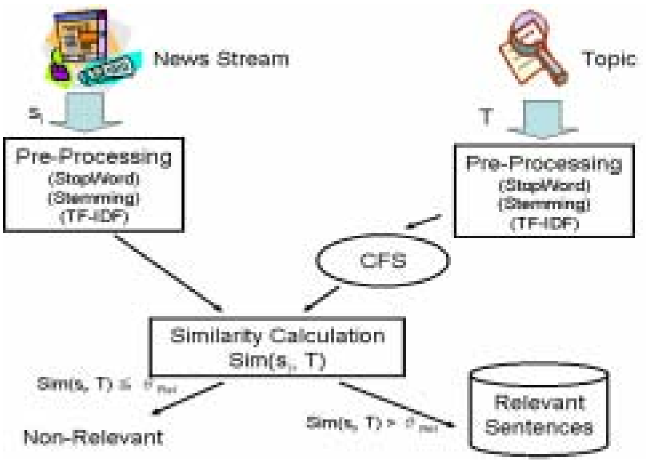


Figure 3. Architecture of relevancy detection system

expanded profile and the word vector of a sentence exceeds a threshold, the sentence is regarded as relevant. Figure 3 shows the architecture of our relevancy detection system. The title, description and narrative field were used to adjust profiles. Only the topic profile was expanded.

4.2.2. Threshold Learning

In this system, we must set an appropriate threshold to distinguish Relevant sentences from Non-Relevant ones. The threshold was trained by using the corpus of TREC2002 Novelty Track (min_qrels.relevant and max_qrels.relevant). We adopted the threshold where the F measure was maximized.

The number of New sentences in a Relevant sentence set decreases inevitably as the recall becomes low. Therefore, the threshold where the recall is 0.7 was also used.

4.2.3. System Variation

We had three system variations to identify Relevant sentences as shown in Table 2. The profiles were expanded by CFS in R1 and R2, but were not expanded in R3 to compare accuracy with R1 and R2.

Table 2. Relevancy detection system variation

	CFS Expansion	Threshold Learning
R1	O	Maximum F-Measure
R2	O	Recall=0.7
R3	X	Maximum F-Measure

4.3. New Sentence

4.3.1. Novelty Detection System Description

To identify new sentences, we used two measures: sentence score and redundancy score. 1) For calculating sentence score, we used N-window-idf to consider the time window. Local sentence score is calculated by using document frequency for the past N documents. 2) Redundancy score of a sentence is the maximum similarity between the sentence and ones judged to be new in the past. Figure 4 shows the architecture of our novelty detection system.

4.3.1.1. Sentence Score

The sentence score is calculated based on sentence weight proposed by Zechner [4]. We improved it so that it might take novel feature. If news documents are streaming in chronological order, they have the feature that a specific topic concentrates in a small range. Therefore, in order to judge novelty, it is effective not to consider globally, but to consider locally. We used local rarity of a word to use this feature. It is calculable using N-window-idf which is document frequency in past N documents. By using N-window-idf, weights of frequent words decrease and sentence weights represent local information.

$$SentenceScore(s) = \sum_i tf(t_i) \times N - window - idf(t_i)$$

$$N - window - idf(t) = \log \frac{N}{N - window - df(t)}$$

where tf(t_i) is the frequency of the word t_i in the sentence s, N is the window size, and N-window-df(t) is the document frequency of the word t in past N documents.

4.3.1.2. Redundancy Score

To calculate the redundancy score, we used maximum similarity of sentences which are already identified as novelty. The similarity is calculated by cosine measure.

$$RedundancyScore(s) = \underset{NovSi \in NoveltySentences}{Max} Similarity(NovSi, s)$$

4.3.1.3. Novelty Score

We used the sentence score and the redundancy score to identify the novelty. We thought that novelty sentences must have higher information weight and differ from pre-selected novelty sentences. Therefore, we combined these scores:

$$NoveltyScore(s) = \lambda \times SentenceScore(s) - (1 - \lambda) \times RedundancyScore(s)$$

If the NoveltyScore exceeds a threshold, the sentence is regarded as novelty.

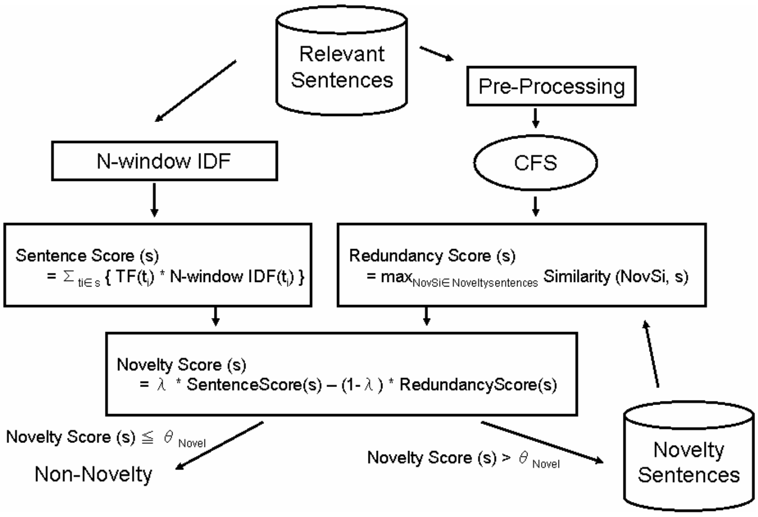

Relevant Sentences
Pre-Processing
N-window IDF
CFS
Sentence Score (s) = Σ ti∈s { TF(ti) * N-window IDF(ti) }
Redundancy Score (s) = max NovSi∈Noveltysentences Similarity (NovSi, s)
Novelty Score (s) = λ * SentenceScore(s) – (1-λ) * RedundancyScore(s)
Novelty Score (s) ≦ θ Novel
Non-Novelty
Novelty Score (s) > θ Novel
Novelty Sentences

Figure 4. Architecture of novelty detection system

4.3.2. Parameter Setting

To identify new sentences, we had to set up three parameters:

1) window size: N
2) ratio of sentence score to redundancy score: λ
3) threshold for judging whether the input sentence is New or not: θ

We set the widow size to 200 based on the number of sentences to a news document. λ and θ were determined by learning using Trec2002 Novelty Track data (min_qrels.new, max_qrels.new) as well as Relevancy Detection System. We adopted λ and θ from which F measure becomes the maximum.

4.3.3. System Variation

Four variations were prepared (Table 3).

Table 3. Novelty detection system variation

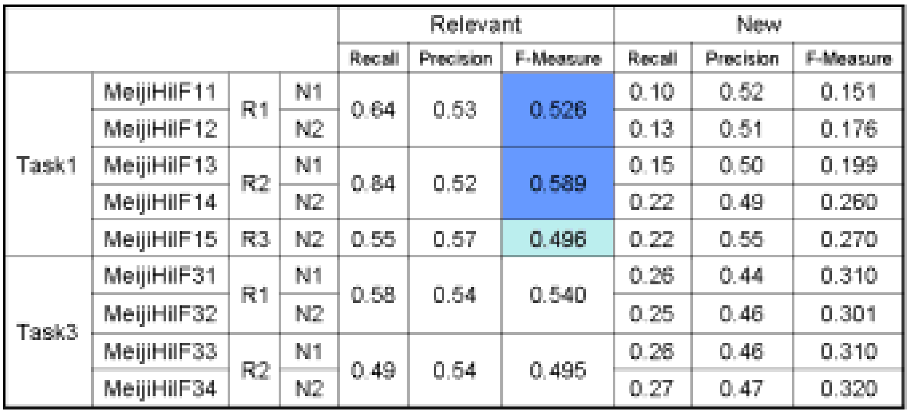
	N-window-idf	CFS Expansion
N1	O	O
N2	O	X
N3	O	-
N4	X	-

In N-window-idf column, [O] means N-Window-idf is used to calculate sentence scores and [X] means basic idf is used instead of N-Window-idf. The df values of basic idf were calculated using about 810,000 news documents in Reuters corpus. In Expand column, [O] means CFS expansion is used to calculate redundancy scores, [X] means expansion is not used and [-] means redundancy scores are not used.

4.4. Result and Discussion

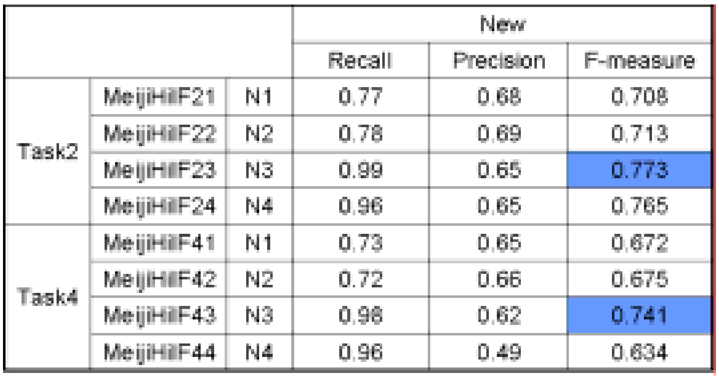
We submitted for Task 1-4. Table 4-5 shows the results. In the Relevancy Detection phase, the validity of expansion by CFS has been shown. Moreover, we presented the validity of N-window-idf, which considered locality, in the Novelty phase.

Table 4. Result of Task 1 and Task 3



				Relevant			New		
				Recall	Precision	F-Measure	Recall	Precision	F-Measure
Task1	MeijiHilF11	R1	N1	0.64	0.53	0.526	0.10	0.52	0.151
	MeijiHilF12		N2				0.13	0.51	0.176
	MeijiHilF13	R2	N1	0.84	0.52	0.589	0.15	0.50	0.199
	MeijiHilF14		N2				0.22	0.49	0.260
	MeijiHilF15	R3	N2	0.55	0.57	0.496	0.22	0.55	0.270
Task3	MeijiHilF31	R1	N1	0.58	0.54	0.540	0.26	0.44	0.310
	MeijiHilF32		N2				0.25	0.46	0.301
	MeijiHilF33	R2	N1	0.49	0.54	0.495	0.26	0.46	0.310
	MeijiHilF34		N2				0.27	0.47	0.320

Table 5. Result of Task 2 and Task 4



			New		
			Recall	Precision	F-measure
Task2	MeijiHilF21	N1	0.77	0.68	0.708
	MeijiHilF22	N2	0.78	0.69	0.713
	MeijiHilF23	N3	0.99	0.65	0.773
	MeijiHilF24	N4	0.96	0.65	0.765
Task4	MeijiHilF41	N1	0.73	0.65	0.672
	MeijiHilF42	N2	0.72	0.66	0.675
	MeijiHilF43	N3	0.98	0.62	0.741
	MeijiHilF44	N4	0.96	0.49	0.634

5. References

[1] T. Takagi, A. Imura, H. Ushida, and T. Yamaguchi, "Conceptual Fuzzy Sets as a Meaning Representation and their Inductive Construction", *International Journal of Intelligent Systems*, Vol.10, pp.929-945, 1995.

[2] T. Takagi, A. Imura, H. Ushida, and T. Yamaguchi, "Multilayered Reasoning by Means of Conceptual Fuzzy Sets", *International Journal of Intelligent Systems*, Vol.11, pp.97-111, 1996.

[3] A. Aizawa, "A Method of Cluster-Based Indexing of Textual Data", *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pp.1-7, 2002.

[4] K. Zechner, "Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences", *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, pp.986-989, 1996.