PERGAMON

Information Processing and Management 39 (2003) 363–389

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

The diversity-based approach to open-domain text summarization ☆

Tadashi Nomoto [a,*], Yuji Matsumoto [b]

[a] *National Institute of Japanese Literature, 1-16-10 Yutaka Shinagawa, Tokyo 142-8585, Japan*
[b] *Nara Institute of Science and Technology, 8916-5 Takayama Ikoma, Nara 630-0129, Japan*

Received 9 November 2001; accepted 12 October 2002

Abstract

The paper introduces a novel approach to unsupervised text summarization, which in principle should work for any domain or genre. The novelty lies in exploiting the diversity of concepts in text for summarization, which has not received much attention in the summarization literature.

We propose, in addition, what we call the *information-centric* approach to evaluation, where the quality of summaries is judged not in terms of how well they match human-created summaries but in terms of how well they represent their source documents in IR tasks such document retrieval and text categorization.

To find the effectiveness of our approach under the proposed evaluation scheme, we set out to examine how a system with the diversity functionality performs against one without, using the test data known as BMIR-J2. The results demonstrate a clear superiority of the diversity-based approach to a non-diversity-based approach.

The paper also addresses the question of how closely the diversity approach models human judgments on summarization. We have created a relatively large volume of data annotated for relevance to summarization by human subjects. We have trained a decision tree-based summarizer using the data, and examined how the diversity method compares with the supervised method in performance when tested on the data. It was found that the diversity approach performs as well as and in some cases superior to the supervised method.

Keywords: Text summarization; Clustering; Decision tree; MDL; Extrinsic and intrinsic evaluation

☆ The present paper is an extensively revised version of our earlier work presented at ACM/SIGIR 2001 (Nomoto & Matsumoto, 2001).

* Corresponding author. Tel.: +81-3-3785-7131; fax: +81-3-3785-4452.
E-mail addresses: nomoto@nijl.ac.jp (T. Nomoto), matsu@is.aist-nara.ac.jp (Y. Matsumoto).

0306-4573/03/$ - see front matter
doi:10.1016/S0306-4573(02)00096-1

1. Introduction

Prior work on automatic text summarization usually come in two flavors: supervised and unsupervised. Supervised approaches (Berger & Mittal, 2000; Chuang & Yang, 2000; Kupiec, Pedersen, & Chen, 1995; Marcu, 1999b) typically make use of human-made summaries or extracts to find features or parameters of summarization algorithms, while unsupervised approaches (Carbonell & Goldstein, 1998; Gong & Liu, 2001; Luhn, 1999; Zechner, 1996) determine relevant parameters without regard to human-made summaries. One of the problems with the former approach has to do with its underlying assumption that human-made summaries are reliable enough to be used as "gold standards" for automatic summarization. Recently, research on human summarization has witnessed some conflicting results about the validity of the assumption.

Nomoto and Matsumoto (1997), for instance, asked a large group of university students to make 10% extracts for a set of texts drawn from various domains in a news paper corpus. They reported a rather modest result of about 25% kappa agreement among students on extracts. Also Salton, Singhal, Mitra, and Buckley (1999) reports low inter-subject agreement on paragraph extracts from encyclopedias. On the other hand, there have been some reports to the contrary. Marcu (1997) found 71% (percent) agreement among judges on sentence extracts from expository articles; Jing, Barzilay, McKeown, and Elhadad (1998) found quite high percent agreement (96%) for extractions from TREC articles. [1]

It is not known at present what factors are involved in influencing the reliability of summarization and therefore we do not know whether there is any principled way of eliciting reliable summaries from humans.

Another problem associated with the approach concerns the portability of a summarization system: deploying the system in a new domain usually requires one to collect a large amount of data, which need to be manually annotated, and then train the system. Besides being costly, annotation demands intensive human labor, which prompted, for example, Marcu (1999b) to address the problem of automating the construction of summarization corpora.

The first part of the paper describes the approach to text summarization which strives to overcome the issues of portability and the quality of human-made summaries mentioned above. We begin by arguing for what we call the *information-centric* approach for summary evaluation. We will then explain mechanisms that drive the summarizer, and evaluate our approach using a data set known as BMIR-J2, under the information-centric paradigm.

The second part of the paper addresses the problem of how well the unsupervised approach laid out here can model human judgments in summarization tasks. We will make an experimental

[1] However, there is a good reason to question the wisdom of using percent agreement for measuring agreement among humans on sentence extraction. The problem with the percent agreement is that it includes not only sentences humans agree to put in a summary but also those they agree not to, which obviously constitute a large part of a text, and therefore tends to inflate agreement. The kappa statistic, in contrast, is equipped with a device to correct it for expected agreement and gives a more conservative figure. Consider, for instance, a task of choosing one out of 10 sentences as a summary. Suppose further that you have two judges completely disagreeing on what they think should go into a summary. Percent agreement gives an astonishingly high figure of 80% for the task, in contrast to the kappa, which is as modest as −11%.

comparison of our approach and an approach based on supervised learning. We will conclude the paper by discussing some of the limitations of the current approach and possible future directions.

2. Information-centric approach to evaluation

Like previous extract-based approaches (Edmundson, 1969; Kupiec et al., 1995; Luhn, 1999), we define a summary as a set of sentences extracted verbatim from a text, which cover major substance of that text.

However, the present approach significantly differs from previous work in taking an *information-centric* approach to evaluation. We evaluate summaries, not in terms of how well they match human-made extracts (Edmundson, 1969; Kupiec et al., 1995), nor in terms of how much time it takes for humans to make relevance judgments on them (Mani et al., 1998), but in terms of how well they represent source documents in usual IR tasks such as document retrieval and text categorization. We are interested in asking whether it is possible to replace documents altogether by corresponding summaries and do as well in IR tasks. Thus an ideal summary would be one whose rank order in retrieved documents is same as that of its source document, or whose assigned category is same as that of its source document. This notion of summary as a perfect surrogate of its source, while left unexplored in research on summarization, permits a simple and objective characterization of how well summaries represent the original documents. [2]

There have been arguments against extractive summarization in the computational linguistics literature, because extracts generally lack fluency or cohesion. However, Morris, Kasper, and Adams (1999) found in a reading comprehension study that humans were able to perform as well reading 20–30% extracts as the original full texts and expert-created abstracts. Humans were able to capture enough of the information from extracts so that they could perform as if they had read their full-length versions.

3. The diversity-based summarization

Assuming that the problem of summarization is one of finding a subset of sentences in text which in some way represents its source text, a natural question to ask is, 'what are the properties of text that should be represented or retained in a summary?' Katz (1996) is enlightening in this regard. In his work on language model, he made an important observation that the numbers of occurrences of content words in a document do not depend on the document's length, that is, the frequencies per document of individual content words do not grow proportionally with the length of a document. Where is the missing mass that accounts for the discrepancy between the document length and frequencies of content words? He resolves the apparent puzzle by showing that it is the number of *different* content words in text that increases with the document length.

[2] One might consider the information-centric evaluation here an extreme form of extrinsic evaluation (Mani et al., 1998; Sparck Jones & Gallier, 1995).

Katz's observation illuminates two important properties of text: *redundancy* and *diversity*. The former relates to how repetitive concepts (or content words) are, the latter relates to how many different concepts there are in the text. While much of the prior work on summarization capitalize on redundancy to find important concepts in the text or its relevance to the query, few of them address the problem of diversity. One exception is Carbonell and Goldstein (1998), who added the diversity component to a criterion for sentence selection, which they call the *maximal marginal relevance* or MMR. MMR selects a sentence both relevant to the query and least similar to sentences selected previously. Mani et al. (1998) report that MMR-based summarizers ranks among the best in the 1998 SUMMAC conference. Radev, Jing, and Budzikowska (2000) also make use of some dissimilarity measure for ranking sentences for multi-document summarization.

Gong and Liu (2001) present yet another attempt to incorporate an MMR-like feature in summarization.

3.1. The method

The above discussion motivates a summarization strategy which takes seriously diversity as well as redundancy of concepts in text. We will show how to construct a generic single-document summarizer along this direction. The summarizer would consist of the following three components:

Find-diversity
Find diverse topical clusters in a text.

Reduce-redundancy
From each topic cluster, identify the most important sentence and take that sentence as a representative of that cluster.

Generate-summary
Form a summary by collecting and putting in some order sentences identified by Reduce-redundancy.

The term "topical cluster" here is to be understood as a set of sentences which are mutually similar by some criterion but may not be necessarily contiguous. Let us look at each of the operations in detail.

3.1.1. Find-diversity

Find-diversity is built upon the K-means clustering algorithm extended with minimum description length principle (MDL) (Li, 1998; Rissanen, 1997). The algorithm presented here is an MDL-version of X-means (Pelleg & Moore, 2000). X-means itself is an extension of the popular K-means clustering algorithm with an added functionality of estimating K, the number of clusters which otherwise needs to be supplied by the user. We call our adaptation of X-means 'X^M means.'

For the remainder of the paper, borrowing in part notations from Pelleg and Moore (2000), we denote by μ_j the coordinates of the centroid with the index j, and by x_i the coordinates of the ith data point, (i) represents the index of the centroid closest to the data point i. $\mu_{(j)}$, for example, denotes the centroid associated with the data point j. c_i denotes a cluster with the index i.

K-means is a hard clustering algorithm that produces a clustering of input data points into K disjoint subsets. It dynamically redefines a clustering by relocating each centroid to the center of mass of points associated with it and re-associating the centroid with points closest to it.

K-means starts with some randomly chosen initial points. As noted in the machine learning literature (Bradley & Fayyad, 1998; Pelleg & Moore, 2000), a bad choice of initial centers can have adverse effects on performance in clustering. In experiments described later in the paper, following Bradley and Fayyad (1998), we repeatedly ran K-means with random initial points and selected the best among clustering solutions on the basis of *distortion*, a measure for the tightness of a cluster. A best solution is one that minimizes the distortion.

We define distortion as the averaged sum of squares of Euclidean distances between objects of a cluster and its centroid. Thus for some clustering solution, or a set $S = \{c_1, \ldots, c_k\}$ of clusters, its distortion is given by:

$$D(S) = \sum_{i}^{k} V(c_i),$$

where

$$V(c_i) = \frac{1}{|c_i|} \sum_{j} (x_j - \mu_{(i)})^2.$$

Here c_i denotes a cluster, x_j is an object in c_i, $\mu_{(i)}$ represents the centroid of c_i, and $|\cdot|$ is the cardinality function.

One problem with K-means is that the user has to supply the number of clusters, and it is known that it is prone to searching local minima (Pelleg & Moore, 2000). X-means overcomes these problems by globally searching the space of centroid locations to find the best way of partitioning the input data. X-means resorts to a model selection criterion known as the Bayesian information criterion (BIC) to decide whether to split a cluster. The splitting happens when the information gain from splitting a cluster as measured by BIC is greater than the gain for keeping that cluster as it is.

Let us graphically illustrate this situation. Fig. 1 shows a K-means solution with four centroids (large dots), which cover four distinct regions of the data space. The split operation examines each

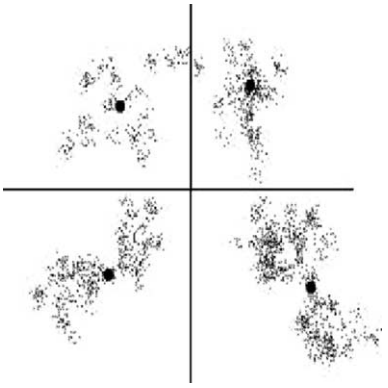

Fig. 1. The initial state with four regions.

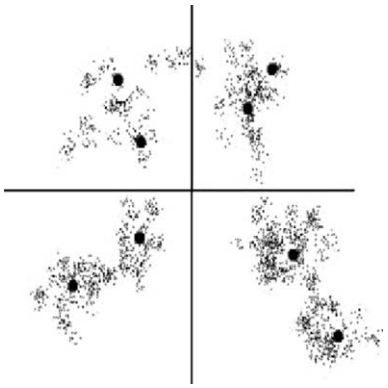

Fig. 2. Each local cluster splits into two sub-regions.

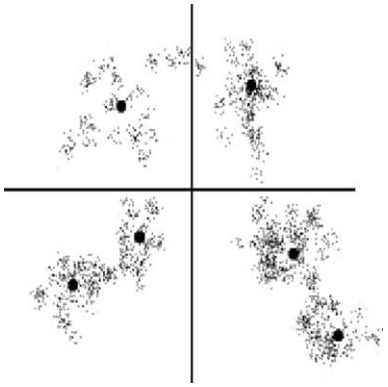

Fig. 3. BIC determines that some of sub-regions are not worth keeping.

of the four clusters, breaks each of them in two, running the regular K-means on each local region with $K = 2$, and decides whether the splitting is worthwhile in terms of BIC. As mentioned above, each call to K-means involves repeated runs of itself with randomly chosen initial centers and selecting the best clustering solution from those runs. (In experiments described later, we performed 200 runs of K-means and selected from those runs a solution with the least distortion.) Fig. 2 shows a state where each local cluster splits into two sub-regions by K-means. In Fig. 3, BIC determines that the upper left and upper right regions are not worth breaking up, leavings us with six clusters.

One modification to X-means involves replacing BIC by minimum description length principle or MDL, a well-studied general criterion for model selection. [3]

[3] One may do equally well with BIC or Akaike information criterion (AIC). There is no a priori reason for choosing one criterion over another. The choice of MDL over others is largely motivated by our familiarity with the paradigm and interest in how MDL fares when combined with clustering.

In general, given the data $x_1, \ldots, x_m$, MDL insists that a model that allows a shortest description for the data is most likely to have given rise to them. (A model here is thought of as the probability distribution of a variable X, where $X = x_j$.) In MDL, the length of a description of data is given as the sum of bits required to encode a model and bits required to encode the data given the model. The best hypothesis or model, h for $x^m = x_1, \ldots, x_m$ is then expressed as follows:

$$h_{\text{best}}(x^m) = \arg\min_{M \in \mathbf{M}} L(x^m : M),$$

where $L(y^m : M) = L(x^m|M) + L(M)$, $\mathbf{M}$ is the set of possible models, and $L(x)$ is a description-length of x. $L(x^m|M)$ denotes the description length of data, given the model M, which is the sum of the maximum log-likelihood estimate of $P(x^m|M)$ and the coding length of parameters involved.

Let us assume identical hyper-spherical Gaussian distributions for input data. Also let each data point represent a multi-dimensional encoding of the sentence in text, i.e. a vector of weights of index terms in the sentence. Then the probability that a given data point x_i belongs to a cluster $c_{(j)}$ can be defined as the product of the probability of observing $c_{(i)}$ and the multi-variate normal density function of x_i, with the covariance matrix $\Sigma = \sigma^2 I$. [4] Therefore we have

$$\widehat{P}(x_i) = \frac{R_{(i)}}{R} \frac{1}{\sqrt{2\pi}\hat{\sigma}^U} \exp\left(-\frac{1}{2\hat{\sigma}^2} \|x_i - \mu_{(i)}\|^2 \right),$$

where $R_{(i)} = |c_{(i)}|$, $\|\cdot\|$ is the Euclidean norm, R is the total number of input data points, U is the number of dimensions, and $\hat{\sigma}^2$ is the maximum likelihood estimate of the variance such that:

$$\hat{\sigma}^2 = \frac{1}{R-K} \sum_i (x_i - \mu_{(i)})^2.$$

Therefore the maximum log-likelihood of the cluster c_j is:

$$\hat{l}(c_j) = \log \prod_{i \in c_j} \widehat{P}(x_i) = \sum_{i \in c_j} \left(\log \frac{1}{\sqrt{2\pi}\hat{\sigma}^U} - \frac{1}{2\hat{\sigma}^2} \|x_i - \mu_{(i)}\|^2 + \log \frac{R_{(i)}}{R} \right)$$

which is equivalent to:

$$-\frac{R_{(i)}}{2} \log(2\pi) - \frac{R_{(i)} U}{2} \log(\hat{\sigma}^2) - \frac{R_{(i)} - K}{2} + R_i \log R_i - R_i \log R.$$

Now define the probability of observing the model M by:

$$P(M) = \frac{1}{|\mathbf{M}|}.$$

Then the description-length of the model M is: $L(M) = -\log |\mathbf{M}|$, where $|\mathbf{M}| = K_{\max}$ in our case. But we can drop it from the MDL formula since it is invariant for any choice of M. This brings us to the final form of MDL:

[4] Thus we assume that the features are statistically independent and have the same variance σ^2. See, for instance, Duda, Hart, and Stork (2001), for details.

Table 1
The X^M-means clustering algorithm

```
X^M-means(c_0, K_max)
begin
C = φ
(c_1^0, c_2^0) = 2-means(c_0)
C = C ∪ {c_1^0, c_2^0}
k = 2
while k < K_max and k does not converge
    begin
    S = {c : c ∈ C, MDL(2-means(c)) < MDL(c)}
    if S is not empty then
      c_best = arg min_{c∈S} MDL(2-means(c))
      C = C \ {c_best} ∪ {c_1^k, c_2^k}^{c_best}
      k = k + 1
    endif
    end
end
```

$$L(D:M) = -\sum_{c_i}^{K} \hat{l}(c_i) + \frac{k}{2}\log R,$$

where M is a model, D is a set of data points in the input space, c_i is a region or cluster as demarcated by M, R is the size of the input data, and k is the number of free parameters in M. The number of parameters here is simply the sum of $K-1$ cluster probabilities, $U \cdot K$ centroid coordinates, and one variance estimate.

Table 1 shows an algorithm for X^M means. It takes as input the entire data region (represented by a single cluster c_0) and the maximum number $K_{\max}$ of clusters one likes to have. It begins then by running 2-means (i.e. K-means with $K=2$) on the entire region, giving birth to two sub-regions (clusters) c_0^1 and c_0^2. Add those to C. Set k, the number of clusters in C, to 2. If $k < K_{\max}$ and there is no convergence, then run 2-means on each local region found in C and test if the splitting produces a pair of child regions whose MDL is smaller than that of the parent. If that is the case, store that information in S. Find a region in S whose splitting results in the smallest MDL. Replace the region with those of children. Increment k by 1. Halt if $k \geqslant K_{\max}$. If not, repeat the whole process. The idea here is that regions which are not represented well by current centroids will receive more attention with an increased number of centroids in them. The algorithm searches the entire data space for the best region to split.

3.1.2. Reduce-redundancy

For reduce-redundancy, we will use a simple sentence weighting model by Zechner (1996) (call it the *Z-model*), where one takes the weight of a given sentence as the sum of tf · idf values of index terms in that sentence. Let the weight of sentence s, or $W(s)$ be given by:

$$W(s) = \sum_{x \in s}(1 + \log(\mathrm{tf}(x)))\mathrm{idf}(x),$$

where x denotes a index term, $\mathrm{tf}(x)$ is the frequency of term x in document, $\mathrm{idf}(x)$ is the inverse document frequency of x. In the Z-model, sentence selection proceeds by: determining the weights of sentences in the text; sorting them in a decreasing order; and finally selecting top sentences. Our implementation of the Z-model further normalized sentence weight for length.

Reduce-redundancy applies the Z-model to each one of the clusters identified by Find-Diversity and tries to find a sentence with the best $W(s)$ score, which it takes as representative of the cluster. Note that this strategy of picking up best scoring sentences is designed to minimize the loss of the resulting summary's relevance to a potential query. This is in contrast to an approach by Radev et al. (2000), where the centroid of a cluster is selected instead.

4. Test data and evaluation procedure

4.1. BMIR-J2

BMIR-J2 (Benchmark for Japanese IR system version 2) represents a test collection of 5080 news articles in Japanese, all of them published in 1994 (Nichi-Gai Associates, 1995). Articles were collected from diverse domains, including economy, engineering, and industrial technology, but they all came from a single news paper source. The collection comes with a set of 60 queries and the associated list of answers indicating which article is relevant to which query to what degree. The degree of relevance falls into one of three categories: A, B, and C. 'A' indicates a perfect match to the query, 'B' some relevance and 'C' no relevance. All of the articles were manually labeled for relevance, and labelings were reviewed by the BMIR-J2 committee, comprising researchers from industry and academia. The collection also features a classification of queries based on sorts of language technologies potentially required to process them (Table 2). As can be seen from the table, the classification gives us a general idea of how difficult the task of retrieval using a given query is. For instance, to properly identify documents relevant to a query of type A requires the morphological analysis of the query, which involves tokenization, lemmatization and assignments of grammatical categories, and also the possible use of a thesaurus, and to deal with a query of type B, one needs some way of analyzing collocation involving numbers, which is more difficult than an A-type query. To find documents relevant to a query of type F, it is claimed, one has to make reference to common sense or knowledge about the world and also some reasoning, which is the hardest of all. C, D, and E-type queries come in between.

Table 2
The classification of queries in BMIR-J2

Type	Explanation	Primary	Secondary
A	Morphological analysis, the use of the saurus	14	0
B	The analysis of collocation involving numbers, e.g. inequality, range	3	1
C	Syntactic analysis	10	1
D	Semantic, discourse analysis	9	2
E	World/common sense knowledge	4	3
F	Semantic, world/common sense knowledge, reasoning	10	3

Figures under the primary (secondary) heading indicates the number of queries that fall under a given type.

Moreover, a set of queries prepared by BMIR-J2 comprises a primary set of 50 queries, for each of which there are five or more relevant documents, and a secondary set of 10 queries, each having one to four relevant documents. The description of a query in BMIR-J2 contains two types of information; query words/phrases used for the retrieval of documents and a short explanation of search needs the query is meant for. In experiments, we used the primary set of queries.

4.2. Experiment setup and procedure

We have conducted experiments using BMIR-J2. Our interest was in finding out whether the diversity-based summarizer as formulated here is superior to some baseline systems proposed in the literature such as the Z-model described earlier and the lead-based system, which generates a summary by picking up the leading portion of a text. Furthermore, we use the Z-model in the hope that a comparison between the Z-model and the diversity models might reveal true effects of the diversity component on performance in summarization.

We treated summaries as if they were stand-alone documents, and performed usual document retrieval sessions using them: which is to retrieve documents for a particular query and rank them according to the cosine similarity to the query. We scored performance in F-measure where for given P (precision) and R (recall),

$$F = \frac{2PR}{P+R}.$$

We performed two sets of experiments which differ in what relevance scheme is employed. One scheme, which we call the *strict relevance scheme* (SRS), takes only A-labeled documents as relevant to the query, while another, called the *moderate relevance scheme* (MRS), takes both A- and B-labeled documents as relevant. BMIR-J2 recommends the latter scheme.

In the experiments, we ran each summarizer at a given compression rate and evaluated its performance on BMIR-J2, using either of the relevance schemes. Summarizers included the Z-model, a diversity-based summarizer with the standard K-means (hereafter, DBS/K), a diversity-based summarizer with X^M-means (hereafter, DBS/X^M), and a lead-based summarizer, or LEAD, which works by a simple heuristic of selecting the initial portion of a text as a summary. DBS/K, which is identical with DBS/X^M except for the diversity component, was introduced to examine the effects of MDL on K-means. To obtain a given compression level α with DBS/X^M, we set $K_{\max}$ to the corresponding number of sentences in text: e.g., for the text of 10 sentences, $K_{\max} = 5$ for $\alpha = 50\%$. Similarly for DBS/K. We note here that the number of sentences DBS/X^M collects could be less than $K_{\max}$, as the optimal number of clusters as determined by MDL need not be $K_{\max}$. However as far as BMIR-J2 is concerned, we find that DBS/X^Mreturns $K_{\max}$ sentences, for most of the time.

One feature specific to the present test domain is the use of a Japanese tokenizer ChaSen (Matsumoto, Kitauchi, Yamashita, & Hirano, 1999), which breaks up sentences into words, each labeled with relevant morphological information such as part of speech. ChaSen is reported to have the accuracy of over 98% (Matsumoto et al., 1999). For index terms, we used everything except for punctuation marks, non-linguistic symbols, particles such as case marker. Furthermore, we did not use any stop-list except for those elements already excluded from the set of index terms.An evaluation procedure takes the following steps:

(1) At a given compression rate ranging between 10% and 50%, run the Z-model, DBS/K, DBS/X^M and LEAD on the entire BMIR-J2 collection, to produce respective pools of extracts.
(2) For each query from BMIR-J2, perform a search on each pool generated, and score performance with the uninterpolated average F-measure.

Since we have two relevance schemes to consider, we did the sequence under each scheme, which brought the total of retrieval sessions to some 2000.

One problem with the use of a summary as a surrogate of its full length source in document retrieval is that the condensation process usually destroys statistical properties of a source text such as term frequency: thus for instance, terms X and Y, which may have different frequencies in the source, could end up having the same number of occurrences in a summary, which would leave us with no way of distinguishing between them in terms of term weight. One way to go about the problem is to extrapolate frequencies of index terms in a summary in order to estimate their true frequencies in its source, which we did using the following formula from Katz (1996):

$$E(k|k \geqslant m) = \sum_{r \geqslant m} \left(\frac{p_r}{\sum_{j \geqslant m} p_j} \right) r = \frac{\sum_{r \geqslant m} p_r r}{\sum_{r \geqslant m} p_r}, \tag{1}$$

where p_r denotes the probability of a given word occurring r times in the document and $m \geqslant 0$. Formula 1 (= (6.4) in Katz (1996)) estimates the average number of occurrences of a word in the documents, each of which contains at least m occurrences of that word. With Formula (1) it is possible to estimate the average frequency in the source of a word observed in a summary. For example, if we observe m occurrences of a word w in a summary, its expected frequency in its source text is given as $E(k|k \geqslant m)$.

In our experiments, we restricted ourselves to index terms with two or more occurrences in the document, so their extrapolated estimates would be $E(k|k \geqslant 2)$.

The 'df' values of index terms in a summary are obtained directly from a pool of summaries.

5. Results and discussion

Tables 3 and 4 give us a detailed picture of how Z, DBS/K, DBS/X^M and LEAD match against one another in the two relevance paradigms, i.e. moderate and strict relevance schemes. As with others, how much of a text is selected is determined by a given compression rate, except for the full-length system (FULL), which runs a query on full-length documents and therefore is immune to compression. (Nonetheless, we put its performance score at each compression rate to allow an easy comparison across the systems.) All the figures are scores in F-measure averaged over the primary set of queries.

Returning to Table 3, we are struck by the lack of significant differences in performance among the systems, which makes a curious contrast with their performance in MRS, where the differences are more apparent. Indeed, as shown below, statistical tests we have performed on the results confirm the differences or the lack of them among the systems.

Table 3
Average performance of Z, DBS/K, DBS/X^M, and LEAD in SRS

Compression rate	(Full)	Z	DBS/K	DBS/$X(M)$	LEAD
10%	(0.230)	0.203	0.211	0.227	0.230
20%	(0.230)	0.231	0.208	0.225	0.244
30%	(0.230)	0.225	0.220	0.220	0.260
40%	(0.230)	0.240	0.222	0.225	0.252
50%	(0.230)	0.234	0.227	0.235	0.258

'Full' denotes the full-length retrieval system, which runs a query on full-length documents. Figures are in F-measure.

Table 4
Average performance of Z, DBS/K, DBS/X^M, and LEAD in MRS

Compression rate	(Full)	Z	DBS/K	DBS/$X(M)$	LEAD
10%	(0.170)	0.145	0.185	0.206	0.178
20%	(0.170)	0.178	0.191	0.208	0.187
30%	(0.170)	0.194	0.209	0.223	0.203
40%	(0.170)	0.214	0.213	0.234	0.221
50%	(0.170)	0.227	0.228	0.233	0.225

Again, 'Full' here denotes the full-length retrieval system. Figures are in F-measure.

The null hypothesis here is that there is no difference in performance between a pair of the systems in either relevance scheme. What we did was to run the two-tailed paired t-test on each pair to see if the null hypothesis holds. The results are listed in Tables 5 and 6. Table 5 is for SRS and Table 6 for MRS. P-values breaking the 5% significance level are marked with an asterisk.

We see from the tables that most of the differences in performance at 10% in MRS (Table 4) are indeed true differences, whereas in SRS any one of the systems is performing just as good as any other. Thus consider, for instance, LEAD (L) and DBS/X^M (X). Tables 5 and 6 demonstrate that in MRS the two systems are significantly different from each other at compression rates from 10% through 30% whereas in SRS, there is no statistical ground for believing that either system is distinct in performance from the other, though they exhibit a moderately significant difference at 30% and 40%.

Table 5
Significance scores (P-values) for SRS

	10%	20%	30%	40%	50%
L:X	0.9067	0.3978	0.0779	0.0666	0.1662
L:Z	0.1712	0.6704	0.0189*	0.4802	0.1992
L:K	0.2832	0.0784	0.2470	0.1609	0.1058
K:Z	0.5953	0.4751	0.8732	0.0605	0.3744
Z:X	0.2778	0.8460	0.7368	0.1842	0.8722
K:X	0.2577	0.1788	0.9952	0.7924	0.1773

The asterisk indicates the 5% significance level. Note that we are looking here at differences in performance on the primary set of queries between each pair of the systems. Below, we let L stand for LEAD, X for DBS/X^M, K for DBS/K, and Z for the Z-model. 'L:X' reads like 'L is compared with X'.

Table 6
Significance scores (*P*-values) for MRS

	10%	20%	30%	40%	50%
L:X	0.0017*	0.0116*	0.0253*	0.0778	0.1368
L:Z	0.0005*	0.6834	0.3490	0.3494	0.6801
L:K	0.4276	0.6217	0.8193	0.3511	0.5448
K:Z	0.0001*	0.5791	0.5931	0.8746	0.7410
Z:X	0*	0.2033	0.0002*	0.0154*	0.0469*
K:X	0.0007*	0.0017*	0.6106	0.0061*	0.3517

Again, we let L stand for LEAD, X for DBS/X^M, K for DBS/K, and Z for the Z-model.

It is interesting to note, moreover, that in MRS, there is a significant difference in performance between DBS/K and K at 10%. Since they differ only in the use of clustering, whatever difference they have would be attributable to clustering alone. The superiority of DBS/X^M over Z in Table 4—which is statistically significant—also counts as further evidence that clustering improves performance of a summarizer when working with the MRS.

A reason, we believe, is this. Since the Z-model selects from a list of sentences ranked globally, i.e., relative to the whole document, it fails to discover sentences which is only relevant to a secondary or marginal concern of the text. It is reasonable to believe that to correctly retrieve B-labeled documents, one needs to pay more attention to sentences which are relevant to secondary subjects of the text. Unlike the Z-model, however, DBS/X^M selects from *locally* ranked groups of sentences and thus would be sensitive to those sentences relevant to marginal concerns of the text, even if they lack relevance to a primary topic.

Another point about DBS/X^M and the Z-model worth mentioning is that their differences in performance get smaller with the compression rate. We presume that this happens because the Z-model selects more of the topically diverse sentences in the text, as the compression rate increases.

Tables 7 and 8 break down performance of DBS/X^M by query type under SRS and MRS, respectively. The general picture is that in either scheme, DBS/X^M performs best on average for queries of type B, and moderately well for queries of type A and C. The results are somewhat contrary to our expectation since finding right documents for a query of type B is supposedly more difficult than that for an A-type query.

Fig. 4 plots performance of LEAD and DBS/X^M against each other. Each panel there displays a query-wise plot of performance of LEAD (x-axis) against that of DBS/X^M (y-axis) at compression rates 10% through 50%. If a point appears above the diagonal line, it indicates that a

Table 7
The break down of average performance of DES/X^M by query type under SRS

Query type	10%	20%	30%
A	0.230	0.230	0.233
B	0.381	0.372	0.399
C	0.245	0.248	0.248
D	0.161	0.162	0.151
E	0.078	0.072	0.074
F	0.040	0.037	0.043

Table 8
The break down of average performance of DBS/X^M by query type under MRS

Query type	10%	20%	30%
A	0.139	0.141	0.145
B	0.273	0.279	0.333
C	0.193	0.197	0.213
D	0.101	0.101	0.116
E	0.081	0.080	0.085
F	0.074	0.070	0.090

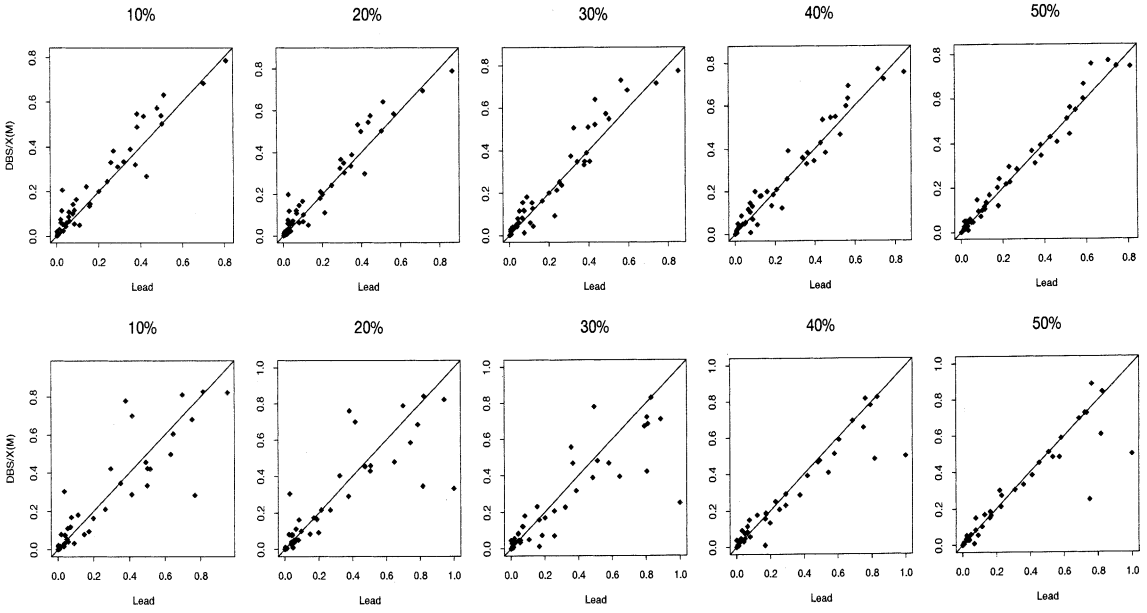

Fig. 4. Comparing performance of Z-model and DBS/X^M against the target retrieval system, which uses full-length documents.

model associated with y-axis performs better at that point than a model associated with x-axis, and worse if a point appears below the line. They break even when a point appears on the line. The top row gives results for SRS, the bottom row for MRS.

Fig. 4 shows that in the region of 10% through 30%, there is a greater variance in performance between the LEAD and DBS/X^M in MRS than in SRS, which is presumably why we get a higher significance level for MRS than SRS. Points apparently settle down along the diagonal line as we go along with the growing compression, indicating that the two systems become increasingly indistinguishable.

Gong and Liu (2001) is an interesting alternative to the summarization paradigm based on clustering. They attempt two different approaches to representing the diversity of contents in a text: one by selecting a representative sentence in such a way that it may not contain any term in a pool of sentences already chosen, and the other by way of singular value decomposition (SVD).

Table 9
Average performance of ATC, ATN, and ANN in SRS

Compression rate	ATC	ATN	ANN
10%	0.072	0.110	0.139
20%	0.062	0.150	0.150
30%	0.113	0.211	0.211
40%	0.182	0.209	0.209
50%	0.221	0.219	0.219

It would be worthwhile to see how their approaches compare against the DBSs. As Gong and Liu (2001) report rather similar results with the non-SVD and SVD approaches, we try here the simpler, non-SVD approach. In this approach, one goes through the following steps to create a summary:

(1) Rank sentences in a text, according to how similar they are to the document they belong to, using some weighting scheme, such as tfidf.
(2) Select a top scoring one and add it to a summary.
(3) Eliminate terms contained in the selected sentence altogether from the document. Return the summary and halt if a predefined number of sentences are collect. If not, go back to step 1.

Below we will look at performance of three summarizers based on the non-SVD approach, namely ATC, ATN, and ANN, which, according to results from Gong and Liu (2001), consistently produce performance better than or equal to SVD-based methods and also are among the best systems reported there. The first and second make use of what they call the *augmented* term weighting scheme, together with idf. [5] However ATC differs from ATN in that the former further normalizes term weights by length of a sentence. ANN uses the augmented term weighting but neither of idf or a normalization factor. Table 9 lists results in the SRS, and Table 10 those in the MRS. In either scheme, it is found that none of the systems ATC, ATN, and ANN, begins to come close in performance to the systems we studied above, including Z and LEAD. This means perhaps that choosing for a summary as lexically distinct sentences as possible is not a good way of representing a text for IR tasks such as one here. We might note that their approach does not allow any lexical overlap between sentences in a summary. We suspect that this may have caused a poor performance of the systems.

To sum up, we studied how the diversity-based methods (DBSs) compare in performance among themselves as well as against some baselines like the Z-model and LEAD in two relevance paradigms, SRS and MRS. It was found that the DBSs differ from others more in MRS than SRS: DBSs consistently produce performance better or equal to the Z-model and LEAD in MRS but not in SRS, indicating that the diversity-based methods are more sensitive to marginally relevant documents than the non-diversity variety.

[5] For a given term i, its augmented weight is defined by: $0.5 + 0.5(\mathrm{tf}(i)/\mathrm{tf}(\max))$, where 'tf' is the term frequency of i in a sentence. The inverse document frequency 'idf' is defined as: $\log(N/n(i))$. N is the total number of sentences in a document, and $n(i)$ denotes the number of sentences in that document that contain i.

Table 10
Average performance of ATC, ATN, and ANN in MRS

Compression rate	ATC	ATN	ANN
10%	0.015	0.109	0.122
20%	0.051	0.152	0.152
30%	0.100	0.191	0.191
40%	0.141	0.234	0.214
50%	0.186	0.227	0.227

6. Modeling human judgments in summarization tasks

In the following, we turn to the question of how well the diversity-based summarizer models (or predicts) subjective judgments humans make when asked which sentence to extract from a text for a summary. In particular, we will explore how the DBSs compare to a supervised approach for summarization, which would be a natural choice, given that the training data is available. [6] In the following, we start off by describing how we had collected human judgments on sentence extraction, and then examine how a supervised approach trained on them fares against DBSs, which are, of course, unsupervised.

7. The decision tree method

We consider here the C4.5 decision tree-based algorithm (Quinlan, 1993) and describe ways in which it could be turned into a variable-length summarizer: we implemented an MDL-based pruning mechanism as we had a promising result from our earlier work (Nomoto & Matsumoto, 2000) that the extension of the decision tree with MDL pruning does improve performance to the degree comparable to AdaBoost.Ml (Freund & Schapire, 1996). See Appendix for technical details about harnessing C4.5 with MDL.

Another modification involved the way the decision tree classifies data. In order to deal with a variable length summary, it was modified to output the probability of class membership rather than a label, which makes it possible to rank sentences according to the probabilistic strength. Let us call a decision tree so modified the *probabilistic decision tree* or simply ProbDT.

Now here is how it works. Given a sentence u, we start by asking ProbDT what class it may belong to. Then it determines a particular leaf node u should be assigned to, as with a decision tree

[6] To our knowledge, no prior work on automatic summarization addressed the question of how supervised and unsupervised approaches compare in performance when they are set to the same task. A general perception seems to be that a learning-based approach works better, since it is able to exploit information about the grand truth provided by humans, which is not available to an unsupervised approach. But it is well known that learning-based approaches to summarization inherently suffer from the problem of "lack of uniqueness of a summary" (Mani et al., 1998) and the cost of creating a corpus large enough to train algorithms on. Besides, the superiority of the learning-based approach to the unsupervised approach is yet to be proved.

Against this background, the present work sets out to investigate how the two varieties of approaches compare, by directly matching one against the other in the same summarization task.

Table 11
Probabilistic classification with the decision tree

$$P(Select|\vec{u}, \mathrm{DT}) = \alpha\left(\frac{\text{the number of "Select" sentences at } t(\vec{u})}{\text{the total number of sentences at } t(\vec{u})}\right)$$

'$P(Select|\vec{u}, \mathrm{DT})$' reads like "the probability that a sentence u is selected as part of a summary, given DT." $\vec{u}$ is a vector representation of sentence u. α is a smoothing function, e.g., Laplace's rule. $t(\vec{u})$ is some leaf node assigned to $\vec{u}$ by DT.

of the usual variety. But this is where the similarity ends. Instead of reporting a class label of the majority of cases on the node, ProbDT now reports the probability that u is to be included in a summary, using the formula given in Table 11. Thus for instance, given sentences u_1 and u_2, ProbDT enables us to say things like "u_1 is more likely to be included in a summary than u_2." Notice that with ProbDT, one could create a summary of any length, based on a probability ranking among sentences.

A feature description of a sentence contains information such as length, position, and linguistic cues. Listed in the following are a set of features used to encode a sentence. (Some of them are devised to specifically deal with Japanese as we use a Japanese corpus for evaluation.)

<LocSen> The location of a sentence X defined by:

$$\frac{\#S(X) - 1}{\#S(\text{Last\_Sentence})}.$$

'$\#S(X)$' denotes an ordinal number indicating the position of X in a text, i.e. $\#S(k\text{th\_sentence}) = k$. 'Last_Sentence' refers to the last sentence in a text. LocSen takes values between 0 and $(N-1)/N$. N is the number of sentences in the text.

<LocPar> The location of a paragraph in which a sentence X occurs given by:

$$\frac{\#\mathrm{Par}(X) - 1}{\#\text{Last\_Paragraph}}.$$

'$\#\mathrm{Par}(X)$' denotes an ordinal number indicating the position of a paragraph containing X. '# Last_Paragraph' is the position of the last paragraph in a text, represented by the ordinal number.

<LocWithinPar> The location of a sentence X within a paragraph in which it appears.

$$\frac{\#S(X) - \#S(\text{Par\_Init\_Sen})}{\mathrm{Length}(\mathrm{Par}(X))}.$$

'Par_Init_Sen' refers to the initial sentence of a paragraph in which X occurs, 'Length(Par(X))' denotes the number of sentences that occur in that paragraph. LocWithinPar takes continuous values ranging from 0 to 1. A paragraph initial sentence would have 0 and a paragraph final sentence 1.

<LenText> The text length in Japanese character i.e. *kana, kanji.*

<LenSen> The sentence length in *kana/kanji.*

Following some work in Japanese linguistics, we assume that sentence endings are relevant for identifying semantic relations among sentences. Some of the sentence endings refer to inflectional categories of verbs such as PAST/NON-PAST, INTERROGATIVE, and others to morphological categories like nouns and particles, e.g. question-markers. Along with Ichikawa (1990), we identified a set of sentence-ending cues and marked a sentence as to whether it contains a cue from the set. [7] Included in the set are inflectional forms of the verb and the verbal adjective, PAST/NON-PAST, morphological categories such as COPULA, and NOUN, parentheses (quotation markers), and sentence-final particles such as -ka. We use the following attribute to encode a sentence-ending form.

<EndCue> encodes one of sentence-ending forms described above. It is a discrete valued feature. The value ranges from 0 to 6.

Finally, one of two class labels, 'Select' and 'Don't Select', is assigned to a sentence, depending on whether it is to be included in a summary. The 'Select' label is for a sentence which would be included in a summary, and the 'Don't Select' label for other cases.

8. Evaluation

8.1. The test data

We asked 112 Japanese subjects (students at the graduate and undergraduate level) to extract 10% sentences in a text which they consider most important in making a summary. The number of sentences to extract varied from two to four, depending on the length of a text. The age of subjects varied from 18 to 45. We used 75 texts from three different genres (25 for each genre); column, editorial and news report. Texts were of about the same size in terms of character counts and the number of paragraphs, and were selected randomly from articles that appeared in a Japanese financial daily (Nihon-Keizai-Shimbun-Sha, 1995). Table 12 provides some statistics on the test corpus. We assigned, on average, seven subjects to each text for an extraction task.

The kappa agreement among subjects was as modest as 0.25. [8] The result is in a way consistent with Salton et al. (1999), which report a low inter-subject agreement on paragraph extracts from encyclopedias, and with Gong and Liu (2001), who also report low inter-subject agreement in newswire summarization. While there are some work (Jing et al., 1998; Marcu, 1999a) which do report high agreement rates, their success may be attributed to particularities of texts used, as suggested by Jing et al. (1998), as well as the particular agreement metric they used, i.e., percent agreement (see footnote 1 for the discussion of its problems). Thus, the question of whether it is possible to establish an ideal summary based on agreement is far from settled, if ever. In the face of this, it would be interesting and perhaps more fruitful to explore another view on summary: that the variability of a summary is the norm rather than the exception.

[7] Word tokens are extracted by using CHASEN, a Japanese morphological analyzer which is reported to achieve the accuracy rate of over 98% (Matsumoto et al., 1999).

[8] The kappa statistic represents one of measures of agreement for nominally scaled data, which takes the form: $(P(A) - P(E))/(1 - P(E))$ It is the ratio $P(A)$ of pairwise agreement among subjects adjusted for the expected agreement ratio $P(E)$ (Siegel & Castellan, 1988).

Table 12
The test data

Text type	#sent	#par	#articles
Column	17.04	4.28	25
Editorial	22.32	7.48	25
News	17.60	6.40	25

The test set consists of a random selection of 25 texts from three genres, column, editorial, and news, all coming from a same new paper source. '#sent' indicates the average number of sentences in a text, '#par' the average number of paragraphs per text.

Table 13
#sent

#sent	#agr $\geqslant 1$	Unmarked
1424	707	717

#sent denotes the number of sentences in the test data (Table 12). #agr $\geqslant 1$ denotes the number of those marked as relevant by one or more subjects.

One consequence of the view is that a sentence marked as important by any *one* of the subjects becomes a potential summary sentence, however marginal it may be. In experiments, we decided to go along with this view, and consequently regarded as correct, every sentence marked as important by one or more subjects. Table 13 gives some idea of how many of the sentences in the test data collected one or more votes from judges for membership in a summary.

8.2. Procedure

We evaluated performance of the decision tree-based summarizer by 10-fold cross-validation: where we split the test data into 10 blocks, reserve one for the test and use others for training. Table 14 lists performance in *classification,* averaged over 10 folds, of the decision tree. Note that Table 14 is concerned with the accuracy of classification, not summarization. As shown in the table, the accuracy of classification for the test data was 59% without the pruning but increased to 61% with MDL. Though the figures are low, either is well above the baseline performance, which is 51%.

Table 14
Tenfold cross-validated error rates for C4.5

	Not pruned	MDL-pruned
Train	0.093 (527)	0.383 (4.6)
Test	0.406 (527)	0.385 (4.6)
Baseline	0.49	

A figure in parentheses represents the size (the number of nodes) of a decision tree generated. The bottom row shows a baseline performance, also in error rate.

A figure in parentheses indicates the size of a generated tree. As is clear from the table, MDL contributes to the significant reduction of the tree almost by a factor of 100, while still improving performance.

As for a diversity-based summarizer DBS/X^M, the only training involved was the estimation of the document frequency of a term used in sentence weighting, which was done using a collection of 14,391 articles selected from the same news paper corpus the test data came from. The summarizer was tested on the entire test data. Also for the purpose of comparison, we considered DBS/K, which as we might recall is identical to DBS/X^M, except that it makes use of the regular K-means instead of X^M-means, where the number of K is given in proportion to a compression rate. For example, at compression rate of 0.2, K was set to 2 for a text of 10 sentences.

Also enlisted as a baseline were the LEAD system and the Z-model. The latter system, as we might recall from the BMIR-J2 experiments, operates by a simple heuristic of selecting top ranking sentences in the text based on tfidf weights. Note that a comparison of the Z-model against DBS/K should reveal effects of the diversity functionality on summarization, since the only difference between the former and the latter is that the former lacks the diversity component which the latter has. The LEAD system, which is again identical to that used in the BMIR-J2 experiments, picks up the leading portion of a text as a summary. DBS/K, DBS/X^M, Z and LEAD were all tested on the entire test set.

9. Results and discussion

Before we go into the results, let us go over some notations we use throughout the section. We mean by a term 'K1' a test data where for each positive, i.e. summary-worthy, sentence, there is at least one judge voting for its inclusion in a summary, and similarly for 'K2' to 'K5' where one needs at least two to five votes from a pool of judges to get a sentence marked as positive. Also, κ indicates a ratio of agreement in κ.

Table 15 shows how the summarizers fare on K1 at compression rates 10% through 50%. [9] It is somewhat surprising to find that the decision tree-based summarizer DT/MDL consistently falls behind DBS/X^M though they come closer as the compression rate increases. [10]

A comparison among DBS/X^M, DBS/K, Z, and LEAD reveals significant effects of the diversity component on performance when the test data contains a large variance in judgment. DBS/X^M outperforms Z and LEAD by a large margin at every compression level. While the

[9] As usual, we give F-measure as: $F = 2PR/(P+R)$ for precision P and recall R.

[10] One thing to note is that recall is given here as the ratio of the number of correct (or summary-worthy) sentences retrieved to that of correct sentences found in a text. But this gives an unfairly low recall value for a summary at lower compression rates such as 20% and 30%. To see this, imagine that you want to compress a text at 20%, 50% of which are marked as "Select." But then it is not possible for a summary 20% the size of its source to contain all of the marked sentences. This would put an upper bound for recall at 40%. One remedy would be to use a modified denominator D for recall:

$$D = \begin{cases} \alpha|T| & \text{if } \alpha|T| < |C_T|, \\ |C_T| & \text{otherwise,} \end{cases}$$

where α is a compression rate, $|T|$ is the length of a text T (in sentences), and C_T is the number of "Select" sentences in T.

Table 15
Performance on K1 at compression rates from 10% to 50%

Rate	MDL/DT	DBS/K	DBS/$X(M)$	Z	LEAD
10%	0.220	0.167	0.340	0.104	0.166
20%	0.353	0.346	0.431	0.231	0.355
30%	0.453	0.436	0.495	0.341	0.433
40%	0.535	0.502	0.547	0.436	0.500
50%	0.585	0.554	0.606	0.511	0.527

DT/MDL denotes a summarizer based on C4.5 with MDL. DBS/$X(M)$ denotes the diversity-based summarizer (DBS) with X^M-means. DBS/K stands for DBS with the regular K-means. Z represents the Z-model summarizer. Performance figures are in F-measure. $\kappa = 0.253$.

difference between DBS/K and LEAD is rather marginal, the fact that DBS/K is performing better than Z, which is DBS/K modulo K-means, demonstrates the effectiveness of the diversity functionality. Note that the superior performance of DBS/X^M over DBS/K implies the superiority of X^M- over K-means.

While Table 15 leads us to believe that DBS/X^M better models K1 than the C4.5 based system, it would be interesting to see what happens if we look beyond K1 and consider data sets K2–K5. Or more generally, it is worthwhile to ask how agreement on labeling among subjects affects performance of the systems.

To this end, we ran experiments on K2 through K5, with the same setup as K1, except that agreement on extracts varied among the data sets. The performance scores for DT/MDL are averaged over 10-fold runs as before. The rest of the systems are tested on the whole training data. The results are listed in Tables 16–19. Also listed are the corresponding kappa rates.

While performance of the summarizers except LEAD generally degrades as agreement increases, the tables make it clear that DBS/X^M is one of those most severely affected by increasing agreement. DBS/X^M leads others in K1 and K2. But beyond K2, it starts lagging behind MDL/DT and LEAD. The LEAD system, in contrast, picks up at K3 and remains a top performing system beyond that.

Now Table 20 gives rankings of the systems for K1–K5 at a compression rate of 10%. It gives a clear picture of how DBS/X^M, starting out the first runner, falls behind LEAD and MDL/DT as the compression increases. But the way top performing systems alternate illuminates one important (albeit domain-specific) property of human judgments: that highly agreed upon sentences are more likely to be found in the beginning of the text than elsewhere, and poorly agreed upon

Table 16
K2 ($\kappa = 0.381$)

Rate	MDL/DT	DBS/K	DBS/$X(M)$	Z	LEAD
10%	0.272	0.146	0.317	0.095	0.228
20%	0.343	0.299	0.389	0.214	0.387
30%	0.366	0.332	0.423	0.288	0.406
40%	0.397	0.386	0.446	0.341	0.409
50%	0.409	0.401	0.444	0.380	0.392

Table 17
K3 ($\kappa = 0.500$)

Rate	MDL/DT	DBS/K	DBS/$X(M)$	Z	LEAD
10%	0.292	0.115	0.258	0.101	0.302
20%	0.293	0.220	0.316	0.164	0.344
30%	0.274	0.245	0.340	0.219	0.337
40%	0.294	0.281	0.343	0.250	0.316
50%	0.297	0.277	0.321	0.271	0.284

Table 18
K4 ($\kappa = 0.600$)

Rate	MDL/DT	DBS/K	DBS/$X(M)$	Z	LEAD
10%	0.313	0.119	0.216	0.091	0.274
20%	0.266	0.195	0.253	0.150	0.345
30%	0.219	0.208	0.265	0.188	0.302
40%	0.227	0.223	0.250	0.206	0.268
50%	0.229	0.214	0.227	0.215	0.227

Table 19
K5 ($\kappa = 0.711$)

Rate	MDL/DT	DBS/K	DBS/$X(M)$	Z	LEAD
10%	0.208	0.125	0.181	0.056	0.340
20%	0.175	0.168	0.204	0.109	0.338
30%	0.136	0.171	0.199	0.142	0.261
40%	0.131	0.194	0.182	0.141	0.218
50%	0.122	0.165	0.165	0.153	0.179

Table 20
The summarizers' rankings for K1 through K5 at 10%

Data set	First	Second	Third	Fourth	Fifth
K1	DBS/$X(M)$	MDL/DT	DBS/K	LEAD	Z
K2	DBS/$X(M)$	MDL/DT	LEAD	DBS/K	Z
K3	LEAD	MDL/DT	DBS/$X(M)$	DBS/K	Z
K4	MDL/DT	LEAD	DBS/$X(M)$	DBS/K	Z
K5	LEAD	MDL/DT	DBS/$X(M)$	DBS/K	Z

sentences are scattered across the entire text. That is why the DBS/X^M performs well on K1 to K2, and badly on K3 to K5, while the opposite is true for LEAD, which is better off on K3 to K5.

The results above suggest some limitation of DBS/X^M in modeling human judgments in summarization tasks, namely, its inability to identify sentences highly agreed upon for their membership in a summary ($\kappa \geqslant 0.5$). The fact that DT/MDL does a far better job of finding them gives us some reason to believe that there are indeed some regularities among highly agreed upon

Table 21
Performance of ATC, ATN and ANN at varying compression rates on K1

Compression rate	ATC	ATN	ANN
10%	0.054	0.135	0.166
20%	0.147	0.270	0.285
30%	0.220	0.361	0.379
40%	0.300	0.449	0.461
50%	0.359	0.508	0.514

sentences which need to be captured but LEAD and DBSs fail to capture. (Notice that MDL/DT performs best on K4 at 10%.) This observation in turn suggests a way of potentially improving DBS/X^M: which is to make use of some learning-based ranking scheme instead of using the tfidf-based ranking.

Before leaving the section, we look at how ATC, ATN and ANN (Gong & Liu, 2001) [11] compare to DBSs and MDL/DT on K1. The results are listed in Table 21. ANN, a top performing system among the three, falls far behind DBS/X^M, MDL/DT, and LEAD, though performing significantly better than the Z, which again suggests that a clustering like device can help boost performance of a summarizer.

10. Conclusions

In the first segment of the paper, we have presented a new summarization scheme where evaluation does not rely on matching extracts against human-made summaries but measuring the loss of information in extracts in terms of retrieval performance. Under this scheme, the diversity-based summarizer (DBS/X^M) was found to be superior to a tfidf-based summarizer (Z) as well as the LEAD model. In particular since DBS/X^M differs from the Z-model only in the use of the diversity component, the difference in performance could be attributed to the sole effects of selecting diverse sentences. We have seen that improvement by the diversity component is more prominent under the MRS than the SRS. But this is just what is expected of performance of the diversity-based approach, because under the MRS, the system needs to be more sensitive to marginally relevant sentences.

In the second segment, we conducted experiments using human judgments in extraction tasks, and found an interesting result that a diversity-based summarizer based on X^M-means was comparable to and even superior to a supervised approach at some compression rates. We also studied whether the extent of agreement among human subjects is in any way linked with performance of summarizers, and empirically demonstrated that this is indeed the case: when the data exhibit high agreement ($\kappa \geqslant 0.5$), the diversity-based summarizer DBS/X^M performs poorly compared to the learning-based system DT/MDL, and particularly so when the compression rate is low. We suspected that the reason has to do with possible regularities in sentences with high agreement, which the tfdif-based ranking scheme, i.e., reduce-redundancy, is unable to exploit.

[11] See Section 5 for details.

In light of the results, one of the issues future work needs to address is to explore ways of embedding a supervised learning-based ranking scheme in the clustering framework. It would also be interesting to look into how much of the results here carry over to the Web pages, which are vastly different in structure/organization from the data we worked with in the paper.

Appendix A

In harnessing a decision tree with MDL, we follow work by Yamanishi (1997) and Rissanen (1997), where MDL is used to choose among possible prunings of a decision tree.

Consider a decision tree **T**, with a set **H** of subtrees of **T** and a set **A** of attributes. Assume further a subtree of **T** to be a model. Then the problem of finding a best pruning for **T** can be tuned into that of finding a best model in terms of MDL, which can be solved by calculating the MDL for each subtree in the **T** and choosing one with the least MDL. A systematic way of doing this is to work on the decision tree bottom up, reducing a subtree to a node if it produces a tree with a smaller MDL score.

The description length of data $X = x^1 \cdots x^n$ (i.e. sentences) under a model M, is defined by:

$$L(x^n : M) = L(x^n|M, \theta) + L(\theta|M) + L(M),$$

where $L(x^n|M, \theta)$ denotes a description length of the data $x^1 \cdots x^n$ under a model M and a set of parameters θ, $L(\theta|M)$ denotes a description length of parameters given M, and $L(M)$ is a description length of a model M itself. The description length of data is usually equated with their maximum log-likelihood estimate.

We note that MDL is about finding a best way of coding data for transmission over a communication channel, and in particular strives to minimize the coding length of X and parameters that define the probability distributions for X.

Given a parameter vector $\theta = (\theta_1, \theta_2, \ldots, \theta_m)$ for M, the number of bits one needs to encode X is: $-\log P_\theta(X)$. But how do we encode θ? Since it involves real numbers, some of its element parameter may not have a finite representation, which poses a serious problem, for it means that θ cannot be transmitted over a communication channel. MDL's answer to this is to split a parameter space $\Theta(m)$ (of which θ is a member) into a finite number of (multi-dimensional) rectangular solids, pick some finitely representable (or rounded) number from each solid, and let it represent a solid it is from. Now this prompts us to ask, how large each solid might be? As is well known in the MDL literature, it suffices that each edge (or dimension) for a given solid is just as long as $O(1/\sqrt{N})$, where N is the size of data. So it is safe to say that the probability of observing a particular rounded θ, or a rectangular solid it represents, in $\Theta(m)$, is $1/\sqrt{N}^m$, whose coding length should be $(m/2)\log N$. We note that, since parameters in θ sum to 1, we need to be working with only $m - 1$ parameters.

We define the quantity $I(u)$ at a particular node u as the sum of the description length of data and that of parameters:

$$I(u) = -\sum_{j=1}^{K} F_u(j) \log \widehat{P}_u(j) + \frac{k}{2} \log N,$$

Table 22
A pruning algorithm based on MDL

```
MDL-Prune(u)
begin
if u is a leaf then
  set L(u) = − log P0 + I(u)
  return L(u)
else
  L(u) = Σ_{v∈D(u)} MDL-Prune(v)
  where D(u) is a set of daughter nodes of u.
  if − log P0 + I(u) ⩽ − log P1 + I(u) + L(u) then
    remove every v ∈ D(u)
  endif
  return min{− log P0 + I(u), − log P1 + I(u) + L(u)}
endif
end
```

where N is the number of cases (examples) that reached u, $F_u(j)$ is the frequency of class j at u, K is the number of classes, $\widehat{P}_u(j)$ is the maximum likelihood estimate of the class j at u, k is the number of free parameters at u, so this would be $K-1$.

The model length of u is given as follows:

$$l_u(m)\begin{cases} -\log P_0 & \text{if } u \text{ is a leaf,} \\ -\log P_1 + l_u(A) & \text{otherwise,} \end{cases}$$

where $l_u(A)$ is the length of coding attributes at u; $l_u(A) = \log |A|$, for a set A of attributes. If the attribute is continuous-valued, then the cost of encoding a threshold, $\log(r)$, is added to $l_u(A)$, where r is the number of possible thresholds at u. P_1 is the probability of observing a non-terminal node in $\mathbf{T}$ and P_0 the probability of observing a leaf in $\mathbf{T}$. Note that $P_1 + P_0 = 1$.

Then the total description length of u is:

$$L(u:m) = I(u) + l_u(m).$$

For any subtree rooted at u in $\mathbf{T}$, its description length $L(u)$ is either: $-\log P_0 + I(u)$ if u is a leaf, or $-\log P_1 + I(u) + \sum_{v\in D(u)} L(v)$, where $D(u)$ is a set of daughter nodes of u.

A subtree with the minimum description length could be found by the following procedure: start with leaves in $\mathbf{T}$ and at each node one visits, prune a subtree rooted at that node if the following condition is met:

$$-\log P_0 + I(u) \leqslant -\log P_1 + I(u) + \sum_{v\in D(u)} L(v)$$

and continue to do so until one reaches the root. A precise algorithm is given in Table 22.

References

Berger, A., & Mittal, V. O. (2000). Query-relevant summarization using FAQs. In *Proceedings of the 38th annual meeting of the association for computational linguistics* (pp. 294–301). Hong Kong.

Bradley, P. S., & Fayyad, U. M. (1998). Refining initial points for *k*-means clustering. In *Proceedings of the fifteenth international conference on machine learning (ICML98)* (pp. 91–99). Morgan Kaufmann.

Carbonell, J., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21th annual international ACM/SIGIR conference on research and development in ir*. Melbourne, Australia.

Chuang, W., & Yang, J. (2000). Text summarization by sentence segment extraction using machine learning algorithms. In T. Terano, H. Liu, & A. L. P. Chen (Eds.), *Knowledge discovery and data mining* (pp. 454–457). Springer.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (second ed). John Wiley & Sons, Inc.

Edmundson, H. P. (1969). New Method in Automatic Abstracting. *Journal of the ACM, 16*(2), 264–285.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the thirteenth international conference on machine learning*.

Gong, Y., & Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM/SIGIR conference on research and development*. ACM-Press, New Orleans.

Ichikawa, T. (1990). *Bunshôron-gaisetsu*. Tokyo: Kyôiku-Shuppan.

Jing, H., Barzilay, R., McKeown, K., & Elhadad, M. (1998). Summarization evaluation methods: experiments and analysis. In *AAAI symposium on intelligent summarization*. Stanford Univesisty, CA.

Katz, S. M. (1996). Distribution of content words and phrases in text and language modelling. *Natural Language Engineering, 2*(1), 15–59.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. In *Proceedings of the fourteenth annual international ACM/SIGIR conference on research and developmnet in information retrieval* (pp. 68–73). Seattle.

Li, H. (1998). *A probabilistic approach to lexical semantic knowledge acquisition and structural disambiguation*. Unpublished doctoral dissertation, University of Tokyo, Tokyo.

Luhn, H. P. (1999). The automatic creation of literary abstracts. In I. Mani, & M. T. Maybury (Eds.), *Advances in automatic text summarization* (pp. 15–21). The MIT Press (reprint).

Mani, I., House, D., Klein, G., Hirshman, L., Obust, L., Firmin, T., Chrzanowski, M., & Sundheim, B. (1998). *The TIPSTER SUMMAC Text Summarization Evalutation Final Report* (Tech. Rep.). Virginia, USA: MITRE.

Marcu, D. (1997). The Rhetorical Parsing of Natural Language Texts. In *Proceedings of the 35th annual meetings of the association for computational linguistics and the 8th European chapter of the association for computational linguistics* (pp. 96–102). Madrid, Spain.

Marcu, D. (1999a). Discourse trees are good indicators of importance in text. In I. Mani, & M. T. Maybury (Eds.), *Advances in automatic text summarization* (pp. 123–136). The MIT Press.

Marcu, D. (1999b). The automated construction of large-scale corpora for summarization research. In *Proceedings of the 22nd international ACM/SIGIR conference on research and development in informational retrieval* (pp. 137–144). Berkeley.

Matsumoto, Y., Kitauchi, A., Yamashita, T., & Hirano, Y. (1999). *Japanese morphological analysis system ChaSen version 2.0 manual* (Tech. Rep.). Ikoma: NAIST. (NAIST-IS-TR99008).

Morris, A. H., Kasper, G. M., & Adams, D. A. (1999). The effects and limitations of automated text condensing on reading comprehension performance. In I. Main, & M. T. Maybury (Eds.), *Advances in automatic text summarization* (pp. 305–323). The MIT Press.

Nichi-Gai Associates. (1995). *CD-Mainichi Shimbun '94: Data shu*. CD-ROM.

Nihon-Keizai-Shimbun-Sha. (1995). *Nihon Keizai Shimbun 95 nen CD-ROM ban*. CD-ROM. (Tokyo, Nihon Keizai Shimbun, Inc.).

Nomoto, T., & Matsumoto, Y. (1997). Data Reliability and Its Effects on Automatic Abstracting. In *Proceedings of the fifth workshop on very large corpora*. Beijing/Hong Kong, China.

Nomoto, T., & Matsumoto, Y. (2000). Comparing the minimum description length principle and boosting in the automatic analysis of discourse. In *Proceedings of the seventeenth international conference on machine learning* (pp. 687–694). Stanford University: Morgan Kaufmann.

Nomoto, T., & Matsumoto, Y. (2001). A new approach to unsupervised text summarization. In *Proceedings of the 24th international ACM/SIGIR conference on research and development in informational retrieval*. New Orleans: ACM.

Pelleg, D., & Moore, A. (2000). *X*-means: Extending *K*-means with efficient estimation of the number of clusters. In *Proceedings of the seventeenth international conference on machine learning (ICML2000)* (pp. 727–734). Stanford University: Morgan Kaufmann.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann.

Radev, D. R., Jing, H., & Budzikowska, M. (2000). Centroid-based summarization of multiple documents: sentence extration, utility-based evaluation, and user studies. In *Proceedings of the ACL/NAAL workshop on summarization*. Seattle, WA.

Rissanen, J. (1997). Stochastic complexity in learning. *Journal of Computer and System Sciences, 55*, 89–95.

Salton, G., Singhal, A., Mitra, M., & Buckley, C. (1999). Automatic text structuring and summarization. In I. Mani & M. T. Maybury (Eds.), *Advances in automatic text summarization* (pp. 342–355). The MIT Press (reprint).

Siegel, S., & Castellan, N. J. (1988). *Nonparametric statistics for the behavioral sciences* (second ed). McGraw-Hill.

Sparck Jones, K., & Gallier, J. R. (1995). *Evaluating natural language processing systems: an analysis and review*. Springer.

Yamanishi, K. (1997). Data compression and learning. *Journal of Japanese Society for Artificial Intelligence, 12*(2), 204–215 (in Japanese).

Zechner, K. (1996). Fast generation of abstracts from general domain text corpora by extracting relevant sentences. In *Proceedings of the 16th international conference on computational linguistics* (pp. 986–989). Copenhagen.