Data Reliability and Its Effects on Automatic Abstracting

Tadashi Nomoto
Advanced Research Laboratory
Hitachi Ltd.
2520 Hatoyama Saitama, 350-03 Japan
nomoto@harl.hitachi.co.jp

Yuji Matsumoto
Nara Institute of Science and Technology
8916-5 Takayama Ikoma Nara, 630-01 Japan
matsu@is.aist-nara.ac.jp

Summary

We discuss a particular approach to automatic abstracting, where an abstract is created by extracting important sentences from a text. A primary purpose of the paper is to demonstrate that the reliability of human supplied annotations on corpora has crucial effects on how well an automatic abstracting system performs. The corpus is developed through human judgements on possible summary sentences in a text. The reliability of human judgements is evaluated by the kappa statistic, a reliability metric standardly used in behavioral sciences. The C4.5 decision tree method (Quinlan, 1993) is used to build a extraction model. We demonstrate that there is a positive correlation of data reliability with a performance of automatic abstracting, and show results indicating that the reliability of human provided data is crucial for improving the performance of automatic abstracting.

1. INTRODUCTION

The traditional approach to automatic abstracting aims at providing a reader with fast access to documents by facilitating a judgement on their relevance to his or her information needs. Another possible use of automatic abstracting can be found in works such as Bateman and Teich (1995) and Alexa et al. (1996), where computer-generated abstracts are used for the editing purposes.

In this paper, we discuss an approach to automatic abstracting where an abstract is created by extracting sentences from a text that are indicative of its content. In particular, the paper focuses on creating abstracts of Japanese newspaper texts. An approach to abstracting by extraction typically makes use of a text corpus with labelled extracts, indicating which sentence is a summary extract. However, as far as we know, no question has ever been raised on the empirical validity of the extracts used. Usually, extracts are manually supplied by

Table 1: Statistics on Corpus

Text Type	Length in char.	# Par.	# Articles
Column	about 640	4-5	352
Editorial	900-1100	6-9	131
News Report	800-1000	6-9	147

the author himself (Watanabe, 1996) or by someone else (McKeown and Radev, 1995) (as in the TIPSTER Ziff-Davis corpus). Or one takes a roundabout way to identify extracts in a text through a human-supplied abstract (Kupiec et al., 1995). In the paper, we will propose a method for identifying summary extracts in a way that allows objective justification. We will do this by examining how humans perform on summary extraction and evaluating the reliability of their performance, using the kappa statistic, a metric standardly used in the behavioral sciences (Jean Carletta, 1996; Sidney Siegel and N. John Castellan Jr., 1988). Based on summary extracts supplied by humans, we construct a collection of texts annotated with information on sentence importance. They will be used as training and test data for a decision tree approach to abstracting, which we adopt in the paper (Quinlan, 1993). In a decision tree approach, the task of extracting summary sentences is treated as a two-way classification task, where a sentence is assigned to either "yes" or "no" category, depending on its likelihood of being a summary sentence. The merit of a decision tree method is that it provides a generic framework in which to combine knowledge from multiple sources, a property necessary for automatic abstracting where information from a single source alone often fails to determine which sentence to extract.

2. METHODOLOGY

2.1. Collecting Data on Summary Extraction by Humans

We conducted experiments with humans to collect data on how they perform on the sentence extraction task. We asked 112 naive subjects (students at graduate and undergraduate level) to extract 10 % of sentences in a text which they consider most important in making its summary. The number of extractions varied from two to four, depending on the length of a text. The age of subjects varied from 18 to 45. The experiments used 75 texts from three different text categories (25 for each category); COLUMN, EDITORIAL and NEWS REPORT. The 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 economics daily in 1995 (Nihon-Keizai-Shimbun-Sha, 1995). Table 1 provides some statistics on the corpus from which extraction tests are constructed. A single test material consists of three extraction problems, each with a text from a different category. Though 85 of 112 subjects were assigned to one test, due to the lack of enough subjects, we had to ask the remaining 27 subjects to work on

five tests. On the average, each test had about 7 subjects assigned to it.

2.2. Measurement of Reliability

The Kappa Statistic Following Jean Carletta (1996), we use the kappa statistic (Sidney Siegel and N. John Castellan Jr., 1988) to measure degree of agreement among subjects. The reason for choosing the kappa over other measures of agreement (Passonneau and Litman, 1993) derives from our interest in discovering a relationship between the reliability or quality of data (as quantified by some metric) and the performance of automatic abstracting. As aptly pointed out in Jean Carletta (1996), agreement measures proposed so far in the computational linguistics literature has failed to ask an important question of whether results obtained using agreement data are in any way different from random data. It has been left unclear just how high level of agreement among subjects needs to be achieved before reliably using data. It could be the case that data with high agreement may still be too noisy to use for a task for which they were collected.

We assume that the kappa coefficient gives a suitable way of measuring the reliability of data, where we take reliability to mean reproducibility of data, or the degree to which data are reproduced under different circumstances, with different coders (Krippendorff, 1980). The kappa coefficient (K) of agreement measures the ratio of observed agreements to possible agreements among a set of raters on category judgements, correcting for chance agreement:

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{1}$$

where $P(A)$ is the proportion of the times that raters agree and $P(E)$ is the proportion of the times that we would expect them to agree by chance. $K = 1$ if there is complete agreement among the raters. $K = 0$ if there is no agreement other than that which is expected by chance. Consider a set of k raters and a group of N objects, each of which is to be assigned to one of m categories. Each of the raters assigns each object to one category. We represent the assignments data as an $N \times m$ matrix (Table 2), where the value (n_{ij}) at each $\text{cell}_{i,j}$ ($0 < i \leq N$, $0 < j \leq m$) denotes the number of raters assigning the ith object to the jth category. Let C_j be the total number of times that objects are assigned to the jth category, i.e., $C_j = \sum_{i=1}^{N} n_{ij}$. S_i measures the proportion of pairwise agreements among the raters on category assignments for a particular object i. S_i gives a measurement of agreement among raters on decisions regarding which category a given object i is to be assigned to. Let us define S_i by Def. 2.

$$S_i = \frac{\sum_{j=1}^{m} \binom{n_{ij}}{2}}{\binom{k}{2}} = \frac{1}{k(k-1)} \sum_{j=1}^{m} n_{ij}(n_{ij} - 1) \tag{2}$$

Table 2: Assignments Matrix

	1	2	⋯	j	⋯	m	
1	n_{11}	n_{11}	⋯	n_{1j}	⋯	n_{1m}	S_1
2	n_{21}	n_{22}	⋯	n_{2j}	⋯	n_{2m}	S_2
⋮				⋮			⋮
i	n_{i1}	n_{i2}	⋯	n_{ij}	⋯	n_{im}	S_i
⋮				⋮			⋮
N	n_{N1}	n_{N2}	⋯	n_{Nj}	⋯	n_{Nm}	S_N
	C_1	C_2	⋯	C_j	⋯	C_m	

Table 3: A hypothetical agreement table

	1	2	⋯	j	⋯	m
a	0	0	⋯	$2m$	⋯	0
b	2	2	⋯	2	⋯	2

For each object i, agreement frequencies n_{ij} must sum up to k, the total number of raters. Note that $0 < S_i \leq 1$. $S_i = 1$ when there is total agreement among the raters for a given category j on the ith row. Suppose that we asked $2m$ raters to assign two objects a and b to one of m categories and found results as in Table 3. For a, there is a complete agreement on the object's category, while for b, decisions are spread evenly over m categories. Since $S_a = 1$ and $S_b = 1/(2m-1)$ $(m > 1)$, we have $S_a > S_b$.

The proportion $P(A)$ of the times that the raters agree is given as the average of S_i across all objects (Def. 3).

$$P(A) = \frac{1}{N}\sum_{i=1}^{N} S_i \tag{3}$$

The expected probability that a category is chosen at random is estimated as $p_j = C_j/(N \cdot k)$. Then, the probability that any two raters agree on the jth category by chance would be p_j^2. $P(E)$ is defined as the sum of chance agreement for each category (Def 4), representing the overall rate of agreement by chance.

$$P(E) = \sum_{j=1}^{m} p_j^2 \tag{4}$$

The values of $P(A)$ and $P(E)$ are then combined to give the kappa coefficient K.

Evaluation Judgements produced by subjects on a summary extraction task can be cast into an assignments matrix in a number of different ways. (Note that a single extraction

Table 4: A matrix representation of a hypothetical example

OBJECTS	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	# subjects
1	3	-	-	-	3	1	1	1	-	-	9
2	-	2	1	2	-	1	-	2	1	-	9
3	-	-	1	-	-	2	-	2	1	3	9
C	3	2	2	2	3	4	1	5	2	3	

Table 5: Kappa coefficients for judgements on sentence importance

Text Type	K	# Texts	# Raters
COLUMN	0.122	25	183
EDITORIAL	0.156	25	184
NEWS REPORT	0.255	25	183

task consists of extracting a specified number of sentences from one text.) We adopt here a representation scheme where we take N to be the number of choices made by a subject for a text and m to be the number of sentences in that text.[1] (Note that since we asked a subject to choose 10% of sentences in the text, the number of extractions made for each text depends entirely on the text's length, but the number of extractions from a given text should be the same across subjects.) Imagine for instance that nine subjects are asked to extract three most important sentences from a text with ten sentences. Under the scheme here, the resulting data could be represented as a matrix of height $N = 3$ and width $m = 10$ with $k = 9$ like one in Table 4, where the first object is thought of as an earliest occurring sentence a subject considers most important, the second object as a second earliest occurring sentence a subject considers most important, and the third object as a third earliest sentence a subject considers most important.

It is important to notice that a matrix is constructed for each extraction task and the agreement coefficient K is determined for each task, not for each sentence in the text. Table 5 lists the K values for subjects' judgements on sentence importance, averaged over texts. The number of subjects assigned to one extraction task varied from 4 to 9. 96% of the time, we had over 6 subjects working on a same task. The average number of subjects per text was 7.33.[2] We find in Table 5, however, that there is only marginal agreement among subjects.

[1] Another possibility is to represent the data as an $N \times m$ matrix of height N=the number of sentences in the text and width $m = 2$ (yes/no), representing a binary judgement about whether a given sentence is relevant for summarizing.

[2] In Table 5, there are more raters than subjects. This happens because subjects are multiply assigned to extraction tasks.

Table 6: A reliability scale based on K (cited in Carletta et al. (1997))

K			reliability
	$<$	0	POOR
.0	–	.20	SLIGHT
.21	–	.40	FAIR
.41	–	.60	MODERATE
.61	–	.80	SUBSTANTIAL
.81	–	1.0	NEAR PERFECT

Level of Agreement and Data Reliability For a behavioral scientist, results in Table 5 would indicate that judgements produced by humans on the summary extraction are not to be trusted: on the reliability scale in Table 6, rates we get for the extraction data are somewhere between SLIGHT and FAIR. However it is not immediately clear how an abstracting program trained on such 'untrustworthy' data would perform. How does the notion of level of agreement or data reliability in a behavioral scientist's sense relate to the performance of automatic abstracting ? This is a question we are going to address in the following sections.

We follow Passonneau and Litman (1993) in assuming that the majority opinion is correct and drop decisions not in agreement with the majority. In fact our approach here provides a principled basis for Passonneau and Litman (1993)'s notion of *majority opinion* through the kappa statistic.

Now a decision on whether or not a sentence should be included in a summary extract is said to be a *majority opinion* if it is *positively* agreed upon by n subjects, where n ranges anywhere from 2 to the total number of subjects assigned to a task.[3] Data with various levels of agreement can be obtained by removing from agreement tables those decisions which are against the majority opinion for various values of n.[4] Of them, only those data whose agreement rate is over a specific K threshold are used as training/test data for automatic abstracting. Table 7 lists average agreement rates for data with thresholds ranging from 0.1 to 0.8. The row represents K thresholds, and the column represents text types. Figures in parentheses are the number of texts with a given threshold.

[3]For the reasons mentioned earlier, we dismiss a negative agreement among the majority altogether, which is in contrast with Passonneau and Litman (1993)'s approach where agreement among the majority, either positive or negative, counts as a majority opinion.

[4]The removal of decisions against the majority consists of the following steps. (a) Let a desirable level of agreement be t ($0 \leq t \leq 1$). For each text, set the size of the majority to 1. (b) Find K. If $K \geq t$, stop. (c) Otherwise increase the size by one and remove decisions against the majority so defined. Go back to (b). Note that there will be no removal of disagreeing decisions if the text has the kappa coefficient greater than or equal to t at the start.

Table 7: Thresholding by the kappa coefficient

K	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
COLUMN	0.25(23)	0.37(21)	0.50(21)	0.55(20)	0.59(18)	0.73(10)	0.75(7)	1.00(1)
EDITORIAL	0.20(24)	0.35(22)	0.49(20)	0.55(20)	0.62(18)	0.68(12)	0.87(5)	0.95(3)
NEWS REPORT	0.26(25)	0.38(25)	0.52(24)	0.62(23)	0.65(23)	0.76(13)	0.82(9)	1.00(5)

2.3. Extraction Method

We make use of a decision tree program C4.5 (Quinlan, 1993) to develop a sentence extraction algorithm. What it does in essence is to classify sentences as either "yes" or "no", based on a prediction it makes about whether a given sentence is to be included in a summary extract.

C4.5 works with 'coded descriptions' of data (or *cases*). A coded description consists of a specification of data in terms of a fixed set of attributes and a category to which the data are to be assigned. We use a corpus of coded texts, where each sentence is represented with a set of attributes and assigned to either a "yes" or a "no" category according to whether the sentence is a summary extract selected by a group of humans with some level of agreement among them. We constructed 15 sets of coded texts from the corpus by varying the threshold value of agreement from 0.1 to 0.8.

2.4. Attributes

Attributes provide ways in which to code a sentence. The trouble is, there are many possible ways of choosing among potential attributes and one has to go through some trial and error experimentation to find a set of attributes that work best for his or her task. The selection of attributes is essentially heuristic and empirical. After some examination, we have settled on the following set of attributes, some of which are variations of those typically found in the summarization literature (Kupiec et al., 1995; Paice and Jones, 1993; Edmundson, 1969; Zechner, 1996).

Text Type: This attribute is categorical and identifies the type of a text to which a given sentence belongs. The possible values are "C" for COLUMN, "E" for EDITORIAL and "N" for NEWS REPORT.

Location in Text: The location attribute records information on how far a given sentence appears from the beginning of the text. The value is the ratio of the number of sentences preceding to the total number of sentences in the text. The assumption is that where a sentence occurs in the text gives an important clue to predicting whether it is an extract chosen by human subjects (Edmundson, 1969).

Similarity to Title: This attribute records information on how similar a given sentence is to the title. We use the normalized tf·idf as a similarity metric (Wilkinson, 1994). The similarity between a sentence S and a title T of the text in which it occurs is given by:

$$SIM(T,S) = \sum_{w \in W(T)} NF(w,S) \cdot IDF(w)$$

$W(T)$ is a set of words in T.[5] For each word w in $W(T)$, we find its normalized word frequency $NF(w)$ in S by:

$$NF(w,S) = \frac{F(w,S)}{MAX\_F(S)}$$

where $F(w,S)$ denotes a frequency of the word w in S and $MAX\_F(S)$ the frequency of the most frequent word in S.

$$IDF(w) = \frac{\log \frac{N}{DF(w)}}{\log N}$$

$DF(w)$ is the number of sentences in the text which have an occurrence of w. N is the total number of sentences in the text. $\log N$ is a normalization factor.

Within Text tf·idf: The within-text tf·idf is a metric to quantify how well a given sentence distinguishes itself from the rest of the text (Zechner, 1996). For a sentence S, its degree of distinction $D(S)$ from other sentences is defined analogously to the similarity function above:

$$D(S) = \sum_{w \in W(S)} NF(w,S) \cdot IDF(w)$$

Attitudinal Construct: Attitudinal constructs in Japanese include modal verbs/auxiliaries, a class of verbal/sentential constructions expressing the speaker's subjective attitude (*hitsuyo-da* 'it is necessary', *kiboo-suru* 'it is hoped') and sentence final particles such as interrogative and communicative markers(*-ka*,*-yo*,*-ne*) (Nagano, 1986; Unetaya, 1987). This attribute is categorical and takes one of the three values, TYPE 1, TYPE 2 and TYPE 3, depending on whether the sentence ends with a verbal of non-attitudinal type (TYPE 1), or with an attidutinal verbal or a modal (TYPE 2), or with a sentence final particle (TYPE 3). The assumption here is that a sentence with attitudinal expressions has more of a chance to be chosen as a summary extract. Unetaya (1987) gives some supporting evidence.

[5]Words here mean nominals, which are identified using a Japanese tokenizer program (Sakurai and Hisamitsu, 1997).

Table 8: Attribute Representations of Sentences

```
C,0.941,0.000,28,1,2.900,0.333,Y
E,0.000,0.717,31,1,6.366,0.000,Y
N,0.167,0.339,26,1,5.966,0.600,N
```

Sentence Length: This attribute records the length (given in character) of a sentence. The idea is that short sentences may not be informative enough to serve as a summary line (Kupiec et al., 1995).

Location in Paragraph: This attribute records the location of a given sentence within the paragraph. The value is continuous and determined similarly to the location attribute above.

Shown in Table 8 are some sample encodings of sentences in terms of the attributes above. Each line encodes a sentence as regards to TEXT-TYPE, LOCATION-IN-TEXT, SIMILARITY, TEXT-LENGTH, ATTIDUDINAL-TYPE, WITHIN-TEXT-TFIDF, LOCATION-IN-PARAGRAPH, and CLASS in this order. The first line for instance represents a sentence which is a column-type text; its location in text is in the rear; its similarity to title is nil; it is 28 character long; its attidutinal type is 1; it has a tfidf value of 2.9; it occurs at one third of the paragraph; and finally its class is Y, meaning that it is judged important.

3. EVALUATION AND DISCUSSION

We discarded data sets with $K > 0.5$ because they lacked a sufficient number of sentences for evaluation: the column-type data has only 19 sentences at 0.8 (Cf. Table 7). This had left us with nine sets of data with associated threshold values, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, and 0.5.[6] Texts contained in the evaluation data ranged in length from 314 to 535 sentences. A part of a generated decision tree is given in Figure 1. See the caption for explanations.

The procedure for evaluation consists in the following steps: (1) choose at random 200 cases of category "no" and 40 of category "yes" from each of the data sets to form evaluation data; (2) divide the data so chosen into a training set and a test set; (3) build a decision tree from the training set, running C4.5 with the default options; and (4) evaluate its performance on the test data. Since the accuracy of evaluation can vary wildly depending on ways in which the data is divided into training and test sets, the re-sampling method of *cross-validation* is used here, which gives the average over possible partitions of the data into training and test sets. In particular, we use a 10-fold cross-validation method where the data are divided into 10 blocks of cases, of which 9 blocks are used for the training and the remaining one for the

[6]Data with the threshold = 0.1, for instance, consists of coded representations of texts whose agreement rate is above or is equal to 0.1.

Figure 1: A partial decision tree: figures in parentheses denote the number of hits (left) and misses (right) a particular path gives. Y and N represents classes 'Yes' and 'No', respectively. The first line says that the decision tree got 11 hits and 2 misses using the condition "location <= 0.045." Meanings of conditions should be clear from the previous discussion on attributes (Section 2.4).

```
location <= 0.045 : Y (11.0/2.0)
location > 0.045 :
|   similarity <= 1.534 :
|   |   attitudinal type = 3: N (0.0)
|   |   attitudinal type = 1:
|   |   |   tf.idf <= 3.189 :
|   |   |   |   similarity <= 0.143 : N (5.0)
|   |   |   |   similarity > 0.143 :
|   |   |   |   |   similarity <= 0.297 : Y (2.0)
|   |   |   |   |   similarity > 0.297 : N (6.0/1.0)
|   |   |   tf.idf > 3.189 :
|   |   |   |   tf.idf <= 6.26 : N (72.0)
|   |   |   |   tf.idf > 6.26 :
|   |   |   |   |   location <= 0.154 :
|   |   |   |   |   |   location <= 0.125 : N (4.0/1.0)
|   |   |   |   |   |   location > 0.125 : Y (2.0)
|   |   |   |   |   location > 0.154 :
|   |   |   |   |   |   similarity <= 0.952 :
|   |   |   |   |   |   |   tf.idf <= 12.37 : N (59.0/1.0)
|   |   |   |   |   |   |   tf.idf > 12.37 :
|   |   |   |   |   |   |   |   location <= 0.357 : N (2.0)
|   |   |   |   |   |   |   |   location > 0.357 : Y (2.0)
|   |   |   |   |   |   similarity > 0.952 :
|   |   |   |   |   |   |   similarity <= 1.08 : Y (4.0)
|   |   |   |   |   |   |   similarity > 1.08 : N (6.0)
|   |   attitudinal type = 2:
|   |   |   sentence length <= 38 : N (6.0)
|   |   |   sentence length > 38 :
|   |   |   |   similarity <= 0.338 : Y (2.0)
|   |   |   |   similarity > 0.338 :
|   |   |   |   |   sentence length > 64 : N (4.0)
|   |   |   |   |   sentence length <= 64 :
|   |   |   |   |   |   similarity > 1.24 : N (2.0)
|   |   |   |   |   |   similarity <= 1.24 :
|   |   |   |   |   |   |   similarity <= 0.447 : N (2.0)
|   |   |   |   |   |   |   similarity > 0.447 :
|   |   |   |   |   |   |   |   tf.idf <= 9.237 : Y (6.0/1.0)
|   |   |   |   |   |   |   |   tf.idf > 9.237 : N (4.0/1.0)
```

Table 9: Human reliability and precision of abstracting by extraction (averaged over 50 runs). Parenthetical figures denote recall rates.

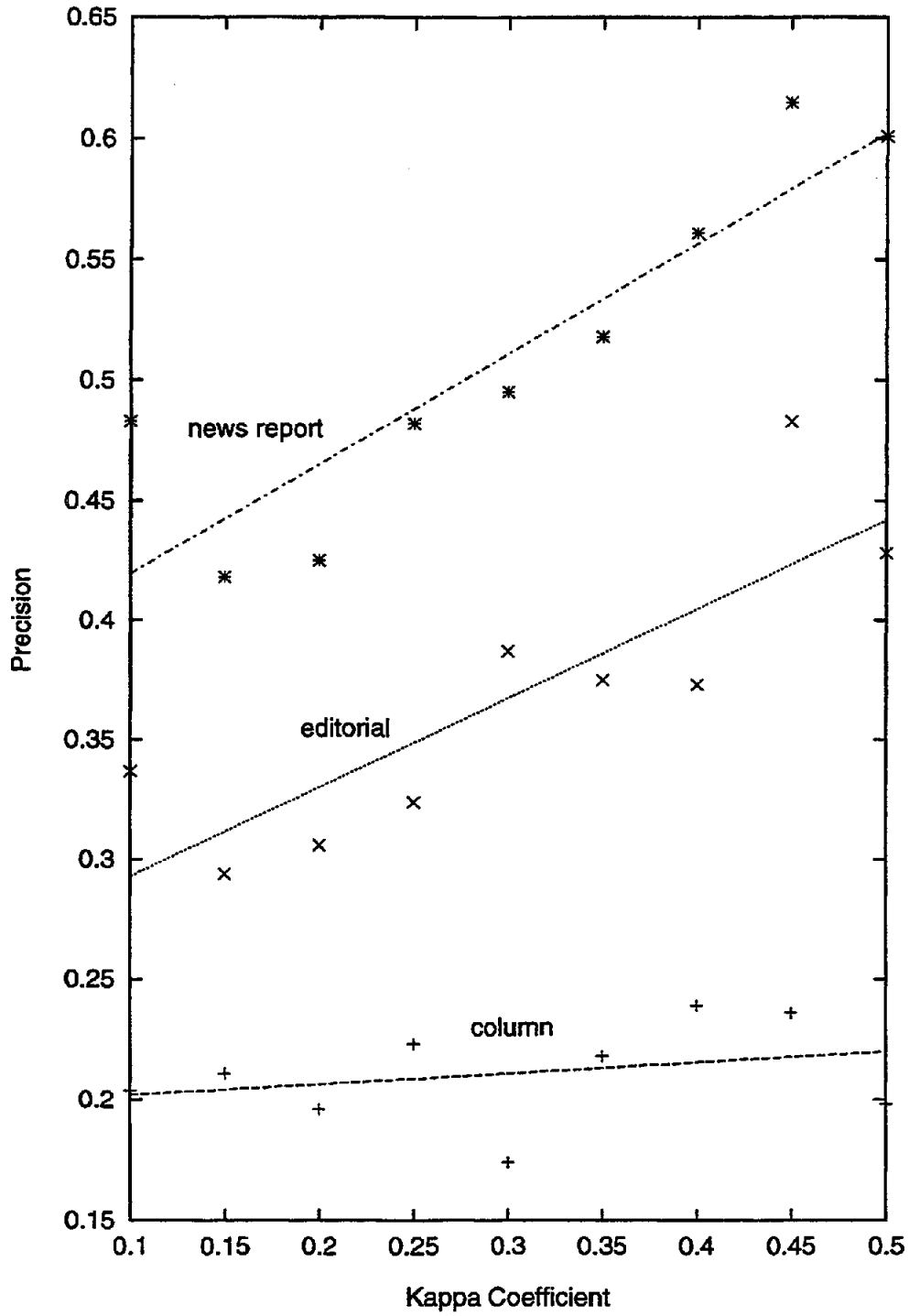
threshold	COLUMN	EDITORIAL	NEWS REPORT
0.10	0.204 (0.113)	0.337 (0.195)	0.483 (0.307)
0.15	0.211 (0.119)	0.294 (0.167)	0.418 (0.262)
0.20	0.196 (0.118)	0.306 (0.189)	0.425 (0.267)
0.25	0.223 (0.127)	0.324 (0.198)	0.482 (0.307)
0.30	0.174 (0.092)	0.387 (0.249)	0.495 (0.322)
0.35	0.218 (0.117)	0.375 (0.271)	0.518 (0.366)
0.40	0.239 (0.138)	0.373 (0.253)	0.561 (0.395)
0.45	0.236 (0.134)	0.483 (0.349)	0.615 (0.466)
0.50	0.198 (0.114)	0.428 (0.316)	0.601 (0.462)

test. Note that the method here gives a rise to 10 possible divisions and an equal number of corresponding decision tree models. The average performance of the generated models is then obtained and used as a summary estimate of the decision tree strategy for a particular set of evaluation data.

Further we use information retrieval metrics, recall and precision, to quantify the performance of the decision tree approach. Precision is the ratio of cases assigned correctly to the "yes" category to the total cases assigned to the "yes" category. Recall is the ratio of cases assigned correctly to the "yes" category to the total "yes" cases. Furthermore, because different samplings of evaluation data from a source data set could produce wide variations in performance, we performed 50 runs of the evaluation procedure on each of the 9 data sets. Each run used a separately (and randomly) sampled set of evaluation data. Results of multiple runs of the procedure on a data set were then averaged to give a representative performance rating for that data set.

Table 9 lists the average precision ratings for the nine data sets. Despite some fluctuations of the figures, the results exhibit clear patterns (Figure 2); the kappa coefficient is strongly correlated with performance for texts of editorial type and of news-report type, but correlation for column-type texts is only marginal. There are also marked differences in performance between text types; the decision tree method performs best on news reports and editorials, but worst on columns. This means that the attributes used are effective only for texts of certain types. The results suggest, further, that if attributes used are indeed a good predictor of summary extracts, their strength as a predictor will be enhanced by the reliability or quality of human judgements. Thus the method's poor performance on column-type texts, despite the fact that texts are becoming increasingly reliable, suggests a need to devise a set of attributes different from those for editorials and news reports.

Figure 2: Relationship between precision and the kappa coefficient for the three text types. The data for each text type are fitted by a least squares regression line: $Y = 0.197800 + 0.0440 * X$ (column); $Y = 0.255844 + 0.3720 * X$ (editorial); $Y = 0.373789 + 0.4570 * X$ (news report).



4. CONCLUSION

We have seen how human reliability can affect the performance of automatic abstracting. Reliability refers to reproducibility or inter-coder consistency of data, which is measured by the kappa statistic, a metric standardly used in the behavioral sciences. It was found that reliability enhances the strength of "good" attributes for a sentence, leading to an improved performance of abstracting models. But we did not discuss an important question of whether the kappa statistic serves as a general tool for distinguishing "good" from "bad" data for training a learning algorithm.

We have also found that a set of attributes vary in effectiveness from one text type to another, though texts under consideration are all from the same domain. But at the moment, it is not clear to us what is a good attribute for representing texts like columns, for which the abstracting model was found not effective. It could be the case that no good attribute exists for columns. In fact humans are not doing well on them either.

Acknowledgements

Many thanks go to the following people, who helped us organize and conduct the testing on summary extraction: Hideaki Takahashi, Sachiko Yoshida, Jun Haga, Takehito Utsuro, and Takashi Miyata. We also thank students of Tsukuba University, Bunkyo University and Nihon Kogyo University for having spared the time to take the summarization tests.

REFERENCES

Melina Alexa, John Bateman, Renate Henschel, and Elke Teich. 1996. Knowledge-Based Production of Synthetic Multimodal Documents. *ERCIM NEWS*, 26:18–20, July. European Research Consortium for Informatics and Mathematics.

John Bateman and Elke Teich. 1995. Selective Information Presentation in an Integrated Publication System: An Application of Genre-Driven Text Generation. *Information Processing & Management*, 31(5):753–767.

Jean Carletta, Amy Isard, Stephen Isard, Jacqueline C. Kowtko, Gwyneth Doherty-Sneddon, and Anne H. Anderson. 1997. The Reliability of a Dialogue Structure Coding Scheme. *Computational Linguistics*, 23(1):13–31.

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

Jean Carletta. 1996. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, 22(2):249–254.

Klaus Krippendorff. 1980. *Content Analysis: An Introduction to Its Methodology*, volume 5 of *The Sage* COMMTEXT *series*. The Sage Publications, Inc.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A Trainable Document Summarizer. In *Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Developmnet in Information Retrieval*, pages 68–73. Seattle, USA.

Kathleen McKeown and Dragomir R. Radev. 1995. Generating Summaries of Multiple News Articles. In *Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Developmnet in Information Retrieval*, pages 68–73. Seattle, USA.

Masaru Nagano. 1986. *Bunshyoron-Sosetsu.* Asakura Shoten.

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

Chris D. Paice and Paul A. Jones. 1993. The Identification of Important Concepts in Highly Structured Technical Papers. In *The Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 69–78. Pittsburgh, USA.

Rebecca J. Passonneau and Diane J. Litman. 1993. Intention-based Segmentation: Human Reliability and Correlation with Linguistic Cues. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 148–155. The Association for Computational Linguistics. Ohio State University, Columbus, Ohio, USA.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

Hirofumi Sakurai and Toru Hisamitsu. 1997. Keitaiso Puroguramu ANIMA no Sekkei to Jissoo. In *Jyoohoo Shori Gakkai Zenki Zenkoku Taikai Kooen Ronbun Shuu*, volume 2, pages 57–56. Information Processing Society of Japan, March 12-14.

Sidney Siegel and N. John Castellan Jr. 1988. *Nonparametric Statistics for the Behavioral Sciences.* McGraw-Hill, Second edition.

Keiko Unetaya. 1987. Chinzitsu no rensa no zanzon-keikoo. In Mayumi Sakuma, editor, *Bunshoo-koozoo no Youyaku-bun no Shosoo*, chapter 6. Kuroshio.

Hideo Watanabe. 1996. A Method for Abstracting Newspaper Articles by Using Surface Clues. In *Proceedings of the 16th International Conference on Computational Linguistics*, volume 2, pages 974–979, August. Copenhagen, Denmark.

Ross Wilkinson. 1994. Effective Retrieval of Structured Documents. In W. Bruce Croft and C. J. van Rijsbergen, editors, *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 311–317. Dublin City University, Springer-Verlag.

Klaus Zechner. 1996. Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences. In *Proceedings of the 16th International Conference on Computational Linguistics*, pages 986–989. Copenhagen, Denmark.