A Machine Learning Approach to Sentence Ordering for Multidocument Summarization and its Evaluation

Danushka Bollegala, Naoaki Okazaki, Mitsuru Ishizuka

University of Tokyo, Japan

Abstract. Ordering information is a difficult but a important task for natural language generation applications. A wrong order of information not only makes it difficult to understand, but also conveys an entirely different idea to the reader. This paper proposes an algorithm that learns orderings from a set of human ordered texts. Our model consists of a set of ordering experts. Each expert gives its precedence preference between two sentences. We combine these preferences and order sentences. We also propose two new metrics for the evaluation of sentence orderings. Our experimental results show that the proposed algorithm outperforms the existing methods in all evaluation metrics.

1 Introduction

The task of ordering sentences arises in many fields. Multidocument Summarization (MDS) [5], Question and Answer (QA) systems and concept to text generation systems are some of them. These systems extract information from different sources and combine them to produce a coherent text. Proper ordering of sentences improves readability of a summary [1]. In most cases it is a trivial task for a human to read a set of sentences and order them coherently. Humans use their wide background knowledge and experience to decide the order among sentences. However, it is not an easy task for computers. This paper proposes a sentence ordering algorithm and evaluate its performance with regard to MDS.

MDS is the task of generating a human readable summary from a given set of documents. With the increasing amount of texts available in electronic format, automatic text summarization has become necessary. It can be considered as a two-stage process. In the first stage the source documments are analyzed and a set of sentences are extracted. However, the document set may contain repeating information as well as contradictory information and these challenges should be considered when extracting sentences for the summary. Researchers have already investigated this problem and various algorithms exist. The second stage of MDS creates a coherent summary from this extract. When summarizing a single document, a naive strategy that arranges extracted sentences according to the appearance order may yield a coherent summary. However, in MDS the extracted sentences belong to different source documents. The source documents may have been written by various authors and on various dates. Therefore we

cannot simply order the sentences according to the position of the sentences in the original document to get a comprehensible summary.

This second stage of MDS has received lesser attention compared to the first stage. Chronological ordering; ordering sentences according to the published date of the documents they belong to [6], is one solution to this problem. However, showing that this approach is insufficient, Barzilay [1] proposed an refined algorithm which integrates chronology ordering with topical relatedness of documents. Okazaki [7] proposes a improved chronological ordering algorithm using precedence relations among sentences. His algorithm searches for an order which satisfies the precedence relations among sentences. In addition to these studies which make use of chronological ordering, Lapata [3] proposes a probabilistic model of text structuring and its application to the sentence ordering. Her system calculates the conditional probabilities between sentences from a corpus and uses a greedy ordering algorithm to arrange sentences according to the conditional probabilities.

Even though these previous studies proposed different strategies to decide the sentence ordering, the appropriate way to combine these different methods to obtain more robust and coherent text remains unknown. In addition to these existing sentence ordering heuristics, we propose a new method which we shall call *succession* in this paper. We then learn the optimum linear combination of these heuristics that maximises readability of a summary using a set of human-made orderings. We then propose two new metrics for evaluating sentence orderings; *Weighted Kendall Coefficient* and *Average Continuity*. Comparing with an intrinsic evaluation made by human subjects, we perform a quantitative evaluation using a number of metrics and discuss the possiblity of the automatic evaluation of sentence orderings.

2 Method

For sentences taken from the same document we keep the order in that document as done in single document summarization. However, we have to be careful when ordering sentences which belong to different documents. To decide the order among such sentences, we implement five ranking experts: Chronological, Probabilistic, Topical relevance, Precedent and Succedent. These experts return precedence preference between two sentences. Cohen [2] proposes an elegant learning model that works with preference functions and we adopt this learning model to our task. Each expert e generates a pair-wise preference function defined as following:

$$\mathrm{PREF}_e(u, v, Q) \in [0, 1]. \tag{1}$$

Where, u, v are two sentences that we want to order; Q is the set of sentences which has been already ordered. The expert returns its preference of u to v. If the expert prefers u to v then it returns a value greater than 0.5. In the extreme case where the expert is absolutely sure of preferring u to v it will return 1.0. On the other hand, if the expert prefers v to u it will return a value lesser than 0.5. In the extreme case where the expert is absolutely sure of preferring v to u

it will return 0. When the expert is undecided of its preference between u and v it will return 0.5.

The linear weighted sum of these individual preference functions is taken as the total preference by the set of experts as follows:

$$\mathrm{PREF}_{total}(u, v, Q) = \sum_{e \in E} w_e \mathrm{PREF}_e(u, v, Q). \tag{2}$$

Therein: E is the set of experts and w_e is the weight associated to expert $e \in E$. These weights are normalized so that the sum of them is 1. We use the Hedge learning algorithm to learn the weights associated with each expert's preference function. Then we use the greedy algorithm proposed by Cohen [2] to get an ordering that approximates the total preference.

2.1 Chronological Expert

Chronological expert emulates conventional chronological ordering [4, 6] which arranges sentences according to the dates on which the documents were published and preserves the appearance order for sentences in the same document. We define a preference function for the expert as follows:

$$\mathrm{PREF}_{chro}(u, v, Q) = \begin{cases} 1 & T(u) < T(v) \\ 1 & [D(u) = D(v)] \wedge [N(u) < N(v)] \\ 0.5 & [T(u) = T(v)] \wedge [D(u) \neq D(v)] \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

Therein: $T(u)$ is the publication date of sentence u; $D(u)$ presents the unique identifier of the document to which sentence u belongs; $N(u)$ denotes the line number of sentence u in the original document. Chronological expert gives 1 (preference) to the newly published sentence over the old and to the prior over the posterior in the same article. Chronological expert returns 0.5 (undecided) when comparing two sentences which are not in the same article but have the same publication date.

2.2 Probabilistic Expert

Lapata [3] proposes a probabilistic model to predict sentence order. Her model assumes that the position of a sentence in the summary depends only upon the sentences preceding it. For example let us consider a summary T which has sentences $S_1, \ldots, S_n$ in that order. The probability $P(T)$ of getting this order is given by:

$$P(T) = \prod_{i=1}^{n} P(S_n | S_1, \ldots, S_{n-i}). \tag{4}$$

She further reduces this probability using bi-gram approximation as follows.

$$P(T) = \prod_{i=1}^{n} P(S_i | S_{i-1}) \tag{5}$$

She breaks each sentence into features and takes the vector product of features as follows:

$$P(S_i|S_{i-1}) = \prod_{(a_{<i,j>},a_{<i-1,k>})\in S_i\times S_{i-1}} P(a_{<i,j>}, a_{<i-1,k>}). \tag{6}$$

Feature conditional probabilities can be calculated using frequency counts of features as follows:

$$P(a_{<i,j>}|a_{<i-1,k>}) = \frac{f(a_{<i,j>}, a_{<i-1,k>})}{\sum_{a_{<i,j>}} f(a_{<i,j>}, a_{<i-1,k>})}. \tag{7}$$

Lapata [3] uses nouns,verbs and dependency structures as features. Where as in our expert we implemented only nouns and verbs as features. We performed back-off smoothing on the frequency counts in equation 7 as these values were sparse. Once these conditional probabilities are calculated, for two sentences u,v we can define the preference function for the probabilistic expert as follows:

$$\text{PREF}_{prob}(u, v, Q) = \begin{cases} \frac{1+P(u|r)-P(v|r)}{2} & Q \neq \oslash \\ \frac{1+P(u)-P(v)}{2} & Q = \oslash \end{cases}. \tag{8}$$

Where, Q is the set of sentences ordered so far and $r \in Q$ is the lastly ordered sentence in Q. Initially, Q is null and we prefer the sentence with higher absolute probability. When Q is not null and u is preferred to v, i.e. $P(u|r) > P(v|r)$, according to definition 8 a preference value greater than 0.5 is returned. If v is preferred to u, i.e. $P(u|r) < P(v|r)$, we have a preference value smaller than 0.5. When $P(u|r) = P(v|r)$, the expert is undecided and it gives the value 0.5.

2.3 Topical Relevance Expert

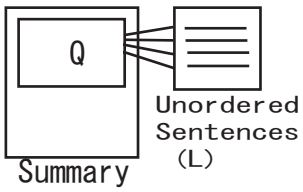

Fig. 1. Topical relevance expert

In MDS, the source documents could contain multiple topics. Therefore, the extracted sentences could be covering different topics. Grouping the extracted sentences which belong to the same topic, improves readability of the summary. Motivated by this fact, we designed an expert which groups the sentences which

belong to the same topic. This expert prefers sentences which are more similar to the ones that have been already ordered. For each sentence l in the extract we define its topical relevance, topic(l) as follows:

$$\text{topic}(l) = \max_{q \in Q} \text{sim}(l, q). \tag{9}$$

We use cosine similarity to calculate $\text{sim}(l, q)$. The preference function of this expert is defined as follows:

$$\text{PREF}_{topic}(u, v, Q) = \begin{cases} 0.5 & [Q = \oslash] \vee [\text{topic}(u) = \text{topic}(v)] \\ 1 & [Q \neq \oslash] \wedge [\text{topic}(u) > \text{topic}(v)] \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

Where, $\oslash$ represents the null set, u,v are the two sentences under consideration and Q is the block of sentences that has been already ordered so far in the summary.

2.4 Precedent Expert

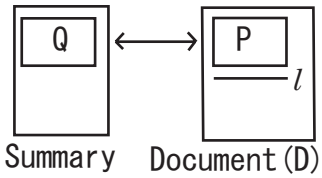

Fig. 2. Precedent expert

When placing a sentence in the summary it is important to check whether the preceding sentences convey the necessary background information for this sentence to be clearly understood. Placing a sentence without its context being stated in advanced, makes an unintelligible summary. As shown in figure 2, for each extracted sentence l, we can compare the block of text that appears before it in its source document (P) with the block of sentences which we have ordered so far in the summary (Q). If P and Q matches well, then we can safely assume that Q contains the necessary background information required by l. We can then place l after Q. Such relations among sentences are called precedence relations. Okazaki [7] proposes precedence relations as a method to improve the chronological ordering of sentences. He considers the information stated in the documents preceding the extracted sentences to judge the order. Based on this idea, we define precedence pre(l) of the extracted sentence l as follows:

$$\text{pre}(l) = \max_{p \in P, q \in Q} \text{sim}(p, q). \tag{11}$$

Here, P is the set of sentences preceding the extract sentence l in the original document. We calculate sim(p, q) using cosine similarity. The preference function for this expert can be written as follows:

$$\mathrm{PREF}_{pre}(u, v, Q) = \begin{cases} 0.5 & [Q = \oslash] \vee [\mathrm{pre}(u) = \mathrm{pre}(v)] \\ 1 & [Q \neq \oslash] \wedge [\mathrm{pre}(u) > \mathrm{pre}(v)] \\ 0 & \text{otherwise} \end{cases} . \tag{12}$$

2.5 Succedent Expert

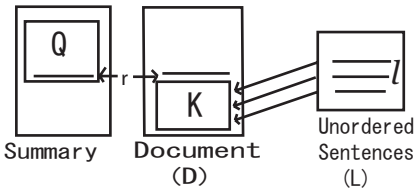

Fig. 3. Succedent expert

When extracting sentences from source documents, sentences which are similar to the ones that are already extracted, are usually ignored to prevent repetition of information. However, this information is valuable when ordering sentences. For example, a sentence that was ignored by the sentence extraction algorithm might turn out to be more suitable when ordering the extracted sentences. However, we assume that the sentence ordering algorithm is independent from the sentence extraction algorithm and therefore does not possess this knowledge regarding the left out candidates. This assumption improves the compatibility of our algorithm as it can be used to order sentences extracted by any sentence extraction algorithm. We design an expert which uses this information to order sentences.

Let us consider the siuation depicted in Figure 3 where a block Q of text is orderd in the summary so far. The lastly ordered setence r belongs to document D in which a block K of sentences follows r. The author of this document assumes that K is a natural consequence of r. However, the sentence selection algorithm might not have selected any sentences from K because it already selected some sentences with this information from some other document. Therefore, we search the extract L for a sentence that best matches with a sentence in K. We define succession as a measure of this agreement(13) as follows:

$$\mathrm{succ}(l) = \max_{k \in K} \mathrm{sim}(l, k). \tag{13}$$

Here, we calculate sim(l, k) using cosine similarity. Sentences with higher succession values are preferred by the expert. The preference function for this expert

can be written as follows:

$$\mathrm{PREF}_{succ}(u,v,Q) = \begin{cases} 0.5 & [Q = \oslash] \vee [\mathrm{succ}(u) = \mathrm{succ}(v)] \\ 1 & [Q \neq \oslash] \wedge [\mathrm{succ}(u) > \mathrm{succ}(v)] \\ 0 & \text{otherwise} \end{cases} . \tag{14}$$

2.6 Ordering Algorithm

Using the five preference functions described in the previous sections, we compute the total preference function of the set of experts as defined by equation 2. Section 2.7 explains the method that we use to calculate the weights assigned to each expert's preference. In this section we will consider the problem of finding an order that satisfies the total preference function. Finding the optimal order for a given total preference function is NP-complete [2]. However, Cohen [2] proposes a greedy algorithm that approximates the optimal ordering. Once the unordered extract X and total preference (equation 2) are given, this greedy algorithm can be used to generate an approximately optimal ordering function $\hat{\rho}$.

let $V = X$
for each $v \in V$ **do**

$$\pi(v) = \sum_{u \in V} \mathrm{PREF}(v,u,Q) - \sum_{u \in V} \mathrm{PREF}(u,v,Q)$$

while V is non-empty **do**
  let $t = \arg\max_{u \in V} \pi(u)$
  let $\hat{\rho}(t) = |V|$
  $V = V - \{t\}$
  for each $v \in V$ **do**
    $\pi(v) = \pi(v) + \mathrm{PREF}(t,u) - \mathrm{PREF}(v,t)$
endwhile

2.7 Learning Algorithm

Cohen [2] proposes a weight allocation algorithm that learns the weights associated with each expert in equation 2. We shall explain this algorithm in regard to our model of five experts.

Rate of learning $\beta \in [0,1]$, initial weight vector $\boldsymbol{w}^1 \in [0,1]^5$, s.t. $\sum_{e \in E} \boldsymbol{w}_e^1 = 1$.

Do for $t = 1, 2, \ldots, T$ where T is the number of training examples.

1. Get X^t; the set of sentences to be ordered.
2. Compute a total order $\hat{\rho}^t$ which approximates,

$$\mathrm{PREF}^t_{total}(u,v,Q) = \sum_{e \in E} \mathrm{PREF}^t_e(u,v,Q).$$

We used the greedy ordering algorithm described in section 2.6 to get $\hat{\rho}^t$.

3. Order X^t using $\hat{\rho}^t$.
4. Get the human ordered set F^t of X^t. Calculate the loss for each expert.

$$\text{Loss}(\text{PREF}_e^t, F^t) = 1 - \frac{1}{|F|} \sum_{(u,v)\in F} \text{PREF}_e^t(u, v, Q) \tag{15}$$

5. Set the new weight vector,

$$w_e^{t+1} = \frac{w_e^t \beta^{\text{Loss}(\text{PREF}_e^t, F^t)}}{Z_t} \tag{16}$$

where, Z_t is a normalization constant, chosen so that, $\sum_{e\in E} w_e^{t+1} = 1$

In our experiments we set $\beta = 0.5$ and $w_i^1 = 0.2$. To explain equation 15 let us assume that sentence u comes before sentence v in the human ordered summary. Then the expert must return the value 1 for PREF(u,v,Q). However,if the expert returns any value less than 1, then the difference is taken as the loss. We do this for all such sentence pairs in F. For a summary of length N we have $N(N-1)/2$ such pairs. Since this loss is taken to the power of β, a value smaller than 1, the new weight of the expert gets changed according to the loss as in equation 16.

3 Evaluation

In addition to Kendall's τ coefficient and Spearman's rank correlation coefficient which are widely used for comparing two ranks, we use sentence continuity [7] as well as two metrics we propose; Weighted Kendall and Average Continuity.

3.1 Weighted Kendall Coefficient

The Kendall's τ coefficient is defined as following:

$$\tau = 1 - \frac{2Q}{{}^nC_2}. \tag{17}$$

Whre, Q is the number of discordant pairs and ${}^nC_2$ is the number of combinations that can be generated from a set of n distinct elements by taking two elements at a time with replacement. However, one major drawback of this metric when evaluating sentence orderings is that, it does not take into consideration the relative distance d between the discordant pairs. However, when reading a text a human reader is likely to be more sensitive to a closer discordant pair than a discordant pair far apart. Therefore, a closer discordant pair is more likely to harm the readability of the summary compared to a far apart discordant pair. In order to reflect these differences in our metric, we use an exponentially decreasing weight function as follows:

$$h(d) = \begin{cases} \exp(1-d) & d \geq 1 \\ 0 & \text{else} \end{cases}. \tag{18}$$

Here, d is the number of sentences that lie between the two sentences of the discordant pair. Going by the traditional Kendall's τ coefficient we defined our weighted Kendall coefficient as following, so that it becomes a metric in $[1, -1]$ range.

$$\tau_w = 1 - \frac{2\sum_d h(d)}{\sum_{i=1}^{n} h(i)} \tag{19}$$

3.2 Average Continuity

Both Kendall's τ coefficient and the Weighted Kendall coefficient measure discordants between ranks. However, in the case of summaries, we need a metric which expresses the continuity of the sentences. A summary which can be read continuously is better compared to a one that cannot. If the ordered extract contains most of the sentence blocks of the reference summary then we can safely assume that it is far more readable and coherent to a one that is not. Sentence n-gram counts of continuous sentences give a rough idea of this kind of continuity.

For a summary of length N there are $N - n + 1$ possible sentence n-grams of length n. Therefore, we can define a precision P_n of continuity length n as:

$$P_n = \frac{\text{number of matched n-grams}}{N - n + 1}. \tag{20}$$

Due to sparseness of higher order n-grams P_n decreases in an exponential-like curve with n. Therefore, we define Average Continuity as the logrithmic average of P_n as follows:

$$\text{Average Continuity} = \exp(\frac{1}{3}\sum_{n=2}^{4} \log(P_n)) \tag{21}$$

We add a small quantity α to numerator and denominator of P_n in equation 20 so that the logarithm will not diverge when n-grams count is zero. We used $\alpha = 0.01$ in our evaluations. Experimental results showed that taking n-grams up to four gave contrasting results because the n-grams tend to be sparse for larger n values. BLEU(BiLingual Evaluation Understudy) proposed by Papineni [8] for the task of evaluating machine translations has an analogical form to our average continuity. In BLEU, a machine translation is compared against multiple reference translations and precision values are calculated using word n-grams. BLEU is then defined as the logarithmic average of these precision values.

4 Results

We used the 3rd Text Summarization Challenge (TSC) corpus for our experiments. TSC[1] corpus contains news articles taken from two leading Japanese

[1] http://lr-www.pi.titech.ac.jp/tsc/index-en.html

newspapers; Mainichi and Yomiuri. TSC-3 corpus contains human selected extracts for 30 different topics. However, in the TSC corpus the extracted sentences are not ordered to make a readable summary. Therefore, we first prepared 30 summaries by ordering the extraction data of TSC-3 corpus by hand. We then compared the orderings by the proposed algorithm against these human ordered summaries. We used 10-fold cross validation to learn the weights assigned to each expert in our proposed algorithm. These weights are shown in table 1. According to table 1, succedent, chronology and precedent experts have the highest weights among the five experts and therefore almost entirely control the process of ordering. Whereas probabilistic and topical relevance experts have almost no influence on their decisions. However, we cannot directly compare Lapata's [3] approach with our probabilistic expert as we do not use dependency structure in our probability calculations. Moreover, Topical relevance, Precedent and Succedent experts require other experts to guide them at the start as they are not defined when Q is null. This inter-dependency among experts makes it difficult to interpret the results in table 1. However, we could approximately consider the values of the weights in table 1 as expressing the reliability of each expert's decisions.

We ordered each extract by five methods: Random Ordering (RO); Probabilistic Ordering (PO); Chronological Ordering (CO); Learned Ordering (LO); and HO (Human-made Ordering) and evaluated the orderings. The results are shown in table 2. Continuity precision, defined in equation 20, against the length of continuity n, is shown in figure 4.

Table 1. Weights learned

Expert	Chronological	Probabilistic	Topical Relevance	Precedent	Succedent
Weights	0.327947	0.000039	0.016287	0.196562	0.444102

Table 2. Comparison with Human Ordering

	Spearman	Kendall	Continuity	Weighted Kendall	Average Continuity
RO	-0.267	-0.160	-0.118	-0.003	0.024
PO	0.062	0.040	0.187	0.013	0.029
CO	0.774	0.735	0.629	0.688	0.511
LO	0.783	0.746	0.706	0.717	0.546
HO	1.000	1.000	1.000	1.000	1.000

According to table 2 LO outperforms RO,PO and CO in all metrics. ANOVA test of the results shows a statistically significant difference among the five methods compared in table 2 under 0.05 confidence level. However, we could not find a statistically significant difference between CO and LO. Topical relevance, Precedent and Succedent experts cannot be used stand-alone to generate a total ordering because these experts are not defined at the start, where Q is null.

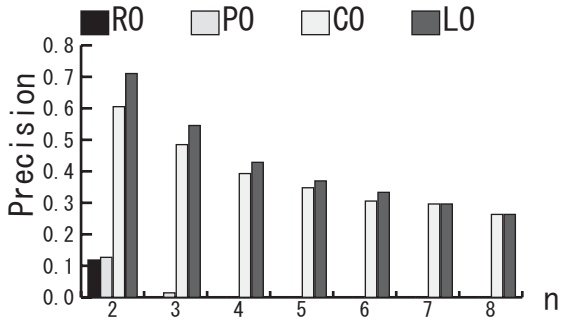

Fig. 4. Precision vs sentence n-gram length

These experts need Chronological and Probabilistic experts to guide them at the beginning. Therefore we have not compared these orderings in table 2.

According to figure 4, for sentence n-grams of length up to 6, LO has the highest precision (defined by equation 20) among the compared orderings. PO did not possess sentence n-grams for n greater than two. Due to the sparseness of the higher order n-grams, precision drops in an exponential-like curve with the length of sentence continuity n. This justifies the logarithmic mean in the definition of average continuity in equation 21. A similar tendency could be observed for the BLEU metric [8].

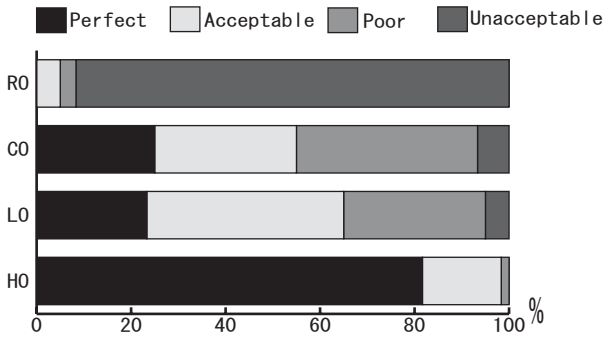

Fig. 5. Human Evaluation

We also performed a human evaluation of our orderings. We asked two human judges to grade the summaries into four categories. The four grades are; *perfect*: no further adjustments are needed, *acceptable*: makes sense even though there is some room for improvement, *poor*: requires minor amendments to bring it up to the acceptable level, *unacceptable*: requires overall restructuring rather than partial revision. The result of the human evaluation of the 60 (2×30) summaries is shown in figure 5. It shows that most of the randomly ordered summaries

(RO) are unacceptable. Although both CO and LO have same number of perfect summaries, the acceptable to poor ratio is better in LO. Over 60 percent of LO is either perfect or acceptable. Kendall's coefficient of concordance (W), which assesses the inter-judge agreement of overall ratings, reports a higher agreement between judges with a value of $W = 0.937$.

Although relatively simple in implementation, the chronological orderings works satisfactorily in our experiments. This is mainly due to the fact that the TSC corpus only contains news paper articles. Barzilay [1] shows chronological ordering to work well with news summaries. In news articles, events normally occur in a chronological order. To evaluate the true power of the other experts in our algorithm, we need to experiment using other genre of summaries other than news summaries.

5 Conclusion

This paper described a machine learning approach to sentence ordering for multidocument summarization. Our method integrated all the existing approaches to sentence ordering while proposing new techniques like succession. The results of our experiments revealed that our algorithm for sentence ordering did contribute to summary readability. We plan to do further study on the sentence ordering problem in future work, extending our algorithm to other natural language generation applications.

References

1. Regina Barzilay, Noemie Elhadad, and Kathleen McKeown. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55, 2002.
2. W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.
3. Mirella Lapata. Probabilistic text structuring: Experiments with sentence ordering. *Proceedings of the annual meeting of ACL, 2003.*, pages 545–552, 2003.
4. C.Y. Lin and E. Hovy. Neats:a multidocument summarizer. *Proceedings of the Document Understanding Workshop(DUC)*, 2001.
5. Inderjeet Mani and Mark T. Maybury, editors. *Advances in automatic text summarization*. The MIT Press, 2001.
6. Kathleen McKeown, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. Towards multidocument summarization by reformulation: Progress and prospects. *AAAI/IAAI*, pages 453–460, 1999.
7. Naoaki Okazaki, Yutaka Matsuo, and Mitsuru Ishizuka. An integrated summarization system with sentence ordering using precedence relation. *ACM-TALIP*, to appear in 2005.
8. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu:a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, 2002.