A Machine Learning Approach to Sentence Ordering for Multidocument Summarization

Danushka Bollegala Naoaki Okazaki Mitsuru Ishizuka
The University of Tokyo

Abstract

Ordering information is a difficult but an important task for natural language generation applications. A wrong order of information not only makes it difficult to understand but also conveys entirely different idea to the reader. In this paper we propose 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

Multidocument summarization(MDS) is the task of generating a human readable summary from a given set of documents. It can be considered as a two-stage process. On the first stage we must extract a set of sentences from the given document set. The second stage of MDS is creating a comprehensible summary from this extract. In this paper we shall concentrate on this second stage of MDS. A good ordering of sentences improves coherence of a summary. Unlike in single document summarization, extracted sentences belong to different documents. Barzilay [1] proposes a chronology oriented approach and Lapata [4] gives a probabilistic text structuring approach to sentence ordering. However, to order a set of sentences correctly, we must consider many other features besides chronology and probabilisitc co-occurances. An algorithm which is able to learn such rules of ordering is needed. Therefore, we used a combination of ordering methods and designed an algorithm which can be trained to order sentences.

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,

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

Here,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. The linear weighted sum of these individual preference functions is taken as the total preference by the set of experts.

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

Here, 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.

2.0.1 Chronological expert

Chronological expert emulates conventional chronological ordering [5, 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:

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

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.

2.0.2 Probabilistic expert

Lapata [4] 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 preceeding 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,

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

She further reduces this probability using bi-gram approximation,

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

She breaks each sentence into features and takes the vector product of features.

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

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

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

Lapata [4] uses Nouns,Verbs and dependency structures as features. Where as in our expert we implemented only Nouns and Verbs as features. We performed backoff smoothing [3] on the frequency counts in equation 7 as these values were sparse. Once these conditional probabilities are calculated, we can define the preference function for the probabilistic expert as follows,

$$\mathrm{PREF}_{prob}(u, v) = \frac{1 + P(v|u) - P(u|v)}{2}. \quad (8)$$

where u, v are two sentences in the extract. When u is preferred to v, i.e. $P(v|u) > P(v|u)$, according to definition 8 a preference value greater than 0.5 is returned. If v is preferred to u, i.e. $P(v|u) < P(u|v)$, we have a preference value smaller than 0.5. When $P(v|u) = P(u|v)$, the expert is undecided and it gives the value 0.5.

2.0.3 Topical relevance expert

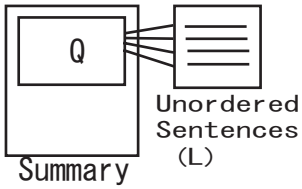

Figure 1: Topical relevance expert

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, $\mathrm{topic}(l)$ as,

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

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

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

2.0.4 Precedent expert

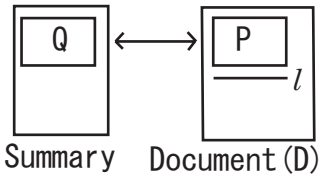

Figure 2: Precedent expert

Okazaki [7] proposes precedence relations as a method to improve the chronological ordering of sentences. He considers the information stated in the documents preceding extract sentences to judge the order. Based on this idea, we define the the precedence $\mathrm{pre}(l)$ of extract sentence l as follows,

$$\mathrm{pre}(l) = \max_{p \in P, q \in Q} \mathrm{sim}(p, q) \quad (11)$$

Here, P is the set of sentences preceding the extract sentence l in the original document. The preference function for this expert can be written as follows,

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

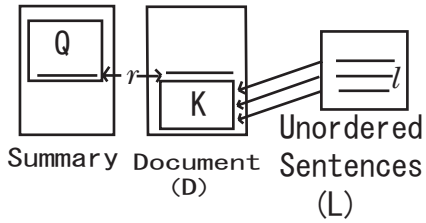

Figure 3: Succedent expert

2.0.5 Succedent expert

When extracting sentences from documents, sentences which are similar to ones already extracted are usually ignored. However, this information is valuable when ordering sentences. We design an expert which uses this information to order sentences. When r is the lastly ordered sentence in the summary so far, we find the block K of sentences that appear after r in the original document. For each of the unordered sentence l, we define its succedence $\mathrm{succ}(l)$ as follows,

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

The preference function for this expert can be written as follows,

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

2.1 Ordering Algorithm

Finding the optimal order for a given total preference 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

2.2 Learning Algorithm

Cohen [2] proposes an 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 $\vec{w}^1 \in [0, 1]^5$, s.t. $\sum_{e \in E} \vec{w}_e^1 = 1$.

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 2.1 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.

   $$\begin{aligned} &\mathrm{Loss}(\mathrm{PREF}^t_e, F^t) \\ &= 1 - \frac{1}{|F|} \sum_{(u,v) \in F} \mathrm{PREF}^t_e(u, v, Q) \end{aligned} \tag{15}$$

5. Set the new weight vector,

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

   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 Results

Preparing 30 sets of extracted sentences based on the TSC-3 extract data, we used 10-fold cross validation to learn the weights assigned to each expert (table 2)

Table 1: Comparison with Human Ordering

	τ_s	τ_k	τ_c	τwk	AC
RO	-0.267	-0.160	-0.118	-0.003	0.024
PO	0.058	-0.019	-0.093	0.003	0.019
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

Table 2: Weights learned

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

and ordered each extract by five methods: Random Ordering (RO); Probabilistic Ordering (PO); Chronological Ordering (CO); Learned Ordering (LO); and HO (Human-made Ordering). We measure closeness of respective orderings to the human-made one and evaluate each method. In addition to Spearman's τ_s and Kendall's τ_k rank correlations which are widely used to compare two ranks, we use a Weighted Kendall coefficient, τ_{wk}, sentence continuity, τ_c, [7] and its extension, Average Continuity (AC).

When reading a summary, readers are more disturbed by closer discordants than by far apart discordants. To reflect this in our evaluation metrics, we use an exponentially weighted Kendall coefficient as follows,

$$\tau_{wk} = 1 - \frac{2\sum_d h(d)}{\sum_{i=1}^{n} h(i)}. \quad (17)$$

Where the weight $h(d)$ imposed on the discordant pair distant d apart is,

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

Figure 4 shows precisions of n-sentence continuity (i.e., sentence n-gram precisions of an ordering against HO). We define,

$$\mathrm{AC} = \exp \sum_{n=2}^{4} \log \frac{\text{number of matched n-grams}}{N-n+1} \quad (19)$$

As it can be seen from table 1 the proposed algorithm(LO) performs better than the existing base line methods. ANOVA test shows that the results are statistically different within 0.05 error margin. In Figure 4, for all lengths of continuity, the proposed method has better precisions than the existing methods. The weights in table 2 can be considered as the influence made by each

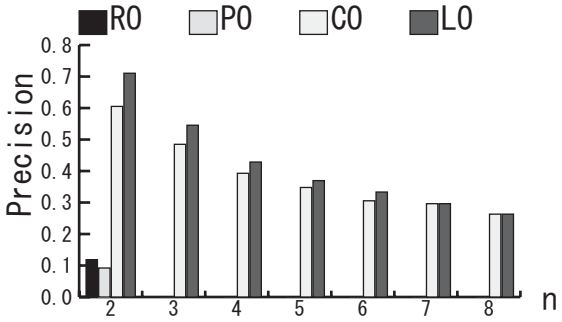

Figure 4: n-gram precision

expert in the final order. We see Succedent and Chronological experts play a major role in the learnt algorithm, where as probabilistic expert has almost no influence. This is due to the naive model used by this expert in calculating sentence conditional probability.

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] Salva M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics Speech and Signal Processing*, 33(3):400–401, 1987.

[4] Mirella Lapata. Probabilistic text structuring: Experiments with sentence ordering. *Proceedings of the annual meeting of ACL, 2003.*, pages 545–552, 2003.

[5] C.Y. Lin and E. Hovy. Neats:a multidocument summarizer. *Proceedings of the Document Understanding Workshop(DUC)*, 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.*, 2005.