Syntax-Driven Sentence Revision for Broadcast News Summarization

Hideki Tanaka, Akinori Kinoshita, Takeshi Kobayakawa,
Tadashi Kumano and Naoto Kato
NHK Science and Technology Research Labs.
1-10-11, Kinuta, Setagaya-ku, Tokyo, Japan
{tanaka.h-ja,kinoshita.a-ek,kobayakawa-t.ko,kumano.t-eq,kato.n-ga}@nhk.or.jp

Abstract

We propose a method of revising lead sentences in a news broadcast. Unlike many other methods proposed so far, this method does not use the coreference relation of noun phrases (NPs) but rather, insertion and substitution of the phrases modifying the same head chunk in lead and other sentences. The method borrows an idea from the sentence fusion methods and is more general than those using NP coreferencing as ours includes them. We show in experiments the method was able to find semantically appropriate revisions thus demonstrating its basic feasibility. We also show that that parsing errors mainly degraded the sentential completeness such as grammaticality and redundancy.

1 Introduction

We address the problem of revising the lead sentence in a broadcast news text to increase the amount of background information in the lead. This is one of the draft and revision approaches to summarization, which has received keen attention in the research community. Unlike many other methods that directly utilize noun phrase (NP) coreference (Nenkova 2008; Mani et al. 1999), we propose a method that employs insertion and substitution of phrases that modify the same chunk in the lead and other sentences. We also show its effectiveness in a revision experiment.

As is well known, the extractive summary that has been extensively studied from the early days of summarization history (Luhn, 1958) suffers from various drawbacks. These include the problems of a break in cohesion in the summary text such as dangling anaphora and a sudden shift in topic.

To ameliorate these problems, the idea of revising the extracted sentences was proposed in a single document summarization study. Jing and McKeown (1999; 2000) found that human summarization can be traced back to six cut-and-paste operations of a text and proposed a revision method consisting of sentence reduction and combination modules with a sentence extraction part. Mani and colleagues (1999) proposed a summarization system based on "draft and revision" together with sentence extraction. The revision part is achieved with the sentence aggregation and smoothing modules.

The cohesion break problem becomes particularly conspicuous in multi-document summarization. To ameliorate this, revision of the extracted sentences is also thought to be effective, and many ideas and methods have been proposed so far. For example, Otterbacher and colleagues (2002) analyzed manually revised extracts and factored out cohesion problems. Nenkova (2008) proposed a revision idea that utilizes noun coreference with linguistic quality improvements in mind.

Other than the break in cohesion, multi-document summarization faces the problem of information overlap particularly when the document set consists of similar sentences. Barzilay and McKeown (2005) proposed an idea called sentence fusion that integrates information in overlapping sentences to produce a non-overlapping summary sentence. Their algorithm firstly analyzes the sentences to obtain the dependency trees and sets a basis tree by finding the centroid of the dependency trees. It next augments the basis tree with the sub-trees in other sentences and finally prunes the predefined constituents. Their algorithm was further modified and applied to the German biographies by Filippova and Strube (2008).

Like the work of Jing and McKeown (2000) and Mani et al. (1999), our work was inspired by the summarization method used by human abstractors. Actually, our abstractors first extract important sentences, which is called lead identification, and then revise them, which is referred to as phrase elaboration or specification. In this paper, we concentrate on the revision part.

Our work can be viewed as an application of the sentence fusion method to the draft and revision

Proceedings of the 2009 Workshop on Language Generation and Summarisation, ACL-IJCNLP 2009, pages 39–47,
Suntec, Singapore, 6 August 2009.

approach to a single Japanese news document summarization. Actually, our dependency structure alignment is almost the same as that of Filippova and Strube (2008), and our lead sentence plays the role of a basis tree in the Barzilay and McKeown approach (2005). Though the idea of sentence fusion was developed mainly for suppressing the overlap in multi-document summarization, we consider this effective in augmenting the extracts in a single-document summarization task where we face less overlap among sentences.

Before explaining the method in detail, we will briefly introduce the Japanese dependency [1] structure on which our idea is based. The dependency structure is constructed based on the bunsetsu chunk, which we call "chunk" for simplicity. The chunk usually consists of one content-bearing word and a series of function words. All the chunks in a sentence except for the last one modify a chunk in the right direction. We call the modifying chunk the modifier and the modified chunk the head. We usually span a directed edge from a modifier chunk to the head chunk [2]. Our dependency tree has no syntactic information such as subject or object.

2 Broadcast news summarization

Tanaka et al. (2005) showed that most Japanese broadcast news texts are written with a three-part structure, i.e., the lead, body, and supplement. The most important information is succinctly mentioned in the lead, which is the opening sentence(s) of a news story, referred to as an "article" here. Proper names and details are sometimes avoided in favor of more abstract expressions such as "big insurance company." The lead is then detailed in the body by answering who, what, when, where, why, and how, and proper names only alluded to in the lead appear here. Necessary information that was not covered in the lead or the body is placed in the supplement.

The research also reports that professional news abstractors who are hired for digital text services summarize articles in a two-step approach. First, they identify the lead sentences and set it (them) as the starting point of the summary. As the average lead length is 95 characters and the allowed summary length is about 115 characters (or 150 characters depending on the screen design), they revise the lead sentences using expressions from the remainder of the story.

We see here that the extraction and revision strategy that has been extensively studied by many researchers for various reasons was actually applied by human abstractors, and therefore, the strategy can be used as a real summarization model. Inspired by this, we decided to study a news summarization system based on the above approach. To develop a complete summarization system, we have to solve three problems: 1) identifying the lead, body, and supplement structure in each article, 2) finding the lead revision candidates, and 3) generating a final summary by selecting and combining the candidates.

We have already studied problem 1) and showed that automatic recognition of three tags with a decision tree algorithm reached a precision over 92% (Tanaka et al. 2007). We then moved to problem 2), which we discuss extensively in the rest of this paper.

3 Manual lead revision experiment

To see how problem 2) in the previous section could be solved, we conducted a manual lead-revision experiment. We asked a native Japanese speaker to revise the lead sentences of 15 news articles using expressions from the body section of each article with cut-and-paste operations (insertion and substitution) of bunsetsu chunk sequences. We refer to chunk sequences as phrases. We also asked the reviser to find as many revisions as possible.

In the interview with her, we found that she took advantage of the syntactic structure to revise the lead sentences. Actually, she first searched for the "same" chunks in the lead and the body and checked whether the modifier phrases to these chunks could be used for revision. To see what makes these chunks the "same," we compared the syntactic head chunk of the lead and body phrases used for substitution and insertion.

Table 1 summarizes the results of the comparison in three categories: perfect match, partial match (content word match), and different.

The table indicates that nearly half of the head chunks were exactly the same, and the rest contained some differences. The second row shows the number where the syntactic heads had the same content words but not the same function words. The pair 会談し kaidan-shi 'talked' and 会談しました kaidan-shi-mashi-ta 'talked' is an

[1] This is the kakari-uke (modifier-modifiee) relation of Japanese, which differs from the conventional dependency relation. We use the term dependency for convenience in this paper.

[2] This is the other way around compared to the English dependency such as in Barzilay and McKeown (2005).

		Ins.	Sub.	Total
1)	Perfect	9	6	15
2)	Partial	6	6	12
3)	Different	1	6	7
	Total	16	18	34

Table 1. Degree of syntactic head agreement

example. These are the syntactic and aspectual variants of the same verb 会談する kaidan-suru 'talk.'

The third row represents cases where the syntactic heads had no common surface words. We found that even in this case, though, the syntactic heads were close in some way. In one example, there was accordance in the distant heads, for instance, in the pair 見つかった mitsuka-tta 'found' and 一部の ichibu-no 'part of.' In this case, we can find the chunk 見つかった mitsuka-tta 'found' at a short edge distance from 一部の ichibu-no 'part of.' Based on the findings, we devised a lead sentence revision algorithm.

4 Revision algorithm

4.1 Concept

We explain here the concept of our algorithm and show an example in Figure 1. We have a lead sentence and a body sentence, both of which have the "same" syntactic head chunk, 到着しました, touchaku-shima-shi-ta, 'arrived.'

The head chunk of the lead has two phrases (underlined with thick lines in Figure 1) that directly modify the head. We call such a phrase a *maximum phrase* of a head[3]. Like the lead sentence, the body sentence also has two maximum phrases. In the following part, we use the term phrase to refer to a maximum phrase for simplicity.

By comparing the phrases in Figure 1, we notice that the following operations can add useful information to the lead sentence; 1) inserting the first phrase of the body will supply the fact the visit was on the 4th, 2) substituting the first phrase of the lead with the second one in the body adds the detail of the IAEA team. This revision strategy was employed by the human reviser mentioned in section 2, and we consider this to be effective because our target document has a so-called inverse pyramid structure (Robin and McKeown 1996), in which the first sentence is elaborated by the following sentences.

[3] To be more precise, a maximum phrase is defined as the maximum chunk sequence on a dependency path of a head.

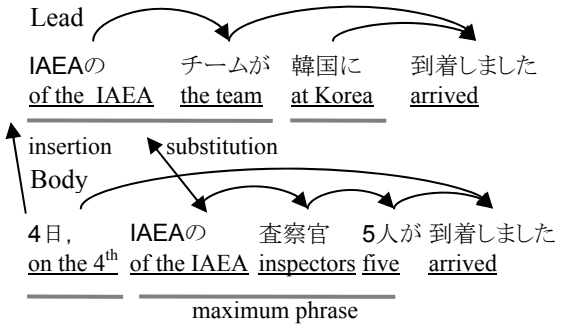

Figure 1. Concept of revision algorithm

Further analyzing the above fact, we devised the lead sentence revision algorithm below. We present the outline here and discuss the details in the next section. We suppose an input pair of a lead and a body sentence that are syntactically analyzed.

1) Trigger search
   We search for the "same" chunks in the lead and body sentences. We call the "same" chunks *triggers* as they give the starting point to the revision.
2) Phrase alignment
   We identify the maximum phrases of each trigger, and these phrases are aligned according to a similarity metric.
3) Substitution
   If a body phrase has a corresponding phrase in the lead, and the body phrase is richer in information, we substitute the body phrase for the lead phrase.
4) Insertion
   If a body phrase has no counterpart in the lead, that is, the phrase is floating, we insert it into the lead sentence.

Our method inserts and substitutes any type of phrase that modifies the trigger and therefore has no limitation in syntactic type. Although NP elaboration such as in (Nenkova 2008) is of great importance, there are other useful syntactic types for revision. An example is the adverbial phrase insertion of time and location. The insertion of the phrase 4 日 yokka 'on the 4th' in figure 1 indeed adds useful information to the lead sentence.

4.2 Algorithm

The overall flow of the revision algorithm is shown in Algorithm 1. The inputs are a lead and a body sentence that are syntactically parsed, which are denoted by *L* and *B* respectively.

The whole algorithm starts with the all-trigger search in step 1. Revision candidates are then found for each trigger pair in the main loop from steps 2 to 6. The revision for each trigger pair is

Algorithm 1[4] (Left figures are the step numbers.)

1: find all trigger pairs between L and B and store them in T.
$T=\{(l, b)\ ;\ l \approx b, l \in L$ and $b \in B\ \}$
2: for all $(l, b) \in T$ do
find l's max phrases and store in P_l.
$P_l=\{p_l\ ;\ p_l \in$ max phrase of $l\}$
3: do the same for trigger b
$P_b=\{p_b\ ;\ p_b \in$ max phrase of $b\}$
4: align phrases in P_l and P_b and store result in A
$A=\{(p_l, p_b)\ ;\ p_l \leftrightarrow p_b\ ,\ p_l \in P_l, p_b \in P_b\ \}$
5: for all $(p_l, p_b) \in A$ do
follow Table 2
end for
6: end for

		Body	
		$p_b = \varnothing$	$p_b \neq \varnothing$
Lead	$p_l = \varnothing$	4: no op.	1: insertion
	$p_l \neq \varnothing$	3: no op.	2: substitution

Table 2. Operations for step 5

found based on the idea in the previous section in steps 4 and 5. Now we explain the main parts.

- Step 1: trigger chunk pair search

We first detect the trigger pairs in step 1 that are the base of the revision process. What then can be a trigger pair that yields correct revisions? We roughly define trigger pairs as the "coreferential" chunk pairs of all parts of speech, i.e., the parts of speech that point to the same entity, event, action, change, and so on.

Notice that the term coreferential is used in an extended way as it is usually used to describe the phenomena in noun group pairs (Mitkov, 2002). The chunk 到着しました touchaku-shimashita 'arrived' and IAEA の IAEA-no 'of the IAEA' in Figure 1 are examples.

Identifying our coreferential chunks is even harder than the conventional coreference resolution, and we made a simplifying assumption as in Nenkova (2008) with some additional conditions that were obtained through our preliminary experiments.

(1) Assumption: Two chunks having the same surface forms are coreferential.
(2) Conditions for light verb (noun) chunks: Agreement of modifying verbal nous is further required for chunks whose content words consist only of light verbs such as ある aru 'be' and なる naru 'become': these chunks themselves have little lexical meaning. The agreement is checked with the hand-crafted rules. Similar checks are applied to chunks whose content words consist only of light nouns such as こと koto ('koto' makes the previous verb a noun) .
(3) Conditions for verb inflections: a chunk that contains a verb usually ends with a function word series that indicates a variety of information such as inflection type, dependency type, tense, aspect, and modality. Some information such as tense and aspect is vital to decide the coreference relation (exchanging the modifier phrases "arrive" and "will arrive" will likely bring about inconsistency in meaning), although some is not. We are in the process of categorizing function words that do not affect the coreference relation and temporally adopted the empirically obtained rule: the difference in verb inflection between the te-form (predicate modifying form) and dictionary form (sentence end form) can be ignored.

- Step 4: phrase alignment

We used the surface form agreement for similarity evaluation. We applied several metrics and explain them one by one.

1) Chunk similarity t, s
$t, s : x, y \in$ chunk $\rightarrow [0, 1]$.
Function t is the Dice coefficient between the set of content words in x and those in y. The same coefficient calculated with all words (function and content words) is denoted as s.

2) Phrase absorption ratio
$a : p_x, p_y \in$ phrases $\rightarrow [0, 1]$
This is the function that indicates how many chunks in phrase p_x is represented in p_y and is calculated with t as in,

$$a(p_x, p_y) := \frac{1}{|p_x|} \sum_{x \in p_x} \max_{y \in p_y}(t(x, y)).$$

3) Alignment quality
With the above two functions, the alignment quality is evaluated by the function
$g : p_x, p_y \in$ phrases $\rightarrow [0, 1]$
$g(p_x, p_y) := \alpha a(p_x, p_y) + (1 - \alpha) s(x, y),$
$\alpha \in [0,1],$
where the shorter phrase is set to p_x so that $|p_x| < |p_y|$. The variables x and y are the last

[4] The sign a ≈ b means the chunk "a" and "b" are triggers. The sign p ↔ q means the phrases "p" and "q" are aligned.

chunks in p_x and p_y, respectively. Intuitively, the function evaluates how many chunks in the shorter phrase p_x are represented in p_y and how similar the last chunks are. The last chunk in a phrase, especially the function words in the chunk, determines the syntactic character of the phrase, and we measured this value with the second term of the alignment quality. The parameter α is decided empirically, which was set at 0.375 in this paper.

In alignment, we calculated the score for all possible phrase combinations and then greedily selected the pair with the highest score. We set the minimum alignment score at 0.185; those pairs with scores lower than this value were not aligned.

- Step 5 (Table 2, case 1): insertion

Step 5 starts either an insertion or substitution process, as in Table 2. If $p_b \neq \varnothing$ (body phrase is not null) and $p_l = \varnothing$ (lead phrase is null) in Table 2, the insertion process starts.
In this process, we check the following.

1) Redundancy check
Insertion may cause redundancy in information. As a matter of fact, redundancy often happens when there is an error in syntactic analysis. Suppose there are the same lead and body phrases that modify the same chunks in the lead and body sentences. If the lead phrase fails to modify the correct chunk because of an error, the body phrase loses the chance to be aligned to the lead phrase since they belong to different trigger chunks. As a result, the body phrase becomes a floating phrase and is inserted into the lead chunk, which duplicates the same phrase.
To prevent this, we evaluate the degree of duplication with the phrase absorption ratio a and allow phrase insertion when the score is below a predefined threshold θ: we allow insertion when
$a(p_b, L) < \theta, p_b \in$ phrase, L : lead sentence,
is satisfied.

2) Discourse coherence check
Blind phrase insertion may invite a break in cohesion in a lead sentence. This frequently happens when the inserted phrase has words that require an antecedent. We then prepared a list of words that contain such context-requiring words and forbid phrase insertions that contain words that are on the list. This list contains the pronoun family such as この kono 'this' and special adjectives such as 違う chigau 'different.'

3) Insertion point decision
The body phrase should be inserted at the proper position in the lead sentence to maintain the syntactic consistency. Because we dealt with single-phrase insertion here, we employed a simple heuristics.
Since the Japanese dependency edge spans from left to right as we mentioned in section 1, we considered that the right phrase of the inserted phrase is important to keep the new dependency from the inserted phrase to the trigger chunk. Because we already know the phrase alignment status at this stage, we follow the next steps to determine the insertion position in the lead of the insertion phrase.
A) In the body sentence, find the nearest right substitution phrase p_r of the insertion phrase.
B) Find the p_r's aligned phrase in the lead p_r^L.
C) Insert the phrase to the left of the p_r^L.
D) If there is no p_r, insert the phrase to the left to the trigger.

- Step 5 (Table 2, case 2): substitution

If $p_b \neq \varnothing$ and $p_l \neq \varnothing$ in Table 2, the substitution process starts. This process first checks if each aligned phrase pair contains the same chunk other than the present trigger. If there is such a chunk, the substitution phrase is reduced to the subtree from the present trigger to the identical chunk. The newly found identical chunks are in trigger table T, and the remaining part will be evaluated later in the main loop. Owing to the phrase partitioning, we can avoid phrase substitutions which are in an inclusive relation.
The substitution candidate goes through three checks: information increase, redundancy, and discourse cohesion. As the latter two are almost the same as those in the insertion, we explain here the information increase. This involves checking whether the number of chunks in the body phrase is greater than that in the aligned lead phrase. This is based on the simple assumption that elaboration requires more words.

5 Revision experiments

5.1 Data and evaluation steps

- Purpose

We conducted a lead revision experiment with three purposes. The first one was to empirically evaluate the validity of our simplified assump-

tions: trigger identification and concreteness increase evaluation. For trigger identification, we basically viewed the identical chunks as triggers and added some amendments for light verbs (nouns) and verb inflections. For the check of an increase in concreteness, we assumed that phrases with more chunks were more concrete. However, these simplifications should be verified in experiments.

The second purpose was to check the validity of using the revision phrases only in body sentences and not in the supplemental sentences.

The last one was to determine how ineffective the result is if the syntactic parsing fails. With these purposes in mind, we designed our experiment as follows.

- Data

A total of 257 articles from news programs broadcast on 20 Jan., 20 Apr., and 20 July in 2004 were tagged with lead, body, and supplement tags by a native Japanese evaluator. The articles were morphologically analyzed by Mecab (Kudo et al., 2003) and syntactically parsed by Cabocha (Kudo and Matsumoto, 2002).

- Evaluator and evaluation detail

We prepared an evaluation interface that presents a lead with one revision point (insertion or substitution) that was obtained using the body and supplemental sentences to an evaluator.

A Japanese native speaker evaluated the results one by one with the above interface. We planned a linguistic evaluation like DUC2005 (Hoa Trang, 2005). Since their five-type evaluation is intended for multi-document summarization, whereas our task is single-document summarization, and we are interested in evaluating our questions mentioned above, we carried out the evaluation as follows. In future, we plan to increase the number of evaluation items and the number of evaluators.

Concreteness	Score
Decreased	0
Unchanged	1
Increased	2

Table 3. Evaluation of increased concreteness

Completeness	Required operations	Score
Poor	More than 2	0
Acceptable	One	1
Perfect	None	2

Table 4. Sentential completeness

E1) The evaluator judged if the revision was obtained from the lead and body sentences with or without parsing errors. Here, errors that did not affect the revision were not considered.

E2) Second, she checked whether the revision was semantically correct or revised information matching the fact described in the lead sentence. Here, she did not care about the grammaticality or the improvements in concreteness of the revision; if the revision was problematic but manually correctable, it was judged as OK. This step evaluated the correctness of the trigger selection; wrong triggers, i.e., those referring to different facts produce semantically inconsistent revisions as they mix up different facts.

The following evaluation was done for those judged correct in evaluation step E2, as we found that revisions that were semantically inconsistent with the lead's facts were often too difficult to evaluate further.

E3) Third, she evaluated the change in concreteness after revision with the revisions that passed evaluation E2. She judged whether or not the revision increased the concreteness of the lead in three categories (Table 3).
Notice that original lead sentences are supposed to have an average score of 1.

E4) Last, she checked the sentential completeness of the revision result that passed evaluation E2. They still contained problems such as grammatical errors and improper insertion position. Rather than evaluating these items separately, we measured them together for sentential completeness. At this time, we measured in terms of the number of operations (insertion, deletion, substitution) needed to make the sentence complete[5].

As shown in Table 4, revisions requiring more than two operations are categorized as "poor," those requiring one operation are "acceptable," and those requiring no operations are "perfect." We employed this measure because we found that grading detailed items such as grammaticality and insertion positions at fine levels was rather difficult. We also found that native Japanese speakers can correct errors easily. Notice the lead sentences are perfect and are supposed

[5] This was not an automatic process and may not be perfect. The evaluator simulated the correction in mind and judged whether it was done with one action.

to have an average score of 2 in sentential completeness. Since the revision does not improve the completeness further but elicits defects such as grammatical errors, it usually produces a score below 2. Some examples of the results with their scores are shown below. The underlined parts are the inserted body chunk phrases, and the parenthesized parts are the deleted lead chunks.

1) Concreteness 2, Completeness 2
民間団体の「コリア・ソサエティ」などが主催する「朝鮮半島平和フォーラム」に（催しに）出席する…
minkan-dantai-no 'private organization', korea-society-nado-ga 'Korea Society and others', shusai-suru 'sponsored', chousen-hantou-heiwa-forumu-ni 'Peace Forum in Korean Peninsula', (moyooshi-ni 'event'), shusseki-suru 'attend'

2) Concreteness 1, Completeness 2
部品に亀裂が入っているのが（ ）見つかった…
buhin-ni 'to the parts' kiretsu-ga 'cracks', haitte-iru-no-ga 'being there' (), mitsuka-tta 'found'

3) Concreteness 2, Completeness 0
ヘリコプターから地上二十メートルの高さから（ ）落下し死亡しました。
Herikoputa-kara 'from a helicopter', chijou-niju-metoru-no-takasa-kara 'from 20 meters high' (), rakka-shi 'fell and', shibou-shima-shita 'killed'

Example 1 is the perfect substitution and had scores of 2 for both concreteness increase and completeness. Actually, the originally vaguely mentioned term 'event' was replaced by a more concrete phrase with proper names, 'Korean Peninsula Peace Forum sponsored by Korea Society and others.' Notice that this can be achieved by NP coreference based methods if they can identify that these two different phrases are coreferential. Our method does this through the dependency on the same trigger 出席する shusseki-suru 'attend.'

Example 2 is a perfect sentence, but its concreteness stayed at the same level. As a result, the scores were 1 for concreteness increase and 2 for completeness.

		Incorrect	Correct	Cor. Ratio
Parse	Succ.	70	353	0.83
	Fail.	31	149	0.83
Sent.	Body	50	464	0.90
	Supp.	51	38	0.43

Table 5. Results of semantic correctness

Score		0	1	2	Ave.
Parse	Succ.	0	55	298	1.84
	Fail.	1	19	129	1.86
Sent.	Body	1	61	402	1.86
	Supp.	0	13	25	1.66

Table 6. Results of concreteness increase

Score		0	1	2	Ave.
Parse	Succ.	78	60	215	1.39
	Fail.	66	55	28	0.74
Sent.	Body	120	110	234	1.25
	Supp.	24	5	9	0.61

Table 7. Results of sentential completeness

Actually, the original sentence that meant "They found a crack in the parts" was revised to "They found there was a crack in the parts," which did not add useful information. Example 3 has a grammatical problem although the revision supplied useful information.As a result, it had scores of 2 for concreteness increase and 0 for completeness. The added kara-case phrase (from phrase) 地上二十メートルの高さから chijou-niju-metoru-no-takasa-kara 'from 20 meters high' is useful, but since the original sentence already has the kara-case ヘリコプターから herikoputa-kara 'from helicopter,' the insertion invited a double kara-case, which is forbidden in Japanese. To correct the error, we need at least two operations, and thus, a completeness score of 0 was assigned.

5.2 Results of experiments

Table 5 presents the results of evaluation E2, the semantic correctness with the parsing status of evaluation E1 and the source sentence category from which the phrases for revision were obtained. Columns 2 and 3 list the number of revisions (insertions and substitutions) that were correct and incorrect and column 4 shows the correctness ratio. We obtained a total of 603 revisions and found that 30% (180/603) of them were derived with syntactic errors.

The semantic correctness ratio was unchanged regardless of the parsing success. On the contrary, it was affected by the source sentence type. The correctness ratio with the supplemental sentence

was significantly[6] lower than that with the body sentence. Table 6 lists the results of the concreteness improvements with the parsing status and the source sentence type. Columns 2, 3 and 4 list the number of revisions that fell in the scores (0-2) listed in the first row. The average score in this table again was not affected by the parsing failure but was significantly affected by the source sentence category. The result with the supplement sentences was significantly worse than that with body sentences.

Table 7 lists the results of the sentential completeness in the same fashion as Table 6. The sentential completeness was significantly worsened by both the parsing failure and source sentence category.

These results indicate that the answers to the questions posed at the beginning of this section are as follows. From the semantic correctness evaluation, we infer that our trigger selection strategy worked well especially when the source sentence category was limited to the body.

From the concreteness-increase evaluation, the assumption that we made also worked reasonably well when the source sentence category was limited to the body.

The effect of parsing was much more limited than we had anticipated in that it did not degrade either the semantic correctness or the concreteness improvements. Parsing failure, however, degraded the sentential completeness of the revised sentences. This seems quite reasonable: parsing errors elicit problems such as wrong phrase attachment and wrong maximum phrase identification. The revisions with these errors invite incomplete sentences that need corrections. It is worth noting that cases sometimes occurred where a parsing error did not cause any problem in the revision. We found that the phrases governed by a trigger pair in many cases were quite similar, and therefore, the parser makes the same error. In that case, the errors are often offset and cause no problems superficially.

We consider that the sentential completeness needs further improvements to make an automatic summarization system, although the semantic correctness and concreteness increase are at an almost satisfactory level. Our dependency-based revision is expected to be potentially useful to develop a summarization system.

[6] In this section, the "significance" was tested with the Mann-Whitney U test with Fisher's exact probability. We set the significance level at 5%.

6 Future work

Several problems remain to be solved, which will be addressed in future work. Obviously, we need to improve the parsing accuracy that degraded the sentential completeness in our experiments. Although we did not quantitatively evaluate the errors in phrase insertion position and redundancy, we could see these happening in the revised sentences because of the inaccurate parsing. Apart from this, we need to further refine the following problems.

Regarding the trigger selection, one particular problem we faced was the mixture of statements of different politicians in a news article. The statements were often included as direct quotations that end with the chunk 述べました nobemashi-ta 'said.' Our system takes the chunk as the trigger and does not care whose statements they are; thus, it ended up mixing them up. A similar problem happened when we had two different female victims of an incident in an article. Since our system has no means to distinguish them, the modifier phrases about these women were mixed up.

We think that we can improve our method by applying more general language generation techniques. An example is the kara-case collision that we explained in example 3 in section 5.1. The essence of the problem is that the added content is useful, but there is a grammatical problem. In other words, "what to say" is ok but "how to say" needs refinement. This particular problem can be solved by doing the case-collision check, and by synthesizing the colliding phrases into one. These can be better treated in the generation framework.

7 Conclusion

We proposed a lead sentence revision method based on the operations of phrases that have the same head in the lead and other sentences. This method is a type of sentence fusion and is more general than methods that use noun phrase coreferencing in that it can add phrases of any syntactic type. We described the algorithm and the rules extensively, conducted a lead revision experiment, and showed that the algorithm was able to find semantically appropriate revisions. We also showed that parsing errors mainly degrade the sentential completeness such as grammaticality and repetition.

Reference

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence Fusion for Multidocument News Summarization. *Computational Linguistics*. 31(3): 298-327.

Katja Filippova and Michael Strube. 2008. Sentence Fusion via Dependency Graph Compression. *proc. of the EMNLP 2008*: 177-185

Hongyan Jing and Kathleen R. McKeown. 1999. The Decomposition of Human-Written Summary Sentences. *proc. of the 22nd International Conference on Research and Development in Information Retrieval SIGIR 99*: 129-136.

Hongyan Jing and Kathleen R. McKeown. 2000. Cut and Paste Based Text Summarization, *proc. of the 1st meeting of the North American Chapter of the Association for Computational Linguistics*: 178-185.

Taku Kudo and Yuji Matsumoto. 2002. Japanese Dependency Analysis using Cascaded Chunking. Proc. of the 6th Conference on Natural Language Learning 2002: 63-69.

Taku Kudo, Kaoru Yamamoto and Yuji Matsumoto. 2004. Applying Conditional Random Fields to Japanese Morphological Analysis, *proc. of the EMNLP 2004*: 230-237.

H. P. Luhn. 1958. The Automatic Creation of Literature Abstracts. *Advances in Automatic Text Summarization. The MIT Press*: 15-21.

Inderjeet Mani, Barbara Gates, and Eric Bloedorn. 1999. Improving Summaries by Revising Them. *Proc. of the 37th Annual Meeting of the Association for Computational Linguistics.*: 558-565.

Ruslan Mitkov 2002, Anaphora Resolution, Pearson Education.

Ani Nenkova. 2008. Entity-driven Rewrite for Multidocument Summarization, *proc. of the 3rd International Joint Conference on Natural Language Generation*: 118-125.

Jahna C. Otterbacher, Dragomir R. Radev, and Airong Luo 2002, Revisions that Improve Cohesion in Multi-document Summaries: A Preliminary Study. *Proc. of the ACL-02 Workshop on Automatic Summarization*: 27-36.

Jacques Robin and Kathleen McKeown. 1996. Empirically designing and evaluating a new revision-based model for summary generation. *Artificial Intelligence*. 85: 135-179.