PAPER

Improving the Readability of ASR Results for Lectures Using Multiple Hypotheses and Sentence-Level Knowledge

Yasuhisa FUJII[†a)], **Kazumasa YAMAMOTO**[†], ***Members***, ***and*** **Seiichi NAKAGAWA**[†], ***Fellow***

SUMMARY This paper presents a novel method for improving the readability of automatic speech recognition (ASR) results for classroom lectures. Because speech in a classroom is spontaneous and contains many ill-formed utterances with various disfluencies, the ASR result should be edited to improve the readability before presenting it to users, by applying some operations such as removing disfluencies, determining sentence boundaries, inserting punctuation marks and repairing dropped words. Owing to the presence of many kinds of domain-dependent words and casual styles, even state-of-the-art recognizers can only achieve a 30-50% word error rate for speech in classroom lectures. Therefore, a method for improving the readability of ASR results is needed to make it robust to recognition errors. We can use multiple hypotheses instead of the single-best hypothesis as a method to achieve a robust response to recognition errors. However, if the multiple hypotheses are represented by a lattice (or a confusion network), it is difficult to utilize sentence-level knowledge, such as chunking and dependency parsing, which are imperative for determining the discourse structure and therefore imperative for improving readability. In this paper, we propose a novel algorithm that infers clean, readable transcripts from spontaneous multiple hypotheses represented by a confusion network while integrating sentence-level knowledge. Automatic and manual evaluations showed that using multiple hypotheses and sentence-level knowledge is effective to improve the readability of ASR results, while preserving the understandability.

key words: *improving readability of ASR results, confusion network, automatic speech recognition, classroom lecture speech, sentence-level knowledge*

1. Introduction

The availability of audio transcripts of speech allows the content of the speech to be more easily understood. In particular, classroom lectures, which are the focus of this paper, benefit from transcripts because they can assist the hearing impaired and can also be used in downstream processing such as summarization [1], indexing [2], browsing systems [3], and so on, for normal students. Therefore, there is much research currently underway on transcribing these lectures [4]–[6].

However, the recognition results from current automatic speech recognition (ASR) systems are not easily understood by people even with perfect speech recognition results, because the speech used in classroom lectures contains many ill-formed utterances with filled pauses, restarts, repetitions, deletion of prepositions and so on. Thus, before making transcripts available to users, their readability needs to be improved to assist the readers in understanding the contents of the lecture material. Operations for correcting transcripts include removing filled pauses and repetition, converting from a spoken style to a written style, inserting punctuation marks such as commas and periods, and discourse markers such as for paragraphs.

In this context, there is currently extensive research on paraphrasing and correcting recognition results [7]–[10]. Shitaoka et al. formulated the problem as a kind of machine translation and applied a statistical method to transform spoken language to written language [7]. Hori et al. used weighted finite state transducers (WFSTs) for the same purpose by representing each component as a WFST [8]. Neibig et al. also used WFSTs in which their method was based on the WFST-based log-linear framework [9], [10].

ASR in the classroom is quite difficult owing to the presence of many kinds of domain-dependent words and the spontaneity of the lecture. In the case of classroom lectures, state-of-the-art recognizers typically achieve a word error rate (WER) of 30-50% [4]–[6]. In this scenario, we need a method that is robust to recognition errors to improve the readability of ASR results. Most previous research focused on manually transcribed texts and therefore did not need to take this problem into account. Without special treatment for the problem, those methods would suffer severe degradation when dealing with ASR results [11].

To make the method robust to recognition errors, it would be more effective to use multiple hypotheses produced by an ASR decoder instead of the single-best hypothesis, because we would then have an opportunity to recover recognition errors in the following post-processing stage by making use of more sophisticated knowledge. A number of methods such as N-best, word graph (lattice), word trellis, and confusion networks are known to represent multiple hypotheses for ASR results [12]–[15]. In this study, we use a confusion network as an intermediate representation between a speech recognizer and a module for improving readability for the following three reasons. First, it can serve posterior probabilities of hypothesized words that represent confidence in the words by the recognizer. Second, it is easy to handle its concise structure like a "sausage", as depicted in Figure 1. Finally, it has been successfully used in many other studies as an intermediate representation between speech recognizers and downstream modules, such as machine translation [16] and retrieval [17].

Although using multiple hypotheses would make methods for improving the readability of ASR results more ro-

Manuscript received July 11, 2011.
Manuscript revised November 26, 2011.

[†]The authors are with the Department of Information and Computer Sciences Toyohashi University of Technology, Toyohashi-shi, 441–8580 Japan.
a) E-mail: fujii@slp.cs.tut.ac.jp

DOI: 10.1587/transinf.E95.D.1101

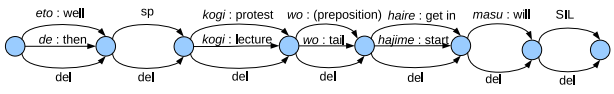


Fig. 1 Confusion Network (*eto kogi wo hajime masu*: Well (I) will start lecture).

bust to recognition errors, we face an additional challenge if we use sentence-level knowledge with multiple hypotheses that are not represented as sentences directly, like N-best. Accurate sentence boundary detection algorithms such as methods using maximum entropy, support vector machines and conditional random fields usually require sentence-level knowledge to extract features from surrounding context.

In this paper, we propose a novel algorithm that infers clean, readable transcripts from spontaneous multiple hypotheses represented by a confusion network while integrating sentence-level knowledge. To integrate sentence-level knowledge, the algorithm uses iterative decoding proposed in [18]. Since this algorithm can integrate sentence-level knowledge efficiently, we can employ a more sophisticated sentence boundary detection algorithm that requires sentence-level knowledge. For the sentence boundary detection algorithm, we used "improved sequential dependency analysis (improved-SDA)" which is an algorithm for dependency analysis and sentence boundary detection designed to work well for spontaneous speech whose sentence boundaries are not predefined [19]. We employed the improved-SDA for sentence boundary detection since it seemed the best among sentence boundary detection methods when dealing with spontaneous speech. In addition, it produced scores from bunsetsu estimation and dependency analysis which may have been useful for sentence cleaning. The scores computed during the process of improved-SDA are integrated into the transcript cleaning process to further improve the performance.

This paper is organized as follows: Sect. 2 explains our proposed method to infer cleaned transcripts from multiple ASR hypotheses represented by a confusion network. Integration of sentence-level knowledge to further improve the transcript cleaning algorithm is described in Sect. 3. Baselines used in the experiments are described in Sect. 4. Experimental results are shown in Sect. 5. Section 6 states the conclusions and outlines some future work.

2. Inferring Cleaned Transcripts from Multiple ASR Hypotheses

In this section, we describe an algorithm to improve the readability of ASR results using multiple hypotheses represented by a confusion network without sentence-level knowledge.

2.1 Formulation

We cast the problem of improving the readability of ASR results as one of finding a clean, readable transcript $\boldsymbol{w}$ given an acoustic observation sequence $\boldsymbol{o}$. The posteriori probability of $\boldsymbol{w}$ given $\boldsymbol{o}$ can be written as

$$P(\boldsymbol{w}|\boldsymbol{o}) = \sum_{\boldsymbol{s}} P(\boldsymbol{w}|\boldsymbol{s},\boldsymbol{o})P(\boldsymbol{s}|\boldsymbol{o}) \approx \max_{\boldsymbol{s}} P(\boldsymbol{w}|\boldsymbol{s})P(\boldsymbol{s}|\boldsymbol{o}), \quad (1)$$

where $\boldsymbol{s}$ stands for a raw transcript corresponding to acoustic observation $\boldsymbol{o}$, $P(\boldsymbol{w}|\boldsymbol{s})$ represents a probability if $\boldsymbol{s}$ is transformed to $\boldsymbol{w}$, and $P(\boldsymbol{s}|\boldsymbol{o})$ is a posterior probability of $\boldsymbol{s}$ given $\boldsymbol{o}$ obtained from a decoder. In this paper, $P(\boldsymbol{w}|\boldsymbol{s})$ is approximated as follows:

$$P(\boldsymbol{w}|\boldsymbol{s}) \approx \frac{1}{C}P(\boldsymbol{w})\delta(\boldsymbol{w},\boldsymbol{s}), \quad (2)$$

where $P(\boldsymbol{w})$ is a language model for clean, readable transcripts, $\delta(\boldsymbol{w},\boldsymbol{s})$ is a function if $\boldsymbol{s}$ can be converted into $\boldsymbol{w}$, the value is 1, otherwise 0, and C is a normalization term to ensure the summation of the probability is equal to 1. These types of (handcrafted) rules are often used when a parallel corpus to train the transformation is not available [20]. For practical reasons, we compute $\delta(\boldsymbol{w},\boldsymbol{s})$ ($\boldsymbol{w} = (w_1, w_2, \ldots, w_N)$, $\boldsymbol{s} = (s_1, s_2, \ldots, s_N)$) as follows (see Sect. 2.3 for the different length):

$$\delta(\boldsymbol{w},\boldsymbol{s}) = \prod_{i}^{N} \delta(w_i, s_i), \quad (3)$$

where $\delta(w, s) = 1$ if there exists the word pair in the translation table, otherwise 0. Since $P(\boldsymbol{s}|\boldsymbol{o})$ is represented by a confusion network, it can be computed as follows:

$$P(\boldsymbol{s}|\boldsymbol{o}) = \prod_{i} P(s_i|\boldsymbol{o}). \quad (4)$$

Using Eqs. (2), (3) and (4), Eq. (1) becomes

$$P(\boldsymbol{w}|\boldsymbol{o}) \approx \frac{1}{C}P(\boldsymbol{w}) \max_{\boldsymbol{s}} \prod_{i}^{N} \delta(w_i, s_i)P(s_i|\boldsymbol{o}). \quad (5)$$

Introducing a weight α to balance between $P(\boldsymbol{w})$ and $P(s_i|\boldsymbol{o})$ into Eq. (5), the final output of the system is the $\hat{\boldsymbol{w}}$, which maximizes the equation below

$$\hat{\boldsymbol{w}} = \operatorname*{argmax}_{\boldsymbol{w}} P(\boldsymbol{w})^{\alpha} \max_{\boldsymbol{s}} \prod_{i}^{N} \delta(w_i, s_i)P(s_i|\boldsymbol{o}). \quad (6)$$

We consider multiple hypotheses from a decoder by a max operator while traditional methods based on statistical methods consider only the single-best hypothesis and thus use only $P(\boldsymbol{w}|\boldsymbol{s})$ [7]–[10].

2.2 Posterior Probabilities from a Confusion Network

In a lattice, we can compute the posterior probability defined as the sum of the posteriors of all paths through a word. Once the confusion network is constructed, words that appear at the same time but on other paths are merged into one class. By summing the posteriors of the words that are merged into the same class, we can compute the posterior of the word class.

$$P(C_i(w)) = \sum_{e \in C_i(w)} P(e), \tag{7}$$

where $C_i(w)$ is the word class of w in bin B_i [†] and it has edges e whose labels are w and whose posterior probabilities are $P(e)$.

The confusion network allows a bin to be dropped by introducing a special word *del*. The posterior probability of this special word *del* is computed as follows:

$$P(del_i) = 1 - \sum_{c \in B_i} P(c), \tag{8}$$

where B_i is the bin to which the special word del_i belongs.

2.3 Transformation Rules

By using Eq. (3), we can deal with any kind of translation. However, in this paper, we deal only with the deletion of filled pauses and the insertion of periods, which should be dealt with initially. In the following sections, we describe each transformation.

2.3.1 Deletion of Filled Pauses

A filled pause is the most dominant phenomenon among the specific phenomena to spontaneous speech [21], and mostly affects the readability of the transcripts. To remove filled pauses, we added entries into the translation table as follows:

$$\delta(del, Filler) = 1, \tag{9}$$

where *Filler* is a word whose part of speech (POS) is *filler* or *interjection*. In our setup, words are composed of surface form, pronunciation and POS tag, so that we can use POS without an additional parser.

2.3.2 Insertion of Periods

The raw output from recognizers lacks the periods and the unit used for recognition (usually a segment between short-pauses) and differs from the actual sentence unit underlying the utterances. Therefore, to improve the readability of the output text, we need to detect sentence boundaries and recover the periods. While periods do not always correspond to pauses, relationships still exist between them [7]. Therefore, initially we added entries into the translation table as follows:

$$\delta(Period, Pause) = 1, \tag{10}$$

where *Pause* is a silence or a short pause. In addition, we added entries into the translation table as follows:

$$\delta(Period, del) = 1. \tag{11}$$

By allowing the conversion from *del* to *Period*, we can also recover the periods that do not correspond to pauses.

Since we do not have manually paraphrased corpora with commas, we do not deal with them in this paper. However, we can add similar rules to those used for insertion of periods, to insert commas.

2.4 Language Model for Clean Transcripts

The language model probability $P(\boldsymbol{w})$ for clean, readable transcripts in Eq. (6) plays an important role in improving the readability of ASR results, because our method assumes that there are no parallel data to obtain transformation rules between raw and cleaned transcripts, unlike [10], and only uses a few heuristic rules for the transformation.

In this study, we train $P(\boldsymbol{w})$ on a large newspaper corpus because the contents of a newspaper can be considered clean and readable, and therefore it is reasonable to try to bring the style of processed transcripts close to that of a newspaper (representative written style).

3. Integration of Sentence-Level Knowledge

3.1 Motivation

While the algorithm described in Sect. 2 has the ability to deal with multiple hypotheses in the transcript cleaning process, it is not accurate for sentence boundary detection because it mostly relies on an N-gram language model trained from newspaper text to detect sentence boundaries. Generally speaking, an N-gram language model is not powerful enough for accurate sentence boundary detection. To improve the sentence boundary detection of the algorithm, we can employ a more sophisticated algorithm that requires sentence-level knowledge [19].

However, we face an additional challenge if we use sentence-level knowledge with multiple hypotheses that are not represented as sentences directly, like the N-best, and are stored compactly by bundling hypotheses locally using some base, like a lattice. Therefore, it is usually difficult or even not feasible to rescore such multiple hypotheses using sentence-level knowledge. Obviously, we can adopt a simple N-best as a representation of multiple hypotheses to use sentence-level knowledge in the algorithm; however, N-best is inefficient for representing thousands of hypotheses.

To solve the problem, we propose to use a recently proposed state-of-the-art iterative decoding algorithm developed for a confusion network to integrate sentence-level knowledge [18]. Using the iterative decoding algorithm, we can use sentence-level knowledge efficiently with multiple hypotheses represented by a confusion network. The actual decoding algorithm we use will be described in Sect. 3.5.

3.2 Improved-SDA

Since we can use sentence-level knowledge efficiently within the process of transcript cleaning owing to the iterative decoding algorithm, we can employ a more sophisticated algorithm that requires sentence-level knowledge to detect sentence boundaries. As the sophisticated sentence

[†]For example, in Fig. 1, *eto*(well), *de*(then) and *del* correspond to word classes and {*eto*(well), *de*(then), *del*} corresponds to a bin.

boundary detection algorithm, we use an improved-SDA, which is an algorithm to determine a sentence boundary and dependency structure simultaneously, given a spoken word stream whose sentence boundaries are not known [19]. Using the improved-SDA, we can expect that sentence boundary detection accuracy will be improved.

3.2.1 *Bunsetsu* Chunking

Since improved-SDA works on a *bunsetsu* (like "phrase" in English) sequence and uses the *bunsetsu* detection result for the accurate sentence boundary detection algorithm, we need to determine *bunsetsu* boundaries prior to applying the improved-SDA. In [19], the *bunsetsu* boundary detection problem was cast as a labeling problem and it was solved by using conditional random fields (CRF) [22]. We also implemented a CRF chunker and used the same configuration with [19] for *bunsetsu* chunking. We use $P(\boldsymbol{b}|\boldsymbol{s})$ to express the probability of *bunsetsu* the chunking result $\boldsymbol{b}$ given a spoken word sequence $\boldsymbol{s}$.

3.2.2 Dependency Modeling

In a typical Japanese dependency analysis, a dependency structure $\boldsymbol{d}$ is represented by a set of head *bunsetsu* $h_1, h_2, \cdots, h_N$ corresponding to modifier *bunsetsus* sequence $\boldsymbol{b} = (b_1, b_2, \cdots, b_N)$. Generally, the dependency analysis is a task for finding the most appropriate dependency structure $\hat{\boldsymbol{d}}$ given the *bunsetsu* sequence $\boldsymbol{b}$. Theoretically, it is written as follows:

$$\hat{\boldsymbol{d}} = \operatorname*{argmax}_{\boldsymbol{d}} P(\boldsymbol{d}|\boldsymbol{b}). \tag{12}$$

$P(\boldsymbol{d}|\boldsymbol{b})$ is the probability for generating the structure $\boldsymbol{d}$ given a *bunsetsu* sequence $\boldsymbol{b}$ and calculated as

$$P(\boldsymbol{d}|\boldsymbol{b}) = \prod_{i=1}^{N} P(b_i \to h_{b_i}|\boldsymbol{\Phi}(b_i, h_{b_i}, \boldsymbol{b})), \tag{13}$$

where $\boldsymbol{\Phi}(b_i, h_{b_i}, \boldsymbol{b})$ is a linguistic feature vector, and $P(b_i \to h_{b_i}|\boldsymbol{\Phi}(b_i, h_{b_i}, \boldsymbol{b}))$ is a link score between b_i and h_{b_i} and trained using dependency parsed data. In [19], the maximum entropy model-based relative dependency model was used to model the link score, which was defined as follows:

$$P(b_i \to h_{b_i}|\boldsymbol{\Phi}(b_i, h_{b_i}, \boldsymbol{b})) = \frac{\exp(\boldsymbol{w} \cdot \boldsymbol{\Phi}(b_i, h_{b_i}, \boldsymbol{b}))}{\sum_{h \in C_{b_i}} \exp(\boldsymbol{w} \cdot \boldsymbol{\Phi}(b_i, h_{b_i}, \boldsymbol{b}))}, \tag{14}$$

where $\boldsymbol{w}$ is a model parameter and C_{b_i} is a set of head candidates for b_i, which is given based on the parsing algorithm and the dependency constraints. If $\boldsymbol{\Phi}(b_i, h_{b_i}, \boldsymbol{b})$ is carefully defined, we can expect that Eq. (14) will work well. To make $\boldsymbol{\Phi}(b_i, h_{b_i}, \boldsymbol{b})$ powerful, explicit feature expansions are usually needed. However, in our algorithm, explicit feature expansions are time-consuming because the algorithm needs to examine a lot of hypotheses. Therefore, instead of expanding features explicitly, we use gate functions to expand features implicitly as follows:

$$P(b_i \to h_{b_i}|\boldsymbol{\Phi}(b_i, h_{b_i}, \boldsymbol{b})) = \frac{\exp(\sum_g^K \boldsymbol{\mu_g} h(\boldsymbol{w_g} \cdot \boldsymbol{\Phi}(b_i, h_{b_i}, \boldsymbol{b})))}{\sum_{h \in C_{b_i}} \exp(\sum_g^K \boldsymbol{\mu_g} h(\boldsymbol{w_g} \cdot \boldsymbol{\Phi}(b_i, h, \boldsymbol{b})))}, \tag{15}$$

where K is a number of gate functions, μ_g is the weight of a gate g, w_g is an internal weight for a gate g and $h(x)$ is a gate function defined as follows:

$$h(x) = \frac{1}{1 + \exp(-x)}. \tag{16}$$

The idea of using gate functions in a maximum entropy model is referenced in [23]. The parameters were trained to maximize the probability defined by Eq. (13) of training data (we used core data from CSJ) using a gradient-based optimization algorithm.

3.2.3 Sequential Dependency Analysis

SDA extracts the dependency structure online by introducing meta-symbols ⟨b⟩ and ⟨c⟩, which express a sentence boundary and an arbitrary *bunsetsu* in the unseen part without changing the formulation of Eq. (12). Please refer to [24] for a detailed explanation of SDA.

In addition to the ⟨b⟩ and ⟨c⟩, we introduce a meta-symbol ⟨n⟩, which expresses a null *bunsetsu*, namely, a link $b_i \to$ ⟨n⟩ means b_i does not have a head. This newly introduced meta-symbol is mainly used for disfluent words that tend not to have heads; therefore, it can be a good indicator for detecting disfluencies (or redundant *bunsetsu*).

3.3 Formulation

Although the initial aim to utilize sentence-level knowledge in transcript cleaning was to improve sentence boundary detection accuracy, the scores computed during the sentence boundary detection process can be used as the scores for transcript cleaning. During the processing of improved-SDA, the *bunsetsu* chunking probability $P(\boldsymbol{b}|\boldsymbol{s})$ and the dependency analysis probability $P(\boldsymbol{d}|\boldsymbol{b})$ are computed. By adding these probabilities into Eq. (6), we obtain the following equation

$$\hat{\boldsymbol{w}} = \operatorname*{argmax}_{\boldsymbol{w}} P(\boldsymbol{w})^{\alpha} \max_{\boldsymbol{s},\boldsymbol{b},\boldsymbol{d}} P(\boldsymbol{d}|\boldsymbol{b})^{\beta} P(\boldsymbol{b}|\boldsymbol{s})^{\gamma} \cdot \prod_{i}^{N} \delta(w_i, s_i|\boldsymbol{b}, \boldsymbol{d}) P(s_i|\boldsymbol{o}), \tag{17}$$

where β and γ are weights for $P(\boldsymbol{d}|\boldsymbol{b})$ and $P(\boldsymbol{b}|\boldsymbol{s})$, respectively. Transformation rules represented by $\delta(w_i, s_i)$ in Eq. (6) are changed to $\delta(w_i, s_i|\boldsymbol{b}, \boldsymbol{d})$ and now depend on an improved-SDA result of $\{\boldsymbol{b}, \boldsymbol{d}\}$.

3.4 Transformation Rules with Improved-SDA

Transformation rules described in Sect. 2.3 are modified to

```
     Input: an observation sequence o
     Output: a clean, readable transcript ŵ
     Variables:
     s, ŝ, s′ : a spoken word sequence
     b, b̂, b′ : a bunsetsu sequence
     d, d̂, d′ : a dependency structure
     w, ŵ, w′ : a clean, readable transcript
     CN = {C_1, C_2, ..., C_|CN|} : a confusion network
     C_i = {C_i1, C_i2 ..., C_i|C_i|} : a set of words representing the bin i
     C_ij : the jth word in the bin i of a confusion network
     x, x̂ : real value
 1   begin
 2     b̂ = φ, d̂ = φ, ŵ = φ
 3     while the next utterance exists do
 4       recognize the utterance and construct confusion network \
           and obtain confusion network CN
 5       x̂ = 0.0
 6       s′ = ŝ, b′ = b̂, d′ = d̂, w′ = ŵ
 7       ŵ = φ
 8       while ŵ ≠ ŵ do
 9         ŵ = ŵ
10         for i = 1 to |CN|
11           for j = 1 to |C_i|
12             s = s′ + GetHypothesis(CN, i, j)
13             b = b′ + Chunking(s, b′)
14             d = d′ + Improved-SDA(b, d′)
15             w = w′ + Transform(s, b, d, w′)
16             Compute score x by Eq. (17) using s, b, d, w
17             if x > x̂ then
18               ŝ = s, b̂ = b, d̂ = d, ŵ = w
19               x̂ = x
20               swap(C_i1,C_ij)
21             end if
22           end for
23         end for
24       end while
25     end while
26   end
```

Fig. 2 Decoding algorithm with iterative decoding.

use the result of improved-SDA. The rules for deletion of filled pauses and insertion of periods are changed as described in the following sections.

3.4.1 Deletion of Filled Pauses

The rule to delete filled pauses is changed as follows:

$$\delta(del, Filler|Filler \rightarrow \langle n \rangle) = 1. \quad (18)$$

This means that *Fillers* only whose heads are $\langle n \rangle$ are transformed to *del*.

3.4.2 Insertion of Periods

Since improved-SDA can detect sentence boundaries accurately, we utilize this information to determine the position where a period is inserted. To reflect the information, we use the following rule:

$$\delta(Period, \langle b \rangle) = 1, \quad (19)$$

where $\langle b \rangle$ is the meta-symbol that indicates a sentence boundary determined by improved-SDA.

3.5 Decoding Algorithm with Iterative Decoding

In this section, we describe an algorithm to find the most plausible clean, readable transcript $\hat{\boldsymbol{w}}$ given an observation sequence $\boldsymbol{o}$ based on Eq. (17). The algorithm is shown in Fig. 2. The algorithm receives utterances one by one, and processes them iteratively (line 3-26). A confusion network CN is constructed at line 4. Iterative decoding finds the most plausible cleaned transcript $\hat{\boldsymbol{w}}$ given the CN and contexts, $\boldsymbol{s}'$, $\boldsymbol{b}'$, $\boldsymbol{d}'$, $\boldsymbol{w}'$, which were the results of the previous iteration (line 6-25). Multiple hypotheses are examined by changing a word at a time (line 10-23). The operator '+' is used to concatenate two partial hypotheses (line 12-15). GetHypothesis(CN, i, j) extracts a word hypothesis from CN by picking the best words from each bin except bin i from which the jth word is extracted. Chunking($\boldsymbol{s}, \boldsymbol{b}'$) chunks $\boldsymbol{s}$ given the context $\boldsymbol{b}'$ using the method described in Sect. 3.2.1 and returns the chunked result. Improved-SDA($\boldsymbol{b}, \boldsymbol{d}'$) conducts improved-SDA on $\boldsymbol{b}$ given a context $\boldsymbol{d}'$ based on Eq. (12) and returns the result. Transform($\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{d}, \boldsymbol{w}'$) applies transformation rules to $\boldsymbol{s}$ given $\boldsymbol{b}$, $\boldsymbol{d}$ and $\boldsymbol{w}'$. If the score of an examined hypothesis $\boldsymbol{w}$ exceeds the score of the current best hypothesis $\hat{\boldsymbol{w}}$, $\boldsymbol{w}$ is used as the new best hypothesis and the examined word C_{ij} is placed at the top of the bin (line 17-21). When the algorithm terminates, $\hat{\boldsymbol{w}}$ is the desired output. Note that this algorithm is globally suboptimal because it considers only the best context in each iteration (line 3-26).

4. Compared Baselines

To confirm the superiority of the proposed method, we compare the proposed method with three baselines, which are described in the following sections.

4.1 Filler Removal

Filled pauses would mostly affect the readability of the transcripts as mentioned in Sect. 2.3.1. Therefore, we removed filled pauses from the transcripts automatically and used it as a baseline. Filled pauses were removed from transcripts based only on the POS information. The words whose POS was "filler" or "interjection" were removed as filled pauses. We will refer to the baseline as *Filler*.

4.2 Single Hypothesis Editing

Our proposed method uses multiple hypotheses to be robust for recognition errors, thus we should compare it with the transcripts that are edited from a confusion network that only contains the single-best hypothesis using the proposed method. To create such a confusion network, we just dropped all other words except single-best words and *dels* from a confusion network. If we need to deal with a plain transcript such as a manual transcript, we create a pseudo confusion network that contains the words included in the

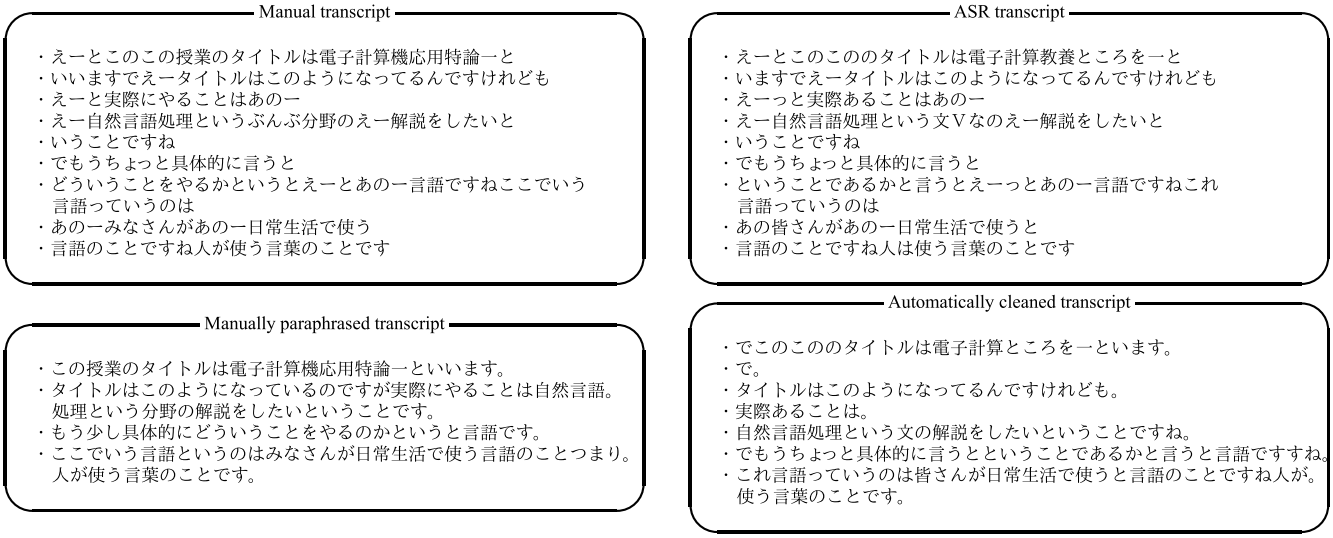


Fig. 3 Examples of transcripts.

Table 1 Statistics of the test sets (average of 8 lectures for CJLC and 4 lectures for CSJ). *APP.* means adjusted perplexity. *OOV* indicates the ratio of OOV.

Corpus	Duration (min.)	#Words		APP.		OOV [%]		#Filler [%]
		Manual	Paraph.	Manual	Paraph.	Manual	Paraph.	
CJLC	67.6	11813	10192	182.6	159.7	3.5	3.9	7.2
CSJ	19.8	4776	3921	80.1	96.4	0.6	0.8	8.6

transcript and *dels* whose pseudo posterior probabilities are c and $1 - c$ in each bin, respectively. We will refer to the baseline as *Single*.

4.3 No Sentence-Level Knowledge

To confirm the effectiveness of sentence-level knowledge, we examine methods that do not use it. In the following experiments, *(w/o snt.)* indicates that sentence-level knowledge is not used (Eq. (6)) while *(w/ snt.)* indicates that sentence-level knowledge is used (Eq. (17)).

5. Experiments

5.1 Setup

We used 8 lectures from CJLC [21] as a test set and 4 lectures from CSJ [25] as a development set. For both corpora, manually transcribed texts (referred to as *manual* in the experimental results) are available, and in addition, we prepared cleaned transcripts of the *manual* transcripts by manually editing and paraphrasing the transcripts to improve the readability (henceforth referred to as *paraphrased*). Examples of the transcripts are shown in Fig. 3. The statistics of the both test sets are shown in Table 1. In the table, "APP." means adjusted perplexity and "OOV" indicates the ratio of OOV. The perplexities were computed using the CSJ model.

We prepared an acoustic model and a language model that were both trained from the CSJ corpus for the ASR. The size of the lexicon was set at 20 k. We used ChaSen with IPADic ver. 2.7.0 as a morphological analyzer. As a decoder, we used SPOJUS++, which has a feature for producing a confusion network [26]. The recognition results, word coverage and word densities of constructed confusion networks are shown in Table 2. We trained a 4-gram language model with Witten-Bell smoothing on the *Mainichi* newspaper corpus (9 years, 214 M words) for clean transcripts using the same lexicon used for the ASR [†]. The same configuration with [19] was used for the improved-SDA except we used Eq. (15) to compute link scores. Our implementation achieved approximately the same performance with [19] for sentence boundary detection and dependency analysis, despite the difference.

Table 2 WER, Word coverage and density of constructed confusion network.

Testset	Reference	WER [%]	Coverage [%]	Density
CJLC	Manual	43.8	80.9	3.83
	Paraphrased	64.2	75.9	
CSJ	Manual	29.0	90.9	4.68
	Paraphrased	49.0	86.9	

We used *WER* to evaluate the transcripts, which was calculated as follows:

[†]Unsurprisingly, most of the colloquial and spoken-style expressions were able to be found in the newspaper articles even though they may have not been frequent words. Since our proposed method does not produce a hypothesis which contains words that are not contained in the vocabulary due to the limited translation rules, it is not needed to use a different vocabulary. However, we can use a more useful different vocabulary without changing the algorithm if needed.

$$WER = \frac{D + S + I}{C + D + S}, \tag{20}$$

where D, I, S, and C denote the number of deletions, insertions, substitutions and matches, respectively. When we calculated *WER*, we considered only surface form.

We used *Recall*, *Precision* and *F-measure* that is a harmonic mean of the *Recall* and *Precision* to evaluate the performance of the insertion of periods. These are defined as follows:

$$Recall = \frac{C}{C + D + S}, \ Precision = \frac{C}{C + I}. \tag{21}$$

All parameters were tuned on the development set (CSJ). α, β, γ in (6) and (17) were 0.15, 0.0, 5.0 for the ASR result-based systems and 0.0, 0.3, 0.0 for the manual transcript-based systems. If the value of a parameter is 0.0, it means that the parameter does not contribute the cleaning process. Before the experiments were conducted, we expected that all parameters would have positive values (> 0.0) to help the cleaning process. Contrary to our expectation, the value of parameter for dependency analysis was 0 for ASR results and the values of parameters for LM and bunsetsu estimation were 0.0 for the manual transcripts. Although this result was different from our expectation, it revealed the interesting characteristic of each component. Namely, the scores from LM and bunsetsu estimation were useful to filter out hypotheses which have relatively higher WER, and they were not needed for the manual transcripts since they were already perfect. On the other hand, the scores from dependency analyses on an erroneous ASR results were not consistent with the final WERs of processed transcripts, however, they were somewhat consistent if the processed transcripts were perfect.

We believe that from the view point of machine learning, ideally, the value of each parameter should be automatically determined through an optimization algorithm and we do not need to concern whether each parameter is effective or not a priori. This means that we can use any scores (features) without concerning the adequateness to be used since the adequateness can be judged by the optimization algorithm. In this experiment, although we used the simplest grid search algorithm to optimize the parameters, we can use more sophisticated algorithms if needed.

The dependency analysis on poor ASR results does not make sense, and therefore, it is difficult to show what extent our dependency analysis worked with our ASR results whose WER was 44% since there is no dependency structure on incorrect transcripts. However, please note that our proposed method works on multiple hypotheses which may obtain correct transcripts. We expected that the dependency analysis produced higher scores for hypotheses which had correct dependency structure, however, it may have not been held because the score of a dependency structure was conditional probability computed by Eq. (13) and there was no meaning to compare them if the condition $\boldsymbol{b}$ was different.

5.2 Automatic Evaluation

5.2.1 Comparison with Paraphrased Transcripts

To evaluate the proposed method, we compared different types of transcripts with manually paraphrased transcripts. The WER for the transcripts derived from the manual transcripts (*Manual-**) must not be 0% because our target was the paraphrased transcript. *Punctuation marks were removed from all transcripts for the evaluation.* Significance tests were conducted using the method described in [27].

The results are given in Table 3. We could observe that the *1-best-raw*, which means the single-best ASR result, obviously produced the worst result (64.2% and 49.0% for CJLC and CSJ, respectively). By removing filled pauses from the transcripts, the *Filler* could improve the *1-best-raw* (64.2% $\rightarrow$ 58.4%†† and 49.0% $\rightarrow$ 42.0%††; where † and †† indicate the method was significantly better under significance levels of 5% and 1%, respectively). The *Single (w/o snt.)* outperformed the *Filler* because it could determine and remove disfluencies by utilizing the posterior probabilities from the confusion networks and the N-gram probabilities from clean transcripts (58.4% $\rightarrow$ 55.2%†† and 42.0% $\rightarrow$ 40.9%†). Using multiple hypotheses, the *Multiple (w/o snt.)* outperformed the *Single (w/o snt.)* (55.2% $\rightarrow$ 54.2%†† and 40.9% $\rightarrow$ 40.0%). Furthermore, the *Multiple (w/ snt.)* was superior to the *Single (w/ snt.)*, but there was no significant difference between them (53.8% $\rightarrow$ 53.5% and 39.5% $\rightarrow$ 39.1%). In addition, the sentence-level knowledge benefited the *Single (w/ snt.)* and the *Multiple (w/ snt.)*, and they

Table 3 Evaluation results of the transcripts [%]. *1-best-raw* means the raw transcript of the best recognition result, *Filler* and *Single* are the two baselines that are described in Sect. 4, *Manual-raw* means raw manual transcript, *Manual-filler* means manual transcript that removed filled pauses using the method described in Sect. 4.1, and *Manual-single* means the transcript that was edited from the manual transcript using the method described in Sect. 4.2. w/o snt. and w/ snt. mean whether sentence-level features are used or not, respectively.

Corpus	Method	Del.	Ins.	Subs.	WER
CJLC	1-best-raw	6.8	18.2	39.3	64.2
	Filler	8.7	11.0	38.7	58.4
	Single (w/o snt.)	16.1	7.3	31.8	55.2
	Single (w/ snt.)	18.7	5.0	30.1	53.8
	Multiple (w/o snt.)	15.4	7.0	31.7	54.2
	Multiple (w/ snt.)	16.7	5.7	31.1	53.5
	Manual-raw	3.5	19.8	14.1	37.4
	Manual-filler	4.0	14.4	13.7	32.1
	Manual-single (w/o snt.)	7.6	10.8	15.8	34.2
	Manual-single (w/ snt.)	7.1	10.3	15.0	32.4
CSJ	1-best-raw	4.7	17.8	26.5	49.0
	Filler	6.5	11.1	24.4	42.0
	Single (w/o snt.)	10.0	7.8	23.0	40.9
	Single (w/ snt.)	12.1	5.8	21.6	39.5
	Multiple (w/o snt.)	9.9	7.4	22.7	40.0
	Multiple (w/ snt.)	11.0	6.1	22.0	39.1
	Manual-raw	2.4	22.3	9.5	34.1
	Manual-filler	2.9	14.0	9.2	26.1
	Manual-single (w/o snt.)	4.5	11.5	10.2	26.2
	Manual-single (w/ snt.)	3.9	11.3	9.4	24.6

outperformed the *Single (w/o snt.)* (55.2% → 53.8%†† and 40.9% → 39.5%†) and the *Multiple (w/o snt.)* (54.2% → 53.5%†† and 40.0% → 39.1%), respectively. If the multiple hypotheses or the sentence-level knowledge is used individually, the improvements were not always significant especially for the CSJ, however, by taking into account both multiple hypotheses and sentence-level knowledge, the *Multiple (w/ snt.)* yielded the best performance among the ASR result-based systems and significant improvements against the *Single (w/o snt.)* where the both knowledge is not used (55.2% → 53.5%†† and 40.9% → 39.1%††). These results clearly show that using multiple hypotheses and sentence-level knowledge makes the ASR results closer to manually cleaned transcripts.

We can see that the numbers of deletion significantly increased in the cases of the *Single* and *Multiple* compared to the *1-best-raw* and *Filler*. We believe that the increase of the number of deletion itself was not a problem since the proposed method mainly benefits the sentence cleaning process by removing not only fillers but also low-confidence, unnecessary and redundant words. However, there is a trade-off between the numbers of deletions of those irrelevant words and correct words because the deletion by the proposed system is not perfect. The effectiveness of the deletion can be evaluated by only a subjective test as we did in Sect. 5.3. As the subjective test revealed, it was true that the proposed method occasionally dropped words which were crucial to understand the content of a video and must be retained in the final transcript since it did not consider the importance of the words. By weighting words depending on their importance, the problem might be mitigated in a similar vein with [28]. This will be one of the our future works.

When using the manual transcripts, the *Manual-raw* was the worst one among all methods (37.4% and 34.1%). The *Manual-filler* reduced the WER by removing filled pauses from raw transcripts compared to the *Manual-raw* (37.4% → 32.1%†† and 34.1% → 26.1%††). The *Manual-single (w/o snt.)* also outperformed the *Manual-raw* significantly (37.4% → 34.2%†† and 34.1% → 26.2%††). However, it was infereior to the *Manual-filler* for the CJLC while it was comprable for the CSJ. The *Manual-single (w/ snt.)* improved the *Manual-single (w/o snt.)* by utilizing the sentence level knowledge (34.2% → 32.4%†† and 26.2% → 24.6%††). The *Manual-single (w/ snt.)* was comparable to the best one (*Manual-filler*) for the CJLC while it was the best one among all methods for the CSJ.

Although we trained the language model using the newspaper corpus, we do not have to be restricted to the newspaper corpus to train the language model. For example, transcripts from the Japanese National Diet Record (JNDR) would be appropriate to train the language model since the transcripts are created by cleaning the raw spontaneous transcripts manually by professional stenographers. As an additional experiment, we combined the newspaper corpus (214 M words) and the JNDR (184 M words) [29], and trained a new language model using the combined corpus (214 + 184 = 398 M words). The result is shown in Table 4. With the new language model, our proposed method provided an almost same result for both corpora (53.5% → 53.4% and 39.1% → 39.0%). The small differences were due to the poor translation model we used and the restricted hypothesis space represented by a confusion network, but not our proposed framework.

Table 4 Evaluation results of the transcripts by the proposed method (*Multiple (w/ snt.)*) when using different language models [%]. JNDR stands for the Japanese National Diet Record.

Corpus	LM	Del.	Ins.	Subs.	WER
CJLC	Mainichi	16.7	5.7	31.1	53.5
	+JNDR	16.7	5.8	30.9	53.4
CSJ	Mainichi	11.0	6.1	22.0	39.1
	+JNDR	11.0	6.1	21.9	39.0

Table 5 Evaluation results of the insertion of periods.

Corpus	Method	Recall	Precision	F
CJLC	Utterance	0.562	0.252	0.334
	Single (w/o snt.)	0.475	0.463	0.467
	Single (w/ snt.)	0.457	0.543	0.489
	Multiple (w/o snt.)	0.487	0.421	0.450
	Multiple (w/ snt.)	0.497	0.517	0.501
	Manual-single (w/o snt.)	0.367	0.602	0.455
	Manual-single (w/ snt.)	0.596	0.656	0.620
CSJ	Utterance	0.824	0.232	0.355
	Single (w/o snt.)	0.474	0.466	0.469
	Single (w/ snt.)	0.645	0.545	0.590
	Multiple (w/o snt.)	0.481	0.390	0.429
	Multiple (w/ snt.)	0.674	0.526	0.590
	Manual-single (w/o snt.)	0.469	0.611	0.530
	Manual-single (w/ snt.)	0.746	0.594	0.661

5.2.2 Evaluation of the Insertion of Periods

In this section, we evaluated the ability to insert punctuation marks. To evaluate the ability, we aligned the transcripts with the paraphrased transcripts that contained manually inserted periods and computed *Recall*, *Precision* and *F-measure* for the periods.

Table 5 gives the evaluation results of the insertion of periods. Since for the results of the *1-best-raw*, *Filler* and the *Manual-filler* must be the same, *Utterance* stands for these methods in the table. We regarded utterance boundaries for ASR as sentence boundaries for the *Utterance*. The utterance was defined as a portion between pauses longer than about 200 msec. The *Utterance* was the worst among all methods (0.334 and 0.355 of F for CJLC and CSJ, respectively). The *Single (w/o snt.)* outperformed the *Utterance* by inserting periods depending on the N-gram information from clean transcripts (0.334 → 0.467 and 0.355 → 0.469). Interestingly, in this evaluation, the *Multiple (w/o snt.)* did not provide any improvement over the *Single (w/o snt.)* (0.467 → 0.450 and 0.469 → 0.429). This might be because inserting periods relying only on the N-gram information, which can only capture local cohesion with multiple hypotheses, causes a number of false alarms and leads to severe degradation of the precision. On the other hand,

Table 6 Statistics of the test set for the subjective test. *APP.* means adjusted perplexity. *OOV* indicates the ratio of OOV.

Duration (min.)	#Words		APP.		OOV [%]		#Filler [%]
	Manual	Paraph.	Manual	Paraph.	Manual	Paraph.	
8.18	1654	1410	148.5	121.2	2.2	2.1	8.8

sentence-level knowledge benefited the inserting periods. As a result, the *Single (w/ snt.)* was superior to the *Single (w/o snt.)* (0.467 → 0.489 and 0.469 → 0.590) and the *Multiple (w/ snt.)* outperformed the *Multiple (w/o snt.)* (0.450 → 0.501 and 0.429 → 0.590). When using sentence-level knowledge, the multiple hypotheses approach was also effective and the *Multiple (w/ snt.)* overcomes or is comparable to the *Single (w/ snt.)* (0.489 → 0.501 and 0.590 → 0.590). Therefore, we concluded that multiple hypotheses and sentence-level knowledge are also beneficial for the insertion of periods.

We did not analyze how the readability varies in accordance with the accuracy of the insertion of periods. The analysis would be one of the future works. However, if the accuracy of the insertion of periods is improved, it means that the performance of the automatic insertion of periods approaches the performance of the manual insertion of periods. If we assume that the manual paraphrased transcripts have the best readability, we can expect that the performance improvement of the insertion of periods improves the readability as well.

5.3 Subjective Test

To assess whether the proposed method really improves the readability and the understandability of the ASR result, we conducted a subjective test. The definitions of the criteria are follows:

- Readabilty: How transcripts are read easily before knowing the contents of the transcripts and it did not matter whether the transcripts conveyed the true meanings or not. Therefore, it just assessed the "readaiblity" of the transcripts.
- Understandability: How the transcript conveyed the true meaning of the original raw transcripts and it did not matter how easily they were read.

The subjective test was conducted for two lectures of CJLC by 10 persons. The statistics of the test set are shown in Table 6 and the recognition results, word coverage and word densities of constructed confusion networks of the test set are shown in Table 7. We can confirm that the test set was representative enough by comparing Tables 6 and 7 with Tables 1 and 2. The procedure of the subjective test was as follows:

1. Read transcripts A and B. Each transcript is about 30 lines (about 850 words) and divided into 4 small blocks (about 7 lines), which roughly reflect topics.
2. Compare each block for readability (paired comparison).
3. Read the manual transcript, and then read and understand each transcript again.
4. Compare each block for understandability (paired comparison).

Table 7 WER, Word coverage and density of constructed confusion network of the test set for the subjective test.

Reference	WER [%]	Coverage [%]	Density
Manual	52.0	74.0	4.00
Paraphrased	69.0	70.1	

Table 8 Subjective test results. R and U mean the evaluations of readability and understandability. † and †† indicate statistical significance of the method under the significance level 0.05 and 0.01, respectively. * means that the subjects were different with other experiments.

Eval.	Method		Count		
	A	B	A	B	?
R	1-best-raw	Filler††	8	70	2
	Filler	Single (w/o snt.)††	26	50	4
	*Single (w/o snt.)	Single (w/ snt.)††	20	53	7
	*Single (w/o snt.)	Multiple (w/o snt.)	34	41	5
	Single (w/o snt.)	Multiple (w/ snt.)††	27	51	2
U	1-best-raw	Filler††	8	57	15
	Filler††	Single (w/o snt.)	60	18	2
	*Single (w/o snt.)	Single (w/ snt.)††	4	65	11
	*Single (w/o snt.)	Multiple (w/o snt.)††	3	66	11
	Single (w/o snt.)	Multiple (w/ snt.)††	19	56	5
	Filler	Multiple (w/o snt.)	38	28	14

If there are no preferences between two pairs, subjects are allowed to use "?" to indicate "I could not distinguish them". We deleted linefeeds for periods from those transcripts to avoid any bias caused by existence of periods.

The results of the subjective test are shown in Table 8. In the table, "count" indicates how many times the method was chosen, while "†" and "††" denote that the method achieved significantly better readability or understandability at the 5% and 1% significance level (z-test), respectively. "*" means that the subjects were different with other experiments. The table shows that the *Filler* significantly improved the readability and understandability compared with the *1-best-raw*. This means that we should always remove filled pauses from transcripts. The *Single (w/o snt.)* significantly improved the readability and degraded the understandability compared with the *Filler*. This result indicated that the *Single* could improve the readability because it could remove disfluent parts better than the *Filler*, while it degraded the understandability because the disfluent part detection was not accurate enough and it deleted some important content words that were crucial to understanding the contents.

The comparison between the *Single (w/o snt.)* and the *Single (w/ snt.)* showed that the sentence level knowledge

was effective for the both criteria, while the comparison between the *Single (w/o snt.)* and the *Multiplee (w/o snt.)* showed that the use of multiple hypotheses was effective especially for improving the understandability. By using the both knowledge, the *Multiple (w/ snt.)* improved the *Single (w/o snt.)* for the both criteria. Although it seems that the *Single (w/ snt.)* would be enough to improve the *Single (w/o snt.)* and was not needed to be used with the multiple hypotheses, the superiority of the *Multiplee (w/o snt.)* to the *Single (w/o snt.)* implied that the hypotheses which were not contained the 1-best result helped the cleaning process, and thereby the combining the both knowledge sources would provide better results [†].

If the *Multiple (w/ snt.)* is compared with the *Filler*, we can induce that the *Multiple (w/ snt.)* was superior to the *Filler* in terms of the readability from the relationships *Filler* < *Single (w/o snt.)* and *Single (w/o snt.)* < *Multiple (w/ snt.)* (where $A < B$ means that A is superior to B). On the other hand, we cannot order them in terms of the understandability from the results. However, we might be able to say that at least there would be no significant difference between *Multiple (w/ snt.)* and *Filler* based on the fact that there was no significant difference between the *Filler* and the *Multiple (w/o snt.)* in terms of the understandability. As mentioned in Sect. 5.2.1, our proposed method benefits the sentence cleaning process by removing not only fillers but also low-confidence, unnecessary and redundant words. Therefore, it had a higher risk to deteriorate the understandability of the transcript than the *Filler* by occasionally dropping content words which were needed to understand the lecture. In that context, the proposed method which successfully kept the understandability of the *Filler* has understandability enough.

The manual (subjective) evaluation showed that the proposed method that uses multiple hypotheses and sentence-level knowledge could improve the readability and understandability compared with the method that does not use either multiple hypotheses or sentence-level knowledge.

6. Conclusions

In this paper, we presented a novel method for improving the readability of ASR results that uses the multiple hypotheses method represented by a confusion network for robustness to recognition errors and sentence-level knowledge such as chunking and dependency parsing for accurate sentence boundary detection. To use the sentence-level knowledge with multiple hypotheses, we proposed to use the novel iterative decoding algorithm proposed for confusion network-based rescoring. Experimental results showed that the effectiveness of using both multiple hypotheses and sentence-level knowledge to improve the readability of the ASR results in automatic and manual evaluations.

In the future, we will address taking a more sophisticated transformation model to deal with the correction of colloquial expressions like [10], which are difficult to consider in an unsupervised manner. In addition, if the construction of a confusion network is problematic, for example, in real-time applications, it is better to use lattices instead of confusion networks to represent multiple hypotheses. In such a case, we can use the algorithm described in [30] to incorporate sentence-level knowledge efficiently instead of iterative decoding used in this paper [18]. This will be another direction of research.

Acknowledgements

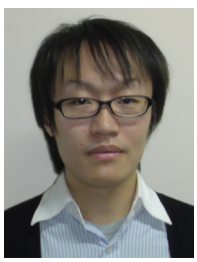
Part of this research was supported by the Global COE Program "Frontiers of Intelligent Sensing" from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] Y. Fujii, K. Yamamoto, N. Kitaoka, and S. Nakagawa, "Class lecture summarization taking into account consectiveness of important sentences," Proc. Interspeech, pp.2438–2441, Sept. 2008.
[2] C. Chelba and A. Acero, "Indexing uncertainty for spoken document search," Proc. Interspeech, pp.61–64, Sept. 2005.
[3] S. Togashi and S. Nakagawa, "A browsing system for classroom lecture speech," Proc. Interspeech, pp.2803–2806, Sept. 2008.
[4] J. Glass, S.C.T.J. Hazen, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT spoken lecture processing project," Proc. Interspeech, pp.2553–2356, Aug. 2007.
[5] S. Kogure, H. Nishizaki, M. Tsuchiya, K. Yamamoto, S. Togashi, and S. Nakagawa, "Speech recognition performance of CJLC: Corpus of Japanese lecture contents," Proc. Interspeech, pp.1554–1557, Sept. 2008.
[6] T. Kawahara, Y. Nemoto, and Y. Akita, "Automatic lecture transcription by exploiting presentation slide information for language model adaptation," Proc. IEEE-ICASSP, pp.4929–4932, 2008.
[7] K. Shitaoka, H. Nanjo, and T. Kawahara, "Automatic transformation of lecture transcription into document style using statistical framework," Proc. Interspeech, pp.2881–2884, 2004.
[8] T. Hori, D. Willet, and Y. Minami, "Paraphrasing spontaneous speech using weighted finite-state transducers," SSPR, pp.210–222, April 2003.
[9] G. Neubig, S. Mori, and T. Kawahara, "A WFST-based Log-linear Framework for Speaking-style Transformation," Proc. Interspeech, pp.1495–1498, 2009.
[10] G. Neubig, Y. Akita, S. Mori, and T. Kawahara, "Improved statistical models for SMT-based speaking style transformation," Proc. ICASSP, Dallas, Texas, USA, March 2010.
[11] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," IEEE Trans. Audio Speech Language Process., vol.14, no.5, pp.1526–1540, Sept. 2006.
[12] A. Stolcke, Y. König, and M. Weintraub, "Explicit word error minimization in n-best list rescoring," Proc. Eurospeech '97, pp.163–166, 1997.
[13] H. Ney and X. Aubert, "A word graph algorithm for large vocabulary continuous speech recognition," Proc. ICSLP '94, pp.1355–1358, 1994.
[14] F.K. Soong and E.F. Huang, "A tree-trellis based fast search for finding the Nbest sentence hypotheses in continuous speech recognition," Proc. ICASSP '91, pp.705–708, 1991.
[15] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech

[†]Please note that the subjects were different among these tests and therefore we need to be careful when comparing the number of counts directly.

recognition: word error minimization and other applications of confusion networkds," Computer Speech and Language, vol.14, no.4, pp.373–400, 2000.
[16] N. Bertoldi and M. Federico, "A New Decoder For Spoken Language Translation based on Confusion Networkds," Proc. ASRU, pp.86–91, 2005.
[17] T. Hori, I.L. Hetherington, T.J. Hazen, and J.R. Glass, "Open-vocabulary spoken utterance retrieval using confusion networkds," Proc. ICASSP, pp.IV-73–IV-76, 2007.
[18] A. Deoras and F. Jelinek, "Iterative decoding: A novel re-scoring framework for confusion network," Proc. ASRU, pp.282–286, 2009.
[19] T. Oba, T. Hori, and A. Nakamura, "Improved sequential dependency analysis integrating labeling-based sentence boundary detection," IEICE Trans. Inf. & Syst., vol.E93-D, no.5, pp.1272–1281, May 2010.
[20] M. Shungrina, "Formatting time-aligned ASR transcripts for readability," Proc. NAACL, pp.198–206, 2010.
[21] M. Tsuchiya, S. Kogure, H. Nishizaki, K. Ohta, and S. Nakagawa, "Developing corpus of Japanese classroom lecture speech contents," Proc. LREC, pp.2061–2065, June 2008.
[22] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," Proc. 18th International Conference on Machine Learning, 2001.
[23] J. Peng, L. Bo, and J. Xu, "Conditional neural fields," Proc. Advances in Neural Information Processing Systems 22, pp.1419–1427, 2009.
[24] T. Oba, T. Hori, and A. Nakamura, "Sequential dependency analysis for online spontaneous speech processing," Speech Commun., vol.50, pp.616–625, July 2008.
[25] S. Furui, K. Maekawa, and H. Isahara, "A japanese national project on spontaneous speech corpus and processing technology," Proc. ASR2000, pp.244–248, 2000.
[26] Y. Fujii, K. Yamamoto, and S. Nakgawa, "Large vocabulary speech recognition system: SPOJUS++," MUSP, March 2011.
[27] S. Nakagawa and H. Takagi, "Statistical methods for comparing pattern recognition algorithms and comments on evaluating speech recognition performance," J. Acoust. Soc. Jpn., vol.50, no.10, pp.849–854, 1994-10-01. (in Japanese).
[28] T. Shichiri, H. Nanjo, and T. Yoshimi, "Minimum bayes-risk decoding with presumed word significance for speech based information retrieval," Proc. ICASSP, pp.1557–1560, 2008.
[29] K. Ohta, M. Tsuchiya, and S. Nakagawa, "Detection of precisely transcribed parts from inexact transcribed corpus," Proc. ASRU, 2011.
[30] A. Rastrow, M. Dreyer, A. Sethy, B. Ramabhadran, and M. Dredze, "Hill climbing on speech lattices: A new rescoring framework," Proc. ICASSP, pp.5032–5035, 2011.



Yasuhisa Fujii graduated from Toyohashi University of Technology with his Bachelor's and Master's degree in 2007 and 2009. Since 2009, he has been studying at the Toyohashi University of Technology as a doctoral student. His research interest is in spoken language processing, signal processing, pattern recognition, and machine learning. He is a member of IPSJ, ASJ, and IEEE.

Kazumasa Yamamoto received his B.E., M.E. and Dr. Eng. degrees in information and computer sciences from the Toyohashi University of Technology, Toyohashi, Japan, in 1995, 1997 and 2000. From 2000 to 2007, he was a research associate in the Department of Electrical and Electronic Engineering, Faculty of Engineering, Shinshu University, Nagano, Japan. Since 2007, he has been an assistant professor in the Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi, Japan. His current research interests include speech recognition and privacy protection for speech signals. He is a member of ASJ, and IPSJ.

Seiichi Nakagawa received Dr. of Eng. degree from Kyoto University in 1977. He joined the faculty of Kyoto University in 1976 as a Research Associate in the Department of Information Sciences. From 1980 to 1983 he was an Assistant Professor, from 1983 to 1990 he was an Associate Professor and since 1990 he has been a Professor in the Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi. From 1985 to 1986, he was a Visiting Scientist in the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, USA. He received the 1997/2001 Paper Award from the IEICE and the 1988 JC Bose Memorial Award from the Institution of Electro, Telecomm. Engrs. His major interests in research include automatic speech recognition/speech processing, natural language processing, human interface and artificial intelligence. He is a Fellow of IPSJ.