

ローカル番組の字幕付与システムの開発

佐藤 庄衛^{1,a)} 尾上 和穂¹ 小林 彰夫¹ 奥 貴裕¹ 一木 麻乃¹ 荒井 孝¹

概要: NHK が地方局展開を進めている字幕付与システムを紹介する。紹介するシステムは番組音声を認識した結果から事前に用意した読み原稿を推定して、人手をかけずに字幕を付与するシステムである。同システムの開発から、生放送番組への字幕付与の現場から寄せられた要望に基づいたシステム設計と実装したアルゴリズムを紹介する。

キーワード: 音声認識, 生放送番組, 字幕付与システム

Recent Developments in Automatic Captioning System for Regional Broadcasting Station

SATO SHOEI^{1,a)} ONOE KAZUO¹ KOBAYASHI AKIO¹ OKU TAKAHIRO¹ ICHIKI MANON¹
ARAI TAKASHI¹

Abstract: This paper describes an automatic captioning system developed for regional broadcasting stations. The developed technology compares the program manuscript with the recognized program speech and instantly identifies the line in the manuscript corresponding to the speech. This makes it possible to eliminate manual correction in the production of closed-captions for broadcasting. This paper also describes our system design based on requirements of our captioning spot.

Keywords: Speech recognition, live broadcasting, automatic captioning system

1. はじめに

音声で伝えられている情報を文字に変換して伝えるテレビの字幕放送は、聴覚障害者やテレビの音の聞き取りが容易ではないと感じる高齢者にとって、放送からの情報を得るための有用なメディアである。NHK は、総務省が掲げた「視聴覚障害者向け放送普及行政の指針」[1] の字幕普及目標の達成に向けて、字幕放送の拡充を進めている。この字幕放送の拡充にあたり、課題となっている番組が生放送番組である。生放送の番組に字幕を付与するためには、番組音声をその場で文字に変換する必要がある。NHK では、番組の性質に応じて、高速入力キーボードなどの人手による方法に加え、音声認識を利用して番組音声から字幕を制作している [2]。

音声認識を利用した字幕制作システムでは、主に番組音声を認識して得られた文字列の認識誤りをオペレータが修正して正確な字幕を放送している。この字幕制作システムは、ニュースやニュースに準じた情報番組の字幕制作であれば、1 名から 2 名の少人数の修正オペレータで運用できるようになった。これまで、この音声認識字幕制作システムは東京だけでなく、大阪、名古屋、福岡、仙台の地域の拠点となる放送局に導入され、それぞれの地域に向けて放送される番組にも字幕が付与されるようになってきた。

一方、これらの放送局以外では、運用コストの課題と、技能を有する修正オペレータの確保の課題から、同システムの導入が容易ではなく、より安価に人手をかけずに運用できるシステムが望まれている。

2. ローカル放送局での字幕制作

ローカル放送局に音声認識字幕制作システムを導入する

¹ NHK
^{a)} satou.s-gu@nhk.or.jp

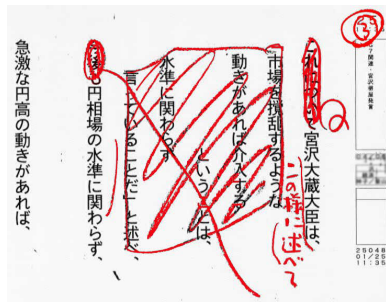


図 1 直前に手書き修正された読み原稿

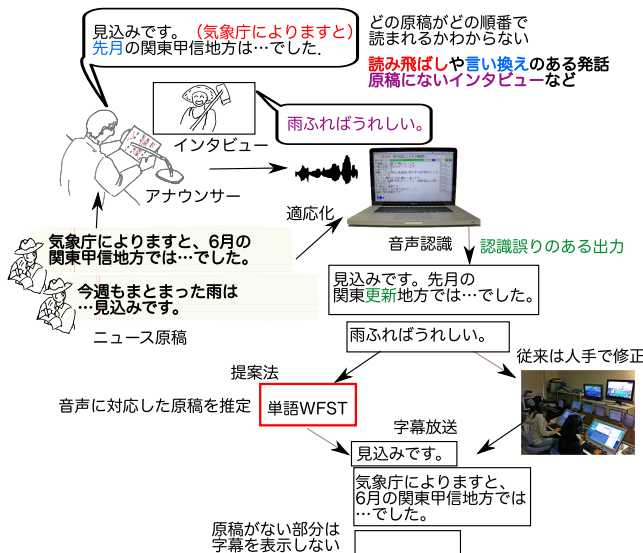


図 2 開発システムにより制作される字幕

際の課題の一つが、認識誤りを発見してそれを即座に修正する技能を有するオペレータの確保である。この即座に素早く行わなければならない作業を軽減し、だれもが余裕を持って行える作業に置き換えるのが本システム開発の課題である。

ニュース番組の多くの項目では、事前に原稿が用意されており、各地の記者が入稿してくる様々な時事に関する原稿がデータベースから参照できるものの、事前にニュース番組中に扱われる時事を確定することは難しい。更に、ニュース番組では、最新の情報に原稿を更新して、番組時間内に原稿を読み終えるために、図 1 のように、原稿を読み替えたり読み飛ばしたりすることがある。また、放送中に、取り上げるニュース項目が予備の項目に変更になったり、原稿を読む順番が変わったりすることもある。さらに、番組中には事前に収録されたインタビューなど原稿がない部分もある。そこで、規模の大きな番組を対象としていた従来のシステムでは、番組音声を認識して認識誤りをオペレータにより修正することでこれらの変更に対応し、番組音声通りの字幕を実現していた。

一方、番組の放送時間の短いローカル放送局のニュースでは、この時間内に複数の項目を放送する必要があるために、ニュースの各項目の大きな変更は難しい。また、番組

内のニュース項目数も少ないため、直前の手書きの原稿修正を手で入力可能である。そこで本開発では、放送音声とほぼ同内容で字幕として放送可能な原稿を用意できることを前提に、システムを構築した。

開発システムでは、番組音声を認識した結果とこの原稿を比較し、どの原稿のどの部分が読まれているかを推定して、その推定原稿を字幕として放送する。これにより、オペレータの作業は、音声を認識してからそれを修正するという迅速さを求められる作業から、音声を認識する前に原稿を修正しておくという時間に余裕のある作業に置き換えられる。この置換えによって、特殊な技能を有するオペレータを確保しなくても、音声認識による字幕制作が実現する。開発システムにより制作される字幕を図 2 に示す。

3. 先行事例

開発システムと同様に原稿の読み位置を推定した字幕制作システムを紹介し、本開発に必要なとされる要件を明らかにする。

スペインの放送局 CARTV では、ニュース項目とその順番、読み原稿が与えられている事を前提とした字幕制作システムを実用化している [3]。このシステムは、原稿の音素列に対する入力音声の強制アライメントを行い、原稿のどの部分が読まれているかを推定し、その原稿文を字幕として放送している。しかし、強制アライメントは、発話音声と音素列に不一致がある場合には著しく精度が低下する。したがって、原稿通り発話された音声では、音素単位で原稿の読み位置を特定できるため、発話から字幕が表示されるまでの遅れ時間が短いという利点があるが、原稿を読み上げる順番に変更があったり、原稿の一部を読み替えたり読み飛ばしたりした場合の精度低下に対応するための仕組みが必要である。

そのため、CARTV のシステムはニュース編集コンピュータシステム (NRCS) と連携した大規模なシステムを構築している。この連携により、放送中に NRCS でニュース項目や項目順に変更がある度にアライメントの音素列が自動的に更新される。しかし、NRCS による音素列の更新が間に合わなかったり、原稿の読み飛ばしや言い換えがあった場合には適切なアライメントができなくなるため、NRCS が管理する番組進行のラップタイムや、ニュース項目の切替タイミング、インサート映像の再生タイミングを利用してアライメント位置を修正している。更に、インサート映像のインタビューなど NRCS では発話内容を管理していない部分もあり、このようなインサート映像の再生時にアライメントが進行しないように、入力を停止するなど複雑な制御も行っている。

このシステムは、現行の番組送出システムに大きな改修を加える必要もあり、システムの開発・導入コストの観点からも、ローカル放送局への導入は現実的ではない。

表 1 本開発システムと強制アライメントシステムの比較

比較項目	アライメント	単語 WFST
簡易な導入	☺	☹
項目の変更	☺	▲
言い換え, 読み飛ばし	☺	▲
字幕表示遅れ	☹→▲	☺

4. WFST を用いた字幕制作

第 2 節で述べた通り, 本開発システムが字幕付与の対象とするローカル放送局の番組は, 放送音声とほぼ同内容で字幕として放送可能な原稿を用意できる. 従って図 2 に示した番組制作フローに基づいて開発システムが取り扱う問題を下記にまとめる.

- 字幕として放送可能な読み原稿が用意できる.
- 放送音声は読み原稿に基づいているが, 読み飛ばしや読み替えがある.
- 読み原稿はどの原稿がどの順番で読まれるか事前に特定できない.
- 読み原稿が用意されていない音声もあるが, この部分に字幕を付与しない.*1
- ニュース番組の場合, 音声認識結果には 5% 以下の認識誤りがある.

上記の条件下で, 適切に字幕を付与するため, 開発システムには音声認識結果の単語列を受理して, その単語列に対応する字幕文を出力する WFST を採用した [4], [5]. 本開発システムと, 強制アライメントに基づく手法の比較を表 1 にまとめた. この WFST は, 従来の字幕制作システムでは人手で認識結果を修正していた部分を置き換える位置づけとなっており, 原稿を用いた音声認識結果の自動修正であるとも言える. また, 音声認識部と字幕送出部には従来のシステムと同じ機器を利用することができるため, 開発コストを低減できる. さらに将来, 原稿を利用できない演出を伴う番組に字幕を付与する際には, この WFST 部を, 従来の人手による修正端末に置き換えることが可能であり, 今後の拡張性も残されている.

4.1 開発システム

図 3 は, ここで構成した WFST の例である. この WFST は各遷移には 音声認識結果の入力単語 w , 字幕として出力する文 c , とその遷移重み s を要素を, $w/c(s)$ の形式で記載した有限状態機械である. ここで, 入出力シンボルに記載された ϵ は, その遷移で入力単語もしくは出力文が無いことを示し, 入力シンボルに記載された $*$ は, 任意の単語を受理可能な遷移であることを示す. 入力シンボルに ϵ

*1 読み原稿が用意されていない部分の例として, インサート映像のインタビュー音声があるが, このような部分では, 映像内に文字を表示したオープンキャプションが付与されるため, クローズドキャプションである字幕放送では送る必要はない.

に記載した遷移は, 認識の脱落誤りと, 原稿の読み飛ばし単語を受理するための遷移であり, 入力シンボルに $*$ を記載した遷移は, 認識の挿入誤りと置換誤りと, 原稿の読み替え単語を受理するための遷移で, それぞれ, 遷移ペナルティ $-p$ が与えられている.

この状態機械は, 時刻 t に音声認識結果の単語 h_t が入力される度に, 各ノード n のスコア $\mathcal{L}_t(n)$ を,

$$\mathcal{L}_t(n) = \max\{(\mathcal{L}_{t-1}(n'(a)) + s_a) \mid a \in \mathcal{A}_n\} \quad (1)$$

を用いて更新し, 探索を進める. ここで, $n'(a)$ は遷移 a の遷移元のノードであり, s_a は遷移 a の遷移重み, \mathcal{A}_n はノード n に流入する遷移の集合である.

出力する字幕文は, 単語 h_t が入力される度に, もしくは, 入力に一定期間以上のポーズがあった時に, 最尤スコアを有するノード,

$$\hat{n}_t = \operatorname{argmax}_{n \in \mathcal{N}} \mathcal{L}_t(n) \quad (2)$$

からそのノードに辿り着いた遷移をトレースバックした遷移 a の系列を求めて確定する. ここで, \mathcal{N} は WFST 上の全てのノードの集合である. このトレースバックで, 既を受理されて字幕文が確定された単語 $h_{t'}$ の時刻 t' から現時刻 t までの a の系列 $v_{t'}^t$,

$$v_{t'}^t = \{a_{t'}, \dots, a_t\} = \{A_{t'}, O_t, A_t\} \quad (3)$$

が得られる. ここで, $A_{t'}$ と A_t は, 出力遷移を含まない遷移区間であり, O_t は出力遷移に挟まれて, 出力遷移毎に分割された遷移区間 o^t の列 $O_t = \{o_1^t, \dots, o_{m_t}^t\}$ である. つまり, o^t は出力遷移に対応する単語列区間となる. ここで, m_t は O_t に内包される出力遷移とそれぞれに対応する区間の数である. 図 4 に o^t の例を示す. この O_t を用いて,

$$\{\forall o \in O_t : E_o < T\} \quad (4)$$

を満たす字幕文出力を確定して, 字幕を放送する. ここで, E_o は, 区間 o の誤りを受理する遷移の通過回数から求めた誤り率であり, 閾値 T は許容する原稿と認識結果の不一致度合いと, 原稿が無い発話と原稿の一致度合いから設定する. この閾値が大きすぎると, 言い換えや読み飛ばし, 認識誤りがあった場合に字幕を付与できなくなり, 小さすぎると, 原稿が無い発話で誤って字幕を付与してしまう. 図 4 の例では, 7 単語中 2 単語が誤りであったため, $E_o = 2/7$ であり, $T = 0.5$ とした場合には, 字幕文「今日の関東地方は晴れです。」が字幕として付与される.

4.2 システム評価

2013 年 6 月 3 日から 14 日に放送された関東ローカル番組「首都圏ニュース 845」9 番組, 460 文, 14,437 単語を対象に, 字幕制作実験を行った [5]. これらの番組は 70 の

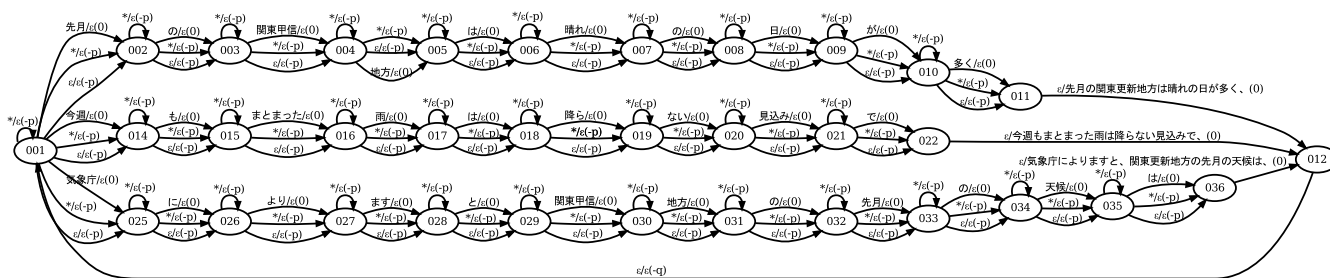


図 3 原稿 WFST の例



図 4 出力する字幕文の確定

表 2 制作された字幕の分類

制作された字幕の分類	文章数
適切・同意	373
不適切	(3)
出力なし	41
判定困難	3

ニュース項目で構成されており、うち 33 項目に複数の版の原稿が用意されていた。これらの版数は、平均は 2.9 版、最大は 7 版であった。

この原稿を用いて言語モデルと語彙を適応化して番組音声認識した所、インタビューなど字幕付与の対象とならない部分も含めて、単語誤り率 6.5% の認識結果が得られた。

この認識結果を開発アルゴリズムに入力し、短い無音を検出される度にトレースバックを行って字幕を付与した。ここでは、 $T = 0.5$ とした結果を表 2 に示す。本開発システムでは、読み替えや読み飛ばしを含む音声を対象に字幕を付与するため、制作された字幕と発話内容との単語誤り率で評価するのが困難である。そこで、制作された字幕が放送音声と同意であり、字幕放送として適切かどうかを基準に制作された字幕文を分類した。

音声認識結果をそのまま字幕とした場合、適切な字幕を付与できたのは、評価の対象となった 420 文中 225 文であったのに対して、開発システムを用いることにより、適切な字幕は 372 文に改善している。不適切と判定された 3 文は、原稿の数字などに誤りがあったものであり、判定困難な 3 文は、複数文にまたがる言い換えがあり、文単位で同意判定が困難だったものである。本システム導入に際しては、稼働中に WFST を最新の情報で更新する仕組みが必要であることが明らかになった。出力なしの 41 文は、全て事前収録のインタビュー部分であり、対応原稿がなく、不適切な字幕を付与しない部分であった。

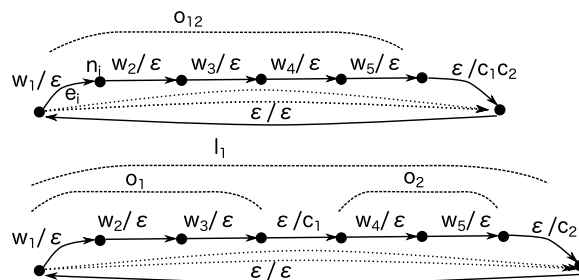


図 5 文中に出力ノードを配置した WFST. 上段は改善前、下段が改善後.

5. 字幕確定の早期化

上記構成では、一つのニュース項目に複数の版の原稿が用意されていても、発話内容に応じて適切な原稿を選択できるが、文が全て読み上げられてからしか字幕を付与できない。一方、字幕付与の現場は、視聴者の要望に応えるため字幕の表示遅れの短縮を望んでおり、本開発システムにも表示遅れの短縮化が必要とされた。

また、上記の実験結果から、原稿を最新の情報に更新するために、放送中も読み原稿を手で確認する必要があることが明らかになった。ここでは、人手で重複する複数の版の原稿を整理して、最新の読み原稿だけを登録できる事を前提に、字幕確定時刻を早期化する手法を紹介する。

5.1 出力記号配置とトレースバック方法の改善

上記の構成では、全てのノード N 中の最尤のノードからトレースバックを行っていた。しかし、字幕出力を確定する主な要因は誤り率 E_0 であり、原稿内の文の類似度が十分に低いとした時には、誤り率 E_0 が小さいパス上に最尤ノードがある事が期待できる。そこで、出力遷移のノード Q に限定してトレースバック開始ノード \tilde{n}_t を

$$\tilde{n}_t = \operatorname{argmax}_{n \in Q} \mathcal{L}_t(n) \quad (5)$$

より求める。さらに、長い文章では、図 5 下段に示すように、文中に出力遷移を適宜配置して、字幕の表示遅れを小さくする。

この場合式 (3) の O_t は、字幕文 l^t で構成され、 $O_t = \{l^t_1, \dots, l^t_{r_t}\}$ に分割できる。ここで、 r_t は O_t に含まれる字幕文数である。さらに、字幕文 l^t は、出力遷移に挟まれ

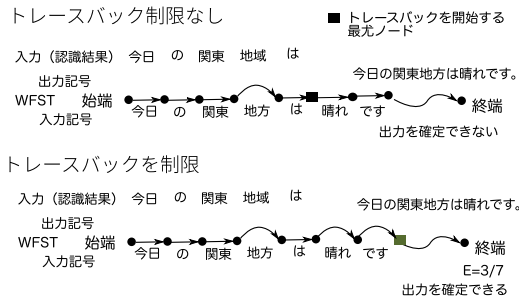


図 6 トレースバックノードの制限による早期確定効果

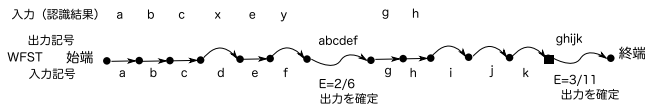


図 7 前方の確定結果を利用した再帰的早期確定

た区間 o_t^k で構成され、 $l_t^k = \{o_1^k, \dots, o_{m_t^k}^k\}$ に分割される。ここで、 m_t^k は文 l_t^k 中の出力遷移の数である。これらの分割を用いて、

$$\{\forall o \in l_t^k : E'(o) < T\} \quad (6)$$

を満たす字幕文出力を確定して字幕を放送する。ここで、 E'_o は、 o_u の単語数を N_u として、

$$E'(o_u) = \begin{cases} 0 & u = 0 \\ 0 & \text{if } E'(o_{u-1}, o_u) < T \\ E'(o_{u-1}, o_u) & \text{otherwise} \end{cases} \quad (7)$$

$$E'(o_{u-1}, o_u) = \frac{N_{u-1}E'(o_{u-1}) + N_u E(o_u)}{N_{u-1} + N_u} \quad (8)$$

である。

トレースバックするノードを制限することにより、図 6 に示す通り、音声認識単語が文の途中までしか入力されていなくても、誤り率 $E < T$ を見込める時には、その文章を確定することができるようになる。

さらに、図 7 に示す通り、文中に出力ノードを配置して式 (7) を適用することにより、文中の前方の区間 o_{u-1} の確定結果を利用して更に早期に字幕を確定可能になる。

5.2 システム評価

原稿を精査できるという条件で、本システムによる字幕制作実験を行った。この実験では、2014 年 4 月最終週に放送された関東ローカル「首都圏ニュース 845」から選んだ 11 のニュース項目を対象とした。ここで、字幕付与の対象となる部分は、108 文、2,302 単語である。用意された事前原稿は、5 単語以下の短い文を接続する文と連結して構成した 100 文、最短 8 単語、最長 61 単語、平均 25 単語であり、原稿内の単語の総計は 2,484 単語である。放送時の読み替えや読み飛ばしのため、原稿の単語数は字幕付与対象部分の単語数より多くなっている。

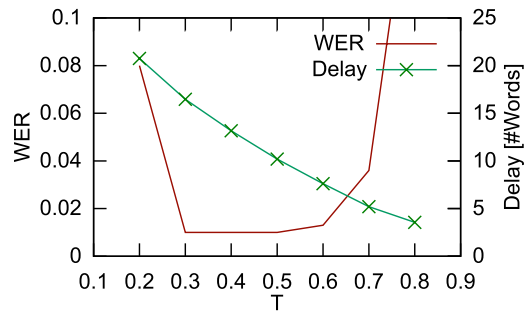


図 8 トレースバック改善による字幕遅延の削減

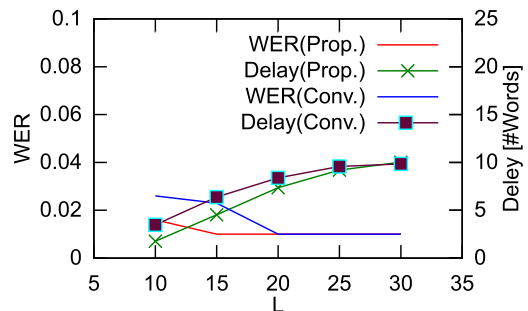


図 9 原稿をチャンクに分割した場合の遅延量削減

この原稿を用いて言語モデルを適応化し、番組音声を認識した結果、字幕付与の対象となる部分で単語誤り率は 3%であった。この認識結果を用いて、4.1 節の方法で字幕を付与したところ、放送音声の書き起こしに対する字幕の単語誤り率は 1%であった。この誤りは、原稿の読み替えと読み飛ばしに起因するもので、文意は同一であり、字幕として適切な出力が得られた。また、認識された単語が字幕として表示されるまでの表示遅延は平均で 31 単語であった。

本節の早期確定システムでトレースバックした場合の字幕の遅延量と単語誤り率を図 8 に示す。横軸は字幕文を確定するための閾値 T である。 $T = 0.5$ とした場合に最も小さい遅延で、誤りのない字幕を付与できた。このとき、遅延量は 4.1 節の方法の 1/3 の 10 単語であった。

図 9 は、原稿を長さ L 単語のチャンクに分割して遅延量の更なる改善を図った実験の結果である。本節の手法に従って字幕の文中に複数の出力遷移を配置した場合を Prop. とし、チャンクを直列に接続せず並列に接続した場合を Conv. とし、チャンクの長さ L を横軸に、単語誤り率と遅延量を記載した。チャンクの長さ L を短くすることにより、どちらの手法でも遅延量が削減される。しかし、並列に構成した場合には、 L を 20 単語以下にすると字幕の精度が低下するため、遅延量を 8 単語までしか削減できない。一方、本節の手法では、字幕の精度を低下させることなく、チャンクの長さ L を 15 単語まで短くすることができ、それにより遅延量を 4.5 単語まで削減できた。

6. 補助機能の開発

6.1 稼働中の WFST の更新機能

最新の情報を伝えるニュース番組では、番組中に数字などの情報が更新される事が多いため、システム稼働中に WFST を更新する仕組みが必要である。この仕組みに要求される条件は、どのタイミングで起こるか分からない情報の更新をいち早くシステムに反映できる事である。進行中のサーチを妨げる事無く WFST を更新するためには、WFST の各ノードに記憶されている通過履歴を保存したまま、遷移を削除したり新たな遷移を追加できなければならない。

開発システムでは、WFST をツリー構造とし、更新文章と同じプレフィックスを有する文が既存のツリーにある場合は、既存の遷移をそのまま利用し、更新部分以降のサフィックスに対応する遷移を新たに構成する。また、更新により無効となった文章は、その文を受理する葉のノードを、トレースバックの対象となるノードの集合 Q から除外する実装とし、効率よく更新に対応できるようにした。

6.2 原稿の連結確認機能

本字幕制作アルゴリズムでは、原稿中に数単語からなる短い文があったり、類似度の高い文対が原稿中にあった場合に、制作される字幕の精度が低下する。例えば、当日のニュース項目を見出し的に紹介する項目には、各ニュースの本記項目と類似する文がある場合がある。これらの文を避けるため、短い文や類似度の高い文を前後の文と連結して一文として、十分な異なりを有する文として整形する必要がある。上記の例では、複数の見出し文を一文にまとめることで、本記の文と十分な異なりを確保できる。更に、文頭のチャンク同士の類似度が高い場合は、チャンクを伸ばして十分な異なりを確保しなければならない。これらの作業は原稿を用意するオペレータが行うが、全ての類似文を手で見つけ出すことは容易ではない。

そこで、本開発システムでは、全ての文対の levenshtein 距離を求め、似ている文対をオペレータに知らせるとともに、全ての文対の levenshtein 距離の最小値を大きくすることのできる文の連結候補をオペレータに提案する。類似する文頭のチャンクについても同様に、levenshtein 距離から類似対をオペレータに知らせるとともに、どこまでチャンクを伸ばしたら十分な異なりを確保できるかを知らせる。

文の連結する際には、読み飛ばされる可能性のある文を他の文と連結するのは不適切であるし、順番が入れ替わる可能性のある文同士を連結するのも不適切である。この不適切の判断は、文意の理解と経験が必要とするため、自動化するのは困難である。開発システムでは、警告と修正案の提示のみを行い、最終判断はオペレータに委ねる構成と

した。

6.3 人手によるバックアップ

開発システムでは、想定外の雑音環境下からのレポートなど、音声認識精度の低下により所望の動作を期待できなくなった場合にも字幕を付与しなければならない。このような場合、最新原稿を入力するオペレータが、手動で原稿を選択して送出できるように構成した。したがって、オペレータは、通常時は原稿の更新を行い、認識精度低下が起こった際には手動で原稿を送出する。本開発では、それぞれを余裕を持って行えるようにシステムを構成した。

7. おわりに

低コストな運用が望まれるローカル番組に字幕を付与するために、番組音声の認識誤りを修正する技能を有するオペレータが不要な字幕制作システムを開発した。開発システムは、字幕として放送可能な読み原稿を用意できる事を前提に、読み原稿がどの順番で読まれるか特定されない原稿を、読み飛ばしや読み替えを交えて発話した放送音声を認識した結果から、音声認識誤りに頑健に読み原稿を特定して、読み原稿を字幕として放送するシステムである。開発システムでは、オペレータに技能を要求せずに、だれでも余裕を持って字幕制作作業を行える。さらに、生放送の字幕付与の課題であった字幕表示遅れを大幅に短縮した。また、開発システムの稼働に必要な補助機能を紹介した。今後、実環境下でのテストを繰り返すとともに、実用化を目指す。

参考文献

- [1] 総務省:視聴覚障害者向け放送普及行政の指針見直し版 (online) 入手先 (http://www.soumu.go.jp/main_content/000254131.pdf) (2014.09.25).
- [2] 今井亨, 奥貴裕, 小林彰夫: 音声認識によるリアルタイム字幕放送の進展, 情報処理学会研究報告, Vol. SLP-88, No. 4 (2011).
- [3] Garcia J.E., Ortega A., Lleida E., Lozano T., Bernues E., Sanchez D.: Audio and text synchronization for TV news subtitling based on Automatic Speech Recognition, BMSB '09. IEEE International Symposium on, pp. 1-6 (2009)
- [4] 佐藤庄衛, 尾上和穂, 小林彰夫, 奥貴裕, 藤田悠哉, 一木麻乃, 荒井孝: ローカル番組への字幕付与方式の提案, 映像情報メディア学会年次大会, 8-2 (2014)
- [5] 佐藤庄衛, 尾上和穂, 小林彰夫, 奥貴裕, 藤田悠哉, 一木麻乃: 原稿 WFST を利用した誤認識修正アルゴリズム, 音響学会春季研究発表会講演文集, 1-8-8, pp. 25-26 (2014)
- [6] 佐藤庄衛, 尾上和穂, 小林彰夫, 奥貴裕, 藤田悠哉, 一木麻乃: 原稿 WFST を利用した誤認識修正アルゴリズム, 音響学会春季研究発表会講演文集, 2-4-8, pp. 65-66 (2014)