リアルタイム字幕生成のための音声ドキュメントへの改行挿入

大野 誠寛† 村田 匡輝‡ 松原 茂樹§

†名古屋大学大学院国際開発研究科
‡名古屋大学大学院情報科学研究科 §名古屋大学情報連携基盤センター
E-mail: ohno@nagoya-u.jp

概要

リアルタイム字幕生成とは，講演や解説などの音声ドキュメントをテキストで提示するものであり，聴覚障害者や高齢者，外国人らによる講演音声の理解を支援するための技術である．講演では一文が長くなる傾向にあり，多くの文がスクリーン上で複数行にまたがって表示されることになるため，テキストが読みやすくなる位置に改行が挿入されている必要がある．本論文では，読みやすい字幕を生成するための要素技術として，日本語講演文への漸進的な改行挿入手法を提案する．本手法では，係り受け，節境界やポーズ，行長などの情報に基づき，統計的手法によって漸進的に改行位置を決定する．日本語講演データを使用した実験によって本手法の有効性を確認した．

キーワード 音声言語，文解析，リアルタイム字幕生成，係り受け構造，節境界，音声コーパス

Linefeed Insertion into Spoken Document for Real-time Captioning

Tomohiro Ohno† Murata Masaki‡ Shigeki Matsubara§

†Graduate School of International Development, Nagoya University
‡Graduate School of Information Science, Nagoya University
§Information Technology Center, Nagoya University
E-mail: ohno@nagoya-u.jp

Abstract

The development of a captioning system that supports the real-time understanding of spoken documents such as lectures and commentaries is required. In monologues, since a sentence tends to be long, each sentence is often displayed in multi lines on the screen, it is necessary to insert linefeeds into a text so that the text becomes easy to read. This paper proposes a technique for incrementally inserting linefeeds into a Japanese spoken monologue as an elemental technique to generate the readable captions. Our method appropriately and incrementally inserts linefeeds into a sentence by machine learning, based on the information such as dependencies, clause boundaries, pauses and line length. An experiment using Japanese speech data has shown the effectiveness of our technique.

key words spoken language, sentence analysis, real-time captioning, dependency structure, clause boundary, speech corpus

1 はじめに

リアルタイム字幕生成とは，講演などの音声ドキュメントをテキストで提示するものであり，聴覚障害者や高齢者，外国人らによる音声理解を支援することを目的とする．近年，字幕の自動生成の実現を目指した研究がいくつか行われており [1]，字幕生成のための音声認識技術について検討が進んでいる [2, 3, 4]．しかしながら，読みやすい字幕を生成するためには，音声を精度よく文字化することだけでなく，文字化されたテキストをどのように提示するかということもまた重要となる [5]．特に，講演では文が長くなる傾向にあり，一文が字幕スクリーン上で複数行にまたがって表示されることになるため，提示されたテキストが読みやすくなるように，適切な箇所に改行が挿入されていることが望まれる．

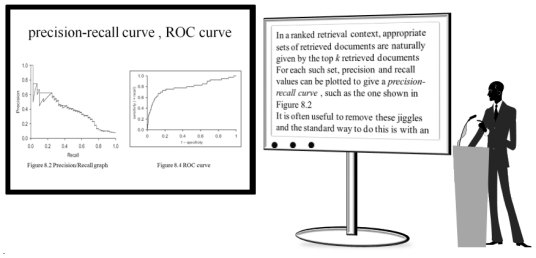

図 1: 講演音声の字幕提示環境

これまでに著者らは，日本語講演音声の書き起こし文への改行挿入手法を提案している [6]．この手法では，節境界，係り受け関係，ポーズ，行長などの情報を用いた統計的手法により，読みやすい位置への適切な改行挿入を実現している．しかし，文を入力単位として改行挿入位置を同定しているため，1 文が長くなる傾向にある講演データでは特に，音声が入力されてから改行位置が同定され字幕が出力されるまでの遅延時間の増加が問題となっていた．また，講演データにおける文境界の判定は難易度が高く，文境界が既知であることを前提とした手法となっていることも課題として残されていた，

そこで，本論文では，読みやすい字幕をより同時的に生成するための基盤技術として，日本語講演音声の書き起こし文への漸進的な改行挿入手法を提案する．本研究では，講演会場での聴衆への字幕情報の提供手段として，字幕のみが複数行表示されるディスプレイの設置を想定している．本手法では，講演全体の文節列を入力とし，節境界が検出されるごとに，それまでに入力された文節列の各文節境界に対して，改行位置を同定する．従来の文単位の改行手法と同様に，節境界，係り受け関係，ポーズ，行長などの情報を用いた統計的手法により，意味的なまとまりを考慮して改行位置を決定するだけなく，節ごとの漸進的な改行位置同定を実現する．

日本語講演データを用いて改行挿入実験を行った結果，人手で改行位置を付与した正解データに対して，再現率で 76.28%，適合率で 73.80%を達成した．文単位の従来手法と比較して，改行位置の再現率・適合率をそれほど低下させることなく，遅延時間を大幅に改善しており，本手法の有効性を確認した．

2　講演テキストへの改行挿入

本研究では，講演会場における字幕提示環境として，プレゼンテーションスライドを表示するスクリーンに併設された，字幕テキスト表示専用のディスプレイの利用を想定する．図 1 に，想定する字幕提示環境を示す．

テレビ番組のクローズドキャプションの場合，通常，画面下部に 2 行程度の字幕が表示され，発声の進行に合わせて表示が切り替わる．一方，本研究では，テキストが行単位で入れ替わり，スクロールしながら常に数行表示される字幕提示システムの利用を前提とする．

例えば環境の問題あるいは人口の問題エイズの問題などなど地球規模の問題たくさん生じておりますが残念ながらこれらの問題は二十一世紀にも継続しあるいは悲観的な見方をすればさらに悪くなるという風に思われます

図 2: 講演音声の書き起こしテキスト

例えば環境の問題
あるいは人口の問題
エイズの問題などなど
地球規模の問題たくさん生じておりますが
残念ながらこれらの問題は
二十一世紀にも継続し
あるいは悲観的な見方をすれば
さらに悪くなるという風に思われます

図 3: 適切な位置に改行が挿入されたテキスト

図 2 に示すように，音声の書き起こしテキストを，改行位置を考慮することなくディスプレイの幅に合わせて表示すると，読みにくいテキストとなる．特に，字幕テキストでは，話者の発声スピードに合わせて読むことが強いられるため，図 3 に示すように読みやすい位置で改行されていることは重要である．

本研究では，字幕生成における改行挿入位置について，以下の前提を設けた．

- ディスプレイの大きさを考慮した行の最長文字数を設定し，各行の文字数をそれ以下とする．
- 日本語では，文節は意味のまとまりの基本単位であることを考慮し，文節境界を改行位置の候補とする．

なお，本論文の以下では，改行が挿入される文節境界を**改行点**（linefeed point）という．

3　節境界に基づく漸進的改行挿入

本手法では，1 講演分の文節列が 1 文節ずつ入力され，節境界が検出されるごとに，それまでに入力された文節列中の各文節境界に対して改行を挿入するか否かを同定し，その結果に従って字幕を漸進的に出力する．

本手法の流れを図 4 を参照しつつ以下に示す．

1. **入力**
   1 講演分の文節列が 1 文節ずつ入力される．図 a) は，入力前の状態を示す．

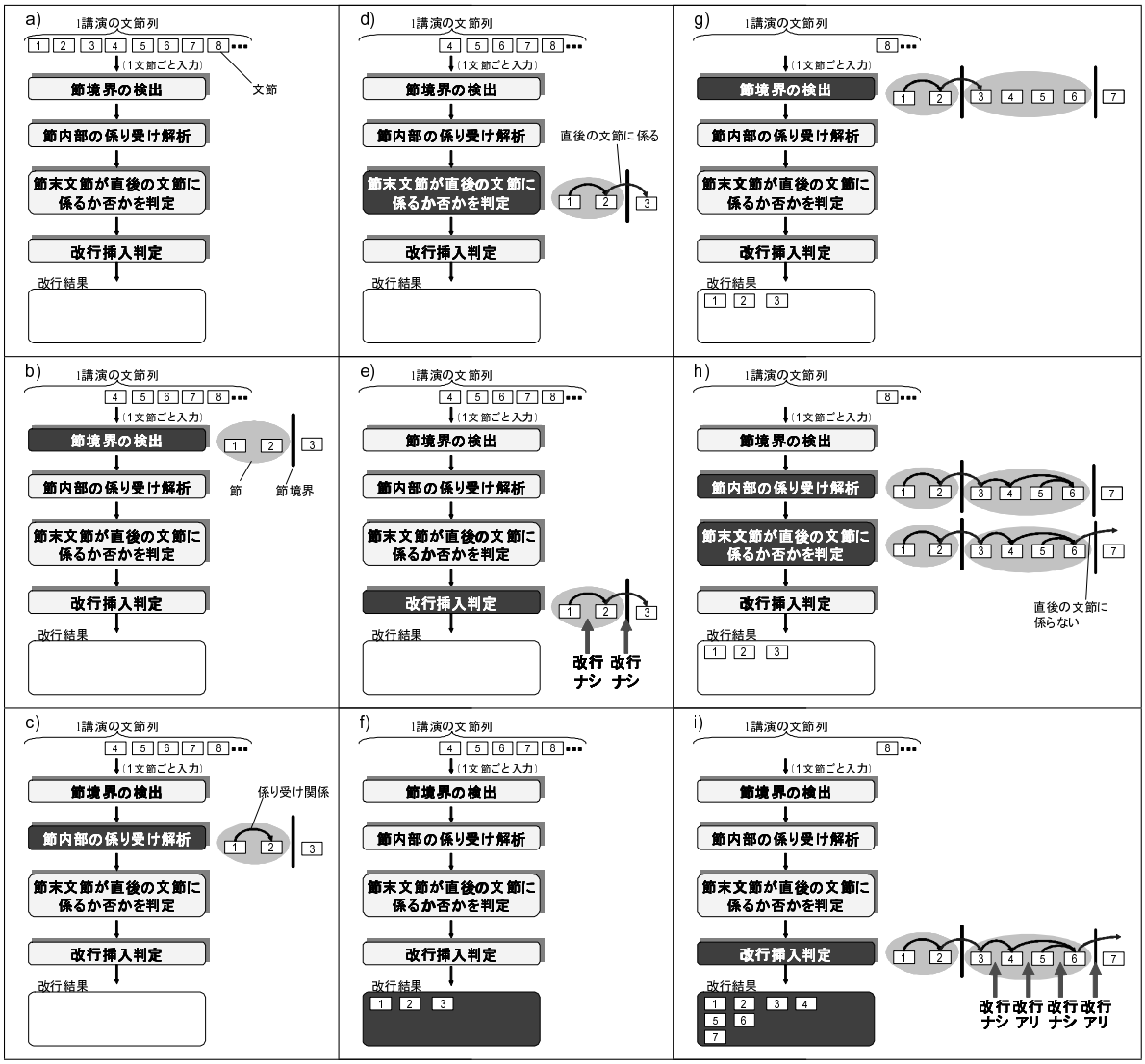

図 4: 漸進的改行挿入の流れ

2. **節境界の検出**
文節が入力されるごとに，節境界解析ツール CBAP[7] を用いて各文節境界が節境界であるか否かを随時判定する．図 b) は，文節 3 が入力されたときに，文節 2 と文節 3 の間に節境界を検出した状態を示す．

3. **節内部の係り受け解析**
節境界が検出され節が同定される度に，その節内部の文節列に対して節境界に基づく係り受け解析手法 [8] を用いて係り受け解析を実行する．図 c）は，文節 1 と文節 2 からなる節の内部の係り受け構造が同定された状態を示す．

4. **節末文節が直後の文節に係るか否かを判定**
節内部の係り受け解析が終わると，節末文節が直後の文節に係るか否かを判定する．節末文節の場合，節内文節の場合とは異なり，係り先がまだ入力されていない可能性が高いため，漸進性を損なわない範囲で出来る限りの係り受け情報を獲得することを考えて，直後の文節に係るか否かの判定のみ行うこととした．なお，この判定は最大エントロピー法を用いて行った．素性は，節境界に基づく係り受け解析手法 [8] において係り受け確率を最大エントロピー法を用いて推定する際に利用されてた素性とほぼ同様のものを用いた[1]．図 d）は，節末文節 2 が直後の文節 3 に係ると判定された状態を示す．

5. **改行挿入判定**
節末文節が直後の文節に係るか否かを判定が終了すると，改行挿入判定が行われていない各文節境界に対して，係り受けや節境界，ポーズ，行長などの情報に基づき，統計的手法によって改行を挿入するか否かを判定する．図 e）は，文節 1 から文節 3 までの各文節境界に対して，改

[1] ポーズ情報の素性を追加した（3.1.2 節参照）．

行を挿入するか否かが判定された状態を示す．

6. **改行結果の出力 (字幕出力)**
   改行挿入判定が終了すると同時に，その改行挿入結果に従って，まだ出力されていない文節列を字幕として出力する．図 f) は，文節 1 と 2，文節 2 と 3 の文節境界には，ともに，改行は挿入されないという結果に従って，文節 1 から 3 までを 1 行にして出力する．

なお，図 g) から図 i) は，図 a) から図 f) と同様に，文節 7 まで入力されたときに行われる処理を示している．

本章の以下では，上述の「5. 改行挿入判定」について詳述する．

3.1 改行挿入判定

改行挿入判定処理の入力は，1 講演の最初の文節から，その時点で検出された節境界の直後の文節までの文節列とする．例えば，図 4-i) における改行挿入判定処理では，文節 1 から文節 7 までの文節列が入力文節列となる．この入力に対して，改行挿入判定済みの結果を覆さない，かつ，1 行あたりの文字数が最長文字数を超えないという条件の下，入力文節列中に挿入されうる改行点の全ての組み合わせの中から，最適な組み合わせを確率モデルを用いて決定する．

以下では，n 個の文節からなる入力文節列を $B = b_1 \cdots b_n$ とするとき，改行結果を $R = r_1 \cdots r_n$ と記す．ここで，r_i は，文節 b_i の直後に改行が挿入されるか（$r_i = 1$）否か（$r_i = 0$）のいずれかの値をとる．なお，計算の都合上，$r_n = 1$ とみなして計算し，文節 b_n の直後に改行が挿入されるか否かは，この時点では決定せず，次の節境界が検出された時点で決定する．入力文を m 行に分割した j 行目の文節列を $L_j = b^j_1 \cdots b^j_{n_j} (1 \leq j \leq m)$ とした場合，$1 \leq k < n_j$ のとき $r^j_k = 0$，$k = n_j$ のとき $r^j_k = 1$ となる．

3.1.1 改行挿入のための確率モデル

本手法では，入力文の文節列を B とするとき，$P(R|B)$ を最大にする改行挿入結果 R を求める．各文節境界に改行が挿入されるか否かは，直前の改行点を除く，他の改行点とは独立であると仮定すると，$P(R|B)$ は次のように計算できる．

$$
\begin{aligned}
& P(R|B) \qquad (1)\\
= \; & P(r^1_1 = 0, \cdots, r^1_{n_1-1} = 0, r^1_{n_1} = 1, \cdots,\\
& \quad r^m_1 = 0, \cdots, r^m_{n_m-1} = 0, r^m_{n_m} = 1|B)\\
\cong \; & P(r^1_1 = 0|B) \times \cdots\\
& \times P(r^1_{n_1-1} = 0|r^1_{n_1-2} = 0, \cdots, r^1_1 = 0, B)\\
& \times P(r^1_{n_1} = 1|r^1_{n_1-1} = 0, \cdots, r^1_1 = 0, B) \times \cdots\\
& \times P(r^m_1 = 0|r^{m-1}_{n_{m-1}} = 1, B) \times \cdots\\
& \times P(r^m_{n_m-1} = 0|r^m_{n_m-2} = 0, \cdots, r^m_1 = 0,\\
& \quad r^{m-1}_{n_{m-1}} = 1, B)\\
& \times P(r^m_{n_m} = 1|r^m_{n_m-1} = 0, \cdots, r^m_1 = 0, r^{m-1}_{n_{m-1}} = 1, B)
\end{aligned}
$$

ここで，$P\,(r^j_k = 1|r^j_{k-1} = 0, \cdots, r^j_1 = 0, r^{j-1}_{n_{j-1}} = 1, B)$ は，1 文の文節列 B が与えられ，$j-1$ 行目の行末位置が同定されているときに，文節 b^j_k の直後に改行が挿入される確率を表す．同様に，$P\,(r^j_k = 0|r^j_{k-1} = 0, \cdots, r^j_1 = 0, r^{j-1}_{n_{j-1}} = 1, B)$ は，文節 b^j_k の直後に改行が挿入されない確率を表す．ただし，計算の都合上，$P(r^m_{n_m} = 1|r^m_{n_m-1} = 0, \cdots, r^m_1 = 0, r^{m-1}_{n_{m-1}} = 1, B) = 1$ として計算する．これらの確率を最大エントロピー法により推定した．最尤の改行結果は，式 (1) の確率を最大とする改行結果であるとして動的計画法を用いて計算する．

3.1.2 最大エントロピー法で用いた素性

本研究では，$P\,(r^j_k = 1|r^j_{k-1} = 0, \cdots, r^j_1 = 0, r^{j-1}_{n_{j-1}} = 1, B)$ ならびに $P\,(r^j_k = 0|r^j_{k-1} = 0, \cdots, r^j_1 = 0, r^{j-1}_{n_{j-1}} = 1, B)$ を最大エントロピー法により推定する際，以下に示す素性を用いた．なお，これらの素性は，改行挿入に有効な素性に関する分析結果 [6] に基づいて設定した．

形態素情報

- 文節 b^j_k の主辞（品詞，活用形）と語形（品詞）

節境界情報

- b^j_k の直後に節境界があるか否か
- b^j_k の直後の節境界のラベル（節境界がある場合）

係り受け情報

- b^j_k が直後の文節に係るか否か
- b^j_k が直前の文節から係られるか否か
- b^j_k が連体節の節末文節から係られるか否か
- 行頭文節 b^j_1 から b^j_k までの間で係り受けが閉じているか否か
- b^j_k が節末文節に係るか否か（b^j_k が節内文節である場合のみ利用）
- b^j_k が行頭からの文字数が最大表示文字数以内の位置にある文節に係るか否か（b^j_k が節内文節である場合のみ利用）

行長

- 行頭から b^j_k までの文字数が以下の 3 分類のいずれであるか

- 2 文字以下
- 3 文字以上 6 文字以下
- 7 文字以上

ポーズ情報

- b_k^j の直後のポーズ時間が以下の 3 分類のいずれであるか
  - 0.2 秒未満
  - 0.2 秒以上 1.0 秒未満
  - 1.0 秒以上 3.0 秒未満
  - 3.0 秒以上

文節の第一形態素

- b_k^j の直後の文節の第一形態素の基本形が「する，なる，思う，問題，必要」のいずれか，もしくはその品詞が「名詞-非自立-一般，名詞-非自立-副詞可能，名詞-ナイ形容詞語幹」のいずれかであるか否か

4 実験

本手法の有効性を評価するため，日本語講演データを用いて改行挿入実験を実施した．

4.1 実験概要

実験データとして，名古屋大学同時通訳データベース [9] に収録されている日本語講演音声の書き起こしデータを使用した．すべてのデータに，形態素情報，係り受け情報，節境界情報が人手で付与されている．実験は，全 16 講演を用いた交差検定により実施した．すなわち，1 講演をテストデータとし，残りの 15 講演を学習データとして改行点の同定処理を実行した．ただし，16 講演のうち 2 講演は事前分析データとして使用したため評価データから取り除き，残りの 14 講演（20,707 文節）に対する実験結果に基づいて評価した．正解の改行データは，人手で改行を付与することにより作成した．正解データの例を図 5 に示す．なお，実験のための最大エントロピー法のツールとしては，文献 [10] のものを利用した．オプションに関しては，学習アルゴリズムにおける繰り返し回数を 1,000 に設定し，それ以外はデフォルトのまま使用した．また，一行の最長文字数を 20 文字とした．

4.2 評価指標

本論文では，各手法の改行挿入位置を評価するため，以下の指標を用いた．

$$\text{再現率} = \frac{\text{正しく挿入された改行数}}{\text{正解の改行数}}$$

それから二番目に
先程伊藤さんからもお話ございましたように
今年は終戦五十年ということで
特別の年でございますので
それに関することを
若干話させて頂きたいと思います

それから現在我々が住んでおります
冷戦後の世界というものは
どういうものかという点につきまして
私の考えを述べさせて頂きたいと思います

図 5: 正解データの例

$$\text{適合率} = \frac{\text{正しく挿入された改行数}}{\text{挿入された改行数}}$$

$$F\text{ 値} = \frac{2 \times \text{再現率} \times \text{適合率}}{\text{再現率} + \text{適合率}}$$

また，文節ごとに，入力時間と出力時間の差を遅延時間として測定し，各手法の漸進性を評価した．ここで，各文節の入力時間は文節の発話終了時間，出力時間は直前の文節境界の改行挿入決定時間とした．ただし，改行挿入決定時間は，解析時間を無視して測定しており，改行挿入するか否かが決定された時に入力された文節の発話終了時間と一致する．なお，各文節の発話終了時間は，連続音声認識エンジン Julius[11] を用いて付与した．

4.3 実験結果

本手法の再現率と適合率を表 1 に示す．なお，比較のために，同様の実験環境下で行われた，文献 [6] における文単位の改行挿入手法（以下，従来手法）の結果についても示す．従来手法の結果は，係り受け解析に，本手法とは異なり，CaboCha[12] が利用されたものであるため，単純には比較できないが，本手法は，再現率と適合率ともに，従来手法を下回った．しかし，本手法は，文境界が未知であることを前提にし，かつ，漸進的に改行点を同定していることを考慮すると，それほど大幅に再現率と適合率が低下しておらず，本手法の有効性を確認した．

次に，各文節の遅延時間の累積割合を図 6 に示す．横軸は遅延時間を，縦軸はその遅延時間未満で出力される文節の全文節数に対する割合を示している．本手法の場合，全体の約 9 割が約 4 秒未満の遅延時間であったのに対し，従来手法の場合，遅延時間が 4 秒未満であった文節数は全体の半数程度であった．本手法は，従来手法と比べて，遅延時間が大幅に短縮していることが分かる．なお，平均遅延時間（=遅延時間の総和/総文節数）は，本手法が 1.58 秒，従来手法が 7.08 秒であった．

表 1: 実験結果

	再現率	適合率	F 値
本手法	79.35% (5,711/7,197)	74.90% (5,711/7,625)	77.06
従来手法	82.71% (5,953/7,197)	80.80% (5,953/7,368)	81.74

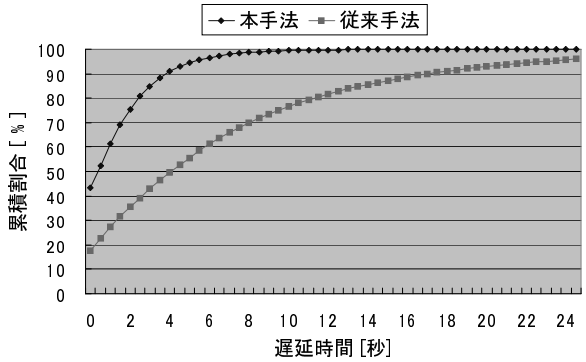

図 6: 遅延時間と累積割合

以上から，本手法は，文単位の改行挿入手法と比べて，改行挿入の再現率・適合率をそれほど低下させることなく，遅延時間を大幅に改善できることを確認した．

5 おわりに

本論文では，聴覚障害者，高齢者，外国人等による音声理解の支援を目的に，日本語講演データへの漸進的な改行挿入手法を提案した．本手法では，係り受け，節境界，ポーズ，行長等の情報に基づき，統計的手法によって読みやすい位置への節単位での漸進的な改行挿入を実現する．日本語講演の書き起こしデータを用いた改行挿入実験では，再現率で 79.35%，精度で 74.90%を示しており，本手法の有効性を確認した．

本論文では，講演の書き起こしテキストに対して，適切な位置に改行を挿入する手法について述べたが，実際のリアルタイム字幕生成に応用するためには，音声認識結果の利用を前提とした，より実践的な方式を検討する必要がある．

参考文献

[1] 今井亨，宮本晃太郎，“放送・教育における音声を利用した障害者支援，” 信学誌，vol.91, no.12, pp.1024-1029, 2008.

[2] G. Boulianne, J.-F. Beaumont, M. Boisvert, J. Brousseau, P. Cardinal, C. Chapdelaine, M. Comeau, P. Ouellet and F. Osterrath, “Computer-Assisted Closed-Captioning of Live TV Broadcasts in French,” Proc. 9th ICSLP, no.Mon2A2O-1, pp.273-276, 2006.

[3] J. Xue, R. Hu and Y. Zhao, “New Improvements in Decoding Speed and Latency for Automatic Captioning,” Proc. 9th ICSLP, no.Wed1CaP-8, pp.1630-1633, 2006.

[4] C. Munteanu, G. Penn and R. Baecker, “Web-Based Language Modelling for Automatic Lecture Transcription,” Proc. 8th Interspeech, no.ThD.P3a-2, pp.2353-2356, 2007.

[5] 中野聡子，牧原功，金澤貴之，中野泰志，新井哲也，黒木速人，井野秀一，伊福部達，“音声認識技術を用いた聴覚障害者向け字幕呈示システムの課題 －話し言葉の性質が字幕の読みに与える影響－，” 信学論（D），vol.J90-D, no.3, pp.808-814, 2007.

[6] 村田匡輝，大野誠寛，松原茂樹，“講演テキストにおける読みやすさを考慮した改行位置同定，” 情報処理学会研究報告，vol.NL-188，pp.37-44, 2008.

[7] 丸山岳彦，柏岡秀紀，熊野正，田中英輝，“日本語節境界検出プログラム CBAP の開発と評価，” 自然言語処理，vol.11, no.3, pp.39-68, 2004.

[8] T. Ohno, S. Matsubara, H. Kashioka, T. Maruyama, H. Tanaka, Y. Inagaki, “Dependency Parsing of Japanese Monologue Using Clause Boundaries,” Language Resources and Evaluation, vol.40, no.3-4, pp.263-279, 2007.

[9] S. Matsubara, A. Takagi, N. Kawaguchi and Y. Inagaki, “Bilingual Spoken Monologue Corpus for Simultaneous Machine Interpretation Research,” Proc. 3rd LREC, pp.153-159, 2002.

[10] L. Zhang, “Maximum entropy modeling toolkit for python and c++,” http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html, 2007, [Online; accessed 6-September-2007].

[11] 河原達也，李晃伸，“連続音声認識ソフトウエア Julius，” 人工知能学会誌，vol.20，no.1，pp.41–49，2005.

[12] T. Kudo and Y. Matsumoto, “Japanese Dependency Analyisis using Cascaded Chunking,” Proc. 6th CoNLL, pp.63-69, 2002.