Vol.2020-AAC-12 No.3
2020/3/13

自動字幕作成システムにおけるモデルの拡張

秋田 祐哉[1]　上乃 聖[2]　三村 正人[2]　河原 達也[2]

概要：我々は，講演や講義などへの効率的な字幕付与を実現するために，音声認識を用いた自動字幕作成システム（サーバ）を運用している．本システムでは，サーバ上で認識用のモデルを対象の音声に適応させた上で認識を実施することが可能で，高精度な音声認識のためにはこの適応は必須となっている．ただし，最近では音声認識もニューラルネットワークによる枠組みが一般的に用いられているが，本システムでは言語モデルについては適応処理が従来のモデルほど容易ではないため，従来の枠組みを利用し続けてきた．今般，本システムでもニューラルネットワーク言語モデルとその適応の枠組みを導入したので，本稿で報告する．

An Expansion of Models in Automatic Captioning System

Yuya Akita[1]　Sei Ueno[2]　Masato Mimura[2]　Tatsuya Kawahara[2]

Abstract: We have been operating an automatic captioning system using the automatic speech recognition (ASR) technology for efficient captioning of lecture and classroom speeches. The system can perform ASR with adapted models to the target speech, which are essential for better ASR performance. Recently, neural network models are commonly used in many ASR systems, while our system still uses the traditional framework for the language model, as a neural network language model (NNLM) is not easy to adapt, compared to the traditional one. In this report, we describe the introduction of NNLM and its adaptation into our system.

1. はじめに

字幕や文字通訳のために音声を文字に書き起こすことは，熟練が必要で，かつ時間的なコストもかかる作業であることから，音声認識技術を用いて自動的に音声を書き起こして字幕や文字通訳に利用する取り組みが進んできている．たとえば，音声認識を用いた自動的な字幕作成はYouTube[*1]で行われている．また音声認識による文字通訳としては，UD トーク[*2]やこえとら[*3]などのアプリケーションがある．我々も，音声認識を用いたオンラインの自動字幕作成システムを運用している [1][*4]．我々のシステムでは，認識対象に合わせてモデルをカスタマイズした上で音声認識を実行できるのが特徴である．音声認識では，特殊な表現や専門用語などの認識を実現するためには，あらかじめ認識のモデル（言語モデル）を対象の音声に適応させておかなければならない．本システムでは，アップロードされた音声と関連テキストから自動的に音声認識がセットアップ・実行され，字幕が作成される．

近年では，音声認識でもニューラルネットワークの技術が一般的に用いられている．しかし本システムでは言語モデルについては従来の統計的言語モデル（N-gram モデル）を使用してきた．この理由として，本システムでは単語の追加などの言語モデルのカスタマイズを前提としているのに対して，ニューラルネットワーク言語モデルでは単純な単語の追加であっても原理的にはネットワークの再学習が必要となることがある．ニューラルネットワーク言語モデルは学習に時間がかかるため，本システムのように処理時間を長くとることができない場合は，モデルの規模や学習

[1] 京都大学 大学院経済学研究科
Graduate School of Economics, Kyoto University
[2] 京都大学 大学院情報学研究科
Graduate School of Informatics, Kyoto University

[*1] https://www.youtube.com/
[*2] https://udtalk.jp/
[*3] http://www.koetra.jp/
[*4] http://caption.ist.i.kyoto-u.ac.jp/

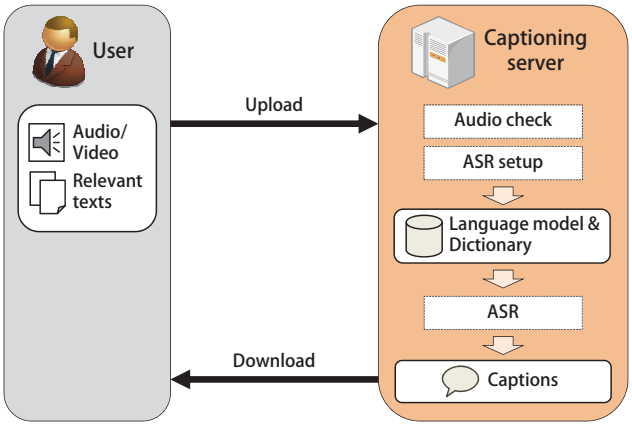

図 1 システムの利用の流れ

の程度に慎重な配慮が必要である．これらの観点から本研究ではニューラルネットワーク言語モデルの導入について検討し，今般実装したので，本稿で報告する．

2. 自動字幕作成システムのあらまし

まず本システムのあらましについて説明する．本システムでは，ユーザにより収録された講義・講演や討論などの音声・映像に対して，事後的に字幕を付与することを想定している．**図 1** にシステムの利用の流れを示す．まず，ユーザがこれらのコンテンツを字幕サーバにアップロードする．音声・映像に加えて，言語モデルを話題に適応させるために，コンテンツの話題と関連するテキスト（たとえば講演予稿やスライド）もアップロードすることができる．字幕サーバではコンテンツからの音声の抽出および検査が行われ，ユーザの指定や関連テキストに応じて自動的に音声認識システムが構成された上で認識処理が実行される．これにより，音声と同期した字幕ファイルがサーバ上に出力される．

本システムでは，音声認識エンジンの構成と使用するモデルが異なるいくつかのプロファイルを用意しており，字幕の作成の際にどのプロファイルを使用するか指定する．現時点でのプロファイルの一覧を**表 1** に示す．講演・スピーチ・討論の 3 つのプロファイルでは，音響モデルとして隠れマルコフモデル（HMM）とディープニューラルネットワーク（DNN）による，いわゆる DNN-HMM モデルを使用し，言語モデルには統計的言語モデル（単語 N-gram モデル）を使用している．モデルの学習データとして，講演・スピーチには『日本語話し言葉コーパス』（CSJ）の学会講演データまたは模擬講演データを，討論には国会音声・会議録を使用している．これらのプロファイルでは，音声認識エンジンとして Julius[*5]を使用している．

講演（E2E）プロファイルでは，これらの枠組みとは全く異なり，End-to-End（E2E）型の音声認識を実行する．

[*5] https://julius.osdn.jp/

表 1 プロファイルの一覧

名称	講演	スピーチ	討論	講演（E2E）
学習データ	CSJ（学会講演）	CSJ（模擬講演）	国会音声	CSJ（学会＋模擬）
音響モデル	DNN-HMM			注意機構付き Encoder-Decoder NN
言語モデル	単語 Trigram			
デコーダ	Julius			

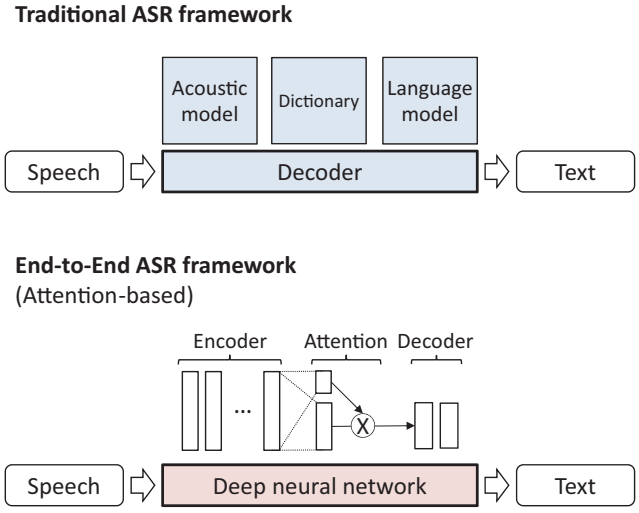

図 2 従来の枠組みと End-to-End の枠組み

End-to-End 型 [2], [3], [4], [5], [6] とは，**図 2** のように，従来の音声認識で個別に構築されていたモデルを 1 つのニューラルネットワークに統合して学習・認識するものである．本システムでは，注意（Attention）機構付きのエンコーダ・デコーダモデルを用いている．音声認識の実行の際は，入力音声をディープニューラルネットワークに投入して，ネットワークの計算（推論）を行うだけで認識結果が出力されるため，実行の制御はシンプルであり，高速に認識を実行できる．ただし End-to-End 型の音声認識では，入力を音声系列，出力を文字（単語）系列としてその対応関係を直接学習するため，きわめて多くの学習データと学習時間が必要となる．このため，本システムでは，現時点では講演を対象としたもののみ実装しており，また適応の対象外である．

これらにより音声を書き起こすことができるが，音声認識結果にはフィラーや口語表現，文末表現などの冗長部分が含まれるため，これらを削除・修正する自動整形 [7] を行う．また，文や節の境界も与えられていないため，句読点の自動推定 [8] も行われる．この結果が最終的な字幕テキストとして出力される．

これまでに述べたシステムは，音声などのコンテンツに対する事後的な字幕付与処理であるが，講義・講演の会場で情報保障などのためにリアルタイムに字幕を作成・表示するシステムも構築している．枠組みとしては，会場における字幕の編集や表示に PC 要約筆記で一般的に用いられ

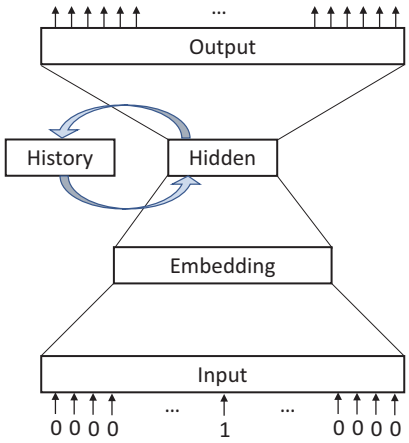

図 3 ニューラルネットワーク言語モデル

ている IPtalk[*6]を使用し，字幕のドラフトとして本システム（サーバ）による音声認識結果をリアルタイムに流し込むものである．

3. ニューラルネットワーク言語モデルの導入

3.1 従来の統計的言語モデル

まず，比較のために従来の統計的言語モデルについて述べる．本システムでは単語 Trigram（3-gram）言語モデルを使用しているが，これは音声認識における言語制約を 3 単語の連鎖確率で規定するもので，この確率は言語モデルの学習テキストにおける単語列 $\{w_i\}$ の統計頻度 C から定められる．

$$P(w_3|w_1w_2) = \frac{C(w_1w_2w_3)}{C(w_1w_2)} \qquad (1)$$

学習テキストに出現しない文脈（単語履歴，(1) 式における w_1w_2）に対処するため，実際にはこの確率が平滑化（スムージング）されて用いられる．N-gram モデルでは，過去の (N-1) 単語に基づいて次の単語の予測，すなわちどの単語がどの程度の確率で出現するかを得ることができる．

3.2 ニューラルネットワーク言語モデル

次にニューラルネットワーク言語モデル [9] について説明する．基本的な構成を**図 3** に示す．ここではモデル化の単位として単語を想定し，語彙（単語の種類の数）を V とする．ニューラルネットワーク言語モデルでは，t 番目の入力単語を w_t とすると，V に含まれるすべての単語について，w_t の次の単語としての確率を出力する．この際，w_t が V 中の何番目の単語であるかによって，対応する次元のみを 1，それ以外の次元を 0 とおいた，いわゆる one-hot ベクトルとして入力が与えられる．通常は入力層の後に次元を圧縮する射影層（Embedding）をおき，続いて隠れ層としてリカレント構造を持つものをおいて，過去の入力（履歴）に関する情報を保持する．実際のユニットとしては LSTM が一般的に用いられ，複数の層を用いることもある．出力層は語彙のサイズ（$|V|$）だけの出力を持ち，語彙中の各単語の確率を出力する．この枠組みから明らかなように，この確率は入力と履歴の状態により変動する．

隠れ層が保持している履歴は，原理的には過去のすべての入力についての情報を保持していることから，ニューラルネットワーク言語モデルは N-gram モデルよりも遠くの過去の情報を用いて単語を予測することが可能である．一般的に文の生成では，直近に用いられた単語だけでなく，以前に出現した単語をもとに次に用いられる単語が決定されることもあるので，長い履歴を保持できることによる予測性能（音声認識性能）の改善が期待される．

ただし，入力と出力の次元数は学習の段階で固定される．このため，いったん学習したモデルについて，たとえば単語を追加するなどのために次元数を変更することは容易ではなく，再学習が必要となる．また，ニューラルネットワークに一般的な傾向として過学習があり，言語モデルでもニューラルネットワークのモデルのみを用いることはかえって性能が低下することが少なくない．さらに，ニューラルネットワーク言語モデルは N-gram モデルと比較して計算コストが大きく，多数の文候補（仮説）を生成しつつ評価する認識処理に単純に適用することは処理の大きな遅延の要因となりうる．

3.3 モデルの線形補間

そこで本システムでは，ニューラルネットワーク言語モデルの使用法として一般的である，N-gram モデルとの線形補間を行う．線形補間は，複数の確率モデルの出力を重み付きで加えるものである．ここで N-gram モデルによる単語 w の確率を P_{Ngram}，ニューラルネットワーク言語モデルによる確率を P_{NN} とすると，最終的な確率 P は

$$P(w) = \lambda P_{Ngram}(w) + (1-\lambda)P_{NN}(w) \qquad (2)$$

で与えられる．ここで混合重み λ $(0 < \lambda < 1)$ を定める必要があるが，本システムでは予備的に調査した値を設定している．

4. 実装と評価

4.1 適応の手順

これらの言語モデルに対して，本システムでは次の手順で適応処理と音声認識処理を行う．

まず，適応のために与えられたテキストに対して，ベースのモデルの学習テキストとまったく同一の前処理（形態素解析や補正処理）を行って単語列に変換する．統計的言語モデルについては，あらかじめベースのモデルの統計量（式（1）における C）を計算しておく．与えられたテキストの単語列についても統計量を計算してこれらと足し合わせ，得られた統計量をもとに N-gram 確率を計算する．

[*6] http://www.s-kurita.net/

ニューラルネットワーク言語モデルについては，ベースモデルの学習テキストの単語列と適応テキストの単語列を結合して学習を行う．これにより，同じテキストから，適応された統計的言語モデルとニューラルネットワーク言語モデルがそれぞれ構築される．

音声認識の際は，ニューラルネットワーク言語モデルの計算が高コストであることから，まず統計的言語モデルのみを言語制約として用いて認識結果をいったん生成する．認識処理の際に検討された文の候補（仮説）には，それぞれ音響モデル・言語モデルに基づくスコアが与えられており，このスコアによって最善の仮説が認識結果として出力されるが，ここでは 2 番目以降のものも多数保持しておく（いわゆる N-best 文）．次に，これらの認識結果の各文についてリスコアリングを行う．すなわち，各文のスコアのうち言語モデルによるスコアについて，式 (2) に基づいてニューラルネットワーク言語モデルを反映させて計算し直し，得られたスコアにしたがって認識結果の N-best 文を並べ直して，最もスコアの高い文を最終的な認識結果として得る．

4.2 評価実験

本システムで導入した枠組みについて，実際の字幕付与データにより性能評価を行った．用いたデータは，京都大学で 2018 年に行われたシンポジウムの講演 1 件（26 分）で，音声認識と IPtalk を用いて字幕付与が行われた．この講演の予稿を使用して言語モデルを適応し，あらためて音声認識を行ってその結果を字幕のテキストと比較することで認識性能の評価とした．

本実験では，システムの音声認識プロファイル（表 1）として講演プロファイルを使用した．ニューラルネットワーク言語モデルの処理は Python で実装されており，ニューラルネットワークのフレームワークとして Chainer[*7]を使用している．モデルが複雑になると学習時間が大きく増加することから，モデルができるだけシンプルになるよう，あらかじめパラメータ（リカレント層の種類，隠れ層の数，射影層のユニット数）の比較検討を行い，隠れ層として LSTM1 層，また射影層のユニット数は 150 とした．なお，適応テキストとベースの学習テキスト（CSJ 学会講演データ）を結合した後の語彙サイズ（つまり入力・出力ベクトルの次元数）は 36,699，データの総単語数は 798 万単語である．モデルの学習に要した時間は 39 分である[*8]．学習エポック数は事前にベースモデルの学習を通じて調整した．

音声認識の結果，従来の統計的言語モデルのみでは単語正解率として 90.5%であったのに対して，ニューラルネットワーク言語モデルの導入（リスコアリング）により 90.8%となった．エラーの削減率は 3.3%となり，導入の効果を確かめることができた．

5. おわりに

本稿では，我々が運用している自動字幕作成システムにて行った言語モデルのニューラルネットワークモデルへの拡張について報告した．ニューラルネットワーク言語モデルは調整の余地が大きく，また End-to-End 型との組み合わせ，あるいは End-to-End 型自体の拡張も考えられることから，今後もシステムの改善に努めていきたい．

謝辞 本研究の一部は学術研究助成基金助成金（課題番号 18K11354）によって行われた．

参考文献

[1] 秋田祐哉，上乃　聖，三村正人，河原達也：音声認識を用いた字幕作成システムの改良，情報処理学会研究報告，2019-AAC-9-34 (2019).
[2] Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K. and Bengio, Y.: Attention-based models for speech recognition, *Proc. Advances in Neural Information Processing Systems*, pp. 577–585 (2015).
[3] Chan, W., Jaitly, N., Le, Q. and Vinyals, O.: Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, *Proc. ICASSP*, pp. 4960–4964 (2016).
[4] Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P. and Bengio, Y.: End-to-End attention-based large vocabulary speech recognition, *Proc. ICASSP*, pp. 4945–4949 (2016).
[5] Audhkhasi, K., Ramabhadran, B., Saon, G., Picheny, M. and Nahamoo, D.: Direct acoustics-to-word models for English conversational speech recognition, *Proc. Interspeech*, pp. 959–963 (2017).
[6] Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., Soplin, N. E. Y., Heymann, J., Wiesner, M., Chen, N., Renduchintala, A. and Ochiai, T.: ESPnet: End-to-End speech processing toolkit, *Proc. Interspeech*, pp. 2207–2211 (2018).
[7] Neubig, G., Akita, Y., Mori, S. and Kawahara, T.: A Monotonic Statistical Machine Translation Approach to Speaking Style Transformation, *Computer Speech and Language*, Vol. 26, No. 5, pp. 349–370 (2012).
[8] 秋田祐哉，河原達也：講演に対する読点の複数アノテーションに基づく自動挿入，情報処理学会論文誌，Vol. 54, No. 2, pp. 463–470 (2013).
[9] Mikolov, T., Karafiat, M., Burget, L., Cernocky, J. and Khudanpur, S.: Recurrent neural network based language model, *Proc. Interspeech*, pp. 1045–1048 (2010).

[*7] `https://chainer.org/`
[*8] 計算は GPU で行い，本実験では Nvidia TITAN X (Pascal) を使用した