

Wikipediaを利用した音声認識用言語モデルの構築および評価

Construction and Evaluation of Language Model for Speech Recognition using Wikipedia

田中和紀¹ 管村昇¹

Kazuki Tanaka¹, and Noboru Sugamura¹

¹工学院大学大学院 工学研究科

¹ Graduate School of Engineering, Kogakuin University

Abstract: 本研究ではインターネット百科事典のWikipediaをコーパスとして利用し、音声認識用言語モデルを構築した。再配布や再利用が可能なWikipediaを用いることにより、音声認識用言語モデルを一般に公開することができる。言語モデル構築にあたって、Wikipediaに対して、不要なデータ除去、読み付与処理などを行った。また、構築した言語モデルを使った音声認識の評価について述べる。

1 はじめに

1.1 音声認識での日本語Wikipedia活用

近年、インターネット上にて、ソフトウェアのオープンソース化やコンテンツのオープンコンテンツ化が盛んにおこなわれている。音声認識分野においても、オープンソースの汎用大語彙連続音声認識エンジンJulius[1]が公開されている。Juliusで音声認識を行うには、音声認識用の言語モデルと音響モデルが必要である。

新聞記事12年分のコーパスを用いた言語モデル（以降、新聞記事言語モデル）[2]が有償配布されているが、有償のため一般ユーザは容易に利用することができない。他にJuliusディクテーション実行キット [3] 付属の言語モデルがある（以降、Julius付属言語モデル）。これは、Webからテキストを収集し構築されたものである。関連研究として、Web全体を言語モデルのコーパスとして用いることができるか評価した研究[4]が存在するが、言語モデルが公開されていない。

本研究では、インターネット百科事典のWikipedia日本語版[5]をコーパスとして用いて、無償配布可能な音声認識用言語モデルの構築および評価を行った。構築した言語モデルを下記のURLにて、一般公開¹している。

http://road-to-dream.net/lang_model/

1.2 Wikipediaのライセンスおよび作成物の取り扱い

Wikipediaデータは、GFDL(GNU Free Documentation License)1.2に基づき利用することができる[6]。加工や再配布が可能なWikipediaを用いることにより、構築した言語モデルを一般に公開することができる。

本研究で構築した言語モデルは、Wikipedia内の日本語文章を統計処理しているため、非透過的複製物（opaque copy）となる。非透過的複製物とは、一般の人々が容易に入手できるテキストエディタや画像ビューワ、ブラウザなどで閲覧できない状態の複製物である[7]。非透過的複製物を公開する際には、WikipediaデータをダウンロードできるURLを明記する必要がある。

2 Wikipediaデータの加工

2.1 Wikipediaデータの加工手順

Wikipediaデータからのコーパス作成手順は参考文献8に準じている。文献では文章形式が整った新聞記事を対象としているが、本研究では文章形式が新聞記事ほど整っていないWikipediaを対象としており、Wikipediaデータ特有の処理を行っている。また、Wikipediaデータの文字コードはUTF-8のため、文献に記載されているツールをEUC-JP対応から

¹ 2009年1月22日より公開し、適時アップデートを行う。

UTF-8対応へ修正した。

図1にWikipediaデータを扱う場合の特有の処理を示し、以下にこの処理について説明する。これらの処理は軽量プログラミング言語であるPerlを用いて作成した。

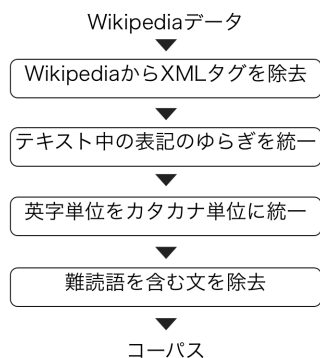


図1 Wikipediaデータ特有の加工処理

2.2 WikipediaからXMLタグを除去

コーパスには、2008年11月28日版のWikipediaデータダンプXMLファイルを用いた[9]。このファイルはXMLとしてタグ付けされており、そのまま日本語コーパスとしては扱えない。そこで、WP2TXT[10]を用いてテキストデータを抽出した。

2.3 テキスト中の表記のゆらぎを統一

日本語Wikipedia上で使われている文字には、「半角英数」、「半角カタカナ」、「全角英数」、「全角カタカナ」、「全角ひらがな(漢字)」などが存在する。次の例文は意味としては同じだが、統計量算出時には異なった文として計算されてしまう。そこで、英字以外を全角文字へ統一した。英字は半角文字へ統一した。

(例) W i k i p e d i a をコーパスとして、活用する。

(例) Wikipedia をコーパスとして、活用する。

2.4 英字単位をカタカナ単位に統一

Wikipediaデータ内には「標高3,776m。」のような英字単位を含んだ文が存在する。後述の形態素解析時に英字単位や英単語に対して読みを付与するが、英字単位に正しく読みを付与できない場合が多くあった。コーパス作成時に表1の英字単位をカタカナ単位へ統一した。処理速度の関係上、全ての英字単位に対応しきれていない。

2.5 難読語を含む文を除去

日本語の発話において、発話しない記号や外国語の文字を含む文をコーパスより除去した。文字単位で除去した場合、所々の文字が無くなった文となり、文法的に間違った文になってしまう。日本語文として正しい統計量を得るために、文単位の除去を行った。次の表2は、除去している読み付与が困難なUnicode範囲である。英単語は形態素解析エンジンを用いて、読みを付与できるか判断し、読みを付与できなければこの段階で文ごとと除去している。

音声認識用言語モデルは全ての単語に読みが正しく付与されている必要がある。XMLタグを除去したWikipediaデータは1021MBであり、読みを付与することができるコーパス量は、801MBとなった。

表1 英字単位のカタカナ単位変換表

英字単位	カタカナ単位
cm	センチメートル
m	メートル
km	キロメートル
kg	キログラム

表2 除去しているUnicode範囲

名称	Unicode範囲 (正規表現)
ギリシャ文字	[Α-Ω α-ω]
ロシア文字	[А-Я а-я]
半角記号	[!-/:-@[-`{-~]
Box Drawing	[┌─┐]
Geometric Shapes	[■-♠]
Mathematical Operators	[∇-ε]
Miscellaneous Symbols	[☀-♀]
Arrows	[←-↔]
Hangul Jamo	₩x11[₩x10-₩xFF]
Number Forms	₩x21[₩x50-₩x8F]
General Punctuation	₩x20[₩x00-₩x6F]
Superscripts and Subscripts	₩x20[₩x70-₩x9F]
Enclosed Alphanumerics	[①-⑩]
Letterlike Symbols	₩x21[₩x00-₩x4F]
Latin-1 Supplement	₩x1E[₩x00-₩xFF]

3 形態素解析と言語モデルの構築

3.1 形態素解析と読み付与

本研究では音声認識エンジンJuliusで利用可能なN-gram言語モデルの構築を行った。N-gram言語モデルは各単語が接続して出現する確率をモデル化したものである。そのため、文を単語単位に分ける必要がある。そこで、形態素解析エンジンMeCab 0.97 [11]を用いて形態素解析を行った。基本となる形態素解析辞書にはipadic-2.7.0 [11]を用いた。

形態素解析時の未知語や英単語の読みに対応するために、はてなキーワードを用いた形態素解析辞書の再構築を行った[12][13]。現状では、次の単語例「乾物：カンブツ／ホシモノ／ヒモノ」のように複数の読みが存在する単語に対して、読み付与処理が完全に対応することができていない。

3.2 言語モデルの構築

言語モデルは、Palmkit 1.0.31 [14]を用いて、比較対象の新聞記事言語モデルと同一条件（前向き2-gramと後ろ向き3-grams, 語彙サイズ：60152, カットオフ：2）となるように処理を行った。その後、音声認識エンジンJulius付属のmkbngam 4.1.1を用いてJulius用のバイナリN-gramファイルへ変換した。

4 言語モデルの評価

4.1 言語モデルの構築結果

表3は、構築したWikipedia言語モデルと既存の言語モデルとの比較表である。N-gram言語モデルを構築しているため、形態素数が多いほど信頼ができる統計量が得られる。Wikipedia言語モデルの形態素数が新聞記事言語モデルの半分となっており、現時点ではWikipediaのコーパス量が不十分であることがわかった。

そこで、Wikipediaコーパス(801MB)にWebから集めたコーパス(313MB)を合わせて、合計1114MBのコーパスから言語モデル（以降、Wikipedia+Web言語モデル）を作成した。WebコーパスはWebからHyperEstraiier 1.4.13 [15]を用いて無作為に集めた後、Wikipediaデータ加工処理と同一の処理を行ったものである。Wikipedia+Web言語モデルの形態素数は222,960,373となった。

4.2 言語モデルの評価結果

構築した言語モデルを用いて音声認識精度の評価を行った。音声認識エンジンにはJulius 4.1.1, 音響モデルには文献2の全世代話者用モデルを用いた。評価用データとして、文献内にある「新聞読み上げ音声」の男女それぞれ3名の24発話と「宇宙に関連したナレーション音声[16]」の男性1名の214発話、合計238発話の音声データを用いた。

図2は、構築したWikipedia言語モデルの音声認識結果である。Wikipedia言語モデルの認識率は71.56%と比較対象の中では低い認識率となった。Wikipedia+Web言語モデルの認識率は74.49%となり、認識率の向上がみられた。

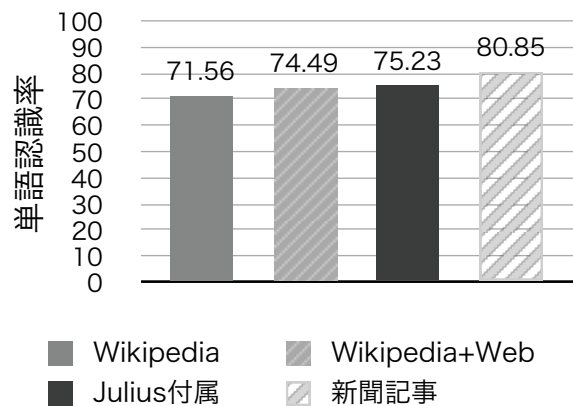


図2 言語モデルの性能評価

表3 言語モデル比較表

	Wikipedia言語モデル	Wikipedia+Web言語モデル	Julius付属言語モデル	新聞記事言語モデル
学習データ	2008年11月時点のWikipediaデータ	WikipediaデータとWebテキスト	Webから収集したテキスト	毎日新聞社の新聞記事CD-毎日新聞 91~2002年版
形態素数	156,139,601	222,960,373	229,665,091	336,728,813
語彙サイズ	60,152	60,152	60,437	60,152

5 まとめ

本研究では、Wikipediaをコーパスとして用いて言語モデルを構築し、性能評価を行った。Wikipediaからは容易に大量の日本語コーパスを得ることができるが、言語モデル用コーパスとしては量が不十分であった。しかしながら、Web上からコーパスを補充することで、一定の認識率を得ることができた。

Wikipediaは百科事典であるため、専門用語に強い言語モデルを構築できた可能性がある。講義音声など専門用語を多用する音声データで評価し、Wikipedia言語モデルの特徴を調査する必要がある。

今後は、Wikipedia言語モデルの特徴調査および読み付与処理の精度向上を行う。また、Wikipedia内の「芸能」、「歴史」、「宇宙開発」などカテゴライズされた情報を活用し、話題毎のコーパスをWeb上から収集する手法を検討する。言語モデルの公開だけでなく、Wikipedia言語モデル作成ツールの公開を行う予定である。

謝辞

本研究を行うにあたり、フリーの百科事典であるWikipediaや音声認識エンジンJulius、連続音声認識コンソーシアムの各種ツール、Wikipediaからテキストを抽出するWP2TXT、形態素解析エンジンMeCab、N-gram言語モデル構築ツールPalmkit、全文検索システムHyper Estraier、などのオープンソースソフトウェアを利用させていただいた。Wikipediaの複数の著者やこれらソフトウェアを開発し、無償公開している作者に対して、心からの感謝を申し上げる。

参考文献

- [1] 李晃伸: 大語彙連続音声認識エンジンJulius, <http://julius.sourceforge.jp/>, (2008年12月アクセス).
- [2] 連続音声認識コンソーシアム2003年度版ソフトウェア, <http://www.lang.astem.or.jp/CSRC/>, (2008年12月アクセス).
- [3] Juliusディクテーション実行キットv3.2, <http://julius.sourceforge.jp/index.php?q=juliuskit.html>, (2008年12月アクセス).
- [4] 伊藤,秋葉,他: WWWは大語彙連続音声認識の学習データとして使えるか?, 日本音響学会秋季研究発表会講演論文集, pp.131-132 (2002).
- [5] 百科事典Wikipedia日本語版, <http://ja.wikipedia.org/>, (2008年12月アクセス).
- [6] Wikipediaのライセンスとデータベース, <http://ja.wikipedia.org/wiki/Wikipedia:データベースダウンロード>, (2008年12月アクセス).
- [7] GNU Free Documentation License, <http://www.gnu.org/licenses/fdl.html>, (2008年12月アクセス).
- [8] 鹿野清宏,伊藤克巨,他: IT Text 音声認識システム 付属CD-ROM内『言語モデルの作成』および『形態素解析・読み付与プログラムの開発』, オーム社 (2001).
- [9] WikipediaデータダンプXML(jawiki-latest-pages-articles.xml.bz2), <http://download.wikimedia.org/jawiki/>, (2008年12月アクセス).
- [10] 長谷部陽一郎: Wikipedia日本語版をコーパスとして用いた言語研究の手法, 言語文化, Vol. 9, No. 2, pp. 373-403 (2006), <http://wp2txt.rubyforge.org/>, (2008年12月アクセス).
- [11] 工藤拓: 形態素解析エンジンMeCab(和布蕪), <http://mecab.sourceforge.net/>, (2008年12月アクセス).
- [12] 鈴木健太郎,西村竜一,他: ウェブ上の言語知識を利用した音声認識用単語辞書の更新手法, FIT2008 (第7回情報科学技術フォーラム), pp.189-190 (2008).
- [13] 緒方淳,後藤真孝,他: PodCastle: 集合知に基づく Web キーワードを考慮した言語モデリング, 日本音響学会2008年 秋季研究発表会, pp.97-100 (2008).
- [14] 伊藤彰則:N-gram言語モデル作成ツールPalmkit, <http://palmkit.sourceforge.net/>, (2008年12月アクセス).
- [15] 平林幹雄: 全文検索システムHyper Estraier, <http://hyperestraier.sourceforge.net/>, (2008年12月アクセス).
- [16] 独立行政法人 情報処理推進機構: 「教育用画像素材集」 - 「理科 地球と宇宙」に存在する全動画 (61件) から抽出した音声データ, <http://www2.edu.ipa.go.jp/gz/index.html>, (2008年12月アクセス).