平成 29 年度　東京外国語大学オープンアカデミー
東京外国語大学語学研究所 企画
『コーパスから見えることば・文化・社会』
2017 年 11 月 14 日（火）　第 6 回

「日本語：数億語のコーパスを作って調べてみるとみえてくる頻出語、頻出表現」

東京外国語大学准教授
望月 源

今日で最終回、6回目のオープンアカデミー「コーパスから見えることば・文化・社会」を始めます。

本日は、コンピュータを使ってコーパスをどのように作るかということに焦点をあてて話をします。また、実際に言語データを集めて作ったコーパスが既にデータとしてはありますので、その中でどういう見え方がするのかという品質や表現を見ていきたいと思います。本日扱う言語は日本語です。

コーパスとは

まず、最初にコーパスとは何かについて、計算言語学、コンピュータ処理の世界から確認しておきたいと思います。これまでの講義で言語学の世界ではコーパスを用いて事例を見つけたり、新しい発見があるなどの話題が出てきたと思いますけれども、私のやっている計算言語学、コンピュータ処理の世界でも昔からコーパスと言うのは言語データとして広く認識され意識されてきました。私たちが普段話している言葉、あるいは電子メールでもブログでもなんでもそうですが普段生み出して、普通に使っている生のものを大量に集めて整理して言語データという形で利用できるようにしたものをコーパスといいます。いろんなものが言語データとして考えられるわけですね。実際にコーパスとして整備されてきたものには、新聞記事、雑誌、小説、教科書、ウェブページなどが挙げられます。こういったものは書き言葉なので「書き言葉コーパス」となり、書き言葉を対象とした研究で書き言葉の特徴が得られるデータとして利用されます。一方、会話とか授業や講演などを言語データとして集めると、これらは話し言葉ですので、話し言葉のデータとか「話し言葉コーパス」となります。話し言葉の場合は発話され

た音声データをそのまま使って「音声コーパス」として使う場合もありますが、通常は、文字にしてテキストの形で保存する書き起こしという作業があります。書き起こし作業は、人が録音音声を聞いてそれを文字にするとか、今は音声認識のプログラムが良くなっているので自動認識した音を直接文字にするなどの前処理をして、間違っているところを修正することもあります。このように作成したものは「書き起こしコーパス」とも言います。メディアが音のままなのか文字にしてからなのかと言う違いがありますが、言語の種類としては書き言葉と違うタイプの「話し言葉コーパス」として使います。

こっちが書き言葉コーパス、こっちが話し言葉コーパスという区別を考えると、その中間的なコーパスとしてよく言われるのが、書き言葉よりも少しくだけた感じで話し言葉に近い書き方をするようなタイプのブログとか Twitter とかのソーシャルネットワークが挙げられます。こういったものは書き言葉と言うより中間のもので、色々と集められています。

結局言語データとしては、自分たちが話しているものとか、その時に使われている言葉を集めることなので、ありとあらゆるものが対象になりうると言えます。平たく言うと、コーパスとは普段私たちが使っている言葉を大量に集めた言語データ、ということになります。なぜこんなものを集めるかと言うと、コーパスには実際に使われている言葉がつまっているので、コーパスを詳しく調べると、おそらく、その言葉の特徴が見えてくるだろうという期待があります。特徴として使い勝手の良いデータがコーパスの分析から見えてくるという期待があって昔から集められ利用されています。

もう一点大事なのは、今はコンピュータで扱える形式のものを集めていることです。コンピュータで扱える形式でコーパスが大量に集められてきたのはここ15年〜20年位のことです。20 数年前は、まだインターネットというものが世の中に普及していない時代で、今のように WWW のホームページが大量にあふれているということもありませんでした。当時インターネットは商用ではなくて研究者用のネットワークだったのです。また、その頃はテキストといっても電子化されている情報はほとんどありませんでした。私が大学院生の頃はあるテキストを処理したくても電子化されていないことがほとんどだったため、紙媒体のテキストを光学文字読み取り装置（OCR）で読み込んでテキストを作ったこともありました。そのような時代には大量のデータを集めてという話は考えとしてはあったのですが、現実的ではありませんでした。

そのうち、新聞記事がコーパスとして利用できるようになりました。ネットニュースなどが存在せず紙媒体しかない時代でしたが、新聞社には毎日の新聞記事がワープロを使った電子データの形で存在していました。つまり言語データとしては電子化されたものがあったので、計算言語学、自然言語処理の分野で著名な先生方や学会から新聞社にデータを研究目的で利用させてもらえないか、という申し入れをして使えるようにしたという経緯があると聞いています。今では多くの新聞社による各新聞記事データが、研究目的に使えるデータとして販売提供されるようになっています。

新聞記事はだいたい年間10万〜20万記事ほどで、10年分くらい集めると2百万記事ぐらいになります。当時としては結構大規模で、新聞コーパスによって新聞の中に出てくる言葉の特徴が見えてくるため、それで要約研究をやったりとか、形態素解析のように文を単語に区切るという技術や、構文解析をする技術を発達させてきたという歴史があります。

コンピュータで言語データであるコーパスを利用できると色々な良いことがあります。最近は言わなくても当たり前のようになりましたが、まず検索が楽になったという事があります。コーパスの中に「ある表現」が出現する場所を素早くとらえることができます。もしコンピュータが利用できないとしたら、色々なテキストの中である表現がどこに出現するか、場所を調べるということはなかなかに難しい問題になります。それから、ある語の出現数を数えるとか、ある語とよく一緒に使われる語にはどんな語があるかとか、そういう情報を取り出して一覧したり確率を計算したりすることができるようになりました。こうしたことが非常に有効なために、様々なコーパスが段々作られるようなりました。初期の新聞社のデータが中心だった時代から、インターネットが各家庭に普及するようになって、誰もがメールを書いたりブログを書いたりするようになると、それまでの時代には考えられないくらいの生産量で言葉が世の中に出されるようになりました。しかも、世に出されたものを広く集めてコンピュータで使える形で整備しやすくなっています。このようなことがあって現在では言語データがコーパスとしてより広く使われるようになりました。

扱える言語データの規模が大きくなった一方で、高性能のコンピュータも身近に使えるようになったので、誰でも比較的いろいろな処理ができる時代にもなっています。うちの学生さんにも、授業の最初の時にこういう話をするのですけれ

ども、例えば今日の新聞記事の中で、「車」という文字が何回出てくるのか調べてみてと言われて、紙の新聞をぽいっと渡されても、まぁ「えっ」と引きますよね。手作業で調べるなどというのは現実的ではありませんが、それがデータの形で提供されて、今日の新聞の全部のデータがあれば、コンピュータの力を借りれば、「車」の数を数えるなどという作業は割と楽にできます。では1年分の記事の中で「車」が何回出てくるかというと、これもコンピュータの力を借りれば十分に可能な作業です。私が前にとったデータをちょっと見てみたいのですけれども、手元には古い新聞記事のデータしかありませんが、1995年には7,085回出てきて、2000年には7,050回「車」と言う単語が出てきます。「車」だけではなくて「なんとか車」というのまで入れるとすごい数になります。コンピュータとコーパスの利用で、このような話ができます。

コーパスの入手

次に、こういったコーパスをもし使おうと思ったら、どうやって手に入れるかと言うことを話します。まず、研究目的の利用で有償・無償で提供されたコーパスが存在します。例えば、国立国語研究所コーパス開発センターのページ http://pj.ninjal.ac.jp/corpus_center/ を参照すると、国立国語研究所が管理・提供しているコーパスが一覧できます。

さまざまなコーパスがありますが、コーパスを利用するということは、そのコーパスの中に出てくる実際に使われている言葉を調べようということなので、自分がどういう目的で研究したいかによってそれに合うコーパスが違うはずです。コーパスを選ぶ場合は、自分の目的に合うものを探すということになります。話し言葉の研究をしようとしているのに書き言葉のコーパスは合わないですよね。ここに一覧されているような既存のコーパスというのもいろいろな人が自分達の研究目的に合うようにいろいろな工夫をして一生懸命集めたものです。

国立国語研究所以外にも入手可能なコーパスは存在します。例えば、楽天とかライブドアとかリクルートとか、こういったIT系の企業では、自社の運営しているいろいろな商品購入サイトにあるレビュー（利用者が商品の評価を書いているレビュー）のデータを研究用に提供していたりします。モノやサービスなどの善し悪しを評価するときの言葉としては、どういう言葉があると良い評価になるのか、どういう言葉が入っていると悪い評価になるのかとか、などを研究するために提供されています。他には皆さんよくご存知のウィキペディアもコーパスとし

て使うことが可能です。ウィキペディアでは、ウィキペディアの記事をひとまとまりのファイルとして一括してダウンロードできるサイトが用意されています(「Wikipedia データベースダウンロード」で検索)。そのサイトに行くとウィキペディアの日本語なら日本語の全データを、一括して自分のコンピューターにダウンロードできるようになっています。言語処理の研究では、このウィキペディアのデータもよく使われています。同じ項目についての、例えば日本語と英語の記事にある説明は、対訳になっているわけでは無いのですけれど、同じことを説明しているので文章が似ていると考えられます。そこで、２つの言語の記事を自動的に、アライメントと言ってどことどこが似ているかをつき合わせる処理をして、対訳データのように考えて翻訳研究などに利用することもできます。また、ウィキペディアは百科事典なので物事の説明の情報源として使われるとか、言葉を調べるだけではなくていろいろな使われ方をしています。こういうのも広い意味ではコーパスです。

あとは必要に応じて市販されているコーパスを購入するということもあります。さきほど述べた新聞記事のデータは、歴史も長いのでもう20年分ぐらいは売られています (http://www.nichigai.co.jp/sales/corpus.html) 。ただし、研究目的であっても高価です。なかなか個人で買えるものではないですが、新聞社と使用権の契約をすることで、比較的自由に研究利用することができます。

それから、コーパスとして出されているわけではないのですが、著作権が切れてパブリックドメインになっている文章を集めた青空文庫というのも利用可能です (https://www.aozora.gr.jp/)。青空文庫から入手した小説をデータとしていろいろなことを調べたりするということもできます。こんな感じで研究に利用するコーパスとして、既存のコーパスや言語データを入手するという方法があります。

既存のコーパスを利用すると良いこと悪いことがいろいろありますが、良いことはまとまったデータが比較的容易に利用できるということです。著作権処理がされていて、コーパスとして使うには契約をしてルールに基づいて使えば良いのです。ただ有料なものもあって比較的料金が高いです。最後にこれが問題なのですが、誰かが用意したものですので自分の目的に合うとは限らないという問題があります。研究によっては既存のコーパスが目的に合わないということもありますし、それよりもっと、別の場所に存在する言語データをどうにかして手に入れて自分の研究に利用した方が良いと言う場合もあります。

WWW のページをコーパスにする

本日は自分でコーパスを作成するということを念頭に話を用意してきているのですが、大きいものとして、Web ページを集めてコーパスにするというものが挙げられます。この利点はデータが大量に存在しているのでたくさん集めることができることです。Web ページは、他人が書いたページなので著作権がそれぞれあります。以前は日本の法律では研究目的であれ、収集して利用するということが著作権に違反するということがあったのですが、今はこの点はクリアになっていて、集めたデータをそのまま複製して配布するということではなく、研究目的でコーパスとして収集して中身を解析して使うということはできるようになっています。コーパスは生のデータなので人が書いたものには著作権があります。そこを注意して使わないといけません。また、特に、成果を発表するときには著作権違反にならないようにする必要があります。

では、Web ページを使ってコーパスを作るとどんな感じなのかという話をします。手順は3段階あります。最初はコーパスのもとになるデータをウェブから収集するという作業です。これは最近の言い方だとクローリングといいます。このクローリングで収集したウェブデータは HTML ファイルと呼ばれ、基本的に HTML (HyperText Markup Language) という言語で書かれています。HTML ファイルは、コンピュータの世界の「テキスト」という形式で書かれているファイルなのですが、Web ブラウザで Web ページを見たときとは異なるウェブページを表示するために必要ないろいろな情報がついています。そのため、2段階目として、HTML ファイルから自分が使いたい部分（ウェブのコンテンツ部分）をデータとして取り出すという作業が必要になります。これは最近の言い方だとスクレイピングと言われます。ここまでの2段階の作業で Web ページのコンテンツ部分を集めたテキストファイルからなるコーパスはできあがりますが、あくまでただのテキストファイルの集合でしかありません。テキスト内の各文を形態素解析したり、構文解析したりといった言語処理技術を施すことでテキスト中の単語や品詞の情報や係り受け関係、構文情報などを付け加えることも可能です。そこで、最後の段階として Web コーパスとしての価値を高めるような加工を施すことも多いです。それによって、色々な目的で使用できるようにデータを整理して Web コーパスを作成することができます。

ここで少し、Web ページである HTML ファイルの構造について話をします。図1と図2に示す「首相官邸」の「総理の一日」の一ページを例にします。

図1. 首相官邸の「総理の一日」のページ (2017年10月27日)

図 2 首相官邸の「総理の一日」のページ（つづき）

今、WWW のブラウザで見ると図1、図2のように見えるページから、各項目名や説明文のようなテキスト部分をデータとして利用したいとします。例えば、マウスを使ってこのページから文章部分をコピーしてどこか他のファイルに貼り付けるという作業をすれば内容は取れます。しかし、そんなことをしていたら、何

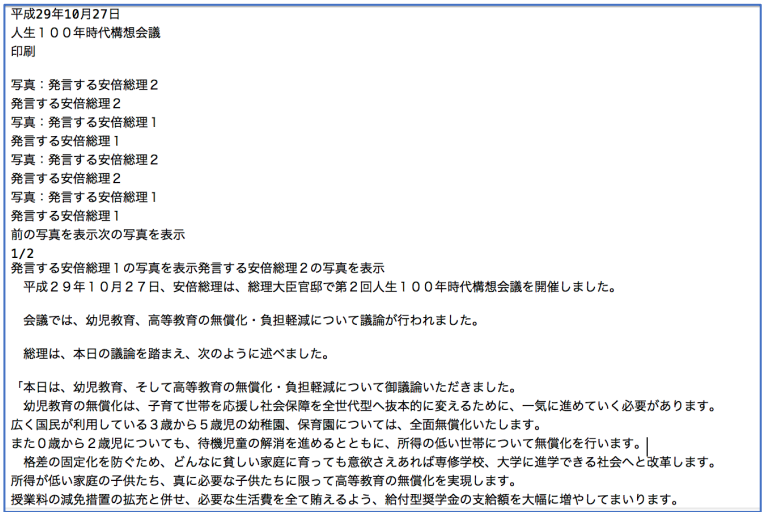
万、何十万、何百万という量は集められないわけです。では、どうするかというと、集めてきた HTML ファイル内に書かれた様々な情報の中から、テキスト部分だけを取り出すという処理をします。実際の HTML ファイルは次の図3のようになっています。この図は実はみなさんも簡単に見ることができます。WWW のブラウザとして Google Chrome を使っている人は図1のように表示されているところで、右クリックをしてみてください。サブメニューが出てきて、その中に「ページのソースを表示」という項目がありますので、クリックしてみてください。新しいページが開いて、そこに次の図3に示すような見慣れない文字の並んだ画面が現れると思います。

図 3 ページのソース画面 HTML の例

このファイルは HTML という専用の言語で書かれた Web ページの設計図といったようなものです。HTML には、ここに何々という画像を表示するとか、この文字列は見出しにする、ここからここまでは一つの段落にする、などの文章構造や付加的な情報を示すためにあらかじめ役割が決められたタグという記号を使って Web ページをつくるための情報が記されています。例えば、 と という一対のタグを使って文字列を囲むと、そのひとかたまりが1つの段落というこ

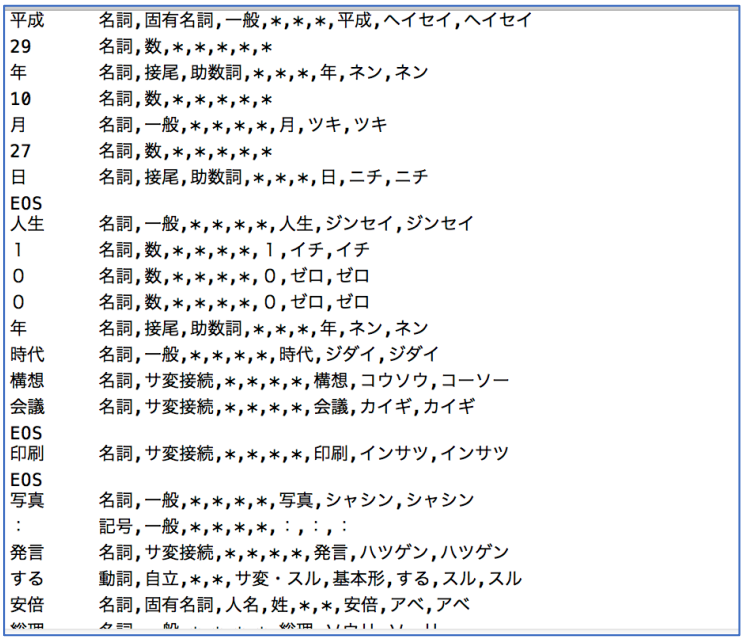
とになります。<h1>と</h1>というタグを使って文字列を挟み込むと、それは見出しを表す文字列ということになります。このようなタグの種類とその意味は決められていますので、「HTML タグ辞典」のような形でまとめられています。さて、Google Chrome や Internet Explorer といった Web ブラウザは、この HTML 言語を理解できる HTML 解読器のようなものということもできます。書かれている指示を順次解読しながらそのとおりにコンテンツを表示していくと全体のページができるのです。Web ページをコーパスとして使おうとする場合には、HTML の中からテキストコンテンツ以外の不要な部分を取り除く必要があります。図3をみると、図2でテキストとして表示されている文と同じ文が含まれていることがわかるでしょうか？例えば「人生100年時代構想会議」や「安倍総理は、総理大臣官邸で第2回人生100年時代懇談会議を開催しました。」という文字列があり、それぞれ< >という形で表現された何かのタグに囲まれているようすが見えると思います。この HTML ファイルから文字部分だけを取り出すプログラムを利用して図4のようなテキスト部分のみのファイルを作ります。

図 4 テキスト部分



次に、テキストデータができたので、これを形態素解析という処理にかけて単語を取り出してみます。図5を見てください。これは mecab という形態素解析をするプログラムによって処理した結果の一部です。

図 5 形態素解析結果



日本語の形態素解析プログラムでは、日本語の文章を入力として与えると、その文を単語に区切って各単語の品詞の情報やよみがな、発音などの情報を解析結果として出力します。図5の各行の左側が1つの単語（形態素）を表し、右側はその単語の品詞情報などを表します。こういう研究を形態素解析といいます。研究成果は一般に公開されていて、Mecab、ChaSen、JUMAN などが有名です。このデータを元にして、プログラムを書けば単語の数を数えるということもできます。ちなみに、2017年10月27日の総理の一日「人生100年時代構想会議」のページの中には、全部で664語あり、出現数の多い順に並べると「、」「の」「を」「する」「ます」「。」の順で、それぞれ33回、31回、29回、25回、19回、18回出現する、といったことがわかります。出現数の上位には内容がわかるような語はないですね。その他の語をいくつかみると、「総理」(13回)、「安倍」(11回)、「発言」(10回)、「無償」(6回)、「教育」(6回)、「議論」(4回)のような語が出てきます。名詞は一般的には文章のトピックに関わるので、名詞だけを見ていても文章の内容がなんとなく想像つくのが面白いです。似たような名詞が出てくる複数のテキストを集めると似たような内容の文章が集まるということになります。私が最初に大学院で言語処理という分野に触れたときには、言葉に興味があって選んだのですが、言語処

理をやればやるほど、言葉が、そもそも文だったものが、単語にバラバラにされて、さらに、それが順番もなくなって数に置き換えられていきました。対象は言葉なのになぁと思って解せない部分もあったのですが、実際やってみると、なぜか結構まとまってくるのですね。だから数を数えるだけでも関心が出てきました。今1つずつの単語を見てきましたが、バイグラムと言って2語の連続で出てくる場合を数えると、より特徴的な言葉の塊が見えてきます。例えばこの文章では「ます 。」「安倍 総理」「発言 する」「する 安倍」「無償 化」といったバイグラムが出現数上位にきます。たくさんのデータを使うと、よく出てくる塊が見えてきて、それがよくある表現ということにだんだん結びついています。1つ1つのWeb ページを処理する要領で、複数のページに広げて沢山集めると1まとまりの巨大なデータになります。WWW のページは億の単位にできます。大きいのでなかなか個人レベルでは難しいですが、基本的にはこんな感じで作るのです。

テレビ字幕からのコーパス作成

次にテレビ字幕の話題に入りたいのですが、ご存知でしょうか。知らない人は全然気がつかないかもしれませんが、地上波デジタル放送には字幕放送というのがあります。テレビ番組表を見たときに、「字」と書いてある番組は、リモコンで字幕をオンにすると、画面に文字が出るようになります。我々の研究グループではこうしたデータを集めているのです。どのくらいあるかと言うと、日本の地上波デジタル放送だけでやっているのですが、東京7局キー局について、総務省によると全体の番組の大体60%位が字幕放送になっているようです。この字幕の文字なんですが、確認したところ、音声の要約はあまりされておらず、しゃべっている音の文字がそのまま書かれています。それで、この間までやっていた朝の連続テレビドラマの『ひよっこ』などでは、そのまま訛りの文字が書かれています。テレビなので話し言葉の情報が多い字幕の文字データを収集しています。字幕の収集でどういうことをやるかと言うと地味な作業が続きます。番組表を自動的に解析して、字幕放送のマークが書いてある番組だけをピックアップして、コンピュータを利用して、実際は普通のパソコンなんですけれども、パソコンの裏にテレビチューナーが刺さっていて、テレビの画像を取り込みます。パソコンの上で、自動予約プログラムが動いていて、字幕の文字が書いてある番組のみを片っ端から録画するのです。録画されて記録されたデータの中から、字幕のデータを自動

抽出するプログラムを動かします。対象の放送局が7局あり、1つのコンピュータにチューナーが4つあるので、2台のコンピュータを並べて24時間動かしっぱなしで記録しています。その後ですが、この記録したデータから字幕データと，番組の情報（番組の説明、放送時間、あとドラマとかバラエティーとかのジャンルの情報など）をファイルで記録します。放送データから取り出した字幕データは図6のようになります。

図6 字幕データの例

```
[Script Info]
; Script generated by Aegisub v2.1.2 RELEASE PREVIEW (SVN r1987, amz)
; http://www.aegisub.net
;
Title: Default Aegisub file
ScriptType: v4.00+
WrapStyle: 0
PlayResX: 1920
PlayResY: 1080
ScaledBorderAndShadow: yes
Video Aspect Ratio: 0
Video Zoom: 6
Video Position: 0

[V4+ Styles]
Format: Name, Fontname, Fontsize, PrimaryColour, SecondaryColour, OutlineColour, BackColour, Bold, Italic, Underline,
StrikeOut, ScaleX, ScaleY, Spacing, Angle, BorderStyle, Outline, Shadow, Alignment, MarginL, MarginR, MarginV, Encoding
Style: Default,MS UI Gothic,90,&H00FFFFFF,&H000000FF,&H00000000,&H00000000,0,0,0,0,100,100,15,0,1,2,2,1,10,10,10,0
Style: Box,MS UI Gothic,90,&HFFFFFFFF,&H000000FF,&H00FFFFFF,&H00FFFFFF,0,0,0,0,100,100,0,0,1,2,2,2,10,10,10,0
Style: Rubi,MS UI Gothic,50,&H00FFFFFF,&H000000FF,&H00000000,&H00000000,0,0,0,0,100,100,0,0,1,2,2,1,10,10,10,0
//
[Events]
Format: Layer, Start, End, Style, Name, MarginL, MarginR, MarginV, Effect, Text
Dialogue: 0,0:00:02.95,0:00:05.30,Rubi,,0000,0000,0000,,{\pos(900,898)}む\N
Dialogue: 0,0:00:02.95,0:00:05.30,Default,,0000,0000,0000,,{\pos(540,1018)}[illegible]➡\N
Dialogue: 0,0:00:05.30,0:00:08.12,Default,,0000,0000,0000,,{\pos(340,838)}[illegible]\N
Dialogue: 0,0:00:05.30,0:00:08.12,Default,,0000,0000,0000,,{\pos(700,1018)}[illegible]\N
```

図6で示した字幕データには、1つのファイルの中にテレビ画面の字幕を表示するためのテレビの解像度がどうなっているとか、字幕がどういうフォントを使うとか、色を何色にするとか、字幕を画面のどの位置に何秒後から何秒表示するとか、そういう情報が全部のっていて、それと実際の字幕のデータが入って

いるという形のデータになっています。研究では、このおおもとの字幕データから、画面に表示される文字列部分だけを取り出して使います。ただし、画面のサイズに合わせて字幕は細切れになっていますので、できる限り1つにまとめて文の単位になるように処理を行います。取り出された字幕データはこのようになっています[1]。

1　0:00:02.95　0:00:11.64　これは字幕データの例です。
2　0:00:11.64　0:00:14.91　(太郎)わぁ　大きな街だなぁ！
3　0:00:11.64　0:00:14.91　(桃子)どんな食べ物があるかしら〜!

各項目がタブという記号で区切られ、最初が文番号、次が表示の開始時間、終了時間で、最後が字幕データ本文です。7局の字幕放送は月に約4,200番組あります。それで私たちのグループは2012年の12月から収集を開始しており、いろいろあってシステムが安定するまでの間、コンピュータのプログラムが停止したとか、停電に見舞われたりとかでデータが取れない時もありましたが、大体平均して月に4,200番組収録した結果、今もそれを続けているのですが、ちょっと古い状態ですが、2017年3月の時点で232,000番組分の蓄積があります。それで文に直したら8900万文ぐらいのデータになります。全部テレビ字幕からとったデータです。では、さっきの形態素解析で単語に区切った場合、何語ぐらいになるでしょうか？この段階で約9億1300万単語になります。2012年末から2017年まで4年延々とテレビのデータを録りためていって、やっと9億語の規模になりました。今年で10億のオーダーに達する見込みです。

内訳をみると公式に番組表に載っているジャンルで13ジャンルあります。アニメーション、カルチャー、スポーツなどが含まれていてニュースとかバラエティーとかの字数が多いです。ニュースは書き言葉に近いですが、バラエティーとかドラマは会話をするので話し言葉が多いです。

テレビ字幕内の単語集計

この規模になると各単語の出現数を数えるのも大変なんですが、数えました。出てきた単語が何種類あるかという異なり語数とそれぞれ何回出現するかとい

[1] 著作権の問題により本物は提示できないため作例です。

う延べ語数を集計するやり方は Web ページのときと同じです。何回出てきたかと言う頻度の多い順に並べた表が次の表1です。

表1 テレビ字幕内の頻出語

	語	頻度	カバー範囲		語	頻度	カバー範囲
1	。	50,851,598	0.057	11	と	10,842,740	0.259
2	の	32,198,588	0.091	12	し	9,103,451	0.269
3	て	22,471,338	0.116	13	ん	8,673,383	0.274
4	に	20,376,283	0.138	14	ます	8,532,087	0.288
5	た	19,639,916	0.159	15	ね	8,135,425	0.297
6	は	18,771,055	0.180	16	?	7,549,670	0.305
7	が	18,899,227	0.201	17	!	7,497,030	0.313
8	を	16,204,739	0.218	18	な	7,466,193	0.321
9	です	13,843,751	0.234	19	…	7,446,579	0.330
10	で	12,399,057	0.247	20	も	7,052,203	0.337
Total frequency of top 10,000 words of 640,697						740,589,219	0.811

これはトップ20までの表です。当たり前なのですけれど、「てにをは」がやっぱり多いというのが日本語の特徴であることがよくわかります。内容語ではなくて機能語が多く、文の区切りを「。」として表示していることが多いので、中には区切りがついていない「。」が入っていない番組もありますけれど、大抵「。」が入っているので「。」がやっぱり多いのです。文数とは一致しないのは「。」が入っていないものもあるからです。ここで見る限りは、今のところこの表にあるものが多いです。あと表内に「カバー範囲」と言う項目が書いてありますが、これは何かと言うと、全9億語の出現のうちのどのくらいの割合で埋まっていくかという

のを表しています。「も」までの上位20位で全体の33%とありますが、これは上位20種類の単語で9億語の出現数の33%をカバーしているということを意味します。例えば、これらの単語20種類をマーカーでマークしていくと33%がマーキングされるということになります。現在のコーパスでは、異なり語数が64万697種類あるのですが、そのうちのトップの10,000種類を集めてくると全体の81%になります。すごく荒く言うと、頻出1万語でテレビのスクリプトの81%は理解できると言う話になります。英語の勉強などで、覚えるべき頻出単語何単語などという言い方をすることがありますが、それと同じように考えると、リアルな実数で実際の4年とか5年分の、テレビの字幕からとったデータから、頻出1万語を覚えると81%カバーできると言えるわけです。

この先の出現頻度の高いものを100位ぐらいまで見ていっても内容がわかるような語はほとんどでてきません。さらにその後の上位語で意味のわかりそうなものをいくつかみてみましょう。105位に「日本」、115位に「今日」、119位に「自分」、114位に「ありがとう」、149位に「僕」、157位に「みんな」というような感じです。ところで、単語の数の数え方というのはちょっと難しくて、さきほど形態素解析で切った状態で数で数えていましたが、表層の文字のままで出ている状態で活用なら活用したままを数える数え方と、活用しているものを基本の形にまとめた形で数える数え方と大きく2つあります。今回お見せした集計は、表層のままで活用しているものを数えているので、例えば「わかる」と「わかって」が出てきた場合に、それぞれ別々のものとして数えています。この数え方で異なり語数がだいたい60万語になりました。もし「わかる」と「わかって」を基本形の「わかる」で辞書的にまとめるともっとまとまってしまうので、全体の異なり語数も少なくなります。さて、これだけデータを集めてこのデータをどう使うのかということは悩ましかったりします。ざっと見る限り、高頻度語はほとんどの字幕に出現する機能語ですので、内容の理解という観点ではあまり意味がないということもいえます。一方、低頻度の語は滅多に出てこないわけですし、出てきても大きな意味はないかもしれないのです。一般的には中頻度の語の中に何かと使い勝手の良い言葉が並んでいると言えるでしょう。

テレビ字幕のNグラム

N=1 幾重にも
N=2 幾重にも 広がる
N=3 幾重にも 広がる 芳じゅん
N=4 幾重にも 広がる 芳じゅん な
N=5 幾重にも 広がる 芳じゅん な 香り
N=6 幾重にも 広がる 芳じゅん な 香り と
N=7 幾重にも 広がる 芳じゅん な 香り と その
N=8 幾重にも 広がる 芳じゅん な 香り と その 味わい
N=9 幾重にも 広がる 芳じゅん な 香り と その 味わい 。

次に1つの単語ではなくて複数の単語の連続 N 個をひとまとまりとして出現数を数えるということを試みます。例えば文として「幾重にも広がる芳醇な香りとその味わい」という文があったとします、形態素解析したところ、「幾重にも/広がる/芳醇/な/香り/と/その/味わい/」と各単語に区切られたとします。ここで N を1から9まで変化させて先頭から連続する N 単語をひとかたまりにすると次の表のようになります。これを単語単位の N グラム（エヌグラム）と呼びます。

	4-grams	頻度		5-grams	頻度
1	て い ま す。	922,441	1	し て い ます 。	198,004
2	です よ ね 。	473,551	2	て い まし た 。	180,974
3	ん です ね 。	418,657	3	ん です よ ね 。	169,054
4	ん です か ?	383,728	4	れ て い ます 。	130,255
5	し まし た 。	359,471	5	じゃ ない です か 。	115,157
6	ん です よ 。	353,923	6	て き まし た 。	114,912
7	と 思い ます 。	285,285	7	ありがとう ござい まし た 。	95,353
8	て い まし た。	248,687	8	よろしく お願い し ます 。	92,480
9	ん です よ ね	209,923	9	な ん です よ 。	86,991
10	い まし た 。	208,025	10	に なり まし た 。	86,235

先程のテレビ字幕コーパスの中で4グラムと5グラムを数えると出現数の上位10位までが次の表のようになります。
あまり面白くないですか？でも重要な特徴としては、見事に文末表現が上位にきています。日本語は文末が重要なので何か意思を表したりとか主張を表したりとか、この文末の形というのがかなり高頻度でパターン化して固まって出てくるということがこのデータからはうかがえます。N をもうちょっと広げて 6 と 7 の場合が次の表のようになります。

	6-gram	頻度		7-gram	頻度
1	されています。	49,239	1	と思うんですよね。	14,552
2	なんですよね	42,644	2	はありませんでした。	9,411
3	していました。	36,664	3	うとしています。	7,608
4	ているんですね。	24,374	4	どうしたんですか？	6,727
5	となっています。	23,636	5	いないいないばあっ！	6,283
6	たんですよね。	21,266	6	んじゃないですかね。	6,246
7	ではありません。	18,590	7	てみたいと思います。	6,194
8	と思うんですね。	18,279	8	んじゃないでしょうか。	5,761
9	んじゃないですか？	18,006	9	だと思うんですね。	5,288
10	んじゃないかなと	17,994	10	んじゃないかと思います	5,119

この中で7グラムの5番目は特殊です。NHK 教育には「いないいないばあっ！」と番組があって、その番組では頻繁に「いないいないばあっ！」という表現が出てきます。そのため、このコーパスでの出現回数が多く出ているのですが、これは一般的な現象ではなくてこの番組のバイアスがかかっている例といえるでしょう。それ以外は、普通の会話の中で文末に出てくる文末のパターンだと思います。やはり、N を増やしても文末表現が上位に出てきました。

ところでどうしてこうした N グラムパターンをとっているかというと、日本語教育のための教材をここから取り出して使おうということを考えています。日本語の会話でよく使う表現というのが、単語ではなくて実際に出てくる塊で、さ

らにその表現があると、その前に出てくる表現はどういうものかと言う事がこのコーパスを使うとリアルなデータとしてわかります。そのため日本語の会話コーパスで実際に出てくる表現を取り出して、さらに分類してデータ化としようということを考えています。例えば、6グラムのところで「されています。」という表現が1位に出てきますが、ではその「されています。」の直前にはどんな言葉が出て来るのかというのを調べてみると、「期待」「展示」「掲載」「販売」「目撃」「公開」という表現が非常に多く出てくることがわかります。「されています。」自体は4万9千回ほどでてくるのですが、その前の語はこの6種類のどれかであることが多いというのは不思議です。これは定型表現とかという言われ方をする決まり文句とかを探していることに相当するのですが、このデータを見る限りは、「されています」とくれば、前に「期待」とか「展示」とか「掲載」といった語が来ることが多いといった具合に予想できるということです。ちょっとあんまりピンとこないですかね。

別の例として「こんにちは」を含む表現と言うのを取り出して調べてみると、「どうもこんにちは」とか「皆さんこんにちは」と言うのが多いです。「皆さん」の漢字とひらがなの違いはありますが。「すみません」を含む表現を取り出してみると、「すみません失礼します」「すみませんでした」とかが、あまり回数は多くないのですが出てきます。「よろしくお願いします」とか「ありがとうございます」とかこういう表現が非常に多いです。我々はよくこの言葉を口にするということを反映しているのだと思います。大事なんですね。

たくさんの N グラムパターンをとってみると、おそらく言語教育のためにも有用なパターンがたくさん含まれていると考えられます。「単純に頻出のパターンさえ覚えれば日本語の理解に役立つので、まずはこれだけを覚えよう」みたいなリストが簡単に作れたら最高なのですが、実際にこれだけのデータをとって見てみると、頻度と有用度の関係はそう単純ではないので、どのようにリストを作るかなど悩ましいことがいっぱいあります。それを含めて研究というのはなかなか簡単にはいかないものです。大量のデータとそこから得られる言語パターンをどうやって捉えたらいいのかなという試行錯誤が続いています。数億語の単位になると単純に数を数えるだけでも非常に大変な作業ですが、プログラムを作ってこんな感じで研究を続けています。

word2vec という語のベクトル表現

最後に、自然言語処理のツールで非常に有名な word2vec というものを使った話をします。コーパス内の語の連続を語の特徴を表す文脈データに見立てて、各語を意味的な特徴を表す数値ベクトルという形で表現します。多くの場合200次元ほどのベクトルで各語を表すのですが、この200次元というのがざっくり言って語の意味を表す200個の要素、というように考えることができます。ある語が数値200個に置き換えられると想像してください。一旦このように語が数字に置き換えられると、ある語とある語の間の意味的な関係性は200個の数値と200個の数値の間の数値計算によって測ることができるようになります。もしも2つの語の意味的な特徴が類似していると、200次元と200次元の数値ベクトル間の距離が近くなります。その特徴を利用するとある語の類語を調べることができるようになります。最後に、ちょっとそれをお見せしようかと思います。

Word	Cosine distance
こんばんは	0.796703
お疲れさまです	0.778086
おかえりなさい	0.762287
おかえり	0.722781
ありがとうございます	0.694695
ごめんください	0.684179
さようなら	0.677760
お邪魔します	0.667552
はーい	0.663395
ごきげんよう	0.654722
おはよう	0.642613
すいませんありがとうございます	0.642137
さようでございますか	0.639013
お疲れさまでした	0.637674
ご苦労さま	0.637072
そういえば…	0.634369
どういたしまして	0.629217
少々お待ちくださいませ	0.628805
ただいま…	0.628740
バイバ～イ	0.628297
いただきまーす	0.626046
はじめまして	0.624270

この表は word2vec の distance というツールを使ってテレビ字幕コーパスの中でドラマのジャンルに絞って「こんにちは」と意味的に近い表現を計算した結果です。

各表現の200個の数値ベクトルが「こんにちは」の数値ベクトルとどのくらい近いかは、コサイン距離という計算式で計算されています。コサイン距離は完全に一致する場合に1.0になります。この中で一番近い「こんばんは」で0.796という数値だということがわかりますね。ざっと見ると、どうでしょう？ 挨拶表現の類似度が高い傾向が見られます。おそらくですが、「こんにちは」という言葉が現れるような場面と、類似した場面では「こんばんは」とか「お疲れさまです」のような会話を開始する時にするような挨拶表現が出てくる傾向があって、それを word2vec のベクトル表現の200次元で捉えられた結果ではないかと想像できます。特別な辞書などを使ったわけではなく、

大量のテキストデータからなるコーパス内の語と語の出現パターンから計算した結果としてこのような関係が得られるということは大変に興味深いと思いませんか？　もう少し違った例をお見せしたいと思います。

Word	Cosine distance
その節は	0.632246
ありがとうございます	0.607437
ありがとう	0.600719
これはどうも	0.596350
お待たせしちゃって	0.590183
お邪魔しております。	0.580752
すいません	0.551259
すみません	0.538490
ありがとうございました	0.533937
来て頂いて	0.533487
お忙しいところ	0.521976
どうも…。	0.520191
お世話さまでした	0.518421
お世話さまです	0.517482

右の表は「どうも」との類似度が高い表現の例です。この結果を見ると、どうもと近い表現として「これはどうも」「どうも…。」が出てきました。しかし、類似表現はこれだけ、他の表現は「どうも」につながる言葉のようです。特に、お礼（どうも「ありがとうございます」、「ありがとう」、「ありがとうございました」など）とお詫び（どうも「お待たせしちゃって」「お邪魔しております」「すいません」など）の表現が多いことがわかります。現実の日本では「どうも」という表現が来ると、それに続いてお礼やお詫びの表現が来ることが頻繁に起きることが伺えます。

Word	Cosine distance
次の方	0.676111
召し上がれ	0.647501
お上がりください	0.638669
お掛けになって	0.625444
遠慮なさらず	0.616894
お上がりください。	0.613021
おかけになって	0.611219
こちらへ。	0.611132
お引き取りを	0.607475
お乗りください	0.600468
中へ。	0.599897
召し上がってください。	0.595524
お構いなく	0.592142
お掛けください	0.577088

最後に「どうも」と1文字違いですが、意味は全然違う「どうぞ」について見てみようと思います。この場合、類似表現ではなく、「どうぞ」につながる表現がたくさん出てきました。

（どうぞ）「次の方」、（どうぞ）「召し上がれ」、（どうぞ）「お上がりください」、（どうぞ）「お掛けになって」、（どうぞ）「遠慮なさらず」などと続く表現が非常にたくさんあることがわかります。これは、我々が普段使っている日本語での「どうぞ」の使い分けを示すデータであると考えることもできるでしょう。「どうぞ」もたくさんのつながりのバリエーションがありますね。同じ「どうぞ」でも、ドラマでなく

て情報番組系のものの場合は、「ご参考になさってください」とか「次回もご覧下さい」とか、こんな感じで変化します。言語データをたくさん集めて分析していくと、場面とかジャンルの変化に応じて同じ言葉と近い言葉も変わっていく、というような言葉の使い分けのようすもわかります。ジャンルとか場面ごとに言葉が違うという様をデータとして取り出すことは言語教育のための重要なデータを取得していることになると考えられます。例えば、言語教育の Can-do という考えがあります。ある表現を覚えると、その表現は、こういう場面で、こんな意味や用法として使えますよ、とか、レストランで食事を頼むときに使うスクリプトとか、ある場面で何を目的にするとどのように喋れば良いかということを中心とした語学教育が行われることが最近では主流になっています。こうした Can-do の教材に成り得るいろんな場面でのいろんな言葉を使った会話がこのテレビ字幕コーパスの中にリアルな事例として入っています。現在は、その場面ごとにどういう表現が出現し、どういう意図で使われているか、ということを調べて、Can-Do を学ぶための事例集に結びつけようと研究をしている最中です。

これだけの規模の日本語字幕データを扱っているところは我々のグループだけなので、他ではやっていないことをやっているのですが、まだまだ成果に結びついていませんので、これからも研究を進めてちゃんとデータを出していかなければならないと思っています。（完）

コラム †　　語の出現頻度とNグラムの出現頻度

コーパス内の語の数え方には、総語数を表す述べ語数と、同じ語は1語として全体で異なる語がいくつかを表す異なり語数があります。今回紹介したテレビ字幕コーパスは、述べ語数で約9億語、異なり語数は約64万語です。また、文の数は約8千9百万文でした。こうした値は実際にコンピュータを使って数えた結果です。数億語規模の数え上げにはどんな高性能マシンが必要なのかと思うかもしれませんが、実は、ハードディスクの容量が 1TB ほど、メモリが 8GB ほどのパソコンで可能です。現代ではそれほど高級なパソコンというわけでもありません。素晴らしい世の中になりました。

講義では N グラムの話もしました。N グラムでは、コーパスの中で連続して出てくる「私/は」「私/が」のような N 語のパターンを観測できるので、語単独よりも言葉の特徴をよく捉えることができます。ただし処理は少し複雑で、文頭から文末まで1語ずつずらしながら N 語のかたまりを取り出す処理が必要です。例えば、10語からなる文「ハワイ/と/いえ/ば/、/やっぱり/パンケーキ/です/ね/。」の2グラムは、文頭から「ハワイ/と」「と/ハワイ」とずらしていき、最後が「ね/。」の9パターンです。Nを1から10までとして全データを取る場合、Nが1の時に10パターン、Nが2の時に9パターン、以下Nが増えるにつれてパターン数が減り、Nが10の時に1パターンです。10語からなる1文内の総パターン数は10+9+8+7+6+5+4+3+2+1で55個です。テレビ字幕コーパスの8千9百万文では、ざっくり言って55倍の48億9千5百万になります。9億語が、1から10のNグラムにしたら約49億パターンにまで膨らんでしまいました。このような少し複雑でデータ量の多い処理にはプログラミング知識も必要です。それでもコーパスを研究対象にする人にとって研究の幅がぐんと広がりますので、テキスト処理のプログラミング技術は習得する価値があります。ぜひ挑戦してみてください。

プログラミングを知るための３冊

・＊・…‡…・＊・＊・…‡…・＊・…‡…・＊・

中植正則、太田和志、鴨谷真知子「Scratchで学ぶ プログラミングとアルゴリズムの基本」日経BP社 2015年

最近盛んな小学生向けプログラミング教室では、機能がわかりやすく、簡単な操作で動くビジュアルプログラミング言語が使われています。本書はその中で代表的な言語である Scratch の入門書です。パズルのような操作で誰でも簡単に基本を学びながら動くものが作れます。

…・＊ ＊・…

(株)アンク「HTML5 の絵本」翔泳社 2012年

現代のWWWページのほとんどは文章の構造とコンテンツをHTML5で、画面のデザインをCSS3、ページの動きがJavaScriptなどのプログラミング言語で分業して書かれています。本書ではWWWの全体的な仕組みをざっくりと理解し、簡単なページが作れる入門書です。

…・＊ ＊・…

高橋京介「絶対に挫折しない iPhoneアプリ開発「超」入門 第7版」ソフトバンククリエイティブ 2018年

iPhone用のアプリケーションを開発するために必要なXcode 10という開発環境の使い方と具体的な例によるSwiftというプログラミング言語の非常に丁寧な入門書です。Macパソコンを持っている人でiPhoneで動くアプリケーションを作ってみたい人におすすめします。

…・＊ ＊・…