修士論文

テキスト分類のための語根レベル畳み込みニューラルネットワークの研究

1810125　　Tie Xinyong

主指導教員　　由井薗　隆也

北陸先端科学技術大学院大学
先端科学技術研究科
(情報科学)

令和 2 年 3 月

Abstract

Text classification is a classic topic in natural language processing. In recent years, deep learning has achieved promising results in natural language processing, such as Convolutional Neural Networks. Convolutional Neural Networks is useful in extracting information from row signals based on computer vision, and it is also achieved to natural language processing. Character level Convolutional Neural Networks (Char-CNN) achieved good results for text classification, which quantizes text and learns text classification using letters as features. However, Char-CNN has a problem that the learning feature is too abstract and loses meaning to make accuracy is lower than word-level models. In Char-CNN, learning a character string as features is higher accuracy than single character. We attempt to find a method of character sequence for improving CNN using etymology. It can avoid dimensional curse and learns with a meaningful feature by the etymons. However, research on etymon-level deep learning is still scarce.

We propose a method that uses etymology to make the etymon as features and to clarify the effects of this method.

This paper evaluates the performance of etymon-level text classification by comparing it to that of word-level and character-level text classifications.

We conduct experiments with text classification in three methods to evaluate the effect of etymon-level. The first is to evaluate the performance using 5 machine learning models and evaluate the accuracy and training time. The second is to learn with a Convolutional Neural Networks using various corpora, to evaluate improve speed in accuracy and loss. Accuracy and loss are recorded each epoch. Third, we perform text clustering experiments using word embeddings and discuss the results.

We collected etymon data from the Online Etymology Dictionary (www.etymonline.com) using a web crawler. The dictionary contains approximately 44,000 words and etymon information. Next, a large corpus (word-level) is collected, and a etymon-level and character-level corpus is created. Then, the corpus based on characters, etymons, and words is converted to a vector, and feed to machine learning models.

First, all the words in corpus were converted to prototypes, and then, replaced by the etymons. For words such as SUMITOMO, which are in the corpus that are not included in the etymology dictionary, they were converted [Unknown]. In Char-CNN, string type data is interpreted as a list of character types, so a direct loop is used to separate character strings and add them to the character base corpus. Next, corpus is changed to a vector using tf-idf or 1-hot. Then, it is divided to 80% for training data and 20% for test data. Finally, fade data into the machine learning model.

The machine learning models are Naïve Bayes, Support Vector Machine, Logistic Regression, K-Nearest Neighbor and Convolutional Neural Networks. We experiment text classification using

five machine learning models and discuss the results.

We experiment on text classification of Convolutional Neural Networks using corpora with different characteristics. Since the etymon is a common semantic code of a word, the etymon-level model does not increase so much even if the corpus vocabulary amount increases. And the corpuses used have different characteristics. For example, British Broadcasting Corporation news from large vocabulary British editors.

Moreover, the etymon embedding demonstrated good clustering performance. We train etymon embeddings using Skip-grams. And evaluate it in text clustering using K-means.

The performance of etymon-level text classification is clarified. Results show that in NB, SVM, LR, and CNN, etymon-level is better than character-level, and has a accuracy close to the word-based. Etymon-level clearly increases accuracy and decreases the loss faster in CNN. In text clustering, etymon-level is better than word-based. Etymon-level is a competitive method to traditional methods for text classification task.

概要

テキスト分類は自然言語処理における典型的な課題である。近年、自然言語処理の分野では深層学習の研究が盛んでおる。ニューラルネットワークを用いたテキスト分類の研究も多く行われ,優れた成果が得られていた。その中、畳み込みニューラルネットワークと呼ばれる学習モデルは計算機視覚（Computer Vision）に基づき、下位層の信号を処理することで画像処理の機械学習が有効であり、自然言語処理に対しても有効である。また、テキストを量子化し、離散の文字を素性としてテキスト分類を学習する文字レベルテキスト分類の実験も行われていた。しかしながら、文字レベルは学習素性が抽象過ぎて意味を失うなどの問題で、正解率は従来の単語レベルモデルより低い。文字レベルでは、文字の列を素性として学習する手法があり、正解率が単文字より高い。ところで、語源学では、単語が持つ共通の文字列を容易に見つけることができる。語根を素性する語根レベルにより、次元の呪い問題を避け、意味持ちの素性で学習するのは精度が高められると考えられる。しかし、語根レベル深層学習の研究はまだ少ない。

本研究では、語源学を活用し、語根を素性とする手法を提案する。また、その新手法の効果を明らかにすることを目的とする。

本研究では、文字レベル、語根レベル、単語レベルの機械学習を用いたテキスト分類の実験を行い、語根レベルの効果を評価する。

語根レベルの効果を評価するため、テキスト分類に巡り、3 つの角度から実験をする。1 つ目は、複数の機械学習モデルを用い、正解率と学習時間を考察して性能を評価する。2 つ目は、多様なコーパスを用い、畳み込みニューラルネットワークのモデルで学習し、正解率と損失率の変化を考察し、学習の効率を評価する。学習過程におけるデータを全部学習するごと(epoch)で正解率と損失率を記録する。3 つ目は、分散表現を用いてテキストクラスタリングの実験を行い、結果を考察する。

実験では、まず、単語の語根情報を収集し、語根辞書を作成する。次に、大規模なコーパス（単語レベル）を収集し、語根レベルと文字レベルのコーパスを作成する。そして、文字レベル、語根レベル、単語レベルのコーパスをベクトルに変更し、学習モデルにフェイドする。最後に、結果を考察する。

まず、語根辞書は、クローラを利用して Online Etymology Dictionary サイトから約 4 万 4 千の単語の語源データを収集し、辞書ファイルを作成する。次に、語根レベルコーパスは、収集された単語レベルコーパスをすべての単語を辞書に照合し、単語の持つ語根を語根コーパスに追加する。辞書に含まれていない単語は、UNKNOW に変更される。文字レベルは、文字列タイプのデータが文字タイプのリストと見成されるため、直接ループを使って文字列を分け、文字レベル

コーパスに追加する。そして、tf-idf や 1-hot を用い、ベクトルに変更する。その中、8 割を訓練データとし、2 割をテストデータとする。最後に、機械学習モデルにフェイドする。

機械学習モデルは、まず、素朴ベイズ（Naïve Bayes)、サポートベクトルマシン（Support Vector Machine)、ロジスティック回帰（Logistics Regression)、K 近傍法（K-Nearest Neighbor）と深層学習の畳み込みニューラルネットワーク（Convolutional Neural Network）5 つの機械学習モデルを用いてテキスト分類を実験し、その結果を考察する。

次は、特色が異なるコーパスを用い、畳み込みニューラルネットワークのテキスト分類を実験する。語根は単語が持つ共通の意味的な符号であるため、語根レベルはコーパス語彙量が増やすとしても素性数がそれほど増やさない。使用されたコーパスは特色が異なる。例えば、語彙が豊かなイギリス人の British Broadcasting Corporation のニュース、あるいは非公式な言葉が収集される通販コメントなど。

最後に、テキストクラスタリングの実験を行う。語根の意味的な特性を探究するため、Skip-gram を用いた分散表現を学習する。評価手段は K-means を用いたテキストクラスタリングを行い、結果を考察する。

以上より、語根レベルのテキスト分類の効果を明らかにする。実験の結果は NB、SVM、LR、CNN において語根レベルは文字レベルをはるかに超え、単語レベルに近い精度を持っておる。深層学習の効率において語根レベルは明白に正解率の向上と損失率の降下スピードが早い。テキストクラスタリングでは、語根レベルが優れている。語根レベルはテキスト分類に適任し、従来の単語レベルと文字レベルに競争力のある手法であると示している。

目次

図目次

表目次

第1章 はじめに

1.1 研究背景

近年、ビッグデータを活用することにより自然言語処理技術は目覚ましい進展を遂げている。大規模なテキストデータを収集し、統計的な機械学習を通じて人間の言語を処理できる人工知能が実現され、分類検索、機械翻訳、対話システムなどの技術が社会に広まっている。

機械学習における自然言語処理では、テキストがベクトルに変更され、訓練データとして学習する手法がある。計算機設備性能の進歩にともない、ニューラルネットワークを用いた自然言語処理の深層学習モデルも広く研究されるようになっておる。

誤差逆伝播法、最急降下法、LSTM、Transformer、BERT などの自然言語処理ための深層学習の研究が行われている一方、コンピュータビジョンを用いた深層学習における自然言語処理の研究も進んでいる[1,2]。畳み込みニューラルネットワーク（CNN）は、データマトリックスを鮮鋭化することを通じて重みを求める学習モデルであり、自然言語処理に対しても有効である[3]。また、CNN は下位層の信号を処理することで学習することが有効である[4]。近年、テキストを量子化し,離散な文字を下位層素性としてテキスト分類の実験も行われていた[5,6]。

しかし、実験結果を比較すると、文字レベルはテキスト分類において従来の単語レベルとの差がある。その原因は、素性数が減少されすぎることや、意味のないアルファベットを素性とし、抽象的になりすぎるためと考えられる。

文字の列を素性とするのは手段の一つであるが、列が長くなると正解率が高くなることではない。逆に、長くなると計算次元が増加する。例えば、2-gram だと次元が 50(文字レベルの次元数)の二次乗になり、3-gram だと 50 の三次乗になる。最適な長さが決定しにくいという課題がある。

言語学には、語源学という分野がある。語源学では、単語が持つ共通の文字列を容易に見つけることができる。本研究は、印欧語族の祖語（Proto-Indo-European）を用いることにより、自然言語処理における次元の呪い問題が避けられる。また、素性が意味を持つことによって学習の正解率を高められると考えた。

また、近年、語根を中間言語とし、サポートベクトルマシンを用いた異言語のテキスト分類の研究が行われ、語根を用いた自然言語処理は機械学習ができることを示した[7,8]。しかしながら、語根を素性として一般化して使用する手法は明らかではなかった。本研究では、さらに語根を素性とした機械学習の効果について検討し、実験を行う。

1.2 研究目的

本研究では、語源学を活用し、語根を素性とする機械学習を提案する。そして、

その手法の効果を明らかにすることを目的とする。

従来の統計的な自然言語処理は単語の情報を扱い、計算することにより実現されてきた。例えば、n-gams や隠れマルコフモデル、あるいは Word2Vec がある（図 1.1 のように言語からベクトルへ）。

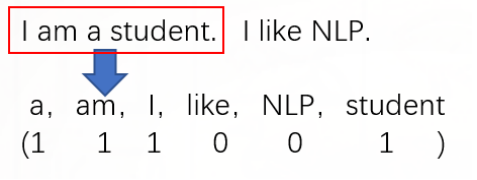


図 1.1：Word2Vec の例

単語を統計する手法以外は、単語列や文字など様々な手法も提案されている。本研究は、単語の持つ共通の文字列を探すことに、語源学の利用を提案する。語根レベルにより、次元の呪いが避けられ、意味の持つ素性で学習するのは正解率を高められると考えられる。しかし、語根レベルを用いた機械学習の研究はまだ少ない。本研究は、その手法を試す一歩である。また、語根は形態素として単語の抽象的な意味を保つである。本研究では語根レベルを用いた意味類似の計算を明らかにする。

自然言語処理分野では次元の呪いは古くからの問題である。豊富な言語情報を統計するために、単語を素性とした項数が多いベクトルが使用される。さらに、言語から変更された疎行列を二次元空間に投射することにより素性数が大幅に削減できる。現在までの手法は単語の統計に基づいた数的な手法である。語根レベルは言語の方面に語の数を減らすことを通じて次元削減する。本研究は、有限な性能の下で効率な計算が出来、さらなる優秀な自然言語処理システム開放に有用である。

本研究では 3 つの実験が行われる。一番目の実験は文字レベル、語根レベル、単語レベルが違う学習モデルにおいての性能の考察する。学習モデルは単純ベイズ、ロジスティクス回帰、サポートベクトルマシン、K 近傍法、畳み込みニューラルネットワーク 5 つある。二番目の実験は、3 つ基礎の手法は特色の異なるコーパスにおいての性能を考察する。実験は畳み込みニューラルネットワークを用いたテキスト分類である。三番目は、ラベルを見せずにテキストクラスタリングの実験を行う。実験は、単語と語根の分散表現を学習し、K-means を用いたクラスタリングを実験する。そして、追加として語根レベルを用いた単語分散表現は類似により意味の計算を観察するため、単語のクラスタリングを実験する。

1.3 論文構成

本論文は以下の通りに構成される。

第 2 章では、過去の文章分類、自然言語処理における深層学習、語源学と本研

究の関連を述べ、また本研究の特色を論じる。
第 3 章では、提案手法について説明する。
第 4 章では、実験データと評価を説明する。
第 5 章では、実験結果を説明し、考察する。
第 6 章では、本論文の結果をまとめ、今後の課題を述べる。

第2章 関連研究

本章では、機械学習を用いたテキスト分類の関連研究について説明する。2.1 節では、テキスト分類の研究を紹介する。2.2 節では、自然言語処理における深層学習の先行研究を紹介し、本研究に関連が強い畳み込みニューラルネットワークテキスト分類に注目する。2.3 節では、語源学を紹介し、語根をどう機械学習に利用するかについて説明する。2.4 節では、語根レベルを利用した機械学習の過去研究を紹介する。最後に 2.5 節では、本研究と過去研究の関連と違いについて議論する。

2.1 テキスト分類

テキスト分類とは 1962 年まで遡り、自然言語処理における古典的な主題であり、最も基礎的な研究課題である[9]。テキスト分類は、図 2.1 のよう、複数のトピックの文章にラベリングを施し、文章を学習サンプルとし、類の未知の文章を判断することである。

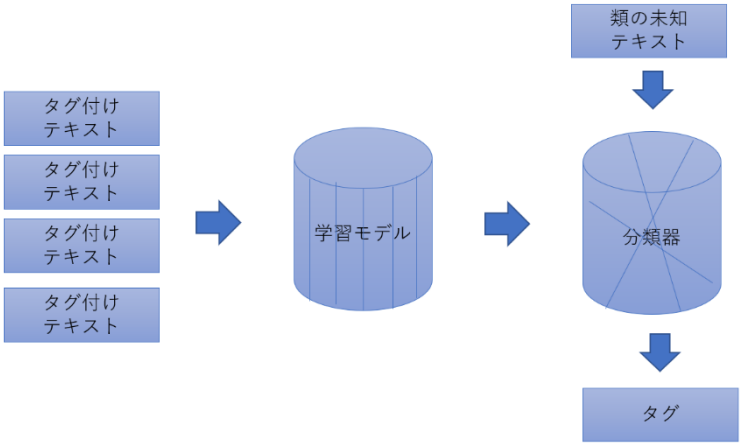


図 2.1：テキスト分類

テキスト分類の研究は、長年を渡り、現在まで様々な手法がある。例えば、CBOW や n-gram など、多くの優れているテキスト分類モデルが提案されている[10]。これらの言語情報を、Naïve Bayes、Support Vector Machine などの統計的機械学習モデルに取り込み、目覚ましい成果が収められている。例えば、Li らはテキスト分類の実験を行い、単純ベイズ F 値 0.921 と得サポートベクトルマシン F 値 0.973 の結果を得た。また、Ravichandran らはサポートベクトルマシンを用いて実験を行い、F 値 91.44%の結果を得た[11,12]。

テキスト分類は機械翻訳、検査エンジンなど様々な自然言語処理技術に関連する。本研究では新しい手法を試すため、テキスト分類の実験を行う。

2.2 文字レベル畳み込みニューラルネットワーク

近年、人工知能の研究が盛ん、様々な優れている機械学習モデルが提案された。自然言語処理は機械学習の脚光を浴び、目覚ましい研究成果が出た。近年、誤差逆伝播法、最急降下法、LSTM、Transformer、BERT などの自然言語処理における深層学習の研究が行われている[1,2]。一方、コンピュータビジョンを用いて自然言語処理のための深層学習の研究も進める。畳み込みニューラルネットワーク（CNN）は、データマトリックスを鮮鋭化することを通じて、重みを求める深層学習モデルの 1 つである。その学習モデルは自然言語処理に対しても有効な手法である[3]。

CNN は下位層の信号を処理することで学習することが有効である[4]。Zhang らは文字レベル畳み込みニューラルネットワーク（Char-CNN）を提案した。彼らはテキストを量子化し,離散の文字を下位層の信号と見なし、文字レベルの文章分類実験を行われていた[5]。モデルの構造は図 2.2 に示す。

まず、彼らは一次元の畳み込みカーネルと一次元の max-pooling を設置し、6 層の ReLU 関数のような畳み込み層とプーリング層を設置する。モデルは確率的勾配降下法（GSD）を利用し、バッチサーズは 128、momentum は 0.9、初期ステップサーズは 0.01。そして、エポック（epoch）ごと一定の量のサンプルをランダムに各クラスに配布する。

次に、テキストを量子化する。まずはテキストを受け入れ、特定したサーズを超える部分を無視し、受け入れた文字列を 1-hot でエンコーディングする。素性表は 26 つ英文字、10 つ数字、33 つ符号や/n などを含めて 70 つある。素性表以外の文字が入ったら、0 ベクトルに見なす。

モデルの構造は図の通りに設置している。モデルには 6 つの畳み込み層と 3 つの全連結層がある。量子化されたテキストは 70 かける特定の長さのマトリックスになり、訓練データとなって訓練する。

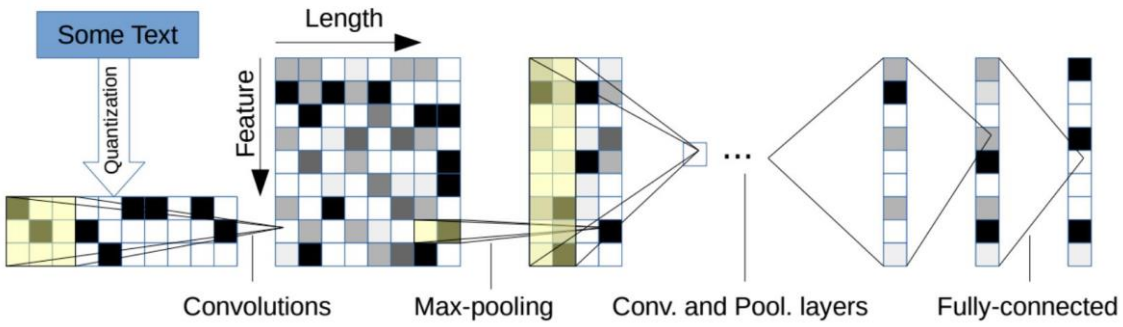


図 2.2：畳み込みニューラルネットワークを用いたテキスト分類

また、それぞれの文字レベルの研究はテキストを量子化し、文字列を訓練データとして機械学習の実験を行う。

2.3 語源学

言語学には、語源学という分野がある。現在使われている言語の一部は古言語から変化したものであり、一部はいくつかの古言語が混在して形成されている。語源学はいくつかの古代のテキストを解読してその他の種類の言語を比較し、1つの言語の発生、変化と消滅を研究し、語源学は語の歴史を掲示することに力を入れる。

語源学はまた、それらの言語の情報を推測し、親族の言語を比較することにより、近い母言語を得ることができ、発見された語根はその原始的な語族にまでさかのぼることができる。例えば、英語は印欧語族の祖語（Proto-Indo-European）を持っている。それで、これらの意味を持つ共通のアルファベット列を容易に見つけることができる。。

英単語では複数の抽象的な意味を含む PIE 語根の組み合わせを通じて構造されている。例えば、「together」を意味する「con」と「to carry」を意味する「fer」を組み合わせ、「confer」になる。従って、「緒に持って行く」は違う場面で「話し合う」「与える」「比較する」意味を持つ。さらに、「the act or fact of verb-ing」を意味する「ence」という名詞の詞綴りを追加し、「conference」になる。「一緒に（論文）を持って行くこと」。

他には、「re」や「trans」などと組み合わせ、「refer」「transfer」などの英単語になる。また、「con」や「in」などを「flu」と組み合わせ、「confluence」と「influence」になり、これらの共通の文字列で英語を構成する。

語根を利用し、形態素の網が構成できる。例えば、図 2.2 は語根「fer」と他の語根の連結網:

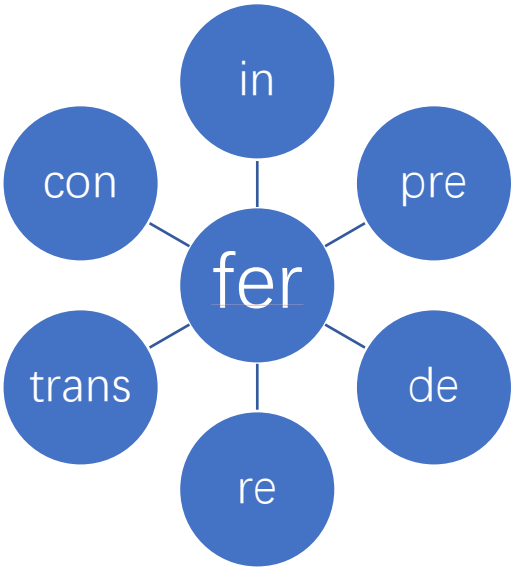


図 2.3：語根「fer」とその他の語根の関係図

「fer」は「to carry」を意味する。
「con」は「together」を意味する。
「in」は「in」を意味する。
「pre」は「before」を意味する。
「de」は「down」「from」「off」「apart」を意味する。
「re」は「back」「again」を意味する。
「trans」は「across」「beyond」「over」を意味する。

本研究では現在インターネットでは公開されたオンライン・エティモロジー・ディクショナリー[1]のウェイブサイトから語根の情報を収集する。

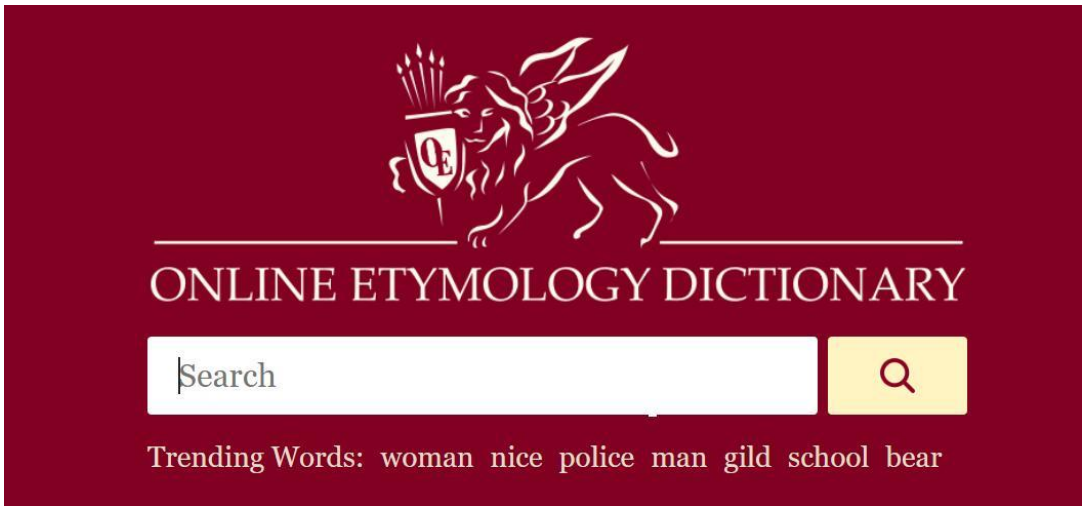
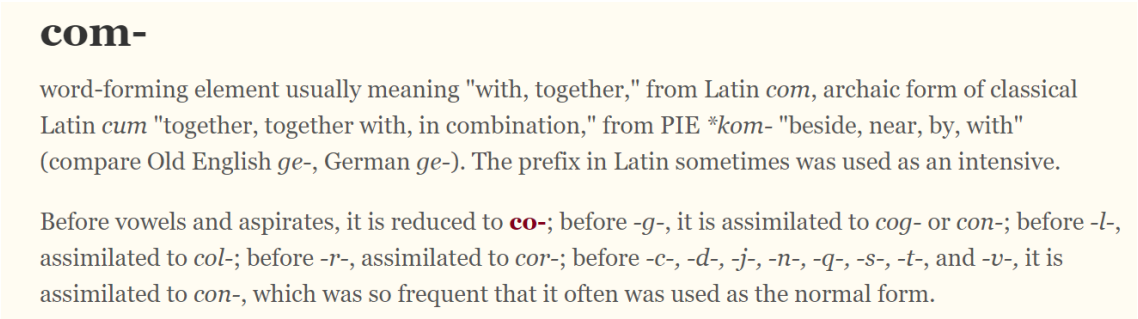


図 2.4：オンライン・エティモロジー・ディクショナリー[2]

[1] オンライン・エティモロジー・ディクショナリー（Online Etymology Dictionary）は、歴史家・作家である Douglas Harper が、作成を開始した語源辞典サイトである。一般的な単語だけでなく、スラングや専門用語も含む単語の由来を保存するために約 3 万語の単語データが収録されている。その内容は「オックスフォード英語辞典」を中心とし、他にミシガン大学による「中世英語辞典」や「バーンハート語源辞典」などから編纂されている。

[2] 図 4.1 はオンライン・エティモロジー・ディクショナリーウェイブサイトのインタフェースである。

オンライン・エティモロジー・ディクショナリーでは、収録されている単語の語根、語源、中古印欧言語と関連単語の情報が記載されている。(図2.5と図2.6)



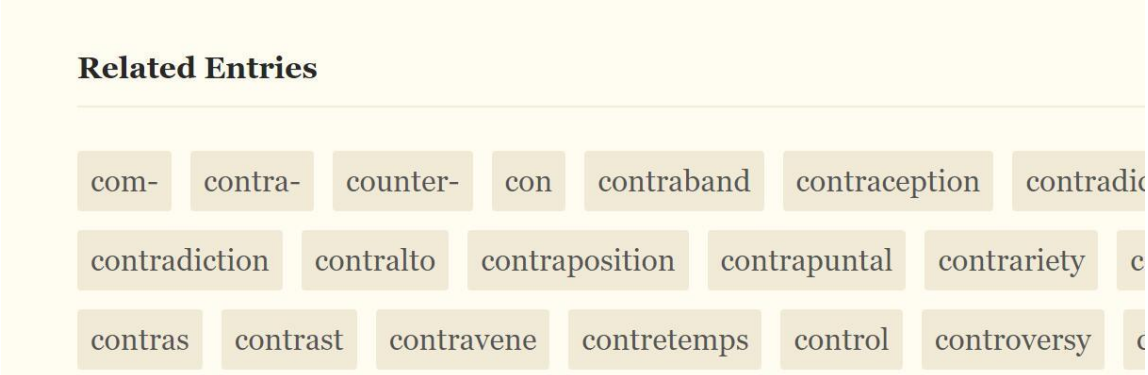
com-

word-forming element usually meaning "with, together," from Latin *com*, archaic form of classical Latin *cum* "together, together with, in combination," from PIE **kom-* "beside, near, by, with" (compare Old English *ge-*, German *ge-*). The prefix in Latin sometimes was used as an intensive.

Before vowels and aspirates, it is reduced to **co-**; before *-g-*, it is assimilated to *cog-* or *con-*; before *-l-*, assimilated to *col-*; before *-r-*, assimilated to *cor-*; before *-c-, -d-, -j-, -n-, -q-, -s-, -t-*, and *-v-*, it is assimilated to *con-*, which was so frequent that it often was used as the normal form.

図 2.5：語源情報

[3]



Related Entries

com- contra- counter- con contraband contraception contradic

contradiction contralto contraposition contrapuntal contrariety c

contras contrast contravene contretemps control controversy c

図 2.6：関連語

[4]

2.4 語源学を用いたテキスト分類

現在まで、語源学を用いた自然言語処理の研究はまだ多くない。2013 年、Nastase らは語根を中間言語として多言語のテキスト分類を実験し、正解率 89%、F 値 80%の結果を得た[7]。

Nastase らは英語とイタリア語がある 4 種類文章をデータとし、語源辞書から共通の祖語を取り出し、CBOW ベクトルに追加し、サポートベクトルマシンを用いた異言語テキスト分類の実験を行った。実験結果が単純 CBOW より F 値が高いことは分かった。

[3] 図 3 はオンライン語源学辞書のとある語根の語源情報である。
[4] 図 4 はオンライン語源学辞書のとある語根の関連語の情報である。

2.5 単語分散表現

Word2Vec は n-gram や HMM より次元削減されたが、まだ疎行列の計算などの問題がある。Mikolov らは単語前後情報のベクトルを二次元マトリックスに投影し、単語分散表現モデルを提案した[13]。単語分散表現においては、単語のコンテスト情報を統計し、意味の計算が実現した。例えば、

KING - MAN + WOMAN = QUEEN

また、Kunser らは単語分散表現を用いて文章の距離を計算し、K 近傍法を用いたテキスト分類の実験を行った[14]。

2.6 本研究の特色

本研究では、以上に述べた先行研究を参考しつつ、語根を素性とする機械学習モデルを提案する。本研究の目的は、単語ではなく、語源から学習する機械学習の試み、その手法の効果を明らかにする。

CNN は画像パターンに対する人間の認知を模倣から来ており、データを鮮鋭化して重みを計算して重要な特徴を選び、その後接合する。本研究の提案手法は、第一層の畳み込み層を省略し、もっと正しく、言語学的に一般的な文字列の重みを決定することに相当すると考えられる。学習モデルの前に自然言語のコンピュータ視覚を加えたことに相当すると考えられる。

また、本研究において、印欧語族の祖語により、英語の機械学習の次元の呪いを避け、意味持ちの素性で学習するのは有効に精度が高められると考えられる。

従って、本研究は自然言語理解、自然言語処理ための機械学習においての次元削減の研究にとって有意義である。

Zhang らの研究では、畳み込みニューラルネットワークが下位層の信号を処理するという特点を利用し、テキストを量子化することを試み、アルファベットを特徴とすることをコンセプトとなる。本研究では、Zhang らの文字レベルの畳み込みニューラルネットワークを用いたテキスト分類の実験を踏まえ、ニューラルネットワークの特性を考察し、モデルの改良を求め、語根レベルにたどり着いた。文字レベルから進化し、完全に語意を無視したことではなく、より意味を重視することを通じてテキスト分類の正解率の向上を求める[5]。

本研究は Nastase らの研究語根を中間言語として多言語の文章分類の実験と異なり、文章内容によりラベルを付け、文章の意味により分類する実験を行う[7]。

単語分散表現表現は、単語の前後にある単語の統計情報を用い、文脈から単語の表現を学習する手法である。それにより、コンテキストにより意味の計算ができる。それに対して、語根は分散表現に適用するかを考察する。例えば、

etymology - etymo + bio = biology

や

confluent - con + re = refluent

本研究は、Kunser らの単語分散表現を用いた文章クラスタリングに参考し、深層学習における語根レベルのコンテキストにより意味類似の効果を考察する[14]。

第3章 語根を素性とする機械学習の提案

3.1 コンセプト

本研究の提案は、語根レベルは文字レベルの実現と同じである。まず、前処理を行い、単語のコーパスは単語から語根に変更され、語根コーパスを作成する。次に、語根コーパスのテキスト文をベクトルに変換し、それぞれの機械学習に送り、学習を行う。機械学習モデルは従来の単語レベルと同じように行う。

提案手法の実験モデルは、図 3.1 のように実装する。まず、テキスト文にある語を語根辞書に照合し、語根に変更する。次に、変更した語根列をベクトルに変更する。最後に、データを機械学習モデルに送り、学習を行う。

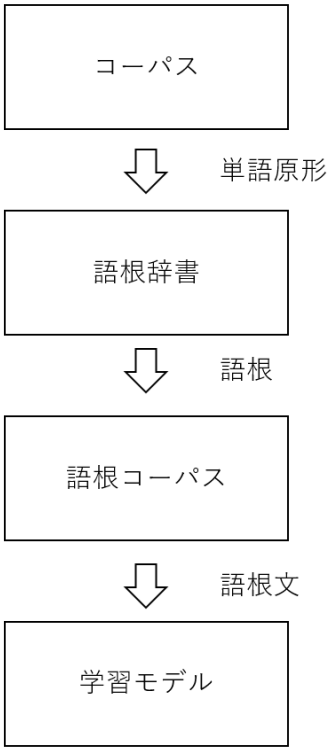


図 3.1：モデルの流れ

3.2 前処理

機械学習モデルにフェイドする前に、テキスト文の前を処理を行う。まず、単語レベルのコーパスを文字レベルと語根レベルのコーパスに変更する。次に、コーパステキストをベクトルに変更し、機械学習を行う。

3.2.1 コーパス作成

コーパスの作成は、まず単語レベルのコーパスをロードし、すべての単語を辞書に照合する。照合される単語が持つ語根を語根コーパスに追加する。また、辞書に含まれていない単語は、UNKNOW に変換される

語根辞書は、クローラを利用して Online Etymology Dictionary サイト（https://www.etymonline.com/）から 44641 語の単語とそれら単語の語根情報を収集し、辞書を作成した。

例：文「The cat sat on the mat」の語根変更。（図 3.2）

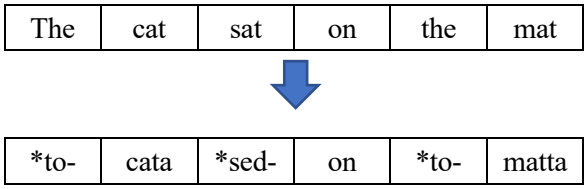


図 3.2：テキストの語根変更例

文字レベルは、string タイプは char タイプのリストと見なすため、直接にループを使って文字リストが作れる。それにより、文字レベルコーパスが作成する。

3.2.2 言語からベクトルへ

Word2Vec を用いて言語をベクトルに変更する。複数の機械学習モデルの実験において TF-IDF を使用し、特色の違いコーパス実において 1-hot ベクトルを使用する、テキストクラスタリングにおいて Skip-gram を使用する。

TF-IDF とは、文書の中に含まれる単語の頻度と重要度を評価する統計手法の一種である。主には情報検索やトテキスト分類などの分野で使用されている。tf-idf は、２つの指標に基づいて計算してくる。まず、tf は Term Frequency、単語の出現頻度である。idf は Inverse Document Frequency、逆文書頻度を指す。そして、この二つの指標をかけて、tf-idf 指標になる。tf-idf は次の式のように定義する：

$$\mathrm{tfidf_{i,j}} = \mathrm{tf_{i,j}} \cdot \mathrm{idf_i}$$
$$\mathrm{tf_{i,j}} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$
$$\mathrm{idf_i} = \log \frac{|D|}{|\{d : d \ni t_i\}|}$$

1-hot ベクトルは自然言語処理において、文書内の単語がベクトルの項目とされ、テキスト内の単語を 1 にする。例は前文の図 1.1 のようにする。

Skip-gram は単語分散表現の手法の 1 つである。文章内の単語を前後の若干の語とペアを組み、ペアの情報を統計し、重みを与える。

3.2.3 データ拡張

単純ベイズ、ロジスティクス回帰、K 近傍法、サポートベクトルマシンの実験において、コーパス規模をさらに拡大するため、K-fold 交差検証を使用する。

交差検証（cross-validation）とは、統計学において検証手法の 1 つである。まず、サンプルデータを分割する。次に、一部分のデータをテストデータとし、残る部分をトレーニングデータとして訓練する。そして、トレーニングデータとした部分を一部分を出してテストデータとし、テストデータとした部分をトレーニングに入れて訓練する。こうやって繰り返して、すべての分割された部分が 1 回ずつテストデータとして訓練したまでは交差検定である。機械学習では、データ拡張の手法と使われ、モデルの妥当性の検証・評価手法の 1 つである。

3.3 機械学習モデル

本節では、実験に使用するモデルを説明する。複数の機械学習モデルの実験、複数のコーパスの実験、テキストクラスタリング 3 つの部分がある。

まず、第一部分は、5 つの機械学習モデルを用いてテキスト分類を実験する。5 つのモデルは単純ベイズ、ロジスティクス回帰、サポートベクトルマシン、K 近傍法、畳み込みニューラルネットワークである。次に、第二部分は 7 つコーパスを使用し、畳み込みニューラルネットワークを用いたテキスト分類を実験する。最後に、第三部分では、語根レベルと単語レベルの分散表現を用いた文章のクラスタリング実験を行う。

3.3.1 では一般的な学習モデルを紹介する。3.2.2 では本研究で構築する深層学習モデルの設置情報を述べる。

3.3.1 モデル紹介

以下は機械学習モデルを紹介する。

単純ベイズ

単純ベイズ分類器は古典的な分類器であるが、現在でも使われており、適切な使い方をすると高い性能を発揮することも多い。単純ベイズ分類器は確率に基づいた分類器であり、事例にF対してP(C|F)が最大となるクラスを出力する[15]。

まず、ベイズの定理を用いて、複数のクラスがある場合は次の式のように定義する：

$$p(C|F_1,\dots,F_n) = \frac{p(C)\,p(F_1,\dots,F_n|C)}{p(F_1,\dots,F_n)}$$

そして、確率が最大のクラスを出力する分類器は次の式のように定義する：

$$\operatorname{classify}(f_1,\dots,f_n) = \operatorname*{argmax}_c\ p(C=c)\prod_{i=1}^{n} p(F_i=f_i|C=c)$$

ロジスティクス回帰

ロジスティック回帰は、ロジットを連結関数として使用する一般化線形モデルであり、ベルヌーイ分布に従う変数の統計的回帰モデルの一種である。

下の式のように定義する：

$$\operatorname{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \alpha + \beta_1 x_{1,i} + \cdots + \beta_k x_{k,i},$$
$$i = 1,\dots,n,$$

サポートベクトルマシン

サポートベクターマシン（SVM）は、1990 年代の終わり頃から自然言語処理において使われ初めてた線形二値分類器であり、高い分類性能を発揮する。訓練データから、各サンプル点の距離が最大化した境界線（超平面）を求める。そして、入力サンプルが超平面のとっちにあるかによって分類する。SVM はカーネル法と組み合わせて用いれば、非線形な分類も可能である。[15]

定義式は次のように：

$$E(w) = \|w\|^2 + C\sum_{i=1}^{n}\Delta(y_i,\hat{y}(x_i;w))$$

K 近傍法

K 近傍法は、機械学習のうちで、最も直観的なアルゴリズムである。あるサンプルの分類は、その近傍の k 個サンプル群のクラスににによって決定される。近傍のサンプルはどれが近いかを判断するために、サンプルのベクトルのユーク

リッド距離を計算する。そして、その k 個近傍のサンプルのうちに、数が多いクラスはそのサンプルの分類と決める。

畳み込みニューラルネットワーク

畳み込みニューラルネットワーク(Convolutional Neural Networks, CNN)は、畳み込み計算を含む奥行き構造を持つ順伝播型ニューラルネットワークニューラルネットワークであり、ディープラーニングの代表的なアルゴリズムの 1 つである。

畳み込みニューラルネットワークの研究は 1980 年代に始まり、LeNet-5 は最初に出現した畳み込みニューラルネットワークである。現在まで、ディープラーニング理論の提案と数値計算設備の改良に伴い、畳み込みニューラルネットワークは急速に発展し、コンピュータビジョンと自然言語処理などの領域に応用されている。

3.3.2 深層学習モデルの設置情報

第一部分と第二部分は同じ構造の畳み込みニューラルネットワークを使用する。

対照となるコーパスの原文や文字文、あるいは語根文をベクトルに変更し、学習データとして畳み込みニューラルネットワークにフェイドする。

構築する畳み込みニューラルネットワークは図 3.3 の過程を用いて構築される。

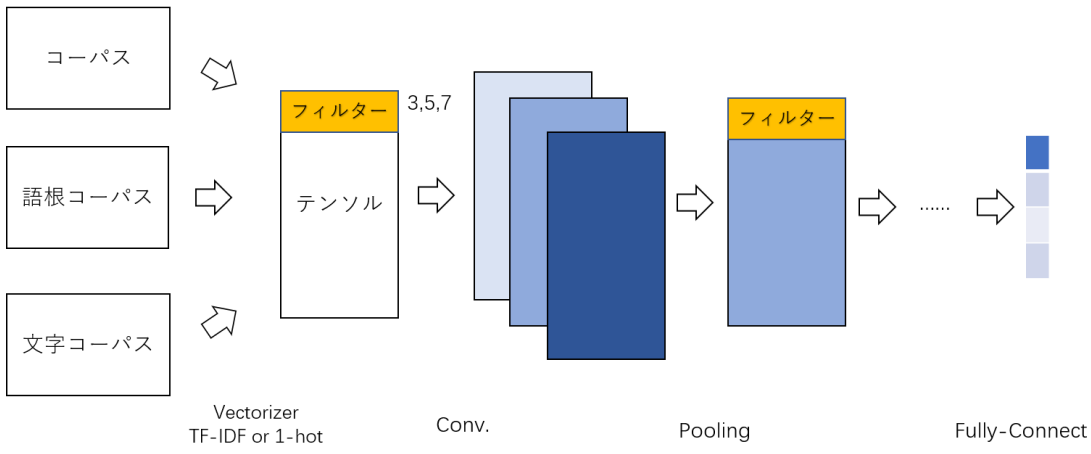


図 3.3：畳み込みニューラルネットワークの構築手順

学習モデルは 3 つの畳み込み層と 1 つの全連結層を使用する。

畳み込み層には 3 つサイズが違うフィルターがある。3 つのフィルターは長さが 3、5、7 であり、幅が入力するベクトルと同じことである。フィルターは正規

分布のランダムテンソルを使用し、標準偏差が 0.02 であり、バイアスが 0.01 である。ストライドは 1。
プーリングは最大プーリングを使用する。
全連結層は長さが文の種類数（本節では 4)、幅が 3 である。
バッチサイズは 128 である。エポックは 100 である。
ロス関数は交差エントロピーである
分類器は softmax 関数である。

第二部分は 7 つコーパスを使用し、畳み込みニューラルネットワークを用いたテキスト分類を実験する。モデルは図 4.4 の畳み込みニューラルネットワークモデルと同じ構造である。エンコーディングは 1-hot を使用し、入力サンプルは句にして特定のサイズを超える部分は無視する。コーパスにより文の長さが違うため、入力サイズは変わる。パラメータは表 3.1 示すように設置される。

モデル

	input	char	etymon-s	etymon	word	epoch
Reuter	sent	max	16	max	max	100
BBC	doc	1024	64	128	64	200
China Daily	doc	1024	64	128	64	500
QA	sent	max	16	max	max	100
IMBD	sent	max	16	max	max	100
Amazon	sent	max	16	max	max	100
Yelp	sent	max	16	max	max	100

表 3.1: コーパスに対してモデルの設置情報

表 3.1 に示す input は、文や文章を 1 つ入力サンプルとすることを表す。そして、char, etymon-s, etymon, word は前文に述べた入力サイズである。epoch はエポック回数である。

最後に、第三部分では、語根レベルと単語レベルの分散表現を用いた文章のクラスタリング実験を行う。
分散表現モデルのパラメータについて、バッチサイズ 128、埋め込みサイズ 128、窓サイズ 4。
テキストクラスタリングモデルは K-means を使用し、K が 5 である。
また、追加として、意味類似の計算を考察するため、単語クラスタリングの実験を行う。ウィキペディアをコーパスとし、分散表現を学習し、K-means で単語と語根をクラスタリングする。sklearn ライブラリの t-SNE を使用して結果を可視化して考察する。

第4章 評価実験

4.1 実験データ

4.1.1 語根辞書

実験では、ウェブクローラを用いてオンライン・エティモロジー・ディクショナリーから 44641 の単語の語根情報を収集し、整理してテキストに保存される。使用する際に、テキストから読み取り、Python の辞書型データでロードし、コーパス変更の照合に利用する。

4.1.2 コーパス

実験では、ロイター（Reuters)、英国放送協会ニュース（BBC News)、中国日報（China Daily News)、問題解答、インターネット・ムービー・データベース（IMBD)、アマゾンレビュー(Amazon reviews)、イェルプレビュー(Yelp reviews)、ウィキペディア（Wikipedia）8 つデータセットが使用される。

次は、コーパスと特色を紹介する。

ロイターは、世界で最も早い通信社の 1 つであり、さまざまな種類のニュースおよび財務データを短い文で提供している。(短い、早い、正しい、経済に関する)。実験では、分野が近い短文のコーパスとして利用される。

英国放送協会ニュース (BBC) は 1922 年に設立された英国最大のニュース放送局であり、英国政府の資金提供による公共メディアである。英国の編集者は同じ単語を違う言い方で表現するを通じて語彙力を見せびらかすため、実験では、単語（多い）と語根（共通だから、少ない）の対照となる。

中国日報は中国の日刊紙でり、中国が世界を理解し、世界が中国を理解させるための重要な窓口である。これは、国際に参入し、外国メディアの再版率が最も高い唯一の中国の新聞であり、最もカタログの多いメディアである。実験では、分野が広く、カタログが多い長文ニュースのコーパスとそて使用される。

問題解答は６つ種類の話題についての問題と答えのコーパスである。実験では対話文として使用される。

インターネット・ムービー・データベースは俳優、映画、テレビ番組、テレビ・スターおよびビデオゲームに関する情報のオンラインデータベースのことである。本実験は、感情によりポジティブとネガティブな映画レビューを収録され、感情分析の二分類問題コーパスとして利用される。

アマゾンレビューは、米国最大の通販サイトから、本、映画、音楽、ゲーム、電子機器、生活用品など、数百万の商品に対する評価が収集されている。実験では、日常な言葉やスラングのコーパス、単語種類が少ない対照として感情分析に使用される。

イェルプレビューは、米国の有名なビジネスレビューウェイブサイトです。2004 年に設立され、レストラン、ショッピングセンター、ホテル、およびさまざまな地域の観光の商人が含まれている。レビューは１から１０までの評価点があるが、実験では、１と１０だけを利用する。

ウィキペディアは、オンライン百科事典プロジェクトです。グローバルネットワーク上で最大かつ最も人気のあるリファレンスツールである。実験ではクローラーを用いてランダムに文を収集する。その文はラベルなしであり、分散表現の訓練に使用される。

各モデルにおいてのコーパス情報は表 4.1 から表 4.5 に示す。

まずは実験第一部分同じコーパスを使用して各機械学習モデルの実験データを説明する。ロイターから 4 つ最も文が多いカテゴリーを選ぶ。各カテゴリーのサンプル数のように 500（総計 2000）が選んでいた。

category	Sample Size
acq	500
earn	500
money-fx	500
grain	500

表 4.1: 各カタログが使用された文章数

文字レベル、語根レベル、単語レベルで処理したコーパスの素性数（また特徴数）は表 4.2 に記載されている。

	Char	Etymon	Word
Vocabulary Size	70	3214	13344

表 4.2: 文字、語根、単語レベルの語彙数

次は第二部分特色が異なる多数のコーパスを利用し、畳み込みニューラルネットワークを用いたテキスト分類を実験する。データはロイター、英国放送協会、チャイナ・デイリー、IMDB、質疑応答文、Amazon、Yelp を使用する。各コーパ

スの情報は表 4.3 のとおりである。

	Class	Sample	Vocabulary	Ety. Vocab
Reuters	4	18250	3569	1358
BBC	5	4819	9413	2201
China Daily	10	1000	10349	2896
QA	6	500	3369	1732
IMBD	2	5000	9163	2001
Amazon	2	1000	2248	1172
Yelp	2	1000	2358	1375

表 4.3: コーパスの分類数、サンプル数、語彙量と語根量の情報

第三部分はラベルを見せずにテキストクラスタリングの実験を行う。実験データは BBC の 5 つカテゴリーのニュースを利用する。サイズは表 4.4 のように各カテゴリーから 100 ずつを選ぶ。

Category	Size
Business	100
Entertainment	100
Politics	100
Sport	100
Tech	100

表 4.4: BBC コーパス情報

最後に、意味論を探求するため、ウィキペディアから大規模なテキストを学習して、単語クラスタリングを行い、可視化する実験を試行する。使用されたウィキペディアの量、語彙、語根数は表 4.5 に記載されている

	Word Size	Vocabulary Size	Etymon Size
Wikipedia	18658642	238997	11248

表 4.5:ウィキペディアコーパス情報

4.2 評価基準

実験では、複数のテキスト分類モデル、違うコーパスを用いたテキスト分類、テキストクラスタリング３つの実験で評価する。

まず、第一部分は文字レベル、語根レベル、単語レベルが単純ベイズ、ロジスティクス回帰、K 近傍法、サポートベクトルマシン、畳み込みニューラルネット

ワーク５つの学習モデルでの正解率とF値を比較し、評価する。
正解率はテストデータを分類器に判断させ、正しく判断したサンプルと全部テスト用サンプルの割合である。これにより、文章分類の性能を評価する。正解率の定義を式にする。

$$\text{正解率} = \frac{\text{分類器が正しく判断できたサンプル数}}{\text{テストデータとして入力したサンプル総数}}$$

F値とは、二分類の統計分析でのテストの精度の尺度であり、精度と再現率を統合した指標である。精度は、分類器が未知のサンプルをクラスにどれぐらい正しく判断できるかを表す量である。召回率はクラスであるものを分類器がどれぐらいカバーできるかを表す量である。F値は1（完全な精度と再現率）で最高値に達し、0で最悪値に達する。本実験では、多分類予測実験を行うため、マイクロ平均を使用する。
F値は以下のように定義する。

まず、表4.6のように予測結果と真の結果の照合をTP、FP、FN、TNと表示する。

	真の結果		
予測結果		正	負
	正	TP	FP
	負	FN	TN

表 4.6: 混同行列

正解率は正や負と予測したデータのうち，実際にそうであるものの割合である。式は次のように定義する。

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

精度は正と予測したデータのうち，実際に正であるものの割合である。式は次のように定義する。

$$\text{Precision} = \frac{TP}{TP + FP}$$

再現率は実際に正であるもののうち，正であると予測されたものの割合である。式は次のように定義する。

$$\text{Recall} = \frac{TP}{TP + FN}$$

F 値は再現率と適合率の調和平均である。式は次のように定義する。

$$\text{F1 Score} = \frac{2\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}$$

また、本実験では、マイクロ平均を使用する。式は次のように定義する。

$$\text{F1-Micro} = \frac{\sum_{i=1}^{N} x_i}{\sum_{i=1}^{N} n_i}$$

次に、第二部分は畳み込みニューラルネットワークを用い、７つ特性が違うコーパスで学習する。評価は、正解率の上、エポックごとの損失率の変化を比較し、文字レベル、語根レベル、単語レベルの学習効率を評価する。
損失関数は交差エントロピーを用い、式は下のように定義する。

$$\mathrm{H}(p, q) = -\sum_{x} p(x) \log q(x)$$

変化は離散な損失率値を回帰関数を求め、その微分式を評価とする。

最後に、第三部分は語根レベルと単語レベルの分散表現を学習し、K-means 用いたテキストクラスタリング表現を評価とする。

第5章 実験結果

5.1 複数のモデルにおける性能の評価

5.1.1 結果

まず、第一部分、文字レベル、語根レベル、単語レベルを用いた単純ベイズ、ロジスティクス回帰、サポートベクトルマシン、K 近傍法、畳み込みニューラルネットワーク 5 つの機械学習モデルの精度結果を表 5.1 に示す。結果単位はパーセントである。

Accuracy	Char	Etymon	Word
MNB	72.2	98.2	99.4
SVM	58.1	80.9	84.7
LR	84.8	99.4	99.6
KNN	99.1	99.7	99.8
CNN	86.6	97.8	99.9

表 5.1: 正解率結果

各基礎手法の純ベイズ、ロジスティクス回帰の F 値結果を表 5.2 に示す。

F1 score	Char	Etymon	Word
MNB	0.528	0.981	0.958
LR	0.679	0.998	0.976

表 5.2: F 値結果

表 5.1 に示す単純ベイズの結果では、精度は文字レベルが 72.192%、語根レベルが 98.206%、単語レベルが 99.400%である。語根レベルは文字レベルより 26.014%高く、単語レベルより 1.149%低い。

ロジスティクス回帰では、精度は文字レベルが 84.792%、語根レベルが 99.492%、単語レベルが 99.586%である。語根レベルは文字レベルより 14.7%高く、単語レベルより 0.094%低い。

サポートベクトルマシンでは、精度は文字レベルが 58.108%、語根レベルが 80.850%、単語レベルが 84.658%である。語根レベルは文字レベルより 22.742%高く、単語レベルより 4.078%低い。

K 近傍法では、精度は文字レベルが 99.109%、語根レベルが 99.730%、単語レベルが 99.802%である。語根レベルは文字レベルより 0.621%高く、単語レベルよ

り 0.072%低い。
畳み込みニューラルネットワークでは、精度は文字レベルが 86.550%、語根レベルが 97.795%、単語レベルが 97.976%である。語根レベルは文字レベルより 11.245%高く、単語レベルより 0.181%低い。
まとめ語根レベルは単純ベイズ、ロジスティクス回帰、サポートベクトルマシン、畳み込みニューラルネットワークにおいて単語レベルより精度が近いことが分かる。

F 値の結果について、表 5.2 に示す単純ベイズの結果では、F 値は文字レベルが約 0.523、語根レベルが約 0.982、単語レベルが約 0.985 である。
ロジスティクス回帰では、精度は文字レベルが約 0.679、語根レベルが約 0.998、単語レベルが約 0.977 である。。
単純ベイズとロジスティクス回帰において語根レベルは単語レベルより優れていることが分かれる。

また、本研究では、素性数の削減により計算量の減少を議論するため、理論的な計算と実際のプログラム実行時間を説明する。
計算環境と設備の情報は Intel Xeon E5-4622 v3 48cpus nodes 384cpus、メモリ 8 TB である。
実行時間を表 5.3 に示す。

単位は時:分:秒である。

Time	MNB	SVM	LR	KNN
Char	0:11:15	45:53:58	0:30:43	27:40:21
Etymon	0:06:22	84:16:26	0:18:46	21:35:32
Word	0:10:54	167:43:03	0:23:09	17:54:42

表 5.3: 機械学習モデルの訓練時間

結果は、ベイズとロジスティクス回帰において、語根は約 6 分、18 分で最短であり、サポートベクトルマシンと K 近傍法におて、約 84 時間、21 時間で文字レベルと単語レベルの中間である。
実験では並列計算を使うのが、サポートベクトルマシはアルゴリズム上並列計算ができない、ゆえに長い時間に訓練した。

5.1.2 考察

文字レベル、語根レベル、単語レベルを 5 つ機械学習モデルでテキスト分類実験を行った正解率結果を図 5.1 に棒グラフで示されている。

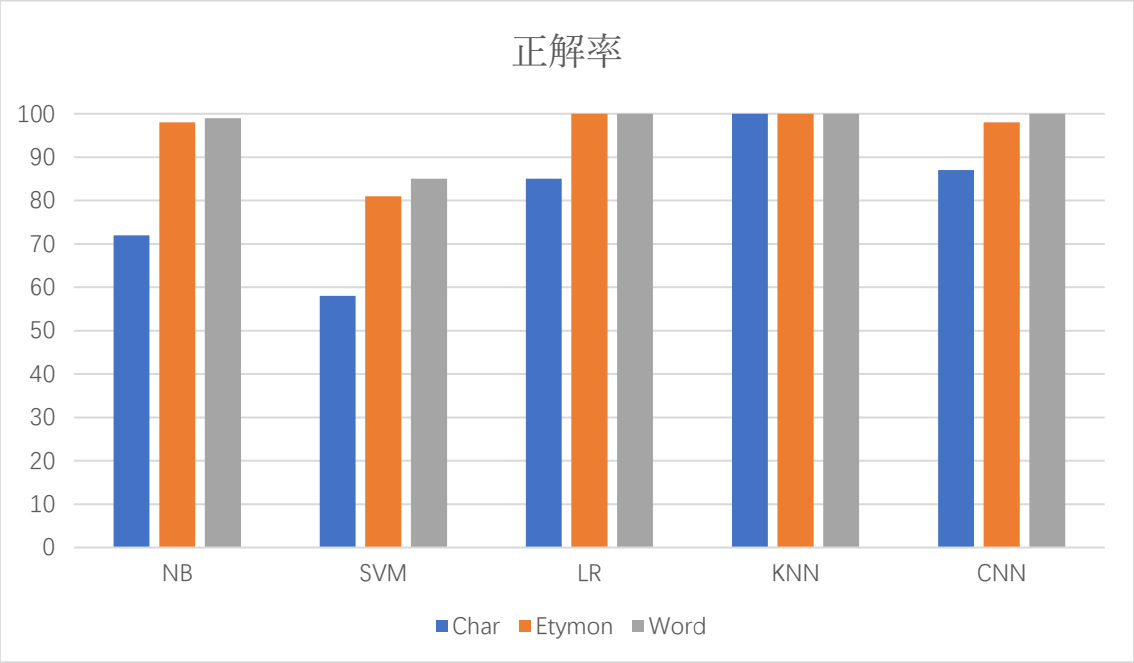


図 5.1：手法正解率比較の棒グラフ

図 5.1 に示す語根レベルの結果では単純ベイズ、サポートベクトルマシン、ロジスティクス回帰、畳み込みニューラルネットワークモデルにおいての精度が単語レベルに近く、文字レベルより優れている。K 近傍法では 3 つのモデルは近いことが明らかにされていた。

機械学習では、データの次元数が少なければ少ないほど分類の正解率が低いということがある。文字レベルは、特徴数を削除過ぎたため、ちゃんと分類することができなくなる。語根レベルは、語の意味を保留し、素性数を最小限まで残したため、大幅に削減しても重要な情報（統計的上、重みが大きい素性）を損失していなく、正解率が単語レベルより近いと考えられる。

実行時間の結果は予測通りに文字レベルと単語レベルの間にあるが。ただし、公式からの推論はあくまでも論理的なものなので、実際に利用した sklearn ライブラリのプログラムの実行時間の割合に合わない。

まずは単純ベイズの計算式：

$$\text{classify}(f_1, \ldots, f_n) = \underset{c}{\text{argmax}}\ p(C = c) \prod_{i=1}^{n} p(F_i = f_i | C = c)$$

公式から見ると、C のクラス数と n のサンプル数は同じとなり、違うのは特徴数、つまり、文字語根語彙の種類数のことである。もし、アルゴリズムは先にすべての特徴数とクラスの確率を統計して、単純ベイズの通りに確率を計算するとなれば、語源基礎と単語レベルの時間割合は 3274/13344、文字レベルと単語レベルの時間割合は 70/13344 のことである。

また、サポートベクトルマシン、パーセプトロンのような重みベクトルをサンプルベクトルとかけて更新し続くアルゴリズムは同じである。論理的上計算量の差は各基礎手法を用いたベクトル長さの差の回数の掛け算。割合は同じ3274/13344 と 70/13344 である。

ニューラルネットワークはブラックボックスであるが、畳み込みニューラルネットワークは枠が決まっているため、計算量が推定できる。Word2Vec の場合は前述と同じ 3274/13344 と 70/13344 の割合である。

単語分散表現と GloVe の場合は 3274/13344 と 70/13344 の二次乗になる。しかし、文を文字や語根に変更すると、語数が何倍増す。本研究では単語分散表現系を議論しない。

5.2 深層学習における性能の評価

5.2.1 結果

7 つ異なるコーパスを用いた畳み込みニューラルネットワークの表 5.4 に示す。

Accuracy	Char	Etymon-short	Etymon	Word
Reuters	81.7	99.7	99.7	98.8
BBC	36.3	34.4	39.2	38.6
China Daily	15.6	23.8	29.9	26.4
QA	67.1	81.0	72.4	81.2
IMBD	79.9	93.7	93.8	95.4
Amazon	82.7	97.7	98.0	98.1
Yelp	82.1	91.0	90.1	94.5

表 5.4: CNN での各コーパスの正解率結果

ロイター（Reuters）においては、語源基礎の精度は 99.657 と 99.665、単語レベルの 98.775 より精度が高く、文字レベル 81.684 より高い。

英国放送協会ニュース（BBC）においては、語源基礎の精度は 34.392 と 39.162、全長語根レベルは単語レベルの 38.627 より精度が高く、文字レベル 36.278 より高い。

中国日報（China Daily News)、においては、語源基礎の精度は 23.847 と 29.931、長語根レベルは単語レベルの 26.396 より精度が高く、文字レベル 15.603 より高い。

問題解答（QA)、においては、語源基礎の精度は 80.983 と 72.375、単語レベルの 81.237 より精度が低く、文字レベル 67.071 より高い。

インターネット・ムービー・データベース（IMBD)、においては、語源基礎の精度は 993.708 と 93.823、ショート語根レベルは単語レベルの 95.443 より精度が高く、文字レベル 79.891 より高い。

アマゾンレビュー（Amazon reviews)、においては、語源基礎の精度は 97.708 と 97.995、単語レベルの 98.093 より精度が低く、文字レベル 82.719 より高い。

イェルプレビュー（Yelp reviews)、においては、語源基礎の精度は 91.026 と 90.099、単語レベルの 94.542 より精度が低く、文字レベル 82.115 より高い。

学習効率について、論文のスペース制限があるため、本論文ではロイターコーパスの学習結果データを代表とし、訓練において正解率と損失率の折り線図だけ示す。詳しい結果は、前 50 エポックの平均正解率と損失率の結果限り、付録に展示する。[5]

図 5.2 と 5.3 に示すのは正解率と損失率の折り線図。

[5] エポックとは、訓練データを何回繰り返して学習させるかの回数のことである。

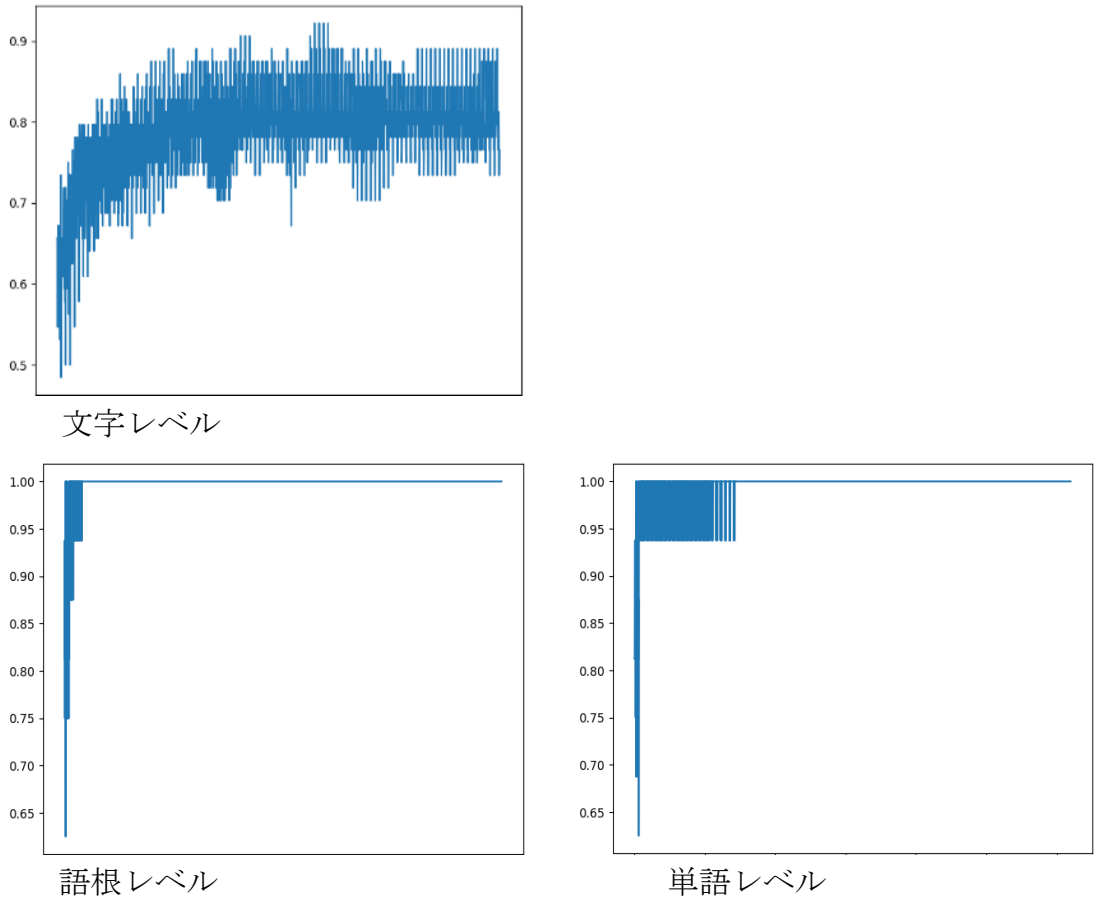


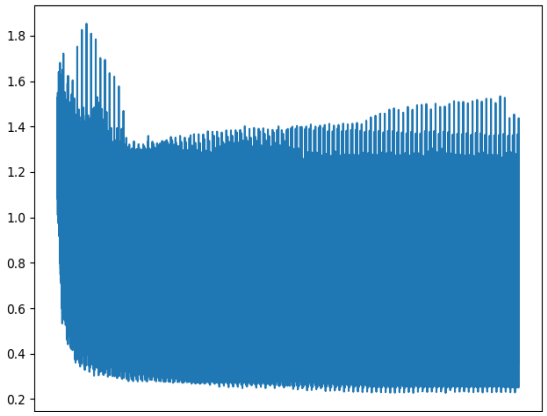
図 5.2：正解率マップ

図 5.2 に示図は、128 サンプル（1 バッチ）ごとの正解率結果の折り線である。

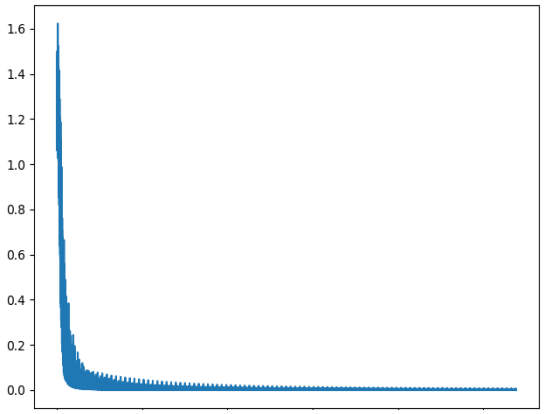
上に示す文字レベルの図は始まりから終わりまで、大きな揺れがあり、最高でも 0.91 の正解率である。

下左の語根レベルの図は最初から揺れがあり、そして穏やかになり、0.99 の高い正解率を維持する。

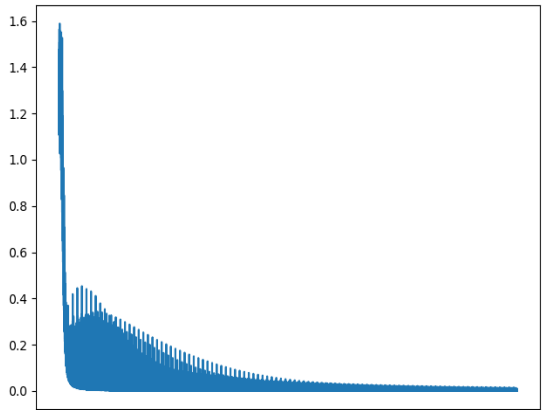
下右の単語レベルの図は、最初から高い正解率を持ち、訓練の繰り返しに伴い数値が揺れ、最後穏やかになる。



文字レベル



語根レベル



単語レベル

図 5.3：損失マップ

図 5.3 に示図は、128 サンプル（1 バッチ）ごとの損失率結果の折り線である。

上に示す文字レベルの図は最初の 1.83 から最後の 0.21 まで、大きな揺れがある。

下左の語根レベルの図は損最初の 1.61 から最後の 0.01 まで、約 1 万回のバッチ処理まで急速に降下し、低い損失率を維持する。

下右の単語レベルの図は、損最初の 1.60 から最後の 0.01 まで、約 6 万回のバッチ処理まで緩めに降下し、低い損失率を維持する。

5.2.2 考察

次に、特色が異なる多数のコーパスを利用し、畳み込みニューラルネットワークを用いたテキスト分類を実験する。一部分の結果は図 5.4 に棒グラフで示されている。

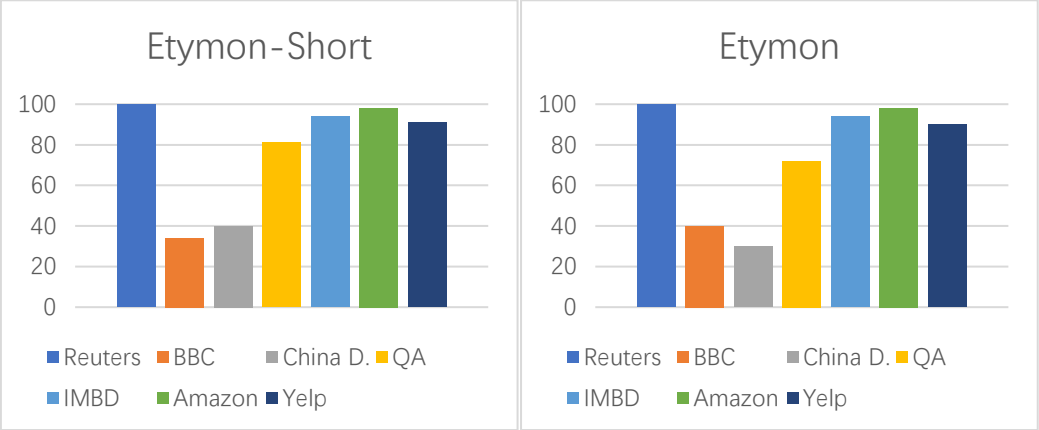


図 5.4：語根レベルのコーパス正解率比較棒グラフ

畳み込みニューラルネットワークにおいては、違うコーパスに対して語根レベルは全体的に単語レベルに近い。語彙が多い場合は語根レベルは単語レベルより高い、少ない場合は低い。文章の長さと関係はない。文章の種類が多い場合は語根レベルは単語より高い。感情分析では特定に区別がない。

図 5.2 と 5.3 に示す図は、文字レベルモデルは最初から低い精度から始め、向上のスピードも遅い。それに対し、語根レベルと単語レベルは向上のスピードが早いであり、学習回数の増加に伴い、ロスが迅速に降下している。さらに、語根レベルは単語レベルの降下よりもっと早く、学習の効率が優れている。

5.3 分散表現を用いたテキストクラスタリング

5.3.1 結果

まず、分散表現の 10 万回バッチ処理訓練の損失率結果を表 5.5 に示す。

Embeddings	Etymon	Word
Loss	0.28	0.28

表 5.5: 分散表現の損失率結果

語根レベルは単語レベルと同じ損失率である。

次は、K-means クラスタリングの結果、サンプルを Python の sklearn ライブラリの t-SNE を用いて可視化された結果を図 5.5 語と図 5.6 に示す：

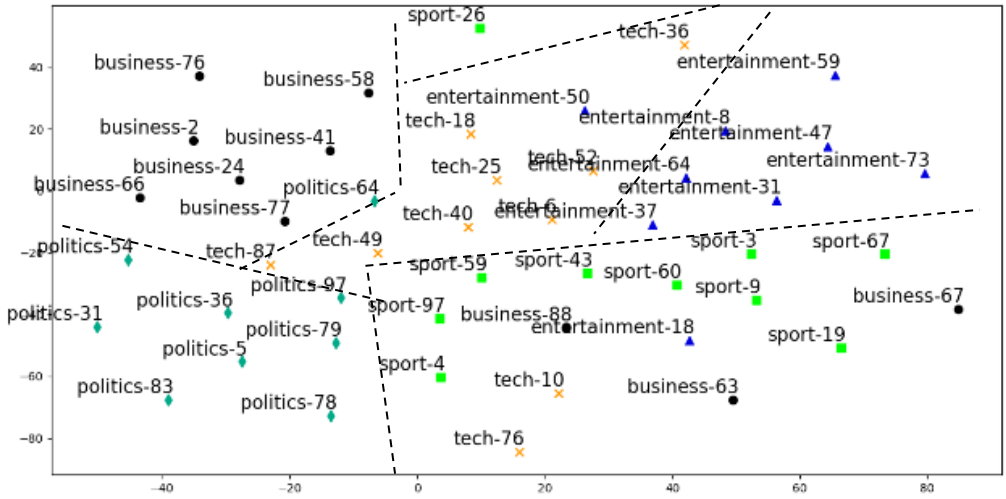


図 5.5：語根レベルのテキストクラスタリング結果

次は単語レベル図 5.6：

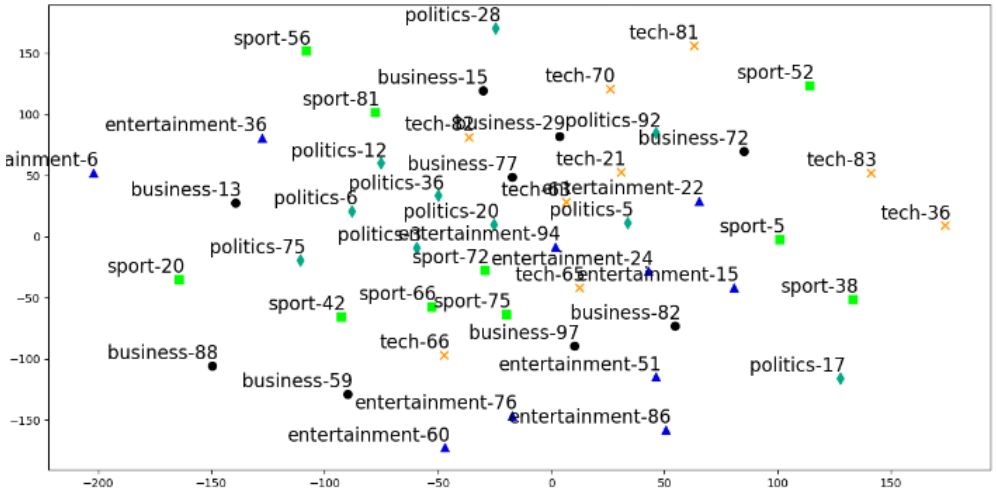


図 5.6：単語レベルのテキストクラスタリング結果

5.3.2 考察

最後に、ラベルを見せずにテキストクラスタリングの実験を行った。図 5.5 と

5.6 に示す群れ図は語根レベルが単語レベルより明白な境界線があり、混同サンプルが少ない。ただし、モデルはブラックボックスなので、原因を探求するのは難しいことである。

第6章 おわりに

6.1 まとめ

本研究では、語源学を活用して語根を素性とする手法を提案する。文字レベル、語根レベル、単語レベルの機械学習を用いたテキスト分類の実験を行い、語根レベルの効果を評価した。

本研究では、語根レベルの効果を評価するため、テキスト分類に巡り、3 つの角度から実験をする。1 つ目は、複数の機械学習モデルを用い、正解率と学習時間を考察して性能を評価する。2 つ目は、多様なコーパスを用い、畳み込みニューラルネットワークのモデルで学習し、正解率と損失率の変化を考察し、学習の効率を評価する。学習過程におけるデータを全部学習するごと(epoch)で正解率と損失率を記録する。3 つ目は、分散表現を用いてテキストクラスタリングの実験を行い、結果を考察する。

実験では、まず、単語の語根情報を収集し、語根辞書を作成する。次に、大規模なコーパス（単語レベル）を収集し、語根レベルと文字レベルのコーパスを作成する。そして、文字レベル、語根レベル、単語レベルのコーパスをベクトルに変更し、学習モデルにフェイドする。最後に、結果を考察する。

（1）NB、SVM、LR、CNN において語根レベルは文字レベルを 8.9%超え、単語レベルに-0.1%～+0.9%近い正解率を持っておる。

（2）深層学習において、語根レベルは正解率が単語レベルに近い上、さらに訓練の繰り返しにおいて損失の降下が早く、学習スピードが速いである。

（3）分散表現の学習においては、語根レベルと単語レベルが同じ損失率を持っている。テキストクラスタリングでは、語根レベルが優れている。

（4）次元が削減されているため、一部分のモデルでは単語レベルよりも学習時間が短いである。

考察としては、語根レベルは語根を用いたため、優れた結果を得た。語根は単語の表す意味を機能しているため、語根レベルを用いた機械学習は語形変化の影響を受けず、重要な特徴を保ち、オーバーフィットを抑える。それで、語根レベルは高い精度と効率を持つと考える。

語根レベルはテキスト分類に適任し、従来の単語レベルと文字レベルに競争力のある手法であり、自然言語処理における次元の呪い問題を改良できる手法と結論している。

6.2 今後

実験の前に、語根レベルの正解率は文字レベルと単語レベルの間にあると予測したが、結果は単語レベルに近く、高い正解率である。語根レベルは自然言語理解においては適任なモデルであるが、言語生成に失敗した。今後は統計的な意味の上で検討し、手法を改良すると考える。さらに、多言語の形態素を素性とし

て機械学習モデルを構築し、意味に基づいて統計的な機械翻訳モデルを実験しようと考える。

参考文献

[1] Laura Aina, Kristina Gulordava, Gemma Boleda. Putting Words in Context: LSTM Language Models and Lexical Ambiguity. ACL2019. Pages 3342-3348. 2019

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv. 2019

[3] Alon Jacovi, Oren Sar Shalom, Yoav Goldberg. Understanding Convolutional Neural Networks for Text Classification. ACL2018. Pages 56-65. 2018

[4] Dimitri Palaz, Mathew Magimai. Doss, Ronan Collobert. Convolutional Neural Networks-based continuous speech recognition using raw speech signal. IEEE ICASSP. Pages 4295-4299. 2015

[5] Xiang Zhang, Junbo Cui, Yann LeCun. Character-Level Convolutional Neural Network for Text Classification. NIPS. 2015

[6] Joonatas Wehrmann, Willian Becker, Henry E. L. Cagnini, Rodrigo C. Barros. A character-based convolutional neural network for language-agnostic Twitter sentiment analysis. IEEE IJCNN. Pages 2384-2391. 2017

[7] Vivi Nastase and Carlo. Bridging Languages through Etymology: The case of cross language text categorization. ACL2013. Pages 653-65. 2013

[8] Vivi Nastase Carlo Strapparava. Word Etymologic as Nature Language Interface. ACL2016. pages 2702-2707. 2016

[9] Harold Borko, Myrna Bernick. Automatic Document Classification. System Development Corporation, Santa Monica, CA. 1962

[10] Alexis Conneau, Ruty Rinott, Guillaume Lample. HXNLI: Evaluating Cross-lingual Sentence R epresentations. ACL2018 Pages 2475-2485. 2018

[11] Ximing LI , Bo Yang. A Pseudo Label based Dataless Naive Bayes Algorithm for Text Classification with Seed Words. ACL2019 Pages 1908-1917. 2019

[12] April Dae C. Bation, Erlyn Q. Manguilimotan, Aileen Joan O. Vicente. Automatic Categorization of Tagalog Documents Using Support Vector Machines. ACL2018 Pages 346-353. 2018

[13] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S.,and Dean, J. Distributed representations of words and phrases and their

compositionality. NIPS, pages 3111-3119. 2013

[14] Matt J. Kusner, Yu Sun YUSUN, Nicholas I. Kolkin N.KOLKIN, Kilian Q. Weinberger. From Word Embeddings To Document Distances. JMLR .2014

[15] 奥村 学, 高村 大也, 言語処理のための機械学習入門, コロナ社 (2010)

付録

ロイターコーパスを用いた CNN 正解率結果（前 50 エポック）

Accuracy(%)	Char	Etymon	Word
1	55.208	38.438	44.357
2	58.229	46.615	54.892
3	60.521	55.035	59.792
4	62.448	61.120	66.875
5	63.979	65.708	72.125
6	65.208	69.271	76.267
7	66.131	72.351	79.479
8	67.847	74.818	82.005
9	68.625	76.921	84.005
10	69.460	78667	85.604
11	70.226	80.180	86.913
12	70.881	81.476	88.003
13	71.369	82.604	88.926
14	71.688	83.571	89.717
15	71.966	84.431	90.403
16	72.200	85.202	91.003
17	72.454	85.895	91.532
18	72.681	86.516	92.002
19	72.844	87.072	82.423
20	73.036	87.589	92.802
21	73.248	88.061	93.145
22	73.689	88.490	93.456
23	73.465	88.890	93.741
24	73.913	89.258	94.002
25	74.131	89.592	94.242
26	74.348	89.904	94.463
27	74.520	90.193	94.668
28	74.662	90.465	94.859
29	74.799	90.718	95.036
30	74.940	90.955	95.201
31	75.094	91.176	95.356
32	75.253	91.383	95.501
33	75.417	91.578	95.638

34	75.539	91.756	95.766
35	75.660	91.923	95.887
36	75.738	92.080	96.001
37	75.822	92.230	96.109
38	75.919	92.371	96.212
39	76.042	92.505	96.309
40	76.159	92.633	96.401
41	76.290	92.759	96.489
42	76.368	92.872	96.572
43	76.407	92.980	96.652
44	76.518	93.087	96.728
45	76.618	93.192	96.801
46	76.712	93.293	96.870
47	76.806	93.389	96.937
48	76.899	93.481	97.001
49	77.981	93.569	97.062
50	77.060	93.654	97.121

ロイターコーパスを用いたCNN損失率結果（前50エポック）

Loss	Char	Etymon	Word
1	1.25	1.28	1.27
2	1.18	1.24	1.19
3	1.11	1.16	1.08
4	1.06	0.07	0.98
5	1.02	0.99	0.88
6	0.98	0.92	0.78
7	0.95	0.86	0.70
8	0.92	0.80	0.63
9	0.90	0.75	0.57
10	0.88	0.71	0.52
11	0.86	0.67	0.48
12	0.85	0.63	0.45
13	0.83	0.60	0.41
14	0.82	0.57	0.39
15	0.81	0.54	0.36
16	0.79	0.52	0.34
17	0.78	0.49	0.32
18	0.78	0.47	0.31
19	0.77	0.46	0.29
20	0.76	0.44	0.28
21	0.75	0.42	0.26
22	0.74	0.41	0.25
23	0.74	0.40	0.24
24	0.73	0.38	0.23
25	0.72	0.37	0.22
26	0.72	0.36	0.22
27	0.71	0.35	0.21
28	0.71	0.34	0.20
29	0.70	0.33	0.19
30	0.70	0.32	0.19
31	0.70	0.32	0.18
32	0.70	0.31	0.18
33	0.69	0.30	0.17
34	0.69	0.30	0.17
35	0.69	0.29	0.16
36	0.69	0.28	0.16
37	0.68	0.28	0.15

38	0.68	0.27	0.15
39	0.68	0.27	0.15
40	0.67	0.26	0.14
41	0.67	0.26	0.14
42	0.67	0.25	0.14
43	0.67	0.25	0.13
44	0.67	0.24	0.13
45	0.67	0.24	0.13
46	0.66	0.24	0.12
47	0.66	0.23	0.12
48	0.66	0.23	0.12
49	0.66	0.23	0.12
50	0.66	0.22	0.11