Fine-tuning for nwjc2vec

nwjc2vec の fine-tuning

新納浩幸（茨城大学工学部情報工学科）*
古宮嘉那子（茨城大学工学部情報工学科）†
佐々木稔（茨城大学工学部情報工学科）‡

Fine-tuning for nwjc2vec

Hiroyuki Shinnou (Ibaraki University)
Kanako Komiya (Ibaraki University)
Minoru Sasaki (Ibaraki University)

要旨

国語研日本語ウェブコーパス (NWJC) を基に作成された分散表現データが nwjc2vec と名付けられて公開されている．NWJC は超大規模コーパスであるため，そこから構築された nwjc2vec の品質はかなり高いと考えられる．ただし分散表現データを実際の自然言語処理システムに利用する際には，そのシステムが対象とする領域に依存した分散表現データが望ましい．これは，一種の領域適応の問題である．ここでは処理対象を新聞記事として，新聞記事 7 年分から構築した分散表現データ mai2vec と nwjc2vec を比較することでこの点を確認する．またこの問題の対処として nwjc2vec に対して少量の新聞記事を利用して fine-tuning を行い，その効果を確認する．

1. はじめに

本論文では分散表現データ nwjc2vec においても領域適応の問題が生じていること，そしてこの問題に対して fine-tuning を行うことを提案する．

分散表現とは単語の意味を密な低次元のベクトルで表現したものである．単語の意味的な類似性を反映したベクトルとなっているために，近年，自然言語処理の多くのシステムで利用され，その有用性は明らかである (岡崎 (2016))．ただし分散表現にはいくつかの点で課題もある．その一つが分散表現を構築する際に利用したコーパスと，システムが処理対象とする領域とが異なる問題，いわゆる領域適応の問題への対処である．例えば分散表現を構築するために利用したコーパスが新聞記事であり，システムが処理対象とする領域がブログなどの場合に，システムの性能が大きく劣化してしまう現象が領域適応の問題である．

一方 nwjc2vec は国語研日本語ウェブコーパス（以下 NWJC）(Asahara et al. (2014)) から構築された分散表現データである (浅原・岡 (2017))．NWJC は超大規模コーパスであるため，そこから構築された nwjc2vec は非常に高品質であると考えられる．実際，いくつかの報告でこの点が確認されている (山木ほか (2017))(新納ほか (2017))．また NWJC は様々な

* hiroyuki.shinnou.0828@vc.ibaraki.ac.jp
† kanako.komiya.nlp@vc.ibaraki.ac.jp
‡ minoru.sasaki.01@vc.ibaraki.ac.jp

コーパスを含んでいるために，広い範囲の領域で利用できると考えられる．つまり前述した領域適応の問題が nwjc2vec では生じないのでは，という疑問がある．

ここでは nwjc2vec のような高品質な分散表現であっても領域適応の問題が生じていることを示す．そのために毎日新聞の記事 7 年分から構築した分散表現データ mai2vec との品質の比較実験を行う．具体的には LSTM を用いた言語モデルの構築を行い，そのパープレキシティを測ることで分散表現データを評価する．LSTM の訓練データと言語モデルの評価データがブログである場合は nwjc2vec の方が品質は高い．しかしそれらが mai2vec と同領域の新聞記事である場合には mai2vec の方が品質が高くなる．これは領域適応の問題が生じていることを意味する．

また nwjc2vec を少量の新聞記事によって fine-tuning を行い，tuning された nwjc2vec を nwjc2vec-ft と名付ける．この場合，LSTM の訓練データと言語モデルの評価データが新聞記事であっても nwjc2vec-ft は mai2vec よりも品質が高かった．これにより分散表現データの領域適応の問題には，fine-tuning を行うことが有効であると考えられる．

2. nwjc2vec

ここでは nwjc2vec の概要を述べる．

nwjc2vec の構築の基になったコーパスは NWJC である．NWJC はウェブを母集団とし収集された文書からなり，全体として約 258 億語からなるコーパスである．1 年分の新聞記事中のプレーンな文のデータが約 2,050 万語[(1)]であることを考えると，NWJC は 1,200 年分以上の新聞記事に相当し，超大規模コーパスといえる．

nwjc2vec は NWJC を unidic を基に形態素解析し，それを word2vec[(2)] を用いて構築されたものである．構築時に使われた word2vec のパラメータは以下の通りである (浅原・岡 (2017))．

表 1　word2vec の実行時のパラメータ

CBOW or skip-gram	-cbow	1
次元数	-size	200
文脈長	-window	8
負サンプリング数	-negative	25
階層化 softmax	-hs	0
最低頻度閾値	-sample	1e-4
反復回数	-iter	15

nwjc2vec は柔軟な利用が可能なように，分散表現をテキストファイルの形式で保存している．1 行は 1 トークンに相当し，以下の形式になっている．

(1) 2008 年度の毎日新聞記事から，文としてなりたつと考えられるものを抽出し，unidic を基に形態素解析したものから算出した．

(2) `https://github.com/svn2github/word2vec`

トークン e_1 e_2 ・・・ e_200

e_i がそのトークンの分散表現の i 次元目の値である．例えば，以下は「意味」に対応する分散表現である．

意味, 名詞, 普通名詞, サ変可能,*,*,*, イミ, 意味, 意味, イミ, 意味, イミ, 漢,*,*,*,
-10.491043 -2.121982 -3.084628 ··· 4.024705 3.570072 12.781445

つまり “意味, 名詞, 普通名詞, サ変可能,*,*,*, イミ, 意味, 意味, イミ, 意味, イミ, 漢,*,*,*,” が 1 トークンである．またベクトル値は word2vec の出力値をそのまま書き出しており，大きさ[(3)]を 1 とする正規化はされていない．

nwjc2vec 全体としては 1,738,455 トークンからなる[(4)]．書字形出現形は 1,541,651 種類存在するので，書字形出現形が同じでも形態論情報が異なるものが多数存在する．従来の単語分散表現は書字形出現形をトークンとしたものが一般的であり，その場合，品詞の違いによる別単語を同一の分散表現にしているという明らかな欠点がある．nwjc2vec ではその欠点を回避できている．

3. nwjc2vec と mai2vec の品質比較

nwjc2vec は超大規模コーパスである NWJC から構築されているために，ある特定のコーパスから構築された分散表現データよりも品質は高い．

ここでは RNN の拡張版である Long Short-Term Memory(以下 LSTM)(Gers et al. (2000)) を用いて言語モデルを構築し，その言語モデルを使って評価用コーパスのパープレキシティを測ることで分散表現の品質を評価する．具体的には LSTM を学習する際に単語の分散表現も一緒に学習されるが，この部分を既存の分散表現データに固定して学習を行う．固定する分散表現だけを変えることで最終的に得られた言語モデルの優劣が分散表現の優劣と考えることができる．

nwjc2vec との比較のために，新聞記事 7 年分から分散表現を構築する．用いたコーパスは毎日新聞’93 年度版から ’99 年度版の 7 年分の記事であり，そこから見出しや表内の文字列等を取り除き，文として認められるものだけを取り出した．取り出した文は 6,791,403 文であった．これを MeCab-0.996 と UniDic-2.1.2 を用いて分かち書きし，これを word2vec にかけることで分散表現を構築した．この分散表現データをここでは mai2vec と名付ける．word2vec 実行時の各種パラメータは nwjc2vec を構築したもの（表 1）と合わせた．最終的に得られた mai2vec のトークン数は 132,509 であった．

言語モデルの学習用のコーパスとしては現代日本語書き言葉均衡コーパス (Maekawa et al.

(3) 本論文ではベクトルの「大きさ」をベクトルの「L2-ノルム」の意味で用いている．
(4) そのテキストファイルは header の 1 行を含め 1,738,456 行である．

(2014)) の Yahoo! ブログと Yahoo! 知恵袋から取り出した 7,330 文のうち 7,226 文を学習用コーパス，104 文を評価用コーパスとした．LSTM の分散表現部分を nwjc2vec に固定して学習できた言語モデルを nwjc2vec-lm と名付け，mai2vec に固定して学習できた言語モデルを mai2vec-lm と名付けた．また参考として分散表現を LSTM 内で学習して構築した言語モデル base-lm も評価する．言語モデルの評価にはパープレキシティを用いる．LSTM の学習は 15 epoch まで行い，各 epoch 毎にモデルを保存し，パープレキシティを測った．そして最も低い値のパープレキシティを評価値とした(5)．

結果を表 2 に示す．nwjc2vec-lm が最もパープレキシティが低く，nwjc2vec が mai2vec と比較して高品質であることがわかる．

表 2　各分散表現データから得られた言語モデルの評価 1

base-lm	mai2vec-lm	nwjc2vec-lm
130.35	124.72	**118.68**

4. nwjc2vec における領域適応の問題

先の実験では学習用コーパスも評価用コーパスも mai2vec とは領域が異なるものである．このため nwjc2vec が有利に作用した結果とも言える．

ここでは学習用コーパスも評価用コーパスも mai2vec と領域が同じである新聞記事として同様の実験を行ってみる．具体的に言語モデルの学習用のコーパスとしては毎日新聞 2007 年度版から取り出した 10 万文を用いる．また評価用コーパスとては，毎日新聞 2008 年度版から取り出した 1 万文を用いる．

結果を表 3 に示す（図 1 参照）．mai2vec-lm が最もパープレキシティが低く，mai2vec が nwjc2vec と比較して高品質であることがわかる．

表 3　各分散表現データから得られた言語モデルの評価 2

base-lm	mai2vec-lm	nwjc2vec-lm
81.52	**64.81**	67.43

LSTM の学習用コーパスと評価用コーパスを mai2vec と同じ領域のものにすると，前章での実験結果が逆転している．これは nwjc2vec であっても領域適応の問題を受けることを意味する．

5. nwjc2vec の fine-tuning

分散表現データの領域適応の問題に対して，ここでは fine-tuning を行うことを提案する．具体的には分散表現データを構築するプログラムにおいて，分散表現データの初期値を既存の分散表現データに設定し，そこから処理領域に合ったコーパスを用いて分散表現データする．

(5) どのモデルも 4 あるいは 5 epoch で学習したモデルが最も品質が高かった．

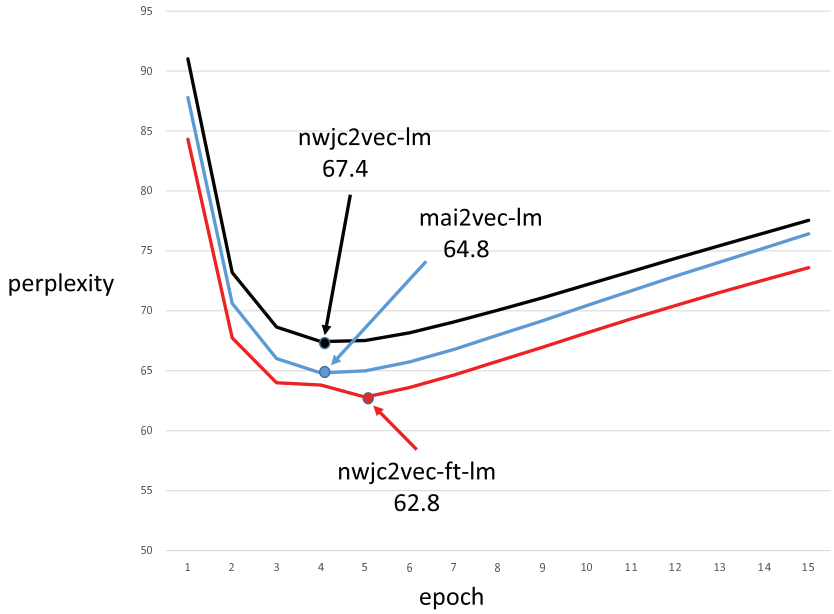

図 1　各分散表現を利用して構築した言語モデルのパープレキシティ

これによって既存の分散表現データを処理領域に合った分散表現データに tuning することができる．

実験として既存の分散表現データを nwjc2vec とし，fine-tuning のための学習用コーパスには毎日新聞 2007 年度版から取り出した 30 万文を用いる(6)．fine-tuning の結果構築された分散表現データをここでは nwjc2vec-ft と名付ける．

nwjc2vec-ft の評価は前章までの評価と同じ方式を用いる．つまり nwjc2vec-ft を用いて前章までの評価実験で用いた学習用コーパスから LSTM モデルを構築し，そのパープレキシティを前章で用いた評価用コーパスから測る．結果を図 1 に示す．明らかに nwjc2vec-ft を用いたモデルのパープレキシティが減少しており fine-tuning の効果が確認できる．

6.　考察

nwjc2vec であっても領域適応の問題は生じる．ただしそれを示した 4 章の実験では，mai2vec との差がそれほど大きくないことも注意すべきである．nwjc2vec の基になったコーパス NWJC が超大規模コーパスであるために，ある程度汎用的には使えると考えている．その上で領域を特化して更に精度を上げるために，本論文で示したようにな fine-tuning が有効であると思われる．

ただしここで行った fine-tuning のやり方には改善すべき点が多い．どの程度の量のコーパスを使えば良いのか，どの程度で学習を打ち切るべきか，学習で用いるパラメータをどのように設定すればよいか，などである．これらの点を今後調査していきたい．

(6) 前章で用いた毎日新聞 2007 年度版から取り出した 10 万文との重複はない．

7. おわりに

本論文では超大規模コーパス NWJC から構築された分散表現データ nwjc2vec であっても，領域適応の問題が生じることを示した．またその対処として fine-tuning を提案し，その効果も確認した．fine-tuning を行う際の様々な設定項目をどうすべきかを調査することが今後の課題である．

謝　辞

本研究の一部は国立国語研究所の共同研究プロジェクト「all-words WSD システムの構築及び分類語彙表と岩波国語辞典の対応表作成への利用」の研究成果を報告したものである．

文　献

岡崎直観 (2016).「言語処理における分散表現学習のフロンティア (＜特集＞ニューラルネットワーク研究のフロンティア)」　人工知能: 人工知能学会誌, 31:2, pp. 189–201.

Masayuki Asahara, Kikuo Maekawa, Mizuho Imada, Sachi Kato, and Hikari Konishi (2014). “Archiving and Analysing Techniques of the Ultra-large-scale Web-based Corpus Project of NINJAL, Japan.” *Alexandria: The Journal of National and International Library and Information Issues*, 25:1-2, pp. 129–148.

浅原正幸・岡照晃 (2017).「nwjc2vec:『国語研日本語ウェブコーパス』に基づく単語の分散表現データ」　言語処理学会第 23 回年次大会発表論文集, pp. 94–97.

山木翔馬・新納浩幸・古宮嘉那子・佐々木稔 (2017).「教師データを用いた語義の分散表現の構築」　言語処理学会第 23 回年次大会発表論文集, pp. 78–81.

新納浩幸・古宮嘉那子・佐々木稔 (2017).「順方向多層 LSTM と分散表現を用いた教師あり学習による語義曖昧性解消」　情報処理学会第 232 回自然言語処理研究会, pp. NL–232–4.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins (2000). “Learning to forget: Continual prediction with LSTM.” *Neural computation*, 12:10, pp. 2451–2471.

Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den (2014). “Balanced corpus of contemporary written Japanese.” *Language Resources and Evaluation*, 48:2, pp. 345–371.