学生の提出レポート解析に文脈ベクトルを使う

吉　武　春　光

1．まえがき

筆者は、吉武(2016)において潜在意味解析の中の１つの手法であるWord2Vec を使うための環境整備を行った。実際に Word2Vec を解析に用いた研究は、2015年から2016年にかけて、筆者が指導していた博士後期課程の学生と修士課程の学生が行った。今回、筆者は、Word2Vec の延長線上にある Doc2Vec という研究手法を用いて複数文の間の類似度を計算する研究を行うことにした。

Word2Vec は、語と語の間の関係をベクトル化するものであったが、文と文の間の関係をベクトル化する手法として開発されたのが Doc2Vec である。筆者は、吉武(2007, 2009) において、学生のレポート文章の解析について論じたが、意味記述の煩雑さのために、研究が進んでいなかった。これを打破するために、今回、Doc2Vec に注目した。

2．Doc2Vec による文脈ベクトルの生成

本章では、Doc2Vec の概要を述べた上で、Doc2Vec を使用するために必要となる、前処理、文脈ベクトル生成処理について述べる。

2.1 Doc2Vec の使い方

Doc2Vec （別名 paragraph2vec または sentence embeddings） は Word2Vec のアルゴリズムを『文、段落、文章などの連続する表現』に拡張したものである(http://radimrehurek.com/gensim/models/doc2vec.html)。

Doc2Vec の C言語版は存在せず、Python Gensim に実装されているもののみである。Gensim 上の Doc2Vec のクラスは基本的に Word2Vec のクラスを継承しているので、Python における Doc2Vec の使用方法は、基本的に Word2Vec と同じである。

なお、Word2Vec に組み込まれているアルゴリズムは “continuous bag of words” (cbow) と “skip-gram” (sg) の２つであったが、Doc2Vec では “distributed memory” (dm) と “distributed bag of words” (dbow) という２つのアルゴリズムが組み込まれており、dbow が default になっている。

2.2 Doc2Vec の入力

Doc2Vec の入力はプログラミング言語 Python の LabeledSentence objects の iterator(イテレータ)となっている。Python の iterator とは、「複数の要素を持ったデータの要素を順番に取り出すとき、より汎用的な手法を提供する方法」のことで、具体的には、Python のタプルやリストや辞書のデータ型のどれでも良いということである。例えば、Doc2vec tutorial のページに挙げてある例では、英文の「some words here」は、クラス LabeledSentence を用いて、下記のように 'some' と 'words' と 'here' を変数 words にリスト構造として入れている。

```
sentence = LabeledSentence(words=[u'some', u'words', u'here'], labels=[u'SENT_1'])
```

上記では、英文の「some words here」に対して 'SENT_1' という文ラベルを付加している。

なお、各々の文に対して、文ラベルを考えるのが煩雑なので、各文に文ラベルを自動的に付与するTaggedLineDocument というクラスも用意してある。TaggedLineDocument の入力は、半角空白で区切られた語が連なったテキストファイルとなっており、文の識別のためのラベルは、先頭文が 0 である文番号を自動的に割り当てるようになっている。入力が複数行ある場合は、第１入力行の最後で改行し、次の行に第２文目を入れる。

解析対象の単位は『改行までの１文』なので、もし解析対象の文が文の途中で改行されている場合は、改行コードを取り除く必要がある。

2.3 ベクトル生成

Python プログラムに読み込ませた語の連なりに対して、クラス Doc2Vec を用いてベクトル生成をさせる。数多くの引数があるが、主要なものを次に挙げる。

本研究では Ubuntu 16.04 というOS上で Python3.5 を使用したが、Doc2Vec のソースコードは /usr/local/lib/python3.5/dist-packages/gensim/models/doc2vec.py に入っている。

表１　Doc2Vec クラスの主要引数

引数名	説明
documents	入力の語の連なり。
dm	default は 1 で “distributed memory” (dm) を使う。 0 を指定した場合は “distributed bag of words” (dbow) モデルを使う。
size	文脈ベクトルの次元サイズ (default は 300)。
window	文脈ベクトルを計算する際に、対象語から調べる前後の語の数。大きな値を指定すると、より広範囲の語まで処理対象とする (default は 8) 。
hs	default は 1 で hierarchical sampling を使う。
min_count	出現回数が min_count より少ない語は無視される (default は 5) 。
workers	並列計算を行う際のスレッド数 (default は 1) 。
alpha	initial learning rate (徐々に min_alpha に近づく)
min_alpha	最終的な learning rate

2.4 繰り返し学習

Doc2Vec では、10回～20回ほどの繰り返し学習が必要となっている。まず、クラス Doc2Vec の中のメソッド build_vocab を用いて内部にボキャブラリーを作っておいてから、クラス Doc2Vec の中のメソッド train を10回

～20回ほど実行させることが必要になる。

2.5 文の類似度の検索

学習が終わったデータに対しては、docvecs プロパティに対して、クラス Doc2Vec の中のメソッド most_similar を使って、類似度が高い文番号と、その類似度の値をタプルにしたものの、任意の上位n個(topn 個)のリストを得ることが出来る。

また、docvecs プロパティに対して、クラス Doc2Vec の中のメソッド similarity を使って、２つの文番号を指定すると、２つの文番号の間の類似度を得ることが出来る。

2.6 予備実験

吉武(2015）で用いた日本経済新聞社の本紙（2014年１年分）を使って、予備実験を行った。記事を MeCab (工藤2002) を使って分かち書きし、語幹を半角スペース区切りにしたものである。MeCab を使って分かち書きを行うプログラムは、プログラミング言語 Python を使って自作した(text2mecab.py 付録１)。なお、Doc2Vec の意味の単位は、「入力された最初から改行コードまで」の文字列である。そこで、日経本紙では１つの記事が複数の文から構成されている場合は、文末のピリオド（句点）の位置で改行せずに、次の文をつなげ、記事の最後でのみ改行コードの挿入を行った。下記の、日経本紙2014年の第２記事（記事番号1）では記事が３つの文から構成されていることが分かる。

但し、動作確認を目的とするために、第１記事（記事番号0）をファイルの最後にコピー追加した（記事番号180497）。更に、第２記事を複製し、第２記事（記事番号1）と第３記事（記事番号2）を同じものとした。

静岡県 は 十 日 、 ヒロ インターナショナル （ 浜松市 、 谷口 一博 社長 ） が 運営 する 飲食店「 * 」で 米国産 牛肉 や 豪州 産 牛肉 など を 松阪牛 と 表示 する て いる た として 、 景品表示法 など に 基づく 改善 指導 を 行う た 。

図１　日経本紙2014年の第１文（記事番号0）（MeCab 処理済）

勤務先から売上金を盗むだことを隠すため、強盗の被害に遭うたと虚偽の通報をするたとして、警視庁東大和署は二日、東京都武蔵村山市、アルバイト、A容疑者（４８）を窃盗と軽犯罪法違反（虚偽申告）の容疑で逮捕するた。　調べによると、A容疑者は先月三十一日午前一時ごろ、勤務先の同市内のガソリンスタンドから売上金約八十二万円入りのバッグを盗む、その後「ナイフを持つた二人組の男にバッグを奪うれるた」と一一〇番通報をするた疑い。　A容疑者がスタンド内に隠すたバッグを別の店員が発見。同署が追及するたところ「借金で困るておる、生活費が欲しいてやるた」と認めるた。

図２　日経本紙2014年の第２文（記事番号1）（MeCab 処理済）

次の図に、python コマンドを使った実行の様子を示す。

```
yositake@dual:/zpool/lz4/hy-svr.hdd/doc2vec/NKM_2004_all_test$ python3
Python 3.5.2 (default, Nov 17 2016, 17:05:23)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import gensim
>>> docsrc =
gensim.models.doc2vec.TaggedLineDocument('NKM_2004_all_test.txt')
>>> m = gensim.models.doc2vec.Doc2Vec(docsrc, size=300, window=10,
hs=1, min_count=2, workers=8, sample=1e-5, alpha=.025, min_alpha=.025)
ここで時間がかかる。
>>> m.docvecs.most_similar(0,topn=3)
[(180497, 0.8200676441192627), (5130, 0.4211444556713104), (46957,
0.41085657477378845)]
記事番号 0 に最も近いのは 記事番号 180497 と表示されており、近似度
は 0.82 であった。
>>> m.docvecs.most_similar(1,topn=3)
[(2, 0.8915579915046692), (7216, 0.4085385203361511), (15521,
0.4077302813529968)]
記事番号 1 に最も近いのは 記事番号 2 と表示されており、近似度は 0.89
であった。
そこで、繰り返し学習を 10 回、行わせた。
>>> m.train(docsrc,total_examples=m.corpus_count,epochs=10)
ここで時間がかかる。
132822934
>>> m.docvecs.most_similar(0,topn=3)
[(180497, 0.9427038431167603), (151616, 0.36943313479423523),
(165145, 0.36827799677848816)]
記事番号 0 に最も近いのは 記事番号 180497 と表示されており、近似度
は 0.94 に上がった。
>>> m.docvecs.most_similar(1,topn=3)
[(2, 0.9340255856513977), (172317, 0.34578341245651245), (114136,
0.34412682056427)]
記事番号 1 に最も近いのは 記事番号 2 と表示されており、近似度は 0.93
に上がった。
>>>
```

図３　日経データを使った予備実験

近似度は 0 から 1 の間の値を取り、1 に近い値ほど似ていることを示している。この結果を見ると、異なる記事の近似度は大きくても 0.34 程度であるが、同一記事（記事番号0 と記事番号180497）の近似度は 0.94 という 1 に近い値になったことが分かった。また、別の同一記事（記事番号1 と記事番号2）の近似度は 0.93 になったことも分かった。

以上の予備実験により、似ている記事を確かに検出できるという確信を得た。

3．解析対象データと解析実験

さて、今回の実験対象は e-Learning システム Moodle 上の吉武の担当科目「情報ネットワーク論」に提出された課題である。[2012年度の課題：経路制御](133答案) と、吉武(2007)で使用したデータ [2005年度の課題：日本の文字コード](78答案)を使用した。

3.1 Moodle からの課題の抽出と前処理

Moodle の課題は、[オンラインテキスト] と [ファイル提出] という２種類の形式がある。

3.1.1 [オンラインテキスト] 形式の場合

[オンラインテキスト] 形式は、ダウンロードしたら HTML 形式になっている。その中から HTML タグと呼ばれる制御コードを取り除く必要がある。更に、2012年度から運用している Moodle バージョン2 以降 では内部漢字コードが UTF8 になっているが、2011年度まで運用していた Moodle バージョン1.9 では内部漢字コードが EUC になっているために、計算機処理するためには、漢字コードを EUC から UTF8 に変換する必要がある。 [2005年度の課題：日本の文字コード] の全78答案に対して HTML タグを取り除き、更に UTF8 に変換を行い、更に MeCab を使って分かち書きし語幹に変換を行った。以下は処理済みの一例である。

日本 の 文字コード は 、 １ ６ ビット を 使用 する た ASCII コード が ある 、 ASCII コード に ヨーロッパ の 各国 用 の 拡張 を 施す た ISO ８ ８ ５ ９ * １ 、 ASCII コード に 拡張 を 施す た JIS * ０ ２ ０ １ の ８ 単位 符号 が ある 。 また JIS * ０ ２ ０ １ の ８ 単位 符号 に は ISO ８ ８ ５ ９ * １ の 拡張 部分 と 重なる コード が ある 。 半角 カタカナ コード の 問題点 は 電子メール で 文字化け する こと だ ある 、 半角 カタカナ に 所属 する 記号 も 文字化け する ので 注意 が 必要 だ ある 。

図４　[2005年度の課題：日本の文字コード] の一例（MeCab 処理済）

3.1.2 [ファイル提出] 形式の場合

[ファイル提出] 形式は、アップロードしてあるファイル形式のままダウンロードされるが、Word 2007 以降で標準の docx 形式以外に、Word 2003 までで標準の doc 形式もある。

docx 形式の Word ファイル中の文字を抜き出すためには、２つの方法がある。

1) docx 形式の Word ファイル中の文字を抜き出すサービスを行っているサイトを利用する方法。

処理対象ファイルを１つずつ指定するため、処理に時間がかかる。

2) プログラミング言語 Perl で書かれた docx2txt.pl を使う方法。

複数のファイルを次々に処理できる。

doc 形式の Word ファイル中の文字を抜き出すためには、プログラミング言語で書かれた処理プログラムも存在するのだが、試してみたら問題が生じたために 1) のサイトのみを使用した。

docx 形式の Word ファイルから文字を抜き出すためには 1) で全てのファイルを処理させ、失敗したもののみを 2) で処理した。

[2012年度の課題：経路制御] の全141答案(docx形式とdoc形式が混在)に対して上記の文字を抜き出す処理を行い、更に MeCab を使って分かち書きし語幹に変換を行った。以下は処理済みの一例である。

経路 制御 と は 、 インターネット 上 で IP パケット を 特定 の 目的 地 に 転送 する ため の 、 パケット の 通り道 （ 経路 ） について の 情報 を 管理 する 、 複数 ある 経路 の うち から 、 最適 だ 経路 を 選択 する 仕組み の こと を いう 。 まず 、 経路 制御 の 必要性 だ が ルータ の ネットワーク インタフェース に は 、 IP アドレス と ネット マスク が 管理者 により 設定 する れる ます 。 その 場合 、 該当 インタフェース に 直接 接続 する れる て いる ネットワーク について は 、 経路 表 に 自動的 に 掲載 する れる 。 しかし 、 ルータ の インタフェース に 直接 繋がる て いる ない 離れる た ネットワーク の 場合 、 その 離れる た ネットワーク が どこ に （ どの インタフェース の 方向 に ） 存在 する の か 、 直接的 だ 判断 は 不可能 だ ある 。

図５　[2012年度の課題：経路制御] の一例（MeCab 処理済）

3.2 実験１　[2012年度の課題：経路制御]

まず、MeCab 処理済の133答案の全てに対して、各々、意味ベクトルの生成を行ってから、繰り返し学習を行い、その結果に対して、各答案に対して最も近似度が高い答案を調べる Python プログラムを作成した(d2v_out.py 付録２)。繰り返し学習の回数は20回とした。

3.2.1 実験結果

得られた近似度を降順にソーティングした結果は次の通りである。

答案番号	答案番号	近似度
37	35	0.897456169
35	37	0.897456169
65	10	0.796049595
10	65	0.796049595
74	62	0.665772617
62	74	0.665772617
…	…	…
18	0	0.284812748
50	39	0.280477494
130	87	0.277597368
17	14	0.262090504

答案番号 37 に最も近似度が高いのは 答案番号 35 だと判り、その時の近似

度は 0.897456169 となっている。また、答案番号 65 と答案番号 10 との近似値は 0.796049595 となっている。答案番号37、答案番号35、答案番号65、答案番号10 の全記事は次の通りである

経路 制御 は 根幹 を なす 技術 だ ある 、 大変 だ 重要 だ 役割 を 持つ て いる 。 経路 制御 の 必要性 を 挙げる て いく 。 まずは 一つ 目 は インター ネットワーク は 、 網 の 目 状 の よう に なる て いる 。 よって 、 目的 の 場所 へ 行く ため に 、 可能 だ 経路 は いくつ も 存在 する という こと だ ある 。 いくつ もの 経路 が 存在 する という こと は 、 すなわち どの 経路 を 通る か という こと を 決める 必要 が ある 。 二つ目 は もしも 経路 に 何らかの 問題 が 発生 する 、 不通 に なる て いる 場合 に 、 どの よう だ 対処 を 取る か 考える て おく 必要 が ある 、 ルーティング は 経路 の 情報 を あらかじめ ネットワーク機器 に 設定 する て おく スタティック ルーティング と 、 経路 情報 を 常に 更新 する ダイナミック ルーティング と に 分かれる 。

簡単 に 原理 を 説明 する て いく と 、 TCP/IP ネットワーク で パケット を 送る う と する とき 、 経路 制御 表 （ ルーティング ・ テーブル ） を 参照 する て パケット の 通り道 （ 経路 ） について の 情報 を 管理 する 、 複数 ある 経路 の うち から 、 最適 だ 経路 を 選択 する 。 ある 端末 から 他 の 端末 へ と パケット を 送る う と する 場合 、 目的 の 端末 が 自 ネットワーク 内 に ない 場合 、 端末 内 に ある 経路 制御 表 を 参照 する 、 パケット を 中継 する せる 端末 を 決定 する 。 経路 制御 表 は インターネット の 状態 を 示す もの だ 、 刻々 と 姿 を 変える て いる 。 その ため 問題 が 存在 する て も 、 また 他 の 経路 から 通じる という こと 。

経路 制御 表 （ ルーティング ・ テーブル ） と は 、 宛先 アドレス に たどり着く ため に は 次に どの ルータ に 行く ば 良い の か 、 を 書く た 表 の こと だ ある 、 駅 の 乗り換え 時刻表 の・ようなもの 。 経路 制御 （ ルーティング ） に は タイプ が ある 、 * ホスト ・ ルーティング * ネットワーク ・ ルーティング * デフォールト ・ ルーティング が 存在 する 。 他 ネットワーク 間 で の 接続 に は ルータ が 用いる られる が 、 IP アドレス を 見る て 相手 の ホスト を 探す 接続 する こと を * ホスト ・ ルーティング と いう 。 しかし その IP アドレス を 持つ ホスト を 含む ネットワーク の ルータ を 探す て ， ルータ 同士 を 接続 する 方法 * ネットワーク ・ ルーティング ） を 取る 必要 が ある 。 単純 に は ， すべて の ネットワーク * アドレス と その ルータ の * アドレス の 一覧 表 を 持つ ば よい の です が ， 世界中 に は 膨大 だ ネットワーク が ある ので ， そのよう だ こと は 不可能 だ ため ルータ に よく 接続 する 相手 ， 最近 接続 する た 相手 に つい て は 経路 制御 表 （ ルーティングテーブル ） を 作成 する て おく ， それ に 合致 する た もの は それ を 利用 する ， その リスト に ない 場合 に は 上位 の ルータ に 接続 する * デフォルト ルート という ） 。 経路 制御 表 において 幾つ か の 経路 の 候補 が ある 場合 に 、 どの 経路 を 選択 する か 決める 手順 を ルーティング ・ アルゴリズム と いう 。 手順 に は 静的 （ スタティック ） と 動的 （ ダイナミック ） が 存在 する 。 静的 （ スタティック ） だ ルーティング は ルータ など が 、 管理者 が 予め 設定 する た 固定 的 だ 経路 表 （ ルーティングテーブル ） に 基づく て 経路 選択 を 行う こと だ ある 。 要は 人 が や 管理 する こと だ 、 手軽 だ は 会う が 維持管理 が 大変 だ ある 。 それ に対して 、 動的 （ ダイナミック ） だ ルーティング は ルータ など が 経路 情報 を 交換 する 合う 、 自動的 に 生成 ・ 更新 する 続ける 経路 表 * ルーティングテーブル * に 基づく て 経路 選択 を 行なう こと 。

図6　[2012年度の課題：経路制御] の記事番号 37 の全記事（MeCab 処理済）

情報 ネットワーク 論 （ 月 2 ）
* AG *　　○○ ○
　経路 制御 は 根幹 を なす 技術 だ ある 、 大変 重要 だ 役割 」 を 持つ て いる 。 経路 制御 の 必要性 を 説明 する 。 まず 一つ 目 に インター ネットワーク は 、 網 の 目上 に なる て いる という こと だ ある 。 よって 、 目的 場所 に 行く ため に 可能 だ 経路 は いくつ も 存在 する という こと だ ある 。 いくつ も の 経路 が 存在 する という こと は 、 すなわち どの 経路 を 通る の か という こと を 決める 必要 が ある 。 二つ目 は もしも 経路 に 何らかの 問題 が 発生 する * 不通 に なる て いる 場合 に 、 どの よう だ 対処 を 取る か を 考える て おく 必要 が ある 、 ルーティング は 経路 の 情報 を あらかじめ ネットワーク機器 に 設定 する て おく * と 、 経路 情報 を 常に 更新 する ダイナミック ルーティング と に 分かれる 。
　簡単 に 原理 を 説明 する て いく と 、 TCP/IP ネットワーク で パケット を 送る う と する とき 、 経路 制御 表 （ ルーティング ・ テーブル ） を 参照 する て パケット の 道路 （ 経路 ） について の 情報 を 管理 する 、 複数 ある 経路 の うち から 、 最適 だ 経路 を 選択 する 。 ある 端末 から ほか の 端末 へ パケット を 送る う と する 場合 、 端末 内 に ある 経路 制御 表 を 参照 する 、 パケット を 中断 する せる 端末 を 決定 する 。 経路 制御 表 は 」 インターネット の 状態 を 示す もの だ 、 姿 を 次々 と 変える て いる 。 その ため の 問題 が 存在 する て も 、 また ほか の 経路 から 通じる という こと だ ある 。
　経路 制御 表 （ * ） と は 、 宛先 アドレス に たどり着く ため に は 次に どの ルータ に 行く ば 良い の か を 書く た 表 の こと だ ある 、 駅 の 乗り換え 時刻表 の・ようなもの 。 経路 制御 （ * ） に は タイプ が ある 、 * ホスト ・ ルーティング * ネットワーク ルーティング * デフォルト ルーティング が 存在 する 。 他 ネットワーク 間 で の 接続 で は ルータ が 用いる られる が 、 IP アドレス を 見る て 相手 の ホスト を 探す 接続 する こと を * ホスト ・ ルーティング と いう 。 しかし その IP アドレス を 持つ ホスト を 含む ネットワーク の ルータ を 探す て 、 ルータ 同士 を 接続 する せる 方法 * ネットワーク ・ ルーティング ） を 取る 必要 が ある 。 単純 に は 、 全て の ネットワーク IP アドレス と その ルータ の IP アドレス の 一覧 表 を 持つ ば 良い の です が 、 世界中 に は 膨大 だ ナット ワーク が ある ため 、 そのよう だ こと は 不可能 だ ため ルータ に よく 接続 する 相手 、 最近 接続 する た 相手 について は 経路 制御 表 （ ルーティング ・ テーブル ） を 作成 する て おく 、 それ に 合致 する た もの は それ を 利用 する 、 そのリスト に ない 場合 に は 上位 の ルータ に 接続 する * デフォルト ルート という ） 。
　経路 制御 表 において いくつ か の 経路 の 候補 が ある ば あい に 、 どの 経路 を 選択 する か 決める 手順 を ルーティング ・ アルゴリズム と いう 。 手順 に は 静的 （ スタティック ） と 動的 （ ダイナミック ） が 存在 する 。 静的 （ スタティック ） だ ルーティング は ルータ など が 、 管理者 が あらかじめ 設定 する た 固定 的 だ 経路 表 （ ルーティング ・ テーブル ） に 基づく て 経路 選択 を 行う こと だ ある 。 要は 人 が 管理 する こと だ 、 手軽 だ は ある が 維持管理 が 大変 だ の が 欠点 だ ある 。 それ に対して 、 動的 （ ダイナミック ） だ ルーティング は ルータ など が 経路 情報 を 交換 する 合う 、 自動的 に 生成 ・ 更新 する 続ける 経路 表 （ ルーティング ・ テーブル ） に 基づく て 経路 選択 を 行う こと が 大事 だ 。

図７　[2012年度の課題：経路制御] の記事番号 35 の全記事（MeCab 処理済）但し、答案中に氏名が入っていたので、氏名を○に変更してある。

経路 制御 の 必要性 ・ 原理 ・ ルーティングアルゴリズム について
* AF *　○○ ○○○

経路 制御 の 必要性 について だが 、 まず インターネット は 網 の 目 状 に なる て いる ため 、 自分 が つながる たい と 思う 相手 の もと に 行く ため に は 様々 だ 網 の 目 の 中 から そして 様々 だ 経路 の 中 から 探し出す ない ば なる ない 。 その ため に 経路 制御 を 利用 する こと だ 、 どの 経路 を 通る べし だ の か を 判断 する せる 必要 が ある 。 また 、 もしも その 経路 が 不通 に なる て いる 場合 に は どの よう に する て 対処 する べし だ の か も 考える て おく 必要 が ある 。 そこ に 動的 だ 経路 制御 アルゴリズム を 採用 する こと に よって 、 動的 に 制御 する こと が できる 。 これ が 必要性 として あげる られる 。

次に 経路 制御 の 原理 について だが 、 経路 制御 は 電車 の 乗換駅 に 例える こと が できる だ う 。 インターネット に つながる た コンピュータ は 駅 に 該当 する 、 駅 に は 乗換 の ため に 時刻表 が 置く てある 。 これら の 前提 を 踏まえる て 、 インターネット の ネットワーク で は 、 パケット を 送る とき に あらかじめ 路線 や 時刻 を 調べる という こと を 行う ない 。 まず 最初 に パケット を インターネット に 送り出す 。 そう する と 、 送り出す れる た パケット は 、 駅 にて 時刻表 に 相当 する 経路 制御 表 を 調べ だ し 、 次に どの 経路 に 乗る べし か を 判断 する 。 その 経路 の 終点 に たどり着く と 、 そこで 再び 経路 制御 表 を 自ら 調べ だ し 、 次に 乗る べし 経路 を 設定 する 。 この 作業 を 繰り返す 、 目的 地 に たどり着く の だ ある 。 この 、 経路 制御 表 は インターネット に 路線 の 状態 を 示す もの だ ある 、 刻々 と 変化 する て いる 。 もし 事故 （ 不通 ） の ため に 途中 で 路線 が 通れる ない という 状況 に なる ても 、 この 経路 制御 表 が 無事 に 通れる 経路 を 常に 示す て くれる ので 、 それ に従って 迂回 する こと が 未然 に できる の だ ある 。

それでは この 経路 制御 表 の 構成 について だが 、 この 経路 制御 表 は 、 宛先 アドレス に たどり着く ため に は 次に どの ルーター に 行く ば いい の か を 書く た 表 だ ある 。 種類 として は 、 デフォルト ルーティング ・ * が 種類 として あげる られる 。 実際 に は 、 この デフォルト ルーティング ・ * を 組み合わせる て 使用 する 。

最後 に ルーティングアルゴリズム について だが 、 ルーティングアルゴリズム と は 経路 制御 表 において 経路 の 中 に いくつ か の 候補 が ある た 場合 に 、 どの 経路 を 選ぶ べし だ の か を 決める 手順 の こと だ ある 。 この 手順 に は 二つ の ルーティング が 存在 する て いる 、 それ が 静的 ルーティング と 動的 ルーティング だ ある 。 静的 だ ルーティング は 人間 が 行先 ・ 経由 ・ セグメント 生き の ルーター 経由 など を 指示 する 。 静的 だ ルーティング に は 周り の ネットワーク 環境 の 変化 に 応じる て それ に 追従 する て 設定 を 行う 必要 が ある 場合 が ある 、 矛盾 の ない ルーティング を 行う よう に 気 を 配る 必要 が ある 。 それ は 、 比較 的 に 気軽 に 利用 できる という 利点 だ ある が 、 維持管理 が 大変 だ ある という 欠点 も ある 。

次に 動的 だ ルーティング だ ある が 、 この 動的 ルーティング は 、 自ら 宛先 へ の 距離 情報 を 隣る 合う ルーター 同士 で 交換 する 、 これ により 経路 制御 表 を 完成 する せる 、 票 の 中 で 最も 距離 が 短い もの を 選択 する 機能 を 持つ 。 この 「 距離 」 として 定義 する れる もの の 違い で その 種類 ごと に 距離 情報 交換 用 の プロトコル が 存在 する 。

図８　[2012年度の課題：経路制御] の記事番号65 の全記事（MeCab 処理済）但し、答案中に氏名が入っていたので、氏名を○に変更してある。

経路 制御 の 必要性 ・ 原理 ・ 経路 制御 表 ・ ルーティングアルゴリズム について
* AF * ○○ ○
経路 制御 の 必要性 について である が 、まず インターネット という もの は 網 の 目 状 に なる て いる ため 、ある 相手 の 場所 に 行く ため に は 様々 だ 網 の 目 の 中 から いくつ も ある 経路 から 探し出す ない ば なる ない 。 その ため に 経路 制御 を 利用 する こと だ 、どの 経路 を 通る か を 判断 する せる 必要 が ある 。 また 、 もしも その 経路 が 不通 に なる て いる 場合 に は どの よう に する て 対処 を する べし だ の か を 考える て おく べし だ ある が 、 そこ に 動的 だ 経路 制御 アルゴリズム を 採用 する こと だ 、 動的 に 制御 を 行う こと が できる 。 これ が 第 一 に 必要性 として あげる られる 。
次に 経路 制御 の 原理 について である が 、 経路 制御 は 鉄道 の 乗換駅 に 例える られる こと と なる 。 さらに インターネット に つながる た コンピュータ は 駅 に 該当 する 、 駅 に は 乗換 の ため に 時刻表 が おく て ある 。 これら 前提 を 踏まえる て 、 インター ネットワーク で は 、 パケット を 送る 時 に 、 あらかじめ 路線 や 時刻 を 調べる たり する という こと を 行う ない 。 まず 最初 に パケット を インターネット に 送り出す 。 そう する と 、 送り出す れる た パケット は 、 駅 にて 時刻表 に 相当 する 経路 制御 表 を 調べる 出す 、 次に どの 経路 に 乗る べし か を 判断 する 。 その 路線 の 終点 に たどり着く と 、 そこ で 再び 経路 制御 表 を 自ずから 調べ だ し 、 次に 乗る べし 路線 を 決定 する 。 この 作業 を 繰り返す 、 目的 地 に たどり着く の だ ある 。 この 、 経路 制御 表 は インターネット に 路線 の 状態 を 示す もの だ ある 、 刻々 と 変化 する て いる 。 もし 事故 （ 不通 ） の ため に 途中 で 路線 が 通れる ない という 状況 に なる て も 、 この 経路 制御 表 が 無事 に 通れる 経路 を 常に 示す て くれる ので 、 それ に 従って 迂回 する こと が 未然 に できる の だ ある 。
それでは この 経路 制御 表 の 構成 について である の だ が 、 この 経路 制御 表 は 、 宛先 アドレス に たどり着く ため に は 次に どの ルータ に 行く ば いい の か を 書く た 表 だ ある 。 種類 として は 、 ホスト ルーティング ・ * が 種類 として あげる られる 。 実際 に は 、 この ホスト ルーティング ・ * を 組み合わせる て 利用 する 。
最後 に 、 ルーティングアルゴリズム について である が 、 ルーティングアルゴリズム と は 経路 制御 表 において いくつ か の 経路 の 候補 が ある 場合 に 、 どの 経路 を 選択 する か を 決める 手順 の 事 だ ある 。 これ に は ふたつ の ルーティング が 存在 する 、 静的 ルーティング と 動的 ルーティング が ある 。 静的 だ ルーティング は 人間 が 行く 先 、 経由 、 セグメント 行き の ルータ 経由 など を 指示 する 。 静的 だ ルーティング に は 、 回り の ネットワーク 環境 の 変化 に 応じる て それ に 追従 する て 設定 を 行う 必要 が ある 場合 が ある 、 矛盾 の ない ルーティング を 行う よう に 気 を 配る 必要 が ある が 、 比較 的 に 気軽 に 利用 できる 。 しかし 、 維持管理 が 大変 だ ある という 問題点 も ある 。
　次に 動的 だ ルーティング だ ある が 、 この 動的 ルーティング は 、 自ずから 宛先 へ の 距離 情報 を 隣る 合う ルータ 同士 で 交換 する 、 これ により 経路 制御 表 を 完成 する せる 、 表 の 中 で 最も 距離 が 短い もの を 選択 する 機能 を 持つ 。 この 「 距離 」 として 定義 する れる もの の 違い で その 種類 ごと に 距離 情報 交換 よう の プロトコル が 存在 する 。

図９　[2012年度の課題：経路制御] の記事番号10 の全記事（MeCab 処理済）但し、答案中に氏名が入っていたので、氏名を○に変更してある。

3.2.2 実験結果の考察

原文を見ると分かるように、答案番号 35 と 37 は酷似した内容であった。また、答案番号 10 と 65 も酷似した内容であった。[2012年度の課題：経路制御] は、講義資料を元にまとめるだけで答案が作成出来るものであったので、ある程度、答案が似ても仕方ないと思われる。しかし、これだけ似ていると答案を写したのかと疑いたくなってしまう。

3.3 実験２　[2005年度の課題：日本の文字コード]

次に、吉武(2007)で使用したデータ [2005年度の課題：日本の文字コー

ド](78答案) に対して、MeCab 処理を行い、実験１で使用した Python プログラム (d2v_out.py)を使って、意味ベクトルの生成を行ってから、繰り返し学習を行った。

3.3.1 実験結果

次に、繰り返し学習を20回と指定して、近似度を計算させた。得られた近似度を降順にソーティングした結果は次の通りである。

答案番号	答案番号	近似度
42	1	0.979680717
1	42	0.979680717
44	3	0.978571534
3	44	0.978571534
69	47	0.975920677
47	69	0.975920677
…	…	…
12	25	0.759830475
28	64	0.735853076
38	32	0.709882736
10	38	0.616832972

答案番号 42 に最も近似度が高いのは 答案番号 1 だと判り、その時の近似度は 0.979680717 となっている。また、答案番号 44 と答案番号 3 との近似値は 0. 978571534 となっている。答案番号 42、1、44、3 の全記事は次の通りである。

日本 の 文字コード は 、 シフト JIS 、 EUC-JP 、 ISO * 2 0 2 2 * JP が ある ます 。 ASCII コード に ヨーロッパ 各国 の 言語 を 拡張 する た 、 ISO 8 8 5 9 * 1 コード が ある 、 日本 に は ASCII コード に カタカナ 用 の 拡張 を 施す た JIS * 0 2 0 1 が ある ます 。 しかし 、 それぞれ の 拡張 部分 に は 重なる ところ が ある 、 ISO 8 8 5 9 * 1 と JIS * 0 2 0 1 の 8 単位 符号 で は 同時に 使用 できる ない という 問題 が 生じる ます 。 JIS * 0 2 0 1 では 「 * 」 と 表示 する れる もの が 、 ISO 8 8 5 9 * 1 では 「 * で 表示 する れる ので 、 電子メール など を 使用 する と JIS * 0 2 0 1 の 重なる て いる コード が 文字化け を おこす こと が ある ます 。 この よう だ 問題 により インターネット の 世界 で は 半角 カタカナ 、 重 なり コード を 使用 する こと は 禁止 する れる て いる ます 。

図10　[2005年度の課題：日本の文字コード] の記事番号 42 の全記事 (MeCab 処理済)

日本 の 文字コード は 、 1 6 ビット を 使用 する た ASCII コード が ある 、 ASCII コード に ヨーロッパ の 各国 用 の 拡張 を 施す た ISO 8 8 5 9 * 1 、 ASCII コード に 拡張 を 施す た JIS * 0 2 0 1 の 8 単位 符号 が ある 。 また JIS * 0 2 0 1 の 8 単位 符号 に は ISO 8 8 5 9 * 1 の 拡張 部分 と 重なる コード が ある 。 半角 カタカナ コード の 問題点 は 電子メール で 文字化け する こと だ ある 、 半角 カタカナ に 所属 する 記号 も 文字化け する ので 注意 が 必要 だ ある 。

図11　[2005年度の課題：日本の文字コード] の記事番号 1 の全記事 (MeCab 処理済)

日本 の 文字コード の 種類 は 、 * シフト JIS * EUC-JP （ UNIX 用 * ISO * 2 0 2 2 * JP （ 電子メール 用 ） 半角 カタカナ と 呼ぶ れる コード の 問題点 は 、 電子メール に 入れる た 半角 カタカナ は 文字化け を 起こす もの が 多い 。 インターネット の 世界 で は 使用 が 禁止 する れる て いる 。

図12　[2005年度の課題：日本の文字コード] の記事番号 44 の全記事 (MeCab 処理済)

日本語 の 文字コード JIS X 0201 * ASCII コード に カタカナ 用 の 文字コード を 施す た もの * JIS X 0208 （ 漢字 コード ） 、 シフト JIS 、 EUC-JP 、 ISO-2022-JP 問題点 半角 カタカナ は JIS X 0201 の * 単位 符号 の 右 半分 に 表記 する れる て いる 文字 （ つまり 半角 カタカナ と 「 」 、 。 など の 記号 ） を 入力 する と 、 それ が インターネット の 世界 で 表示 する れる た 際 に 文字化け を する ため 、 使用 禁止 に なる て いる 。

図13　[2005年度の課題：日本の文字コード] の記事番号 3 の全記事 (MeCab 処理済)

3.3.2 実験結果に対する考察

残念ながら、答案番号 42 と答案番号 1の近似度が高いというのは、納得できないものであった。更に、答案番号 44 と答案番号 3の近似度が高いというのも、納得できないものであった。

その原因であるが、[2005年度の課題：日本の文字コード]は、課題がシンプルであったので、どの答案も似通っていて文字数が少ないものが多かった。一方、実験１ [2012 年度の課題：経路制御] では答えるべき内容が多かったので、文字数が多い答案が多かった。そのため、実験１ [2012 年度の課題：経路制御]があれだけ高精度の近似度を出していることから判断して、文字数が少ない答案は深層学習の多次元ベクトル計算には向いていないと思われる。

４．あとがき

本研究では、前講で述べた Word2Vec を発展させた Doc2Vec を使い、実際に学生が提出したレポートの文面の近似度を調べた。確かに、近似度が 0.9 ぐらいと判断されたレポートは、目視でも文面を僅かだけ修正したものではないかと推測された。長らく進展していなかった近似度の研究が、新しい手法で大きく進展した意義は大きい。

実験１と実験２で使用した Doc2Vec のパラメータは、試行して良い結果が得られた値を使用した。パラメータ決定に関しては、検討が必要かもしれない。

なお、Deep Learning は、計算させる度に結果の僅かな変動が生じるものらしい。実際の実験結果でも、小数点以下５桁程度は変動しているようである。なので、小数点以下３桁程度を有効値と見なす方が良さそうである。これは、気持ち悪いものであるが、仕組み上、仕方ないそうである。

本研究では Doc2Vec の有効性を確認できたが、Deep Learning の手法は、他にも幾つもある。今後は、他の手法へも挑戦したい。また、Deep Learning の手法を画像処理などにも適用してみたい。

参考論文

吉武春光(2007): " SDRTによる談話の意味記述", 西南学院大学商学論集, Vol. 53, No.3 & 4, pp. 211-238, 2007.2月

吉武春光(2009): " Prologを使ったSDRT表記の実現", 西南学院大学商学論集, Vol. 55, No. 4, pp. 257-304, 2009.3月

吉武春光(2016): "日経記事の解析に文脈ベクトルを使うための環境整備", 西南学院大学商学論集, Vol. 62, No.3&4, pp. 263-284, 2016.3月

工藤 拓(2002): MeCab: Yet Another Part-of-Speech and Morphological Analyzer

http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html

Genism　https://radimrehurek.com/gensim/

RadimRehurek: Gensim のDoc2Vec サイト

http://radimrehurek.com/gensim/models/doc2vec.html

RadimRehurek: Doc2vec tutorial

https://rare-technologies.com/doc2vec-tutorial/

付録

1）text2mecab.py

```
#!/usr/bin/env python3
import MeCab
import csv
import sys
argvs = sys.argv
t = MeCab.Tagger("")

with open(argvs[1]) as csvfile:
    dataReader = csv.reader(csvfile, delimiter=',')
    for row in dataReader:
        if row[0] != "articleid":
            result = t.parse(row[6])
            for col in result.split('¥n'):
                if col != "EOS" and col != "":
                    parts = col.split(',')
                    print(parts[6], end=' ')
            print()
```

2）d2v_out.py

```
#!/usr/bin/env python3
import gensim
import sys
import csv

docsrc = gensim.models.doc2vec.TaggedLineDocument('../2014-all.mecab')
m = gensim.models.doc2vec.Doc2Vec(docsrc, size=300, window=10, hs=1,
min_count=3, workers=4, sample=1e-5)

print('¥n 繰り返し学習の開始')
for epoch in range(20):
        print('第 {} 回'.format(epoch + 1))
        m.train(docsrc,total_examples=m.corpus_count,epochs=20)
        m.alpha -= (0.025 - 0.0001) / 19
        m.min_alpha = m.alpha

length = len(list(m.docvecs))
print("max line number=", length)
f = open('out.csv', 'w')
writer = csv.writer(f, lineterminator='¥n')
allout = []

for x in range(length):
    each = m.docvecs.most_similar(x,topn=1)
    out = (x, each[0][0], each[0][1])
    allout.append(out)
    writer.writerow(out)
f.close()

from operator import itemgetter
allout.sort(key=itemgetter(2,0), reverse=True)

f = open('out_sorted.csv', 'w')
writer = csv.writer(f, lineterminator='¥n')
```