Nグラムコーパスを用いたIT用語の意味ベクトルの獲得

渡邊 和弥　　馬　青
龍谷大学大学院理工学研究科数理情報学専攻
t14m006@mail.ryukoku.ac.jp, qma@math.ryukoku.ac.jp

1　はじめに

Mikolov らによって提案された word2vec が単語の意味の加算や減算をすることができて注目を浴びている [1]。word2vec では文脈から出現単語を予測することで単語の共起情報に基づいて分散表現を学習する。そうして得られた分散表現を利用して、単語の意味の足し引きや、意味の近い単語を求めることができる。本稿ではこのような分散表現を意味ベクトルと呼ぶ。

word2vec で精度の高い意味ベクトルを獲得するためには大量な文章（または文）を学習データとして用いる必要がある。しかしながら、このようなデータを大規模に（たとえばWeb検索で）収集するのは必ずしも容易なことではない。一方、Web データに基づく大規模な単語 N グラムは工藤らや矢田により作成され公開されている [3][4]。word2vec に文を与えることは文脈情報を与えることとある意味で等価であることから、学習データとしては文章または文データの代わりに単語 N グラムデータを用いることが考えられる。実際、英語とフィンランド語については単語 N グラムデータを用いた研究がすでになされている [2]。

本研究では日本語について単語 N グラムの学習データとしての有効性を確認することを目的としている。具体的には、word2vec を用いて 10 個の IT 用語とそれらの類語の計 20 個の単語について、Web 検索で収集した文データ、２種類の単語 N グラムデータ、さらにはこれら三つを併合したデータをそれぞれ学習データとして用いた場合の評価実験を行った。

2　コーパス

評価実験に用いた IT 用語および類語は表 1 に、それらに関する４種類のコーパスデータは表 2 にまとめている。ただし、表 2 の中の項目「Google」,「Susumu」,「Web」,「併合」はそれぞれ、Google の N グラムデータ [3]、矢田の N グラムデータ [4]、Web 検索から収集したデータ、上記３つを併合したデータを表している。また、N グラムデータについてはスペースの都合上データ数のもっとも多い 4,5,6-gram のもののみを示している。また、表中の数字は該当用語の出現回数、すなわち、N グラムのデータは該当用語を含むグラムの数、Web 検索データは該当用語を含む文の数、併合データは上記３つの数の合計である。なお、たとえば IT 用語の「スパムメール」が「スパム」と「メール」の２語に分かれていたり、「スパムメール対策」のように他の単語との複合語となっている場合は結合または分割処理を施した。

表 1: IT 用語と類語

	用語	類語
1	暗号化	符号化
2	コールセンター	カスタマーセタンター
3	可用性	稼働率
4	セキュリティホール	脆弱性
5	アドイン	アドオン
6	プロジェクトマネジメント	プロジェクト管理
7	IC カード	スマートカード
8	迷惑メール	スパムメール
9	セキュリティソフト	アンチウイルスソフト
10	メインフレーム	ホストコンピュータ

表 2: 4 種類のコーパスデータ

	Google			Susumu			Web	併合
	4-gram	5-gram	6-gram	4-gram	5-gram	6-gram		
暗号化	22402	29424	34084	8095	13514	33283	3664	158006
符号化	4846	4173	3516	3428	3326	3940	6028	35030
コールセンター	15621	18510	20431	6699	8362	13836	1824	98304
カスタマーセンター	1689	1744	1748	740	844	778	1798	14797
可用性	2687	2843	2513	1102	1464	1215	3298	21070
稼働率	3771	2844	2228	3405	4186	4433	1806	26693
セキュリティホール	2997	2980	2871	1023	1090	1330	1649	16553
脆弱性	13101	15832	17093	4672	6779	11590	3428	80685
アドイン	1103	976	953	498	448	572	2400	9260
アドオン	2890	2847	2905	1681	1544	1045	3216	24712
プロジェクトマネジメント	5325	5735	6063	1611	1575	2191	1870	29846
プロジェクト管理	4284	4610	4976	1399	1217	1804	1862	23960
IC カード	10980	10807	10194	6767	8379	12982	3748	74158
スマートカード	863	762	722	267	187	197	1292	6053
迷惑メール	19650	20830	19306	7230	11242	18587	6066	120078
スパムメール	6236	5970	5741	2610	3148	2548	3449	44103
セキュリティソフト	5845	5993	6031	1737	1905	2635	2322	32553
アンチウイルスソフト	1215	1109	132	502	603	93	1528	6749
メインフレーム	2742	2447	2383	1496	1332	1505	1396	16734
ホストコンピュータ	757	636	628	602	620	566	1325	6898

2.1 Google の N グラムデータ

言語資源協会から発行されている Google の工藤らによって作成された「Web 日本語 N グラム第 1 版」という N グラムコーパスである。これは、Web から抽出した約 200 億文の日本語データから作成されたもので、Mecab により形態素解析された 1〜7-gram のデータである。

2.2 Susumu の N グラムデータ

矢田によって作成され web 上で無料で公開されている N-gram コーパスである。これは、形態素区切りと文字区切りの 1〜7-gram のデータであり、本研究では形態素区切りのものを使用している。

2.3 Web データ

IT 用語をそれぞれ検索クエリとして Google で検索して取得した。取得した Web データからタイトルと本文のみを抽出し、それらを分かち書きにして用いた。

2.4 併合データ

上記 3 種類のデータ、すなわち Google の N グラムデータ、Susumu の N グラムデータ、Web データを併合したものである。

3 word2vec

word2vec は単語の意味ベクトルを得ることを目的としている。word2vec では、その学習モデルとして Continuous Bag-of-Words（CBOW）と

Skip-gram が提案されており、また、学習の効率化を図るために階層的ソフトマックスとネガティブサンプリングという 2 通りの手法が用いられている。

CBOW では対象単語 w_t の前後 k 単語 $w_{t-k}, ..., w_{t-1}, w_{t+1}, ..., w_{t+k}$ の Bag-of-Words 表現を入力とし w_t を推定し学習していくモデルである。一方、Skip-gram は文章内の対象単語 w_t が与えられてその前後 k 単語 $w_{t-k}, ..., w_{t-1}, w_{t+1}, ..., w_{t+k}$ を推定し学習していくモデルである。

階層的ソフトマックスでは、すべての単語を一括で学習するのは計算量が膨大になってしまうので単語を階層的なグループに分けて各グループごとに学習する。階層的なグループは、単語とその出現回数を用いてハフマン符号化を行い、それぞれの単語にハフマン符号を割り振ることによって作成される。ハフマン符号化を行うメリットは、出現頻度の高い単語ほど短い符号が割り当てられるところにある。一方、ネガティブサンプリングでは、出力層で正解ニューロン以外のニューロンを更新しない代わりにランダムに 5 個ぐらい「偽の入力」を選び、その偽の入力で正解の出力が出る確率が下がるように学習をする [5]。

本研究では学習モデルとしては Skip-gram、学習の効率化には階層的ソフトマックスを用いることにした。

4 評価実験

4.1 実験条件と評価方法

実験では Google が開発した word2vec のツール WORD VECTOR estimation toolkit v 0.1b を用いた[1]。本ツールでは、学習モデルの選択 (デフォルト : Skip-gram)、ベクトルのサイズ (デフォルト : 100)、文脈の前後単語数（デフォルト : 前後 5 語ずつ）など多数のパラメータがあるが、実験ではすべてデフォルトの値をそのまま使用した。

実験および評価は以下の手順で行う。まず、2 章で述べた 4 種類のコーパスをそれぞれ学習データとして word2vec に学習させ 20 個の IT 用語のベクトルを得る。次に、得られたベクトルに対してベクトル間のコサイン類似度をもとめ、各用語の類似度上位 5 個の単語を出力する[2]。上位 5 個の単語に類語が出現していればその出現順位を用いて平均逆順位 Mean Reciprocal Rank (MRR) を式 (1) で計算する。この指標で各種の学習データの有効性評価を行う。

$$\mathrm{MRR} = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{rank_k} \tag{1}$$

ただし、N は用語の数で、$rank_k$ は正しい類語が出現した順位である。今回の実験では N は 20 である。また、$rank_k$ は、各用語について出力された類似度上位 5 個の単語の中に正しい類語が出現していればその出現順位である。なお、上位 5 件に正しい類語が出現しなかった場合は $rank_k$ を ∞、すなわちその逆数を 0 として計算する。

4.2 結果

まず、それぞれのコーパスで学習させる際の、データの提示順番についての評価実験を行った。その結果、用語ごとのデータ（その用語を含む N グラムまたは文）の順で学習させるより、すべての用語のデータをランダムな順で学習させた方が、MRR の値が高いことが分かった。表 3 はその一例として、Google の N グラムデータを学習データとした場合の実験結果を示す。以上の結果を踏まえ以降に示す実験結果はすべてデータをランダムな順で学習させた場合のものに限定する。

表 3: データを用語順とランダム順で学習させた場合の MRR 値

	用語順	ランダム順
2-gram	0.262	0.177
3-gram	0.225	0.647
4-gram	0.275	0.850
5-gram	0.129	0.775
6-gram	0.185	0.704
7-gram	0.025	0.175

表 4 は Google の N グラムデータと Susumu の N グラムデータの各種 N グラムを単独に学習させた場合と、Web データとこれらの併合データを学

[1] https://code.google.com/p/word2vec/

[2] 実際はここまでの結果を word2vec がまとめて出力してくれる。

習させた場合の MRR 値を示す。表 5 は、Google の N グラムデータと Susumu の N グラムデータの各種 N グラムを混合に学習させた場合と Web データを学習させた場合の MRR 値を示す。たとえば 2 ～ 9-gram は 2-gram から 9-gram までのデータを意味している。なお、2 章で述べたように、コーパスの元データに対して必要に応じて分割処理を施している。その結果、数は極めて少数であるが、8-gram, 9-gram のデータが存在している。そのために表 5 に示しているように 9-gram までのデータを学習に用いた。

表 4: 4 種類のコーパス（N グラムデータは各種 N グラム単独）を用いた場合の MRR 値

	Google	Susumu	Web	併合
2-gram	0.177	0.017	0.400	0.475
3-gram	0.647	0.127		
4-gram	0.850	0.633		
5-gram	0.775	0.613		
6-gram	0.704	0.617		
7-gram	0.175	0.092		

表 5: 各種 N グラム混合を用いた場合の MRR 値

	Google	Susumu
2 ～ 9-gram	0.487	0.625
3 ～ 9-gram	0.471	0.560
4 ～ 9-gram	0.429	0.500
5 ～ 9-gram	0.467	0.504
6 ～ 9-gram	0.692	0.452
7 ～ 9-gram	0.108	0.033
8 ～ 9-gram	0.083	0.060

これらの結果からまず、Google と Susumu の N グラムデータのどちらにおいても、4,5,6-gram をそれぞれ単独に用いた場合の MRR 値は Web データまたは併合データを用いる場合のそれらよりはるかに高いことがわかった。また、Google と Susumu の N グラムデータのどちらにおいても、N グラムの混合データの 2 ～ 9-gram, 3 ～ 9-gram, . . . , 6 ～ 9-gram を用いた場合の MRR 値は Web データや併合データを用いる場合のそれらより高かった。しかしながら予想に反し、N グラムの混合使用の性能が単独使用より劣っている。これについての再確認や原因究明は今後の課題の一つである。また、7-gram の単独使用の MRR 値も低かった。これは 8-gram, 9-gram と同様、データの数が少ないことが原因と考えられる。

5 おわりに

word2vec で精度の高い意味ベクトルを獲得するためには学習データとして大量な文章（または文）が必要とされてきた。しかしながら、このようなデータを大規模に収集するのは必ずしも容易ではない。一方、Web データに基づく大規模な単語 N グラムは工藤らや矢田により作成され公開されている。word2vec に文を与えることは文脈情報を与えることとある意味で等価であることから、学習データとしては文章または文データの代わりに単語 N グラムデータを用いることが考えられる。その考え方を確かめるために、本研究では 10 個の IT 用語とそれらの類語の計 20 個の単語について、Web 検索で収集した文データ、2 種類の単語 N グラムデータ、さらにはこれら三つを併合したデータをそれぞれ学習データとして用いた場合の評価実験を行った。その結果、単語 N グラムデータの有効性が確認できた。

謝辞

本研究は科研費 (25330368) の助成を受けたものである。

参考文献

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *Proceedings of Workshop at ICLR*.

[2] F. Ginter and J. Kanerva. 2014. Fast Training of word2vec Representations Using N-gram Corpora. *SLTC*.

[3] 工藤, 賀沢. 2007. Web 日本語 N グラム第 1 版. 言語資源協会.

[4] 矢田. 2010. N-gram コーパス - 日本語ウェブコーパス 2010. http://s-yata.jp/corpus/nwc2010/ngrams/.

[5] 西尾. 2014. word2vec による自然言語処理. オライリー・ジャパン.