1D2-4in2

語順を基にした分散的意味表現による観光文書表現の検証

Validating Representation of Tourism Documents by Distributed Semantic Representation Based on Word Order

納村聡仁 Akinori Osamura　沼尾正行 Masayuki Numao　福井健一 Ken-ichi Fukui

大阪大学　産業科学研究所
The Institute of Scientific and Industrial Research, Osaka University

In this article, we propose an application of Doc2Vec acquiring vector representations of documents based on the word order to a tourism data. Conventional document representation based on term frequency cannot represent concept relationship among words. In this experiment, we utilize tourism curation data and acquire vector representation about the contents of each article by Doc2Vec. Then, we extract topics by a clustering and analyze validity of the vector representation by comparison with tags.

1. 序論

1.1 研究背景

現在、観光地の情報収集、予約、現地での交通など、モバイル端末により自由に組み替えることが可能になり、団体から個人へとその観光形態は変化した。この団体から個人への観光形態の変化と共に、個人に即した情報提供の必要性が増してきた。つまり、過去の履歴から個人の特性に合わせて提示すべき情報を見つけてくる手法、レコメンデーションの研究が盛んになっている。レコメンデーションの研究は次の 2 つに分類できる。利用ユーザと類似度が高いユーザの情報を基に推薦する協調フィルタリング、そしてユーザとコンテンツのプロファイルを構築し，それらをマッチングすることで推薦するコンテンツベースレコメンデーション [1] である。

しかし、それぞれの推薦方法には問題がある。協調フィルタリングは、あくまで趣味思考が似ていると思われる他ユーザの情報のみを参考にしているので、新たにシステムを利用し始めた利用者、そして推薦対象として新たにシステムに導入されたアイテムにおいて情報量の不足による推薦が困難となる cold-start 問題がある。一方、コンテンツベースレコメンデーションは、文章中の単語の頻度、重みを考慮した指標である Term Frequency Inverse Document Frequency（TFIDF)[2] による、コンテンツの特徴解析、ユーザ属性や過去の履歴といったユーザ情報の抽出を行い推薦を行っている。しかしこの手法では、単語の頻度情報を基に解析しているので、類義語や同義語を全く別のものとして扱ってしまい、例えば、家とマンションを類義語として、車と自動車を同義語として扱わずに文章の分析を行ってしまう。そのため、ユーザ情報で車が好きと分かっていても、自動車の記事を推薦することはできない。

このような、語順に基づく単語間の概念関係の獲得に対して様々な研究がなされているのだが、例えば、前もって、単語それぞれに対する概念関係を獲得し、文書の特徴量抽出を行う方法である。DBpedia[*1] などのような構造化されたテキストデータならば、データ間の関係がリンクづけられているので、パターンもしくはルールを用いて知識獲得を行うことができる。そして、構造化されていないテキストデータならば、例えば「男が箸でラーメンをすすっている。」と「男がうどんを箸ですすっている。」の文章における「うどん」と周りの単語との共起関係、「ラーメン」と周りの単語との共起関係を基にクラスタリングを行い、「ラーメン」と「うどん」との類義関係を獲得する。しかし、こういった概念関係は無数にあり、データベースとして巨大なものが必要となる。

また、語順を考慮しテキストの特徴を抽出する N-gram[3] がある。N-gram は文書中の出現単語から、N個の単語を 1 つの組み合わせとするトークンをすべての組み合わせにおいて生成する。これにより語順は考慮するもののデータが大きくなりすぎ、推薦システムには適さない。他にも、文書の解析手法として、tfidf によるコーパスの類似関係などを基に次元圧縮を行い、特徴ベクトルを獲得する Latent Semantic Analysis(LSA)、文書は確率生成されるモデルであることに基づき、文書やそこに出現する語彙の潜在トピックを求める手法である Latent Dirichlet Allocation(LDA) などがある。LSA は、tfidf では捉えられなかった単語の類似関係などを、各単語ベクトルの類似性などを基に次元圧縮を行うため各単語間の類似関係などを獲得できた。しかし、LSA の生成方法はあくまで単語間の頻度情報、文書全体の共起性によるため、どの単語と関係が強いかは求まるが、単語周辺の情報を利用していないためより実用的などういう使われ方をするのかを学習できない。また、LDA は、文書、単語それぞれにトピックを持つため、トピックという一つ上の階層を通して、類似関係などを見ることができる。しかし、あくまで単語の出現確率分布により類似関係を計算するため、人間がイメージする、単語、文書の特徴をとらえきれておらず、単語同士、文書同士の演算を行うには不十分と考える。

1.2 研究目的

本論文ではこのコンテンツベースレコメンデーションにおける問題を解決するために、文章の表現ベクトルを獲得する Doc2Vec[4] に注目した。この手法は、語順を基に数個の単語からその次の単語を予測できるような表現ベクトルを獲得する Word2Vec[5] を文章に拡張したものである。この手法により、概念階層 (タクソノミー)、そして類義語や同義語の関係性をベクトル空間内に保存することができるため、従来法では、獲得することができなかった概念も含めた分析を行うことが期待できる。つまり、似たような単語は似たような語順で

福井健一, 大阪大学, 産業科学研究所沼尾研究室,
〒 567-0047 大阪府茨木市美穂ヶ丘 8-1,
Tel: 06-6879-8426，Fax: 06-6879-8428,
E-mail: fukui@ai.sanken.osaka-u.ac.jp

*1 http://wiki.dbpedia.org/

扱われることから、先ほどの車と自動車の関係性などを獲得できる。また、獲得した表現ベクトルにより、単語同士のみでなく、ニュース記事同士の類似度、本同士の類似度、人のプロフィールと本の類似度なども算出することができる。これらの手法は、近年自然言語処理の分野で注目されており、現在のGoogle の検索エンジンへの利用や、推薦システムへ利用した研究なども見られるようになってきた。しかし、観光推薦において、応用事例はまだ見られない。そのため、例えばコンテンツの特徴ベクトル、そして SNS データのようなユーザを特徴づけるテキストの表現ベクトルの類似度を利用した推薦などが考えられ、これらの手法を利用すれば、観光推薦の発展、しいては観光産業の発展につながることが期待される。

本論文ではまず、この観光推薦の発展のために、Doc2ec により観光文書の表現ベクトルを学習し、ベクトルが記事の内容を適切に反映しているか、つまり、似た記事のベクトル同士は類似度が高くなっているかを検証した。本実験ではデータとして、あらかじめ人手により記事の属性を表すタグを付与された観光キュレーションデータを用い、Doc2Vec により各記事の表現ベクトルを獲得した。このとき、似た記事同士は、似たベクトルを持ちクラスタを作っているはずである。そこで K-means 法によりクラスタリングを行い、同じタグを持つ記事同士が同じクラスタ内に属することを検証した。また、クラスタ内の記事の内容を実際に見比べることにより、実際に似た記事同士でクラスタを作っているかを定性的にも評価した。

2. 分散的意味表現法

2.1 Word2Vec：単語の分散的意味表現法

本節では、Word2Vec、つまり単語の分散的意味表現法の概念を紹介する。Word2Vec の単語ベクトルの学習フレームワークは図 1 である。

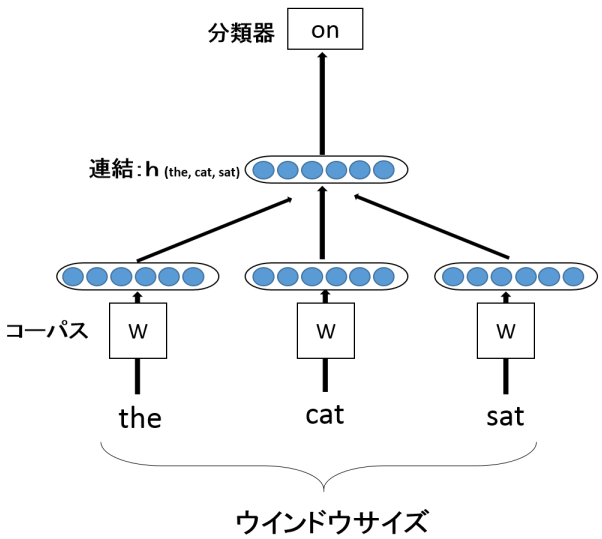

図 1: 単語ベクトルの学習フレームワーク

図 1 は、4 番目の単語 (‘on’) を予測するために、(‘the’, ‘cat’, ‘sat’) の 3 単語の前後関係を用いていることを表す。また、入力単語は行列 W のコーパスの中に格納されている。このようにして、Word2Vec は文章の一部の単語を基に、次に来る単語を予測する。このフレームワークにおいて、全ての単語は個々のベクトルを持ち、コーパス W の中に行列として格納されている。

コーパスは、単語とベクトルが一対一対応となっているため、単語を指定することで、その単語ベクトルを取り出すことができる。各単語ベクトルは、最初ランダムに決められている。形式的に説明すると、まず目的の単語を予測するために使われる単語の数であるウインドウサイズ分の一連の学習用単語列を用意する。それぞれのベクトルを連結し、次式を最大化するように単語ベクトルを修正していく。

$$\max_{w_1,...,w_T} \frac{1}{T} \sum_{t=k}^{T} \log p(w_t|w_{t-k}, ..., w_{t-1}) \tag{1}$$

w_tは t 番目の単語ベクトル、k はウインドウサイズ (学習の範囲)、T は文書中の出現する単語数、p は周辺単語の表現ベクトルの連結が、対象単語の表現ベクトルとなる尤度を表す。このとき、$p(w_t|w_{t-k}, ..., w_{t-1})$ で表される予測タスクは下式のように表される。

$$p(w_t|w_{t-k}, ..., w_{t-1}) = \frac{\exp(h \cdot w_t)}{\sum_{j=1}^{T} \exp(h \cdot w_j)} \tag{2}$$

ここで、h は、$w_{t-k}, ..., w_{t-1}$の連結ベクトルである。このとき、右辺はソフトマックス関数と呼ばれる正規化関数である。正規化関数とは、ここで言う、周辺単語を表す h と目的単語を表す w_tとの内積により求めた類似度を $[0, 1]$ に正規化する関数である。この類似度がより近くなるように、周辺単語のベクトルを修正していく。上記の学習を、ウインドウをずらし繰り返していくことで、Word2Vec は語順を基に各単語ベクトルを学習することができる。

この手法は、似ている単語は似ている語順で使われることを基に単語の意味表現を学習することで、従来の頻度ベースの手法では獲得できなかった類義語や同義語といった概念関係の獲得を可能にした。例えば、家とマンションとの類似関係、車と自動車との同義関係を獲得することができる。

2.2 Doc2Vec：文書の分散的意味表現法

本節では、Word2Vec の拡張版で、文書の特徴ベクトルを学習する Doc2Vec に関して紹介する。フレームワークは図 1 の入力にパラグラフベクトルを追加したものになる。つまり、文章毎に共有するパラグラフベクトルを用意し、単語ベクトルの学習の際共有して学習させることでパラグラフベクトルを獲得する。

このフレームワークにおいて、全てのパラグラフは個々のベクトルを持ち、コーパス D の中に行列として格納されている。コーパスは、パラグラフとベクトルが一対一対応となっている。各パラグラフベクトルは、最初ランダムに決められている。Word2Vec との違いとしては、式 (2) が下式のように、入力にパラグラフベクトル v_pが追加されたのみである。このように、文書中の単語ベクトルを学習する間、入力としてパラグラフベクトルを入れ続けることで、文書の特徴をベクトルとして表現できる。このパラグラフベクトルにより、テキスト間の類似度を計算することができるようになった。例えば、ニュース記事同士の類似度、本同士の類似度、人のプロフィールと記事の類似度も測定することができる。

今回、本手法を用いることで、コンテンツベースレコメンデーションの従来法である tfidf の課題であった、語順の問題に関して解消できると考えた。

3. 実験

Doc2Vec による観光文書の正確な特徴ベクトル獲得を検証する。まず、観光記事の特徴ベクトルを Doc2Vec により学習し、クラスタリングを行った。そして各記事が持つタグとの比較することによる定量的な検証、実際に各クラスタ内の記事を見ることによる定性的な評価を行いその有用性を検証した。

3.1 学習データ

実験において、図 2 のような「沖縄ホテルのここに泊まってよかったランキング 23 選」などが掲載されている観光キュレーションサイト Find Travel[*2] の観光記事約 5000 記事 (2015 年 11 月付け)、そして Wikipedia データ [*3] 約 100 万記事の 2 種類を用いて実験を行った。キュレーションサイトとは、キュレーターと呼ばれる人により、インターネット上の情報を収集しまとめ、新たな価値を持たせたサイトである。昨今、このキュレーションは注目され始め、グノシー、SmartNews といったニュースキュレーションアプリなどもローンチされている。今回用いた観光キュレーションサイト Find Travel は、キュレーターが、インターネット上の観光記事をまとめ、地域ごと、目的毎などによって分類分けを行ったサイトである。そのため、各記事には「ラーメン」、「京都」といった各記事の内容を表すタグを、1 記事あたり平均 3 個持っている。クラスタリングは高頻度なトピックしか評価できないので，マイナーな低頻度タグは評価の対象外とした。このとき、除いたタグのみを持つ記事の総数が、今回の Find Travel の総記事数の約 1%未満となるように、閾値を 40、つまり 40 記事以下についているタグを除いた。さらに今回、Wikipedia データを学習の際に利用することを提案した。観光記事が約 5000 記事と少ないため、十分な単語ベクトルの学習が行われない可能性を踏まえ、Wikipedia データを単語ベクトルの学習に用いた場合との比較を行った。

図 2: 沖縄ホテルのここに泊まってよかったランキング 23 選

3.2 学習

また、観光記事と Wikipedia データの両方を入力として、パラグラフベクトルの学習を行った。そして、各記事のパラグラフベクトルを K-means 法によりクラスタリングを行った。

3.3 定量評価

今回のデータは複数タグを持つため，Precision は必然的に小さい値にあるため Recall で比較した。各クラスタ内の記事のタグを基に、同じタグを持つ記事は密集していることを表す Recall を求め、各クラス t について F 値が最大となるクラスタの平均を取り、Doc2Vec により各記事の特徴をベクトル化できているかを評価した。今回の評価指標とした Recall は下式のようである。

$$Recall(t, C_i) = \frac{N_{t,i}}{N_t} \tag{3}$$

ここで、t は (正解) タグ番号, C_iはクラスタ, $N_{t,i}$は第 i クラスタに含まれる第 t タグの記事数を, N_tは第 t タグを持つ記事数を表すとする。

パラグラフベクトルの次元数、K-means 法によるクラスタリングのクラスタ数を変化させた。結果が、図 3〜4 である。

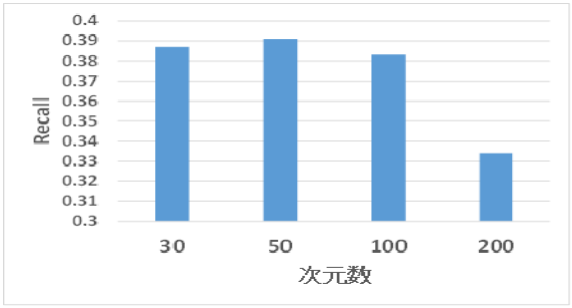

図 3: 観光文書 Find Travel のみを用いて次元数を変化させた結果 (クラスタ数:50)

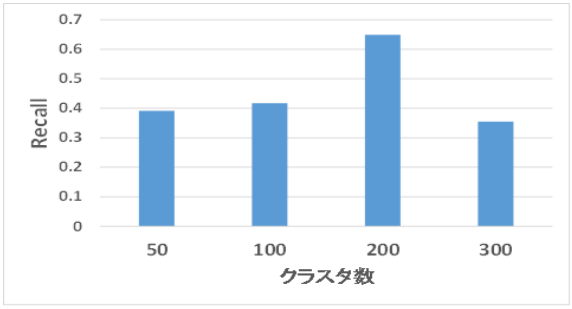

図 4: 観光文書 Find Travel のみを用いてクラスタ数を変化させた結果 (次元数:50)

これより、Find Travel のみを用いた場合、次元数 200 次元、クラスタ数 50 における Recall 0.677 が最適値であるとわかった。この結果は、あるタグを持つ記事約 100 記事をランダムにクラスタに分類したときの Recall の期待値 0.151 と比べると高い値になっている。また、Wikipedia データも用いた学習による最適値は 0.636 となり Find Travel のみを用いた場合よりも小さくなった。

4. 定性評価

次に、クラスタ内の記事の内容を実際に見比べることにより、実際に似た記事同士でクラスタを作っているかを定性的にも評価した。その結果が、表 1 である。id はランダムに選んだクラスタを、クラスタ内タグ TOP3 は、クラスタ内のタグの頻度での TOP3 を、内容は、クラスタ内で見られた記事を基に各クラスタにおける内容をまとめたものである。

*2 http://find-travel.jp/
*3 https://dumps.wikimedia.org/jawiki/latest/jawiki-latest-pages-articles.xml.bz2

表 1: クラスタ内の記事

id	クラスタ内タグ TOP 3	トピック
c1	1:デート,2:クリスマス,3; 夜景	オススメデートスポット
c2	1:ラーメン,2:グルメ,3; 庭園	オススメラーメン、グルメスポット
c3	1:高級ホテル,2:旅館,3; 観光スポット	オススメ高級宿泊スポット

c1 は、「旬のイチゴが食べ放題！千葉・房総のおすすめイチゴ狩り施設 5 選」などの記事が含まれており，また頻度上位のタグはデート、クリスマス、夜景であった。このことから c1 はオススメデートスポットに関するクラスタであると言える。c2 は、「行徳でラーメン好きが集まるお店 7 選。」や「鹿児島グルメおすすめ 24 選！郷土料理が食べられるお店や人気店などご紹介」などの記事が含まれ、頻度上位のタグは、ラーメン、グルメ、庭園であった。これより、c2 はオススメラーメン、グルメスポットに関するクラスタであると言える。c3 は、「横浜の高級ホテル 20 選。夜景を独り占めできるラグジュアリーな空間。」や「幕末維新の歴史がある宇部！泊まるなら超おすすめなホテルはここ！」などの記事が含まれ、頻度上位のタグは、高級ホテル、旅館、観光スポットであった。これより、c3 はオススメ高級宿泊スポットに関するクラスタであると言える。以上より，クラスタに分類された記事の内容から，意味のあるクラスタが抽出できていることが確認できた。

5. 考察

結果として、Recall は 0.677 となり、あるタグを持つ記事約 100 記事をランダムにクラスタに分類したときの Recall の期待値 0.151 と比べると高い値になっている。1 に近づくほど同じタグが密集していることを表しているのだが、この値になった原因として主に、3 つの要因が考えられる。ひとつ目が、データの質の問題。キュレーションサイトの記事なので、記事はそれぞれ各コンテンツ毎にまとめられているが、まとめられているからこそ、コンテンツに関するサイトの営業時間、アクセスといった情報も記事には含まれていた。このことから、コンテンツの表現に難しさがあったのではないかと考える。そして、データ量の問題。Doc2Vec の十分な学習のため、経験的に数万〜数百万記事が必要となってくる。今回、5000 記事を用いて実験を行ったためデータ量が足りなく、単語ベクトル、パラグラフベクトルの学習が十分ではなかったのではないかと考えられる。最後に、タグの問題。今回、5000 記事に対して約 100 種類のタグがあり、一つの記事に複数のタグが付いていた。つまり、各記事に掲載されている様々なサイトの内容に関係するタグがつけられているのだが、例えば石川県の白山の記事において石川県のタグが付いているがテキストとして石川県のことはアクセスのみに掲載されている、などのように、必ずしも全てのタグの情報が各記事のテキスト情報として重要視されなかったのではないかと考える。また、各タグの頻度はバラバラで、多いタグで約 450 記事に付与するタグもあった。タグ頻度にバラつきがあることで、タグの出現回数に偏りが出てしまい recall に影響が出るのではないかと考える。TOP10 のタグを表 2 にて示す。

今回、データ量が十分ではないという問題に対して、Wikipedia データを同時に学習させることでパラグラフベクトルの学習の手助けになるのではないかと考えたが、結果として効果は見られなかった。これは、単語ベクトルとしての学習は行われてはいたが、観光に関係のない情報を基に学習してしまい結果として精度の向上には繋がらなかったのではないかと推察している。そのため、例えば別の観光記事を学習データとして追加すれば、精度の向上が見られる可能性は十分にある。

表 2: タグ頻度 TOP10

順位	頻度	タグ
1	454	ご当地グルメ
2	363	寺・神社
3	354	観光
4	307	グルメ
5	298	デート
6	287	ホテル
7	283	ランチ
8	263	カフェ・喫茶店
9	251	人気観光スポット
10	239	お散歩・街歩き

また、定性評価を行うことにより、いちご狩りの記事とクリスマス関連の記事が同じクラスタに位置するなど、興味深い結果も得られた。Doc2Vec という教師なし学習により、自動でデート関連の記事をまとめることができたことがわかる。

6. 結論

本研究では、将来の観光に関するコンテンツベースレコメンデーションの発展のため、従来法である tfidf の課題である、語順を考慮していないため単語の概念関係を考慮していない点を解決する Doc2Vec の利用を提案し、実際に観光記事を用いてその有用性を検証した。

その結果、同じタグの密集度合いを示す recall は、ランダムにクラスタに分類し得られた結果よりも高い値となり、また、実際に記事を見てみると, デート関連の記事がまとまっているなどが見られた。このことから、単語の概念を考慮した観光記事推薦への足掛かりが得られた。

今後の課題として、まずはマルチクラスの評価基準の導入が考えられる。今回はあくまで、同じタグがどれだけ密集しているのかのみを検証しており、将来的に推薦システムに導入するためには、各タグ毎の分離度も測定可能な、より厳密な評価基準の導入が必要となってくる。また、より Doc2Vec の有用性を検証するため、tfidf、N-gram、LSA、LDA などとの比較が必要となってくる。

参考文献

[1] P. Resnick and H. R. Varian. Recommender systems: Communications of the ACM, Vol. 40, No. 3, pp. 56-58, (1997).

[2] 徳永健伸：情報検索と言語処理, 東京大学出版会, (1999).

[3] 藤村滋, 豊田正史, 喜連川優：文章構造を考慮した自由回答意見からの要望抽出, 言語処理学会第 12 回年次大会, (2006).

[4] Quoc V, Le and Tomas Mikolov. Distributed Representations of Sentences and Documents, In Proceedings of The 31st International Conference on Machine Learning, pp. 1188-1196, (2014).

[5] Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the International Conference on Learning Representations (ICLR 2013), pp. 1-12, (2013).