連載：情報を計測し，法則化する～今に活かせる計量情報学の経験則

テキスト間の類似度の測定

難波　英嗣*

キーワード：コサイン類似度，Jaccard 係数，Dice 係数，Simpson 係数，ニューラルネットワーク

1. はじめに

あるテキストと別のテキストの内容がどのくらい似ているのか（類似度）を測定することは，自然言語処理分野における重要な課題である。この類似度を測る尺度（類似度尺度）はこれまで数多く提案され，また様々な場面で利用されてきた。その応用例として情報検索，テキスト分類，文書クラスタリングがある。最近では，機械翻訳システム[1]やテキスト要約システム[2]の出力を評価する際に，正解テキストとどのくらい似ているのかを測るのにもこの技術が使われている。

本稿では，基本的なものから最新のものまで様々なテキスト間の類似度尺度を紹介する。なお，テキスト間の類似度を測定する技術には，テキスト間の引用等の関係を利用する方法とテキスト中の単語を利用する方法との 2 種類ある。前者については，本連載における伊神氏の記事を参照されたい[3]。本稿では，後者に焦点を当てて説明する。

2. 基本的なテキスト間の類似度尺度

2.1　集合間の類似度

テキスト A とテキスト B に含まれる単語を，集合 A および集合 B の要素と考えた時，Jaccard 係数，Dice 係数，Simpson 係数といった集合間の類似度尺度を用いて，テキスト間の類似度を測ることができる。テキスト A およびテキスト B に含まれる単語を，それぞれ A＝{x, y, v, w} および B＝{x, y, z}とした時，Jaccard 係数，Dice 係数，Simpson 係数を用いたテキスト A とテキスト B の類似度は以下のように計算される。

Jaccard 係数

$$\frac{|A \cap B|}{|A \cup B|} = \frac{2}{5} = 0.4$$

Dice 係数

$$\frac{2|A \cap B|}{|A| + |B|} = \frac{2 \cdot 2}{7} = 0.57$$

Simpson 係数

$$\frac{|A \cap B|}{\min(|A|, |B|)} = \frac{2}{3} = 0.67$$

2.2　コサイン類似度

2.1 節で述べた尺度は，単純で分かりやすい反面，テキスト中でそれほど重要でない単語であっても非常に重要な単語であっても，すべて等しく扱っているため，適切な類似度が計算できないという問題がある。この問題を解決するための尺度である，テキスト中の単語の重要度を考慮したコサイン類似度をここでは紹介する。

まず，テキスト中の単語の重要度を測る手法について述べる。その基本的な統計量として，tf–idf が広く知られている。テキスト A に出現する単語 t の tf–idf の値は，以下に示す 2 つの統計量 tf(term frequency)[4] と idf(inverse document frequency)[5] を乗算することで得られる。

- tf：テキスト A 内における単語 t の頻度。「テキスト中で何度も繰り返し言及される概念は重要な概念である」という仮定に基づく。
- idf：N 件のテキストが存在し，この中で単語 t が出現するテキスト数が df(t)である時，1+log(N/df(t))の値。どのテキストにも出現するような単語 t は df(t)の値が N に近づくため，idf は 0 に近い値になる。逆に，特定の文書にしか出現しない単語 t は，df(t)の値が 1 に近づくため，idf は大きな値になる。

この tf–idf を用いることで，ひとつのテキストから，テキスト中の各単語とその重要度の対の集合が得られる。

次に，コサイン類似度について述べる。各単語を軸，その単語の重要度をその軸の座標と考えると，テキストはひとつのベクトルとみなすことができる。今，テキスト A とテキスト B のベクトルを，それぞれ $\vec{a}$ と $\vec{b}$，これらのベクトルの間の角度を θ としたとき，内積の定義式から以下の式が導出できる。

＊なんば　ひでつぐ　中央大学理工学部
〒112–8551　東京都文京区春日 1–13–27
E-mail: nanba@kc.chuo-u.ac.jp
https://orcid.org/0000-0001-7191-9856

（原稿受領　2020.5.11）

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}||\vec{b}|}$$

テキストAとテキストBの内容が似ている時，同じ単語が近い確率で各テキスト中に出現すると考えられる。この時，$\vec{a}$と$\vec{b}$はベクトルの向きが非常に近いため，θの値は0に，その結果$\cos\theta$は1に近づく。もし，テキストAとテキストBの間に，共通する単語が1語もない場合，$\vec{a}$と$\vec{b}$＝0となるため，$\cos\theta$は0となる。この$\cos\theta$をコサイン類似度と呼ぶ。

テキストAとテキストBのベクトルが以下のように与えられているとする。

	単語x	単語y	単語z
$\vec{a}$	2	4	3
$\vec{b}$	1	1	0

この時，コサイン類似度は以下のようになる。

$$\cos\theta = \frac{2\times1+4\times1}{\sqrt{2^2+4^2+3^2}\sqrt{1^2+1^2}} = 0.79$$

3. ニューラルネットを用いたテキスト間の類似度尺度

3.1 基本的な類似度尺度の問題点

2節で述べた類似度尺度には，次のような問題点がある。以下(1)～(3)において，(1)と(2)はどちらも猫が好きであるが，(3)は猫が嫌いという内容であるため，意味的には(1)と(2)が近く，(3)だけが異なる。

(1) I love cats.
(2) I like cats.
(3) I hate cats.

しかし，loveとlikeは，意味は類似していても単語が異なる。このため，(1)と(2)は3単語中2単語が一致，(1)と(3)も3単語中2単語が一致し，(1)と(2)，(1)と(3)の間のコサイン類似度は，いずれも0.67となる。このように，単語が異なっていても意味は類似している場合は，そうでない場合と区別できるような類似度尺度が提案されている。3.2節では，単語の意味を表現する手法について述べ，3.3節でこの手法を用いた類似度尺度を紹介する。

3.2 単語の分散表現

以下の2文を考えてみよう。(a)にはいずれも同じ文字列が当てはまる。

(文1) 昨日は＿＿＿(a)＿＿＿でパエリアを食べた。
(文2) ＿＿＿(a)＿＿＿は安いし雰囲気もいいね。

文1に「パエリア」という単語があることから，(a)はスペイン料理が食べられる場所であることが推測できる。文2からは，(a)が個人宅ではなく，レストランであろうことが推測できる。ここでは2文だけしか例に用いていないが，もし非常に大量のテキストがあり，この2文以外にも(a)が様々な文脈で出現する例文がたくさん得られれば，かなり正確に(a)を言い当てることができるかもしれない。このように，ある単語は，その周辺にどのような単語がどのような頻度で出現するのかという情報から，ある程度意味を表現することができる。また，テキスト集合中のすべての単語について，それらの周辺単語の情報を収集すれば，周辺単語の類似性によって，単語の意味を表すことができる。これは，1954年にHarris[6]が提示した分布仮説と呼ばれているものである。その後しばらくは，計算機の処理能力や計算機が扱えるデータの問題などから実証するまでには至らなかったが，1990年代後半になって，この仮説が正しいことがLin[7]やLee[8]によって確認された。2013年には，Mikolov[9]がニューラルネットを用いたWord2Vecという手法を開発し，これが深層学習を用いた自然言語処理の基礎技術として広く使われるようになった。Word2Vecでは，ある単語の周辺単語を用いて，その単語の意味を表現しているという点ではHarrisの分布仮説に従ったものであるが，ニューラルネットを使って，各単語の意味を数百次元のベクトルで表現することで，loveとlikeの意味的な類似性を2.2節で紹介したコサイン距離を用いて計算することを可能にしただけでなく，以下のような意味的な演算も出来るようになったことにより，非常に注目を集めた。

- king−man+woman＝queen（kingのベクトルからmanのベクトルを引き，womanのベクトルを加えて得られたベクトルと最も類似する単語としてqueenが得られる）
- 1/2(good+best)＝better（goodとbestのベクトルを加え，2で割って得られたベクトルと最も類似する単語としてbetterが得られる）

また，play−study+studied＝playedのような現在形から過去形への変換も可能であることから，単語のベクトルには意味だけでなく文法的な情報も含まれていると推測される。

3.3 単語の分散表現を用いたテキスト間の類似度尺度

前節で述べたWord2Vecを用いたテキスト間の類似度尺度が数多く提案されている。そのうちのひとつがSCDV[10]である。これは，Word2Vecで計算された単語のベクトル空間に対してGMM（Gaussian Mixture Models）でクラスタリングを行うことで，意味が類似した単語を統合し，2.2節で述べたtf−idfを考慮して，テキスト中のすべての単語のベクトルを統合してひとつのベクトルを得る。その後，ベクトル（テキスト）間の類似度を，コサイン距離で測る。

Paragraph Vector[11] は，Word2Vec を考案した Mikolov らによって，単語の分散表現をテキストの分散表現に変換する手法である。この手法でも，厳密に同じではないが，tf–idf に相当するような情報を単語の分散表現と組み合わせている。Paragraph Vector には，Doc2Vec[12] や sent2vec[13] といった名称で，いくつかの実装が公開されている。ただし，Paragraph Vector の計算には非常に膨大な時間がかかるので，あまり大規模なテキスト集合には適用できないという問題がある。

この他，2018 年には SWEM[14] という，単純にテキスト中の全単語のベクトルを平均したり，ベクトルの各要素の最大値のみ抽出したりするといった複数の手法が提案されている。テキスト中の全単語の情報を使うので，tf に相当する情報は使われているが，idf に相当する情報は考慮されていない。非常に高速に動作するため，多くの論文で比較手法として使われている。

近年では，Word2Vec よりもさらに性能の良い BERT[15] と呼ばれるモデルが提案されており，BERT を用いたテキスト間の類似度尺度 BERTScore[16] も提案されている。これは，BERT を用いた単語ベクトルと idf を組み合わせた尺度である。

4. おわりに

本稿では，テキスト間の類似度を測る複数の尺度を紹介した。テキスト間の類似度を計算することは古典的な研究課題であるが，非常に多くの場面で利用されること，その場面が現在益々増えていること，さらに利用可能な技術が次々と開発されていることから，自然言語処理分野では今なお活発に研究が行われている。その成果は論文と同時にプログラムも公開されているので，興味のある読者は探して使ってみることをお勧めしたい。

註・参考文献

1) Papineni, K. et al. BLEU: a method for automatic evaluation of machine translation. IBM Research Report, RC22176 (W0109-022), 2001.
2) Lin, C.W. ROUGE: a package for automatic evaluation of summaries. Proceedings of the ACL-04 Workshop Text Summarization Branches Out, 2004, p.74–81.
3) 伊神正貫．文献の関係性の分析：書誌結合，引用分析，自然言語処理．情報の科学と技術．2020, vol.70, no.4, p.208-210.
4) Luhn, H.P. A statistical approach to mechanized encoding and searching of literary information, IBM Journal of Research and Development, 1957, vol.1, issue 4, p.309-317.
5) Spärck Jones, K. A statistical interpretation of term specificity and its application in retrieval, Journal of Documentation, 1972, vol.28, no.1, p.11-21.
6) Harris, Z. Distributional structure. Word, vol.10, no.23, p.146-162.
7) Lin, D. Automatic retrieval and clustering of similar words. Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics, 1998, p.768-774.
8) Lee, L. Measures of distributional similarity, Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, 1999, p.25-32.
9) Mikolov, T. et al. Distributed representations of words and phrases and their compositionality. Proceedings of the 26th Neural Information Processing Systems, NIPS 2013, 2013, p.3111-3119.
10) Mekala, D. et al. SCDV: sparse composite document vectors using soft clustering over distributional representations, Proceedings of EMNLP 2017, 2017, p.659-669.
11) Mikolov, T.; Le, Q. Distributed representations of sentences and documents. Proceedings of the 31st International Conference on Machine Learning, ICML 2014, 2014, p.1188-1196.
12) Doc2vec: https://radimrehurek.com/gensim/models/doc2vec.html (accessed 2020-05-11)
13) Sent2vec: https://github.com/epfml/sent2vec (accessed 2020-05-11)
14) Shen, D. et al. Baseline needs more love: on simple word-embedding-based models and associated pooling mechanisms. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, p.440-450.
15) Devlin, J. et al. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2019.
16) Zhang, T. et al. BERTScore: Evaluating text generation with BERT. Proceedings of the 8th International Conference on Learning Representations, 2020.

Series: Measuring and formulating information today — Empirical rules of informetrics: Measurement of similarity between texts. Hidetsugu NANBA (Faculty of Science and Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551 JAPAN)

Keywords: cosine similarity / Jaccard coefficient / Dice coefficient / Simpson coefficient / neural network