Yahoo!知恵袋データセットからの流行語抽出

堺, 雄之介
九州大学工学部電気情報工学科

伊東, 栄典
九州大学情報基盤研究開発センター

出版情報：電気・情報関係学会九州支部連合大会講演論文集. 72 (10-2P-08), pp.586-587, 2019-09-28.
電気・情報関係学会九州支部
バージョン：

Yahoo!知恵袋データセットからの流行語抽出

堺 雄之介 *　伊東 栄典 **
(九州大学 *工学部電気情報工学科 **情報基盤研究開発センター)
* y.sakai.a96@s.kyushu-u.ac.jp, ** ito.eisuke.523@m.kyushu-u.ac.jp

1　はじめに

大衆の動向が把握できれば商機につながる。そのため，Google トレンドでは利用者が入力する検索語の傾向が提供されている。また Twiiter に投稿したテキストを分析による流行語分析も行われている。ヤフー社が提供する Yahoo!知恵袋では，身近な話題から大きな話題まで自由に質問と回答が行われている。本研究では Yahoo!知恵袋の質問文を対象に，分野毎かつ月毎の流行語の分析を行う。単純な単語の出現頻度による分析に加え，単語の分散表現による類似語抽出からの類似単語集約によるトレンド推移も分析した。

2　Yahoo!データセット

Yahoo!データセット [1] は国立情報学研究所がヤフー株式会社から提供を受けて研究者に提供しているデータセットである。このデータセットには「Yahoo!知恵袋」において解決済みとなった質問と回答を，ヤフー株式会社が「Yahoo!知恵袋」のデータベースから抽出したものである。質問および回答に含まれるデータ項目は，質問・回答の ID，質問のカテゴリ，質問・回答のタイトルおよび本文，投稿および解決の日時，ベストアンサーフラグ，画像付きフラグ，ならびに投稿デバイスである。

第 1 版の提供データ期間は 2004 年 4 月 1 日から 2009 年 4 月 7 日である。質問件数と回答件数は表 1 に示す。本研究で対象とした質問データは全質問件数の内 6,476,939 件である。本研究で使用した質問文を含むデータのフォーマットを表 2 に示す。

表 1: 提供データの質問件数と回答件数

項目名	件数
質問件数	16,257,413
回答件数	50,053,894

表 2: Yahoo!データセット第 1 版のデータ項目

No.	項目名
1	質問番号
2	カテゴリ名
3	カテゴリパス
4	質問タイトル
5	質問本文
6	質問者 ID
7	付随回答の回答数
8	質問のステータス
9	質問投稿日
10	質問解決日
11	投票制になった日
12	役に立つ質問に選択されているかどうか
13	質問する際にかける知恵コイン
14	「BA にふさわしくない」に投票された数
15	総投票数
16	画像の枚数
17	モバイルフラグ
18	自動カテゴライズ使用可否
19	補足有無
20	お礼有無
21	補足内容
22	補足日付
23	お礼内容
24	お礼日付
25	お礼アイコン

3　データ処理と単語の分散表現取得

本研究では流行語抽出のために 2 つの方法を適用する。1 つ目は単語の出現頻度を数え上げる方法である。2 つ目は単語の出現頻度に加え，その単語と関連する単語の頻度も考慮する方法である。関連語を算出するために単語の分散表現を使う。データ処理の流れを以下と図 1 に示す。

1. Yahoo!知恵袋の「質問文」を形態素解析ツール Mecab [2] で分かち書き文に変換する。
   - 新語対応のため形態素解析に IPA-Neologd 辞書 [3] を用いる。
   - Mecab での解析の際，分かち書き文に残す品詞を制限する。流行語は名詞が多いため，今回は名詞のみに制限する。
2. 分かち書き文書群から単語の出現頻度 (TF) を得る。TF のカウントには scikit-learn[4] を用いる。
3. 分かち書き文書群をコーパスとして FastText [5] に入力し，単語の分散表現を得る。
   - 分散表現（ベクトル）の次元数は 300 次元とした。

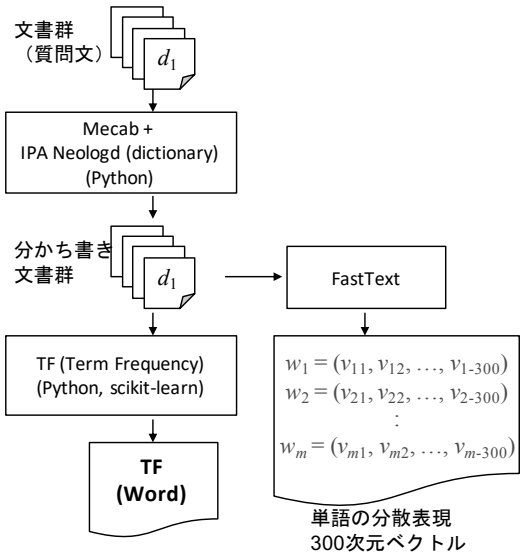

図 1: データ処理の流れ

4　出現頻度による流行抽出

本論文で検討した流行語の抽出方法を述べる。

4.1　方式 1:出現頻度のみ

方式 1 は単語の出現頻度のみで流行語を決めるもので，従来から用いられている素朴な方式である。質問文集合について，期間 p における単語 w の出現頻度 $tf(w,p)$ を求める。期間 p の出現頻度 $tf(w,p)$ が上位となる単語が，その期間の流行語である。

4.2　方式 2:類似単語の出現頻度を考慮

方式 2 として，類似単語の出現頻度も考慮する方法を提案する。1 つの物事を表す単語が 1 つしかない場合は少ない。省略語や類似する単語などで表現される場合が多い。

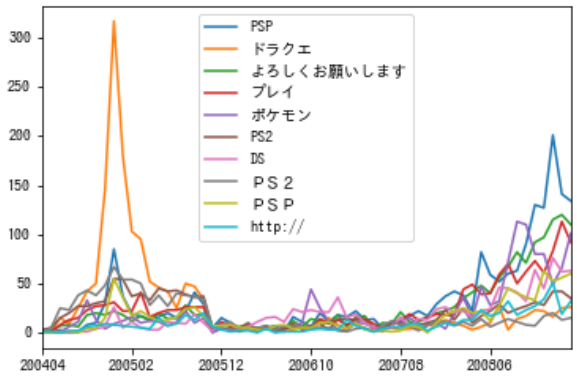

図 2: 出現頻度のみによる分析の結果

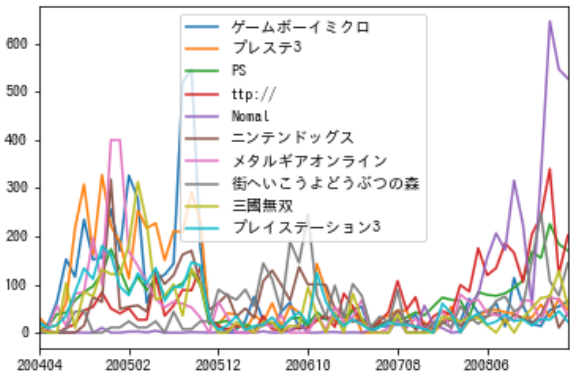

図 3: 類似した単語の出現頻度を考慮した分析の結果

例えば「オリンピック」と同様の単語に「Olynpic」や「五輪」がある。意味的に近い単語に「オリパラ」がある。「オリンピック」の頻度に，類似単語の「Olynpic・五輪・オリパラ」の頻度を加えることで,「オリンピック」の流行度をより良く計れるのではないかと考えた。

方式 2 では，期間 p における単語 w の出現頻度 $tf(w,p)$ に，w の類似語 t の値 $tf(t,p)$ にを加える。ただし w と t の類似度 $sim(w,t)$ を乗じて加える。これを $new\_tf(w,p)$ とする。$new\_tf(w,p)$ の算出方法を式 (1) に示す。式 (1) の T は，質問文に単語 w と共起出現する単語の集合である。

$$new\_tf(w,p) = tf(w,p) + \sum_{t \in T} sim(w,t) * tf(t,p) \ . \quad (1)$$

単語 w と t の類似度 $sim(w,t)$ は，fatstText が出力した単語の分散表現（300 次元ベクトル）のコサイン類似度とする。fastText が算出する単語の分散表現（ベクトル）では，意味的に近い単語は近い値のベクトルとなることが多い。十分な文章量を持つコーパスを与えれば近いベクトルが出力されると期待できる。式 (2) にコサイン類似度の計算式を示す。

$$sim(w,t) = \frac{\sum_i v_{w,i} \cdot v_{t,i}}{\sqrt{\sum_i v_{w,i}^2}\sqrt{\sum_i v_{t,i}^2}} \ . \quad (2)$$

5 実験と考察

最初の実験として，Yahoo!データセット [1] の中の「ゲーム」カテゴリの質問文に対して方式 1 と方式 2 を適用した。なお流行語の推移粒度の期間 p は 1 ヶ月ごとにした。

方式 1 のトレンドを図 2 に示す。どちらも全ての単語は表示できないため，全期間（2004 年 4 月 1 日〜2009 年 4 月 7 日）における出現頻度が上位の単語 10 個に限定している。また方式 2 のトレンドを図 3 に示す。こちらも全期間（2004 年 4 月 1 日〜2009 年 4 月 7 日）における new_tf 値が上位の単語 10 個に限定している。

図 2 を見ると，プレイステーションに関する単語が個別に上位になっている。ゲームでは「ドラクエ」と「ポケモン」が高い頻度になっている時期がある。形態素解析処理の不具合から「https://」や「よろしくお願いします」が名詞として認識されている。

一方，図 3 では「プレステ 3」と「プレイステーション 3」が個別に出現しているものの，類似語が個別に高頻度としてカウントされていない。図では表示されていないものの，30 位までの単語を見ると，当時人気になりつつあるゲーム「東方プロジェクト」に関する単語が集約されて上位に入っている。流行語分析には方式 2 が優れていると思われる。

6 おわりに

本研究では単語の分散表現を用いた類似語抽出を用い，類似単語も考慮した流行語の抽出方式を提案した。実際に Yahoo!知恵袋 (第 1 版) のゲームカテゴリにおける質問文を対象に提案方式を適用した。その結果，素朴な単語頻度による流行語抽出よりも，より意味を考慮した流行語抽出が出来た。

今後は他カテゴリへの適用と検証を行いたい。また最新の Yahoo!知恵袋データ (第 3 版) に適用したい。単語の分散表現を得るためのコーパスに知恵袋の質問文を用いた。Wikipedia 等の他のコーパスを用いた場合も比較したい。

参考文献

[1] NII and Yahoo! Japan: Yahoo!データセット（第 1 版），`https://www.nii.ac.jp/dsc/idr/yahoo/chiebkr3/Y_chiebukuro.html`.

[2] 工藤拓，山本薫，松本裕治:Conditional Random Fields を用いた日本語形態素解析，情報処理学会研究報告自然言語処理（NL），Vol. 2004, No. 47, pp. 89–96（オンライン），`https://ci.nii.ac.jp/naid/110002911717/` (2004).

[3] Toshinori, S.: Neologism dictionary based on the language resources on the Web for Mecab (2015).

[4] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E.: Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830 (2011).

[5] Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C. and Joulin, A.: Advances in Pre-Training Distributed Word Representations, *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)* (2018).