

卒業論文

Twitterを用いた新聞記事への  
自動ソーシャルアノテーションツールに関する研究

平成 26 年 2 月 7 日提出

指導教員

伊庭斉志 教授

ダヌシカ ボレガラ 講師

電子情報工学科

03-120443

村上 晋太郎

## 目 次

<b>1</b>	<b>序論</b>	<b>1</b>
1.1	背景	1
1.2	目的	2
<b>2</b>	<b>手法</b>	<b>3</b>
2.1	記事と関連するニュース記事の収集	3
2.2	内容語の抽出	4
2.2.1	tf-idf	4
2.3	関連度の計算	5
2.3.1	言い換えの関係の語の検出	6
2.3.2	互いに類似する語の検出	6
2.3.3	WordNet を用いた類似語の検出	6
2.3.4	Dekang Lin の手法の利用	7
2.3.5	Latent Space でのモデル化	8
2.4	ニュース断片と Twitter 上の投稿の結びつけ	9
2.4.1	安定結婚問題	9
2.4.2	一対多型安定結婚問題	12
2.5	形態素解析	14
<b>3</b>	<b>実験</b>	<b>15</b>
3.1	概要	15
3.2	実験条件	15
3.2.1	実験サンプル	15
3.2.2	$\kappa$ 評価尺度 shakudo	16
3.3	実験結果	18
3.3.1	内容語の抽出	18
3.3.2	アラインメント	18
<b>4</b>	<b>考察</b>	<b>19</b>
4.1	内容語の抽出	19
4.2	アラインメント	19
<b>5</b>	<b>今後の課題</b>	<b>20</b>
5.1	データ収集の自動化	20
5.2	内容語の抽出	20
5.3	言い換え語・類似語の検出	20
5.4	ユーザー評価の実施	20
<b>6</b>	<b>結論</b>	<b>20</b>

# 1 序論

## 1.1 背景

2000年代後半から、Facebook, Twitter, Google+などのソーシャルネットワーキングサービスが盛んに利用されるようになった。2012年には、Facebookのアクティブユーザ数は10億人<sup>1</sup>を突破し、Twitterでもアクティブユーザー数は1.4億人<sup>2</sup>となっている。Facebookでは一日当たり0.5ペタバイトのデータが増加しており、Twitterへの投稿数は一日当たり3.4億投稿となっている。このように、ソーシャルネットワーキングサービスは利用者数の面でも、データ量の面でも莫大な資源となっており、様々な利用が期待される。

一方、近年インターネット上で閲覧することが出来るニュース記事が増加している。日本では、朝日新聞社<sup>3</sup>、読売新聞社<sup>4</sup>、毎日新聞<sup>5</sup>等の大手新聞者がインターネット上でニュース記事を配信している。海外でも、Washington Post<sup>6</sup> Las Angeles Times<sup>7</sup>等、大手新聞社が同様のサービスを提供している。

ソーシャルネットワーキングサービスの書き込みの中には、このようなインターネット上のニュース記事に言及しているものが多数存在する。このような書き込みには、一般のインターネットユーザーの個人的な意見、見解が記されている。このようなインターネット上のニュース記事と、ソーシャルネットワーキングサービスへの書き込みを結びつけることで、報道機関のプロの書き手とソーシャルネットワーキングサービス上で活動する一般社会の書き手を結びつけることができるようになる。このようなツールが実現すれば、報道機関のプロの書き手は自分の書いた記事が一般社会にどのような影響を与えているのかについての情報を、より簡単に得る事が出来るようになる。一方、一般の書き手にとっては、ニュース記事に情報が付与されるようになり専門的な内容の記事を読みやすくなったり、ニュース記事からより有用な情報を引き出せるようになる。

また、ニュース記事を学術論文に、ソーシャルネットワーキングサービス上の投稿を学術論文に対するコメントに置き換えた場合も、同じ手法を用いることができ、その場合にも同じ効果が期待できるという点で、応用性も高いと考える。

以上のような背景から、本研究ではソーシャルネットワーキング上の情報とインターネット上のニュースを自然言語処理の手法を用いて結びつけることを目標とする。

---

<sup>1</sup><http://ja.wikipedia.org/wiki/Facebook>

<sup>2</sup><http://www.724685.com/twitter/tw13050310.htm>

<sup>3</sup><http://www.asahi.com>

<sup>4</sup><http://www.yomiuri.co.jp>

<sup>5</sup><http://mainichi.jp>

<sup>6</sup><http://www.washingtonpost.com>

<sup>7</sup><http://articles.latimes.com>

## 1.2 目的

本研究では、ソーシャルネットワーキングサービス「Twitter」上の投稿をインターネット上のニュース記事に自動付与(アノテーション)するツールを作成することを目的とする。本研究に置けるツールでは、ニュース記事単位を Twitter 上の投稿の付与の対象とするのではなく、ニュース記事の中の文単位を付与の対象とする(図1)。結果、ユーザはニュース中のどの部分に対して、Twitter 上でどのような反応が起こっているのかが分かるようになる。このようなツールが実現すれば、ニュース記者にとっては自分の書いた記事についてのユーザーの意見が簡単に見れるようになり、フィードバックをすぐに得る事ができるようになるという点で有用である。一方、読者にとっては、難しい内容のニュース記事でも他のユーザーによる情報付与があることにより、理解の助けとすることが出来る。Twitter 上の投稿を Web 上のニュース自体に結びつけるタスクは、先行研究で既に取り扱われているが、Twitter 上の投稿を Web 上のニュースの中の断片に結びつけるというタスクはまだ取り扱われていない。その点が、本研究の新規性である。

コーヒー1日4杯以上、死亡リスク高め 米研究チーム

毎日4杯以上のコーヒーを飲む55歳未満の人は、飲まない人に比べ、死亡率が高いとする疫学調査結果を、米サウスカロライナ大などが米医学誌に発表した。研究チームは「若い人はコーヒーを毎日3杯までに」と注意を呼びかけているが、コーヒーの功罪に結論が出るにはまだ時間がかかりそうだ。

チームが、米国の約4万4千人にコーヒーを飲む習慣を書面で尋ね、その後17年ほど死亡記録などを調べた。その結果、55歳未満に限ると週に28杯以上コーヒーを飲む人の死亡率は、男性では1.5倍、女性は2.1倍になっていた。55歳以上では変化はなかった。ただし今回の研究では、飲用習慣が変わる可能性や、いれ方によって成分に影響が出る可能性などは考慮されていない。

コーヒーは世界で最もよく飲まれている飲み物の一つだが、健康影響はよくわかっていない。

@Tom 自分は毎日5杯飲んでるので早死にしようです。苦笑

@Mary サウスカロライナ大ってどこにあるのかな

@David アメリカのコーヒーなんてコーヒーじゃないと思う

@Ann 私はアメリカンコーヒーが好きだな。毎日飲んでます

図1: Twitter を用いたソーシャルアノテーションツール

本研究では Twitter とインターネット上のニュース記事を対象にツールの構築を進めていくが、その本質は「大きな文章中のセンテンスの集合と、小さな文章の集合のアラインメント」である。そのため、「背景」でも述べたように、本研究で得られた成果を、Twitter やインターネット上のニュース記事に限らずに応用することが可能である。例えば、ニュースを学術論文に置き換え、Twitter 上の投稿をその論文に対するレビューに置き換えることで、その学術論文のどの部分にどのような反応が起きているかを知る事が出来るツールを構築可能である。

## 2 手法

本研究では、ニュース上の記事の断片に対して、ソーシャルネットワーキングサービス「Twitter」上の投稿から関連度の高いものを結びつけるツールを作成する。この章では、そこで用いる手法について説明する。

本研究のタスクは、以下のようなステップに分けることができる。

### 1. 記事と関連するニュース記事の収集

Twitter 上の投稿の中から、対象のニュース記事と関連性が高いと思われるものを抽出して収集する。

### 2. 内容語の抽出

Twitter 上の投稿の文中から、Twitter 上の投稿とニュース断片の関連度の計算にほとんど影響しないと判断できる単語を除去する。

### 3. 関連度の計算

Twitter 上の投稿と、ニュースの断片との関連度を計算して数値化する。

### 4. ニュース断片と Twitter 上の投稿の結びつけ

計算された Twitter 上の投稿とニュース断片の関連度をもとに、ニュース断片への Twitter 上の投稿の結びつけを行う。

以下で、上記のそれぞれのステップについて詳しく説明をする。

## 2.1 記事と関連するニュース記事の収集

記事と関連するニュースの収集に関しては、同じタスクを取り扱った先行研究がすでにあるため本研究では深く取り扱わない。以下、本研究における暫定的な手法について説明する。

Twitter 上においては、あるニュースに関する投稿をする際に、そのニュースのタイトルを併記するという慣習がある。そこで、本研究では、対象のニュースのタイトルを含む Twitter 上の投稿を、その記事と関連するニュースとして自動的に収集するようにした。しかし、Twitter 上の投稿の中には、

- タイトルを併記せずにそのニュースに言及しているもの
- 誤植などで不完全な形でタイトルを併記してそのニュースに言及しているもの
- そのニュースに言及はしていないが、偶然タイトルを含んでいるもの

が存在する。また、他のソーシャルネットワーキングサービス上で同様の慣習が根付いているとは限らないため、応用の範囲も狭められる。今後先行研究をもとに記事の収集を別の手法に置き換えることが望ましいと考える。これについては今後の課題としたい。

## 2.2 内容語の抽出

自然言語の文の特性を調べる際に、bag-of-words[1] という考え方がる。bag-of-words では、ある文の特性を解析する際に、その文にどのような単語がどのくらい含まれているのかを計算して、それをもとに文の特性を解析し、クラスタリング等のタスクを行う。本研究でも、web ニュースの断片に含まれる単語と、Tweet に含まれる単語の集合を比較して、web ニュースの断片と Tweet の関連度を計算する。

一般に、自然言語の文中には、その文章の持つ素性を計算する際にほとんど情報をもたないような単語が多く含まれる。副詞や格助詞などがそうである。反対に、その文章の持つ素性を計算する際に有用な情報を与える単語を内容語と定義する。非内容語である単語をあらかじめ除去し、内容語を抽出しておくことで、関連度の計算の精度を高めることが出来ると考えられる。関連度の計算の前処理として、この内容語の抽出を行う。

内容語の抽出には、二通りの方法が考えられる。一つ目は、形態素解析ツールを使用し、単語の品詞を分析した上で、副詞や格助詞などの非内容語であると考えられる単語を除去する方法である。二つ目は、統計的な手法を用いて、その単語のその文章における重要度を計算し、一定の重要度以下の値を持つ単語を非内用語として除去する方法である。

一つ目の方法は比較的簡単に実現でき、一定の精度も期待できるが、

- 形態素解析ツールに依存するため他言語に拡張できない
- ソーシャルネットワーキングサービス特有の、ネットスラングなどの語彙に対応できない

といった問題点が存在する。そこで、本研究では二つ目の、統計的手法を採用する。

### 2.2.1 tf-idf

本研究では、統計学的手法として tf-idf[2] を採用する。tf-idf は、文書中である単語が、その文章をどの程度特徴づけているかの重み度合いである。tf-idf は、以下の式によって定義される。

$$\text{tfidf}_{w,d} = \text{tf}_{w,d} \cdot \text{idf}_w \quad (1)$$

$$\text{tf}_{w,d} = \frac{n_{w,d}}{\sum_k n_{w,d}} \quad (2)$$

$$\text{idf}_w = \log \frac{|D|}{|\{d | t_w \in d\}|} \quad (3)$$

ここで、 $n_{w,d}$  は単語  $w$  の文書  $d$  における出現回数、 $|D|$  は総ドキュメント数、 $|\{d | t_w \in d\}|$  は単語  $w$  を含む総ドキュメント数である。

この式の意味するところは、「どのような文書にも普遍的に出現するような単語の重みは低くなる」ということと、「ある文書中で頻繁に言及されるような単語があれば、その文書中でのその単語の重みは高くなる」ということである。

本研究では、tf-idf の高い単語は内容語であると判断し、tf-idf の低い単語は内容語では無いと判断する。

本研究では、まず新聞記事を 100 件以上集め、idf の計算を行う。ここで、収集する記事が多ければ多いほど idf の精度は上がる。この idf をもとに、web ニュースの断片と tweet の関連度を

計算する際に tf-idf を計算し、単語を tf-idf でソートする。そして、その中で閾値以上の単語を内容語とする。この閾値を調整すると、内容語の数と質が変わるので、実験によって調整する。

## 2.3 関連度の計算

内容語の抽出を行ったのちに、Twitter 上の投稿とニュース断片の関連度の計算を行う。関連度の計算にも、bag-of-words の考え方を採用し、文中に含まれる単語を分析する。

単語をもとにニュース断片と Twitter 上の投稿の関連度を計算する際の指標として、以下の三つが挙げられる。

### 1. 完全に一致する単語

web ニュースの断片と Twitter 上の投稿の間に完全に一致する単語がある場合、そのセンテンスと Tweet の関連度は高いと考えられる。図 1 の例では、@Mary というユーザーが投稿した tweet に「サウスカロライナ」という単語が含まれているので、同じ単語を含む一つ目のセンテンスとの関連度が高くなる。

### 2. 言い換えの単語

web ニュースの断片と Twitter 上の投稿に言い換えの関係にある単語が含まれている場合、そのセンテンスと Tweet の関連度は高くなると考えられる。図 1 の例では、@David というユーザーが投稿した tweet に「米国」と言い換えの関係にある「アメリカ」という単語が含まれているので、「米国」という単語が含まれる、5 行目から始まるセンテンスとの関連度が高くなる。

### 3. 互いに類似する単語

web ニュースの断片と tweet の文に、同じ文脈で使われるような単語が含まれている場合、そのセンテンスと Tweet の関連度は高くなると考えられる。図 1 の例では、@Tom というユーザーが投稿した Tweet に「早死に」という単語が含まれており、これは「死亡率」という単語と同じ文脈で使われる単語であると判断できるので、一行目から始まるセンテンスとの関連度が高くなる。

これらの単語が含まれていたら値が高くなるようにセンテンスと Tweet の関連度を定義する。このようにすることで、ニュース web ニュースの断片と Twitter 上の単語を比較し、のアライメントをとることができる。

上記の完全に一致する単語、言い換えの単語、互いに類似する単語の中でも、単語によってニュース記事と Twitter 上の投稿の関連度に対する影響は異なると考えられる。そこで、本研究ではこの影響の違いを tf-idf で重み付けすることで対応する。また、「一致する単語」、「言い換えの単語」、「互いに類似する単語」の三つの集団でも、ニュース記事と Twitter 上の投稿の関連度に対する影響は異なると考えられる。これは定数で重み付けをすることで対応する。この定数は、実験で調整する。

以上より、Tweet  $t$  とニュース断片  $s$  の関連度を以下の式によって定義する。

$$\text{Rel}(s, t) = \alpha \sum_{\{w|w \in A\}} w * \text{tfidf}_{w,d} + \beta \sum_{\{w|w \in B\}} w * \text{tfidf}_{w,d} + \gamma \sum_{\{w|w \in C\}} w * \text{tfidf}_{w,d} \quad (4)$$

ここで、 $A$  は  $s$  と  $t$  の中で完全に一致する単語の集合であり、 $B$  は  $s$  と  $t$  の中で言い換えの関係にある単語の集合であり、 $C$  は  $s$  と  $t$  の中で互いに類似の関係にある単語の集合である。同じ単語が複数回出てきた場合も、それは別々の単語として処理する。すなわち、tf-idf の高い単語が  $s$  と  $t$  の両方に複数回にわたって出現していたら、それだけ関連度  $\text{Rel}(s, t)$  は高くなる。 $\alpha, \beta, \gamma$  は定数であり、実験によりアラインメントの結果を見ながら調整する。「完全に一致する単語」の方が、「言い換えの単語」や「互いに類似する単語」よりも関連度に大きな影響力を持つと考えられる。また、「言い換えの単語」は「互いに類似する単語」よりも関連度に大きな影響力を持つと考えられる。そのため、 $\alpha > \beta > \gamma$  となることが予想される。

上記であげた蜜の集合のうち、 $A$  の「完全に一致する単語」は容易に発見することができる。それに対して、 $B$  の「言い換えの単語」や、 $C$  の「互いに類似する単語」を識別するためには、統計的な処理が必要となる。この手法については後述する。

### 2.3.1 言い換えの関係の語の検出

本研究では「言い換えの単語」の検出方法として、WordNet<sup>8</sup>を利用する。WordNet は、英語の概念辞書である。WordNet では、単語が synset という同義語のグループにまとめられている。よって、同一の synset に含まれる単語は言い換えの単語であると判断できる。WordNet には英語版の他にも、独立行政法人情報通信研究機構（NICT）によって日本語版の物が提供されている。また、Python 等のプログラミング言語向けのフロントエンドも提供されており、プログラムから動的にデータ資源にアクセスすることが可能である。

### 2.3.2 互いに類似する語の検出

「互いに類似する単語」の検出にはいくつかの方法が考えられるが、本研究では言い換えの関係の語の検出と同じく、WordNet を利用して類似する単語の検出を行う。しかし、WordNet を用いた類似語の検出にはいくつかの問題点が存在するため、考えられる代替案として Dekang Lin の手法 [3]、Latent Space でのモデル化 [4]、Co Clustering [5] を後に紹介する。

### 2.3.3 WordNet を用いた類似語の検出

WordNet では同義語同士が Synset というグループにまとめられているが、各々の Synset は他の Synset と繋がりを持っている。Synset 同士の繋がりにはいくつかの種類がある。Synset 同士の繋がり種類は、単語の品詞によって異なる。以下に、名詞の Synset における Synset 間の繋がり例を示す。

- 上位語  
すべての  $X$  が  $Y$  の種類の一であるなら  $Y$  は  $X$  の上位語である。
- 下位語  
すべての  $Y$  が  $X$  の種類の一であるなら  $Y$  は  $X$  の下位語である。
- 同族後  
 $X$  と  $Y$  の上位語が同じなら、 $Y$  は  $X$  の同族語である。

---

<sup>8</sup><http://nlpwww.nict.go.jp/wn-ja/>



- 全体語  
X が Y の一部であるなら、Y は X の全体語である。
- 部分語  
Y が X の一部であるなら、Y は X の部分語である。

以上のような繋がりを用いて、WordNet 中では Synset 同士が階層構造を形成している。Synset のネットワークの例を図 2 に示す。

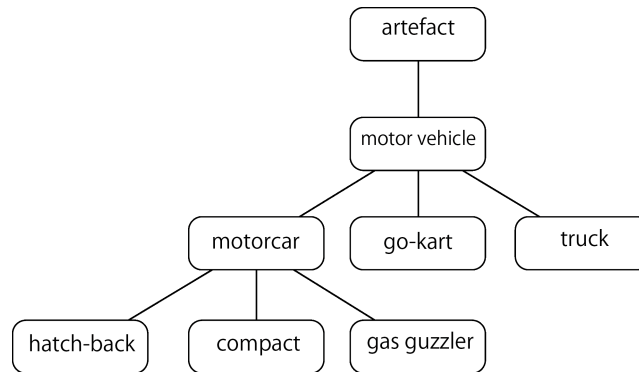


図 2: WordNet における Synset 同士の繋がり の例

図 2 において、artefact は motor vehicle の上位語であり、motorcar, go-kart, truck はともに motor vehicle の下位語であるので、互いに同族語である。

このようなネットワーク上では、意味の近い単語ほど短い経路でたどり着くことができ、意味の遠い単語ほどたどり着くのに長い経路を必要とすると考えることができる。このネットワーク上での距離を測定することで、Synset 間の類似度を測定することが出来る [8]。本研究ではこの性質を利用して単語間の類似度を測定する。そして、類似度が一定の閾値以上の単語同士を互いに類似する単語として集計する。この類似度は実験により調整する。

#### 2.3.4 Dekang Lin の手法の利用

WordNet による類似度の測定は効果的であるが、人手によって構築された意味辞書に依存しているという欠点がある。英語圏には既に完成度が高い WordNet が存在するが、それをもとに作られた日本語圏の WordNet はまだ語彙数が少ない。このように、WordNet を用いた類似語の検出は言語に依存してしまう。また、ソーシャルネットワーキングサービス上ではネットスラング等の辞書上に無い新しい語彙が用いられる場合もあり、その場合も WordNet では対応することが出来ない。

以上のような理由から、本研究のツールにおける類似語の検出には意味辞書に依存せずに、収集したテキストコーパスから学習できるような手法が望ましいと考えられる。

このような「互いに類似する語」の検出方法として、Dekang Lin の手法 [3] の利用が挙げられる。これは、単語同士がどれだけ「似ているのか」を計算するための手法である。Dekang Lin の手法では、単語同士の類似度は以下の式で定義される。

$$\text{sim}(w_1, w_2) = \frac{\sum_{(r,w) \in T(w_1) \cap T(w_2)} (I(w_1, r, w) + I(w_2, r, w))}{\sum_{(r,w) \in T(w_1)} I(w_1, r, w) + \sum_{(r,w) \in T(w_2)} I(w_2, r, w)} \quad (5)$$

ここで、 $I$  は相互情報量であり、以下の式で定義される。

$$I(w, r, w') = -\log(P_{MLE}(B)P_{MLE}(A|B)P_{MLE}(C|B)) - (-\log P_{MLE}(A, B, C)) \quad (6)$$

$$= \log \frac{\|w, r, w'\| \times \|\ast, r, \ast\|}{\|w, r, \ast\| \times \|\ast, r, w'\|} \quad (7)$$

ここで、 $\|w, r, w'\|$  という記法は、単語  $w$ ,  $w'$ 、単語間の関係  $r$  に対して  $w$  が  $w'$  の  $r$  であるという関係  $(w, r, w')$  が成り立つ箇所の総数である。例えば、"I have a brown dog" という文章では、 $w$  を "dog"、 $w'$  を "have"、 $r$  を "object of" として、(dog, objectof, have) という関係が成り立っているので、 $\|\text{dog, objectof, have}\| = 1$  となる。 $w$ ,  $w'$  は  $\ast$  (ワイルドカード) で置き換えられることもあり、その場合、 $\ast$  に全ての単語をあてはめた総数が  $\|w, r, w'\|$  の値となる。

$P_{MLE}$  は最尤法に基づいて計算される確率であり、 $P_{MLE}$  の引数となっている  $A, B, C$  はそれぞれ以下のような事象である。

$A$ : 無作為に選出した語が  $w$  である

$B$ : 無作為に選出した関係が  $r$  である

$C$ : 無作為に選出した語が  $w'$  である

$P_{MLE}$  は以下の式を満たすことが示されている。

$$P_{MLE}(B) = \frac{\|\ast, r, \ast\|}{\|\ast, \ast, \ast\|} \quad (8)$$

$$P_{MLE}(A|B) = \frac{\|w, r, \ast\|}{\|\ast, r, \ast\|} \quad (9)$$

$$P_{MLE}(C|B) = \frac{\|\ast, r, w'\|}{\|\ast, r, \ast\|} \quad (10)$$

これらの定義の意味するところは、同じ単語と同じ関係にある二つの単語は、似ている単語であるということである。例えば、「車」と「電車」は共に「乗る」という動詞の目的語になるので、似ている単語と見なすことができる。

本研究では、この定義に従って  $\text{sim}(w_1, w_2)$  を計算し、この値が閾値を越えたペアを互いに類似する単語とみなす。この閾値を調整すると、検出される類似語のペアの質が変わるので、実験によって最適な値を求める。

### 2.3.5 Latent Space でのモデル化

意味辞書に依存せずに「互いに類似する語」のもう一つの検出方法として Latent Space でのセンテンスのモデル化 [4] が挙げられる。この手法では、

$$X_{ij} = \text{tfidf}_{i,j} \quad (11)$$

によって定義される  $M \times N$  行列  $X$  を、 $M \times K$  行列  $P$ ,  $N \times K$  行列  $Q$  を用いて

$$X \approx P^T Q \quad (12)$$

と近似的に分解することにより、次元数の低い素性を抽出する手法である。ここで、 $P$ ,  $Q$  はランダムに初期化された上で

$$P_{:,i} = (Q\tilde{W}^{(i)}Q^T + \lambda I)^{-1}Q\tilde{W}^{(i)}X_{i,:}^T \quad (13)$$

$$Q_{:,i} = (P\tilde{W}^{(i)}P^T + \lambda I)^{-1}P\tilde{W}^{(i)}X_{:,i}^T \quad (14)$$

によって段階的に求められる。 $W^{(i)}$  は、

$$W_{i,j} = \begin{cases} 1 & \text{if } X_{i,j} \neq 0 \\ w_m & \text{if } X_{i,j} = 0 \end{cases}$$

によって定義される  $W$  の  $i$  行目を成分とする対角行列で、

$$W^{(i)} = \text{diag}(W_{:,i}) \quad (15)$$

と表される。

行列  $P$  の各列には、 $X$  に現れていた、各単語の「どの文書でどのくらい重要になるか」という特徴量が低次元に圧縮されて格納されている。これは、単語の情報が抽象化されて入っていると言える。この  $P$  の中で、コサイン類似度の高い列同士のペアを見つければ、その列に割り当てられている単語同士は互いに類似する単語と言えるようになる。本研究では、コサイン類似度が閾値を越えたペアを類似語とみなす。この閾値を調整すると、検出される類似語のペアの質が変わるので、実験によって最適な値を求める必要がある。

## 2.4 ニュース断片と **Twitter** 上の投稿の結びつけ

この段階では、式 4 により算出された web ニュース断片と Twitter 上の投稿の関連度をもとに、ニュース断片と Twitter 上の投稿の結びつけを行う。これを、アラインメントと定義する。ニュース断片と Twitter 上の投稿の結びつけでは、全ての Twitter 上の投稿がニュース断片と結びつくような状態になってはいけない。逆に、全ての Twitter 上の投稿がニュース断片に結びつかないような状態になっても、ツールとしての有効性が失われる。したがって、この段階では Twitter 上の投稿がニュース断片にほどよくばらけて結びつくようにコントロールする必要がある。本研究では、このようなことが実現可能な手法として、安定結婚アルゴリズム [9] の拡張である、一対多の安定結婚アルゴリズム [10] によるアラインメントを採用した。

### 2.4.1 安定結婚問題

この節では安定結婚問題の概要とその解法を説明する。

安定結婚問題 [9] は、安定マッチング問題の一つである。安定結婚問題では、 $N$  人の男性と  $N$  人の女性がいる状況で、「不倫」の起こることの無いように  $N$  組の結婚を成立させることを目

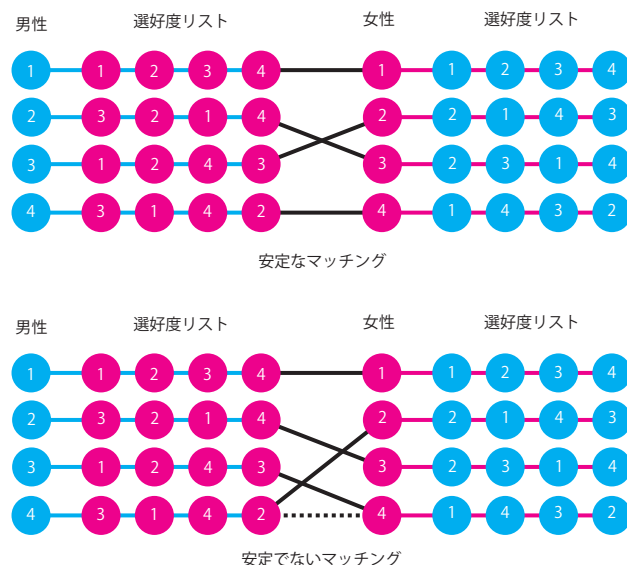


図 3: 安定結婚問題の例

的とする。不倫の起こらない  $N$  組の結婚の組み合わせを、「安定なマッチング」という。安定結婚問題では、 $N$  人の男性はそれぞれ、どの女性を好むかを順位付けした長さ  $N$  の選好度リストをもっている。 $N$  人の女性もそれぞれ同様に、どの男性を好むかを順位付けした長さ  $N$  の選好度リストを持っている。ここでいう「不倫」とは、 $N$  組のカップルが成立している中で、互いに現在組んでいるカップルよりも好きなペアが発生することを指す。このようなペアを「ブロッキングペア」と定義する。

安定結婚問題の例を図 3 に示す。図 3 上段の安定なマッチングの例では、ブロッキングペアが存在しないので不倫は起こらない。それに対して、図 3 下段の不安定なマッチングの例では、男性 4 と女性 4 のペアがブロッキングペアとなるので、不倫がおこってしまう。

安定結婚問題の全ての事例に対して、安定なマッチングが必ず存在することが証明されている。そのようなマッチングを見つける手法として、Gale-Shapley アルゴリズムというものが知られている。Gale-Shapley アルゴリズムの疑似コードを以下に示す

---

```

1  # 定義群
2  # -----
3  # males:  $N$  人の男性が格納された配列
4  # females:  $N$  人の女性が格納された配列
5  #  $match(m, f)$ : 男性  $m$  と女性  $f$  を結婚させる
6  #  $unmatch(m, f)$ : 男性  $m$  と女性  $f$  を離婚させる
7  #  $x.preference\_list$ :  $x$  の選好度リスト。
8  # 男性なら  $N$  人の女性が、女性なら  $N$  人の男性が先頭から好きな順に格納されている
9  #  $x.is\_matched$ :  $x$  が誰かと結婚していれば True を、そうでなければ False を返す
10 #  $x.partner$ :  $x$  が誰かと結婚している場合、その相手を返す
11 #  $x.prefer(a, b)$ :  $x$  が  $a$  よりも  $b$  を好むなら True を、そうでなければ False を返す
12
13 # 最初は全ての男性は独身である
14 for male in males:
15     for female in females:
16         male.has_proposed[female] = False
17
18 while True:
```

```

19     for male in males:
20         if male.is_matched: continue
21         # 独身の男性は、まだ求婚したことがない女性に順番に求婚していく
22         for female in male.preference_list:
23             if male.has_proposed[female]: continue
24             # 求婚をする
25             male.has_proposed[female] = True
26             # 女性が独身なら結婚をする
27             if not female.is_matched:
28                 match(male, female)
29                 break
30             # 女性が既婚であり、もしも女性が現在のパートナーよりも男性
31             #   maleの方が好ましいと思った場合、
32             #   現在の結婚を解消して男性maleの求婚を受け入れる
33             if female.prefer(male, female.partner):
34                 unmatched(female.partner, female)
35                 match(male, female)
36                 break
37             # 求婚が受け入れられなかった場合、一つ選考順位の低い女性に求婚する
38             continue
39         # 全ての男性が結婚している状態になったら終了する
40         finished = True
41     for male in males:
42         if not male.is_matched:
43             finished = False
44     if finished: break

```

上記アルゴリズムが終了した際に得られるマッチングは安定である。証明の概略を以下に示す。

以下、便宜的に、 $x$  が  $y$  より  $z$  を好むことを  $y <_x z$  と表す。

*Proof.* Gale-Shapley アルゴリズムで得られたマッチングが安定であることの証明

得られたマッチングが安定ではないと仮定する。

すなわち、二つのペア  $(m_1, f_1), (m_2, f_2)$  のブロッキングペア  $(m_1, f_2)$  が存在して、 $f_1 <_{m_1} f_2 \wedge m_1 >_{f_2} m_2$  であると仮定する。

(i) 男性  $m_1$  は好きな順番に女性にプロポーズをしている。 $f_1 <_{m_1} f_2$  より、 $m_1$  は  $f_2$  にすでにプロポーズをしている。

(ii) また、女性は相手を変える際には、より選好度の高い男性にしか変えない。(i) より  $f_1$  はすでに  $m_1$  にプロポーズされているので、 $m_1 <_{f_1} m_2$  である。

これは仮定の  $m_1 >_{f_2} m_2$  に矛盾するので、背理法より得られたマッチングは安定である。

□

また、

1. 一人の男性は同じ女性に二度以上プロポーズをしない
2. 女性は婚約すると独身に戻らない
3. 女性はプロポーズされる際その相手が悪くなる事はない

ということから、Gale-Shapley アルゴリズムは有限回数で終了することが言える。さらに、Gale-Shapley アルゴリズムは  $O(N^2)$  時間で終了することが知られている。

上記の説明では、便宜上男性が女性に順番に求婚していくという形式をとったが、女性が男性に順番に求婚していくように置き換える事も可能である。男性が女性に順番に求婚していき、最終的に得られるマッチングを男性最良安定マッチングと呼び、女性が男性に順番に結婚していき最終的に得られるマッチングを女性最良安定マッチングと呼ぶ。

#### 2.4.2 一対多型安定結婚問題

この節では、安定結婚問題の拡張である、一対多型安定結婚問題について説明する。本研究では Tweet とニュース断片のマッチングを安定結婚問題として定式化することでアラインメントをとるが、一般に Tweet とニュース断片の数は異なるため、Gale-Shapley アルゴリズムを適用することが出来ない。一つのニュース断片に複数の Tweet が結びつく場合もあるし、逆に一つも Tweet が結びつかない場合も想定される。

このような問題を「一対多型安定結婚問題」として定式化することを考える。

一対多型安定結婚問題に関する選考研究 [10] では、一対多型安定結婚問題を定式化している。その概要を以下に示す。

##### 定義 2.1. 解の定義

男性の集合  $M$ , 女性の集合  $F$ , そして各々の女性  $f \in F$  に対する定員 (*capacity*) と呼ばれる非負整数  $c_f (f \in F)$  が与えられている。このとき、写像  $\phi: M \rightarrow F$  のうち、

$$|\phi^{-1}| \leq c_f, \forall f \in F \quad (16)$$

となるものを一対多型結婚問題の解 (*solution*)、あるいは縁組と呼ぶ。

また、一対多型結婚問題における安定性を以下のように定義している。

##### 定義 2.2. 安定性の定義

以下の条件 (i) と (ii) を同時に満足する  $m \in M$  と  $f \in F$  の組を、不安定組 (*unstable couple*) と呼ぶ。

$$(i) \exists n \in \phi^{-1}(f); n <_f m \quad (17)$$

$$(ii) \phi(m) <_m f \quad (18)$$

不安定組が存在するとき、縁組み  $\phi: M \rightarrow F$  は不安定 (*unstable*) であると言い、そうでないとき、安定 (*stable*) であるという。

ここで、男性  $m \in M$  の女性に対する選好を全順序集合  $(F, \leq_m)$  で表現し、同様に女性  $f \in F$  の男性に対する選好を全順序集合  $(M, \leq_f)$  で表現する。

上で定義されている不安定組は、安定結婚問題のブロッキングペアに相当するものである。

一対多型結婚問題における安定な縁組みを見つける問題を一対多型安定結婚問題と呼ぶ。この問題を Tweet とニュース断片に適用することで Tweet とニュース断片のアラインメントをとる。

一対多型の安定結婚問題に置ける選好度リストの例を表 1 に示す。また、表 1 の選好度リストのもとでの一対多型安定結婚問題の解の例を図 4 に示す。

表 1: 一対多型安定結婚問題における選好度リストの

男性 ID	選好度リスト				女性 ID	選好度リスト							
1	1	4	3	2	1	8	1	5	6	4	2	7	3
2	4	2	1	3	2	3	1	8	2	7	6	5	4
3	4	2	1	3	3	2	1	7	4	5	3	8	6
4	3	4	2	1	4	5	8	3	1	2	6	4	7
5	3	2	4	1									
6	3	2	4	1									
7	4	1	3	2									
8	2	3	4	1									

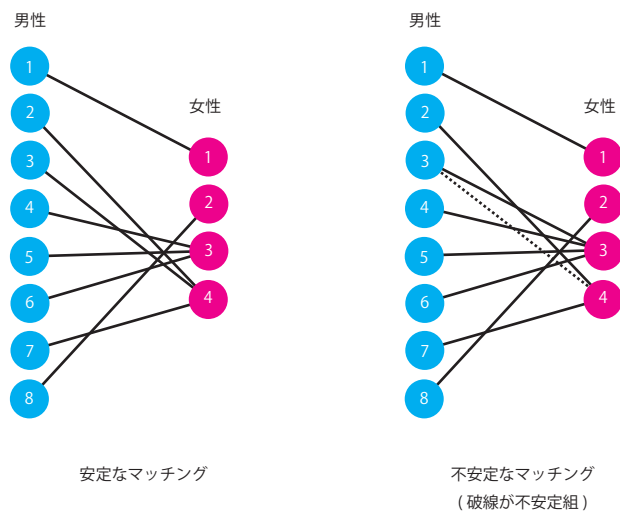


図 4: 一対多型安定結婚問題の結果の例

ここで、便宜的に ID が  $n$  の男性を  $m_n$ , ID が  $n$  の女性を  $f_n$  と表す。

図 4 の右側の例は、全ての組み合わせが定義 2.2 の安定性の条件を満たしているので、安定な解であると言える。それに対して、左側の例では、組み合わせ  $(m_3, f_4)$  について、

$$(i) m_7 \in \phi^{-1}(f_4) \wedge m_7 < m_3 \quad (19)$$

$$(ii) f_2(= \phi(m_3)) <_{m_3} f_4 \quad (20)$$

が成り立ち、 $(m_3, f_4)$  は不安定組となる。

先に挙げた選好研究 [10] では、Gale-Shapley アルゴリズムを拡張することで一対多型安定結婚問題の解を導いている。そのアルゴリズムの概要を以下に示す。

---

```

1  # 定義群
2  # -----
3  # males:  $N$  人の男性が格納された配列
4  # females:  $N$  人の女性が格納された配列
5  # match( $m, f$ ): 男性  $m$  と女性  $f$  を結婚させる
6  # f.queue: 女性  $f$  と結婚している男性のリスト
7  # f.pop(): 女性
   #  $f$  と結婚している男性の中で、 $f$  にとって最も選好度の低い男性を離婚させる
8  # x.preference_list:  $x$  の選好度リスト。
9  # 男性なら  $N$  人の女性が、女性なら  $N$  人の男性が先頭から好きな順に格納されている
10 # x.prefer( $a, b$ ):  $x$  が  $a$  よりも  $b$  を好むなら True を、そうでなければ False を返す
11
12 p = {} # 各男性が何番目の女性を選ぶかを保管するハッシュテーブル
13 for m in males:
14     # 最初は一番好きな女性に求婚する。
15     p[m] = 0
16     # 一番好きじゃない女性に求婚するまで繰り返す
17     while 0 <= p[m] < len(females):
18         f = m.preference_list[p[m]]
19         match(m, f)
20         # もしも女性と結婚している男性のリストが
21         # capacity の範囲内に収まっていればそのまま続ける
22         if len(f.queue) <= f.capacity:
23             break
24         # capacity を越えていれば、最も選好度の低い男性を離婚させる
25     else:
26         m = f.pop()
27         p[m] += 1

```

---

上記のアルゴリズムにより、男性の集合  $M$ 、女性の集合  $F$  に対して  $O(|F| \cdot |M| \cdot \log(|M|))$  の時間計算量で一対多型の安定結婚アルゴリズムを解くことができる。

## 2.5 形態素解析

本研究の最終的な目的は、言語に依存しないツールの構築であるが、実験段階では便宜的に日本語を対象とする。日本語は、英語等の言語と違い、単語間の区切りの位置が明確でない。そのため、どの文字からどの文字が一つの単語であるのかを分析する必要がある。今回は、MeCab[6] という形態素解析ツールを使う。MeCab は、Conditional Random Fields を用いて日本語の単語区切りと、それぞれの単語の品詞を推定するツールである。例えば、「東京都に住む」という



文章は、「東京 都 に 住む」という区切り方に分割され、「東京」は名詞、「都」は接尾辞、「に」は格助詞、「住む」は動詞であると、品詞が解析される。

### 3 実験

#### 3.1 概要

本研究では

- 実験 A:tf-idf による内容語の抽出
- 実験 B:tweet とニュース断片の関連度の計算
- 実験 C:tweet とニュース断片のアラインメント

の三つの実験を行った。

実験 A では、tf-idf によりインターネット上のニュース記事と twitter の書き込みから内容語を抽出するシステムを構築し、その性能を評価した。内容語の抽出に関しては、サンプル記事と、その記事に言及している tweet に現れている各単語について tf-idf を計算し、ソートすることにより、内容語と判定する基準である閾値がどのような値になるのかを検討した。

実験 B では、ニュース記事と tweet の間の関連度を「一致する単語」、「言い換えの単語」、「類似の関係にある単語」を指標として計算し、人間の手によって評価された関連度と比較することでその性能を評価した。

実験 C では、実験 B で評価したニュース記事と tweet の関連度の計算手法をもとに、ニュース記事と tweet の間のアラインメントを安定結婚アルゴリズムにより計算した。このアラインメントの結果も、人間の手によって生成された関連度と比較することでその性能を評価した。

#### 3.2 実験条件

##### 3.2.1 実験サンプル

本研究ではサンプルとして、Twitter 上のアカウント”@Google\_news.jp”によって取り上げられているニュースを使用した。”@Google\_news.jp”は、様々な報道機関が提供しているインターネット上の記事のタイトル・URL を紹介しているアカウントである。

本研究では、収集したニュース記事のタイトルを本文中に含む tweet をその記事に言及する tweet とみなし使用する。ただし、”@Google\_news.jp”のような単なるニュース紹介のアカウントの tweet は、個人の意見や感想を含まないために除外する。記事に言及する tweet は記事のタイトルや URL を内部に含む場合が多いので、その部分は tweet の本文から除外して分析をする。

tf-idf において idf を計算するためのデータセットとしては、”@Google\_news.jp”のニュース記事から 20 記事を集めた物を使用する。

また、ニュース断片と Tweet の関連度の評価、ニュース断片と Tweet のアラインメントの計算のためのニュース記事のサンプルは、2014 年 12 月 1 日から 2014 年 12 月 31 日の間のニュー

表 2: 本研究で取り扱ったニュース記事の一覧

救助中ヘリから落下、心肺停止の 1 人を翌日発見
石破氏なお秘密報道に疑問 「処罰対象ではないが…」
レーシック手術の被害情報 5 年で 80 件、目の痛みや矯正のしすぎ／消費者庁
秘密法案、与党が夜にも成立強行 会期 2 日延長、攻防は最終局面
人類最古の DNA 抽出… 4 0 万年前の人骨から
「ふなっしー」酷似「きゃべっしー」に苦情殺到
甘利氏が治療で T P P 会合欠席へ 早期の舌がん、一時辞意も慰留
自殺原因は過大なノルマ・罵声…日本郵便を提訴
軽自動車保有税上げ 平均数千円、15 年 10 月までに
米中会談は 5 時間半、防空圏「認めない」と副大統領、「日中ホットライン確立を」

表 3: 二者によるクラス分類の例

		B	
		Yes	No
A	Yes	20	5
	No	10	15

スの中で、多くの Tweet に言及されているものを 10 記事選択して取り扱った。本研究で取り扱ったニュース記事のタイトルを表 2 に示す。

また、15 人の協力者に実験に参加してもらい、人手による Tweet とニュース断片の関連度の評価や Tweet とニュース断片のアラインメントのサンプルの収集を行った。

### 3.2.2 $\kappa$ 評価尺度 shakudo

人手による関連度評価やアラインメントの結果は、 $\kappa$  係数 [11] により評価した。 $\kappa$  係数は、人手によるクラス分類の際の、個人ごとによるずれを評価する尺度である。二人の異なる分類者の間で、 $\kappa$  係数は以下の式によって定義される。

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \quad (21)$$

ここで、 $Pr(a)$  は実際に観測された一致率である。それに対して、 $Pr(e)$  は二人のクラス分類が偶然一致する確率である。上記のように定義することで、 $\kappa$  係数では、偶然の要素を除いた二者の一致度合いを測定することが出来る。

分類者 A と分類者 B による、二値のクラス分類の例を表 3 に示す。この例では、全体の一致率は

$$\frac{20 + 15}{20 + 5 + 10 + 15} = 0.7 \quad (22)$$

より、0.7である。一方、分類者 A が Yes に分類する確率  $P_{A,yes}$ , No に分類する確率  $P_{A,no}$ , 分類者 B が Yes に分類する確率  $P_{B,yes}$ , No に分類する確率  $P_{B,no}$  はそれぞれ以下のように求めることができる。

$$P_{A,yes} = \frac{20 + 5}{20 + 5 + 10 + 15} = 0.5 \quad (23)$$

$$P_{A,no} = \frac{10 + 15}{20 + 5 + 10 + 15} = 0.5 \quad (24)$$

$$P_{B,yes} = \frac{20 + 10}{20 + 5 + 10 + 15} = 0.6 \quad (25)$$

$$P_{B,no} = \frac{5 + 15}{20 + 5 + 10 + 15} = 0.4 \quad (26)$$

したがって、両方の分類者が「偶然」同時に yes に分類する確率は  $P_{A,yes} * P_{B,yes}$ , 両方の分類者が同時に no に分類する確率は  $P_{A,no} * P_{B,no}$  で表すことができ、両者の分類結果が偶然一致する結果  $Pr_e$  は以下のように求まる。

$$Pr_e = P_{A,yes} * P_{B,yes} + P_{A,no} * P_{B,no} = 0.5 \quad (27)$$

以上より、この例に置ける分類者 A と分類者 B の間の  $\kappa$  係数は

$$\kappa = \frac{0.7 - 0.5}{1 - 0.5} = 0.4 \quad (28)$$

となる。

一般に二人の分類者の一致度合いは、 $\kappa$  係数が 0 であれば一致なし (偶然と同程度の一致)、0 から 0.4 の間であれば弱い一致、 $\kappa$  係数が 0.4 から 0.6 の間であればそこそこの一致、 $\kappa$  係数が 0.6 から 0.8 の間であれば良い一致、 $\kappa$  係数が 0.8 から 1 の間であれば非常に高い一致度であるとみなされる。

多段階のクラス分類の場合、上記の定義の通りに  $\kappa$  係数を計算すると、クラス数が多くなるほど二者の分類結果が一致する確率が低くなってしまうので、 $\kappa$  係数を多段階クラス用に拡張した重み付け  $\kappa$  係数を使用する。重み付け  $\kappa$  係数では、 $Pr(a)$  は以下の式で求められる。

$$Pr(a) = \frac{1}{N} \sum_{i=1}^k \sum_{j=1}^k w_{i,j} x_{i,j} \quad (29)$$

ここで、 $N$  は全サンプル数、 $k$  はクラス数、 $i, j$  はクラスの ID、 $x_{i,j}$  は片方の分類者がクラス  $i$  に分類し、もう片方の分類者がクラス  $j$  に分類したサンプルの総数である。 $w_{i,j}$  は、重み付け係数で、以下により定義される。

$$w_{i,j} = 1 - \frac{(i - j)^2}{(k - 1)^2} \quad (30)$$

これにより、両者のクラス分類が一致しなかった場合のクラス分類も、一致するクラスと近い場合は「惜しい例」として、重み付けをされた上で  $Pr(a)$  に加算される。なお、重み付け  $\kappa$  係数で  $k = 2$  とした場合は重み付けをしなかった場合の  $\kappa$  係数と完全に一致する。

本研究では、Twitter 上の投稿とニュース断片の関連度の計算を 4 段階の多段階クラス分類、Twitter 上の投稿とニュース断片のアラインメントを、結びつけるか結びつけないかの二値のクラス分類としてモデル化し、 $\kappa$  係数を適用することでその結果を評価する。

## 参考文献

- [1] 高村大地, 奥村学. 言語処理のための機械学習入門 (コロナ社), 2010
- [2] Gerard Salton. Introduction to Modern Information Retrieval (Mcgraw Hill, Inc.), 1986
- [3] Dekang Lin. Automatic Retrieval and Clustering of Similar Words. COLING '98 Proceedings of the 17th international conference on Computational linguistics - Volume 2 Pages 768-774
- [4] Weiwei Guo, Mona Diab. Modeling Sentences in the Latent Space. ACL '12 Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1 Pages 864-872
- [5] Inderjit S Dhillon, Subramanyam Mallela, Dharmendra S. Modha. Information-Theoretic Co-clustering. KDD '03 Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining Pages 89-98
- [6] 工藤 拓, 山本 薫, 松本 裕治. Applying Conditional Random Fields to Japanese Morphological Analysis, 情報処理学会研究報告. 自然言語処理研究会報告 2004, Pages 89-96
- [7] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze: Introduction to Information Retrieval, Cambridge University Press, 2008.
- [8] Ted Pedersen, Siddharth Patwardhan and Jason Michelizzi. WordNet::Similarity - Measuring the Relatedness of Concepts, HLT-NAACL-Demonstrations '04 Demonstration Papers at HLT-NAACL 2004 Pages 38-41
- [9] D.Gale and L.S.Shapley. COLLEGE ADMISSIONS AND THE STABILITY OF MARRIAGE (1962), The American Mathematical Monthly Vol. 69, pp. 9-15.
- [10] 奥居 哲, 柴田 祥一, 岡田 稔, 川島 信. 安定結婚アルゴリズムに基づく卒業研究配属の事例研究. 社団法人情報処理学会 研究報告 kennkyuuhoukoku 2003, pp. 67-72.
- [11] David M. W. Powers. The Problem with Kappa