# 意見分析に適した意見タグ獲得改善への取り組み

三澤賢祐 †§　成田和弥 †§　伊藤友博 †　柴田知秀 ‡§　河原大輔 ‡§　黒橋禎夫 ‡§
† 株式会社 Insight Tech　‡ 京都大学　§ 科学技術振興機構 CREST
{kensuke_mitsuzawa,kazuya_narita,tomohiro_ito}@insight-tech.co.jp
{shibata,dk,kuro}@i.kyoto-u.ac.jp

## 1　はじめに

株式会社 Insight Tech が運営する Web サービスである不満買取センター [1] では一般消費者から多岐にわたる事物・事象・話題 [2] への不満意見の収集をしている。不満意見は商品やサービスに対する改善策を考案するための有益な情報となり得るため、企業のサービス改善やマーケティング用途に利用されている。

不満買取センターに投稿された不満意見のうち一部は学術研究向けに公開 [3] している [2]。以下、本データセットを FKC コーパスと呼ぶ。FKC コーパスでは、不満意見というネガティブな意見に特化しているため、評判情報を扱う研究向けには、一般的な Web 言語資源と比較し、データ抽出などの手間が少なくノイズが少ないデータと言える。不満買取センターの利用にあたっては、性別、生年、職業のようなユーザーデモグラ情報の登録が必須であるため、一般的な Web 言語資源では得られづらい豊富なユーザーデモグラ情報も利用可能である。

FKC コーパスには投稿された不満意見本文と投稿者のユーザーデモグラ情報や、各投稿に対するカテゴリ情報が付与されており、意見分析に利用可能なデータであるといえる。ここで言う「意見分析」とは、意見のバリエーションを観察し、仮説検証や問題発見のために意見を定量的に分類する作業を指す。不満意見のようなテキストに対する意見分析を行うために、一般的には、分類と集計のために任意のラベル付与や、単語共起を用いた文脈観察を行なうが、こうした方法にはそれぞれ問題がある。ラベル付与は非常に時間がかかる作業であり、単語共起では、多くの場合、仮説検証や問題発見に必要な文脈情報が不足してしまい背景理解に時間がかかってしまう。

そこで、FKC コーパスには意見分析を主目的に自動付与された分析補助情報「意見タグ」が付与されている [4]。意見タグを利用すれば、図 1 に示すように「連日の温度が 38 度近くで夏の暑さは気持ちが悪くなる。」というテキストから、対象が「夏の暑さ」であり、評価内容が「気持ちが悪くなる」であると、構造化して獲得できる。意見タグは、自動的に獲得しており、かつ文脈情報を残しているため、少ない労力で意見分析の実施が可能である。

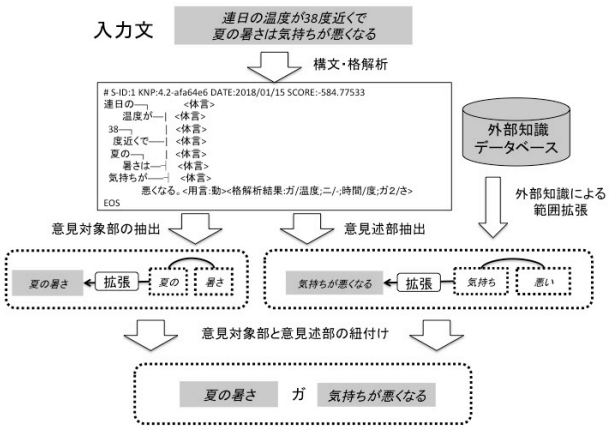


図 1. 意見述部および意見対象部の抽出の流れ

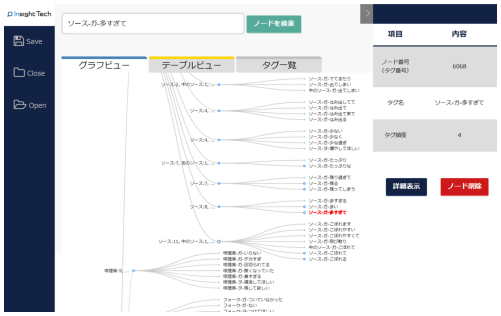


図 2. Insight Tech が開発した意見分布可視化ツール

意見タグの一応用例として、図 2 に株式会社 Insight Tech が開発した意見可視化ツールの例を示す [4]。このツールでは、テキストデータに付与された意見タグを、密ベクトルを利用し意味的な類似度を考慮しながら、グルーピングしている。このツールを利用し、分析者は大量のテキストを読むことなく、言及された意見の傾向と分布を観察することが可能である。

本論文では、意見タグの獲得性能を向上させるための取り組みを紹介する。分析者が文脈情報を理解するための十分な情報を獲得することで、分析補助情報としてのさらなる活用が期待できる。

[1] http://www.fumankaitori.com
[2]「あるファミリーレストラン店」という特定の対象から「人間関係の不満」という身近な内容までと、扱う範囲は幅広い。
[3] http://www.nii.ac.jp/dsc/idr/fuman/fuman.html
[4] http://insight-tech.co.jp/data/news/pdf/IT_NEWS_RELEASE_2017-11-13.pdf

# 2 意見タグの概要

## 2.1 意見タグの定義

「意見タグ」とは、テキストから分析に利用できる箇所だけを部分的に抜き出したフレーズのことである[5]。意見タグは「意見対象部」と「意見述部」、そして2つをつなぐ格助詞の3要素で構成されている。本論文中では意見タグを「意見対象部-格助詞-意見述部」の順番で表記し、また表示上の都合により平文で示すが、意見タグには部分木や形態素情報といった解析情報も含まれている。

## 2.2 意見タグの生成

意見タグは構文解析と述語項構造解析の結果からルールベースで獲得しており [4]、本研究では形態素解析器に Juman[6]、構文解析と述語項構造解析のために KNP[7] を利用する。

図1に意見タグの生成フローを示す。意見対象部と意見述部はそれぞれ独立に獲得され、最後に述語項構造解析の結果に基づき2つを結合する。

意見対象部は体言を開始ノードとして、構文木上で子ども方向にノードを拡張する。ただし、親子間で「連体修飾」のエッジが存在する時のみに、ノード拡張を実施する。

意見述部は述語を開始ノードとして、隣接ノードに拡張をしている。評価表現を扱う意見述部では、文法情報だけで拡張如何の判断が難しい。図1では「気持ち」と「悪い」に間には、ガ格項と述語の関係が存在するが、分析上は「気持ちが悪い」と1表現であることが望ましい。こうしたコロケーション問題に対処するため、外部知識を利用して拡張を実施する。

なお、意見タグとして採用する格助詞はガ格、ニ格、ヲ格に限定している。これは意見分析時の実用性を考慮し、以上の3つの格助詞の時に文の意味を表現する頻度が高いという経験則に基づいている。

# 3 意見タグ獲得性能とエラー分析

## 3.1 評価

先行研究 [4] では意見タグの定性的な評価にとどまっているため、本論文ではまずは獲得性能の定量評価を実施する。意見タグの獲得性能を評価するために、2.2節の手法で獲得された意見タグの正誤人手評価を実施した。評価にあたっては、不満買取センターにおいて2017年10月以降に投稿された143投稿[8]をランダムに選択し、意見タグを獲得した。次の3つの基準を設けて、それぞれ2値評価を実施した。また、次の精度指標を設定し、それぞれの基準について精度を算出した。表1に各項目の精度を示す。

| 正誤項目 | 精度 |
|---|---|
| 述語格ペア保持 | 0.920 |
| 日本語流暢性 | 0.824 |
| 情報保持性 | 0.629 |

表1. 意見タグの精度評価結果

精度 = (正しい意見タグ数)/(獲得された意見タグ数)

**述語格ペア保持**

テキストから述語と格の関係が正しく獲得されていることを評価する。

[例文] メニューが沢山ある割に、
入口の食券機で選ばなければいけない。
(誤例) メニュー-ガ-選ばなければいけない。
(正例) メニュー-ガ-沢山ある

**日本語流暢性**

獲得された意見タグのフレーズが日本語として自然であることを評価する。次の誤例では「なる」という機能語だけでは意味が成立しないため、誤りである。

[例文] 角がケガの原因になる。
(誤例) 角-ガ-なる
(正例) 角-ガ-原因になる

**情報保持性**

獲得された意見タグがテキストに言及された情報を十分に保持していることを評価する。

[例文] ティラミスと言うよりもカフェラテと
言った方が正しいような。
(誤例) 言った方-ガ-正しいような。
(正例) カフェラテと言った方-ガ-正しいような。

## 3.2 エラー分類

表1に示す評価結果から、特に情報保持性に問題があると言える。情報保持性の低さを分析するために、次のエラータイプを定義し、分類を実施した。

**構文解析誤り**

構文解析の時点で誤りが発生してしまったケース。

**意見対象部不足**

意見対象部の要素が不足しており、誤りと判断されたケース。

[例文] 服の黄ばみがとれないのが不満。
(誤例) とれないの-ガ-不満。
(正例) 服の黄ばみがとれないの-ガ-不満。

[5] 意見性の有無判別は実施しない。投稿者の主観、または事実であっても区別せずに扱う。
[6] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN
[7] http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP
[8] 評価対象は FKC コーパスに存在していないが、今後の更新では追加予定である。

| エラー項目 | エラー数 | 割合 |
|---|---|---|
| 構文解析誤り | 33 | 9.3% |
| 意見述部不足 | 23 | 6.4% |
| 文脈情報分断 | 22 | 6.2% |
| 意見対象部不足 | 20 | 5.6% |
| 複数項分断 | 19 | 5.3% |
| その他 | 19 | 5.3% |

表 2. 意見タグにおけるエラー分布。割合は意見タグの総数 354 に対する比率

**意見述部不足**

意見述部の要素が不足しており、誤りと判断されたケース。

[例文] メニューが頼んだのと違う
(誤例) メニュー-ガ-頼んだのと
(正例) メニュー-ガ-頼んだのと違う

**文脈情報分断**

獲得された意見タグは、単独ではテキストに記述された情報を十分に表現できず、かつ、先行する意見タグが後続する意見タグに対して修飾の役割を果たしているケース。

[例文] 満員電車にハイヒールで
乗る人に不満です。
(誤例) 満員電車-ニ-乗る, 乗る人-ニ-不満です。
(正例) 満員電車に乗る人-ニ-不満です。

**複数項分断**

獲得された意見タグが、単独ではテキストに記述された情報を十分に表現できず、かつ、意見述部が共通する他の意見タグと組み合わせると情報を表現できるケース。

[例文] 喫茶店はすべて禁煙にしてほしい。
(誤例) 喫茶店-ヲ-してほしい。,
禁煙-ニ-してほしい。
(正例) 喫茶店を禁煙-ニ-してほしい。

## 3.3 意見タグエラー分析とルール追加

表 2 にエラー分布を示す。最頻のエラーは構文解析誤りであったが、構文解析器自体の問題であるので、構文解析器の改良により対応する。次いで頻出エラーである意見述部不足、文脈情報分断、意見対象部不足に対処するために、次のように獲得ルールの改善を実施した。このルール追加により、獲得される意見タグが改善例を図 3 に示す。

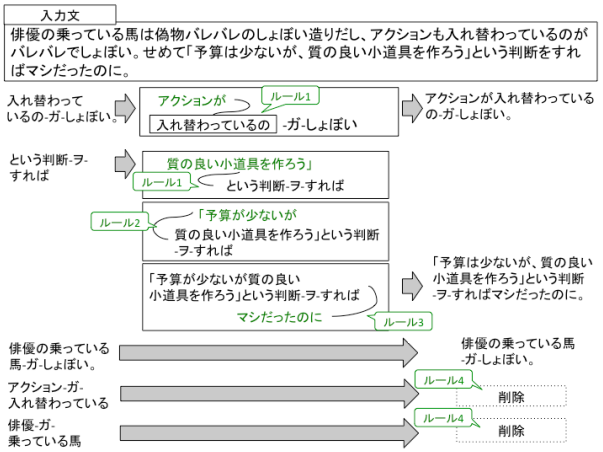


図 3. ルール追加前後の意見タグ。図中のフレーズをつなぐ線は係り受け関係を示す。

### 3.3.1 述語項関係が存在する意見対象部の不足

意見対象部に述語が含まれる場合、意見対象部の文脈が不足してしまう問題がある。例えば、図 3 の「入れ替わっているの-ガ-しょぼい」では、該当述語「入れ替わっている」では、述語項への拡張を実施していないため文脈不足が発生する。

そこで、上記の場合に述語項への拡張を行ない「アクションも入れ替わっているの-ガ-しょぼい」という意見タグを生成する。ただし、意見対象部が長くなりすぎることを防ぐために、この拡張ルールの実施は最大 1 回に制限する。この拡張ルールを**ルール 1** と呼ぶ。

### 3.3.2 直接話法部分の獲得

評判情報を示すテキストにはしばしば他者の言動や筆者の心情を示すために、鉤括弧を利用した直接話法が記述される。分析において、鉤括弧内のすべてを確認しなければ、正しく状況を理解することができない。

そこで、意見対象部内に鉤括弧を利用した話法が検出された場合は、鉤括弧で記述された全てを意見対象部に含む。この拡張ルールを**ルール 2** と呼ぶ。

### 3.3.3 節末で項を持たない述語への拡張

意見タグの獲得ルールは、項を持っている述語に限られている。したがって、図 3 の「いう判断-ヲ-すれば」のように、節末に存在し、項を持たない述語が意見述部から除外されてしまう。

そこで節末述語が項を持たない述語の場合は、意見述部を拡張する。ただし、この問題は節末で特に頻出していたことから、このルールは節末のみに適用するものとする。このルールを**ルール 3** と呼ぶ。

### 3.3.4 意見タグ間に存在する包含関係を除外

ルール 1 とルール 2 により意見タグの範囲が拡張されると、意見タグ間に包含関係が発生してしまう。例えば「アクションも入れ替わっているの-ガ-しょぼい」という意見タグは、意見対象部に「アクション-ガ-入れ替わっている」という意見タグを包含している。

| 評価項目 | ルール追加前 | KNP&ルール追加後 | KNP++&ルール追加後 |
|---|---|---|---|
| 意見タグ数 | 354 | 308 | 324 |
| 述語格ペア保持 | 0.920 | 0.915 | **0.929** |
| 日本語流暢性 | 0.824 | 0.834 | **0.851** |
| 情報保持性 | 0.629 | 0.717 | **0.759** |

表 3. 意見タグの精度評価比較

| エラー項目 | ルール追加前 | KNP&ルール追加後 | KNP++&ルール追加後 |
|---|---|---|---|
| 構文解析誤り | 33 (9.3%) | 30 (9.7%) | 24 (7.4%) |
| 意見述部不足 | 23 (6.4%) | 16 (5.1%) | 11 (3.3%) |
| 文脈情報分断 | 22 (6.2%) | 2 (0.6%) | 4 (1.2%) |
| 意見対象部不足 | 20 (5.3%) | 9 (2.9%) | 4 (1.2%) |
| 複数項分断 | 19 (5.3%) | 20 (6.4%) | 22 (6.7%) |
| その他 | 19 (5.3%) | 10 (3.2%) | 15 (4.6%) |

表 4. 意見タグにおけるエラー分布比較。括弧内割合は意見タグ総数に対する比率

このように意見タグの重複があると、分析者が情報の頻度を誤認してしまうことが起こり得る。そこで、意見タグ間に包含関係がある (部分木同士に包含関係がある) 場合は、短い意見タグを削除する。このルールを**ルール 4** と呼ぶ。

# 4 ルール追加後の意見タグ獲得性能

## 4.1 ルール追加前後での性能比較

3.3 節で述べたルールの追加前後での性能を比較した。また、構文解析器の性能とともに意見タグの精度も向上することを検証するために、形態素解析器に Juman++ [3] を、構文解析器に KNP++ [1] を利用した意見タグの精度評価を実施した。KNP++は形態素解析の N-best 解を受け取り、形態素解析と構文解析の統合的な解析によって、形態素解析の誤り伝播を抑えることができ、Web テキストでも KNP と比べた性能向上が示されている。表 3 に、結果を示す。

ルール追加により、日本語流暢性と情報保持性の項目で精度の向上を確認できた。ルール追加において述語格ペア保持の項目にて精度が低下しているが、これは獲得された意見タグの数が減ったために、意見タグ 1 つあたりの誤りが精度に大きく影響したと考えられる。

構文解析器に KNP++を利用した場合でも、同様の傾向が確認でき、さらに良い精度を確認できた。構文解析が正しく行われ、ルールが想定通りに動作したためであると想定される。

## 4.2 ルール追加後のエラーへの考察

エラー分布を表 4 に示す。「意見対象部不足」、「文脈情報分断」の項目にてエラーが大幅に減少しており、導入したルールが有効に働いていると言える。

一方で「意見述部不足」の項目では多少のエラー率減少にとどまっている。ルール 3 では、節末の述語が述語項を持っていない場合のみに有効である。エラーとなってしまったケースでは、節末の述語が子どもノードをト格項や修飾項、時間項として認識しているケースが見られた。例えば「メニューが頼んだのと違う！」からは「メニュー-ガ-頼んだのと」の意見タグが獲得されている。述語「違う」は「頼んだ」をト格と認識しており、意見タグの獲得時はト格を対象外としているため、このようなエラーが発生している。

# 5 関連研究

構文木の情報を利用して意見分析情報を獲得する研究としては、金山ら [5] が評判分析のために構文木情報から評価フレームを生成する木構造変換モデルを提案している。評価フレームによりテキストの構造化と集計が可能となる一方で、集約のために対象語の修飾語句を表現できない。分析者は意見分析の際に、対象語の修飾語句をも重視しているため、意見タグでは修飾語句も対象範囲内とする [9]。

# 6 おわりに

本論文では、意見分析の補助情報として利用できる「意見タグ」獲得改善に向けた取り組みを紹介した。特に、分析者が文脈情報を把握できるような情報を獲得するために、構文木の子どもノードへの探索ルールを追加し、そのルールの有効を示した。さらに、構文解析器の解析性能が向上するに従い意見タグの獲得性能が向上することも示した。

一方で、述語項を分断してしまったために文脈情報が失われてしまうエラーは今後の課題である。今後は文脈情報を成立させるために必要な格関係の組み合わせを分析し、文脈情報を残しつつ獲得できるような手法を導入する予定である。

# 参考文献


[1] Daisuke Kawahara, Yuta Hayashibe, Hajime Morita, and Sadao Kurohashi. Automatically Acquired Lexical Knowledge Improves Japanese Joint Morphological and Dependency Analysis. pp. 1–10. Association for Computational Linguistics, 2017.

[2] Kensuke Mitsuzawa, Maito Tauchi, Mathieu Domoulin, Masanori Nakashima, and Tomoya Mizumoto. FKC Corpus: a Japanese Corpus from New Opinion Survey Service. In *Novel Incentives for Collecting Data and Annotation from People: types, implementation, tasking requirements, workflow and results*, pp. 11–16, 2016.

[3] Hajime Morita, Daisuke Kawahara, and Sadao Kurohashi. Morphological Analysis for Unsegmented Languages using Recurrent Neural Network Language Model. pp. 2292–2297. Proceedings of EMNLP 2015: Conference on Empirical Methods in Natural Language Processing, 2015.

[4] 三澤賢佑, 成田和弥, 田内真惟人, 中島正成, 黒橋禎夫. 定量調査のための意見調査コーパス構築への取り組み. 言語処理学会第 23 回年次大会 発表論文集, pp. 1014–1017, 2017.

[5] 金山博, 那須川哲哉, 渡辺日出雄. 木構造変換を利用した評判分析手法. 人工知能学会論文誌, Vol. 26, No. 1, pp. 273–283, 2011.


[9] 意見タグにおいても、部分木のうち親ノードだけを集計対象するすれば同じく対象語の集計が可能である。