商品レビューを対象とした極性判定ならびに属性抽出に関する研究動向の調査

北陸先端科学技術大学院大学
情報科学研究科

平賀 一昭

平成 30 年 9 月

課 題 研 究 報 告 書

商品レビューを対象とした極性判定ならびに属性抽出に関する研究動向の調査

1510754 平賀 一昭

主指導教員 白井 清昭

審査委員主査 白井 清昭
審査委員 飯田 弘之
岡田 将吾

北陸先端科学技術大学院大学

情報科学研究科 [情報科学]

平成 30 年 8 月

概 要

インターネットで製品・サービスを購入する際、多くの消費者がレビューを参考にしている。レビューテキストを分析し、消費者の製品／サービスへの満足度・好感度を自動的に分析する評判情報分析は、消費者および製品・サービス提供者の双方にとって有益な情報をもたらす重要な技術である。一方で、多くの消費者がレビューを参考にすることを悪用し、自社製品への肯定的あるいはライバル商品への否定的なレビューの書き込みといったフェイクレビューや、商品と直接関係のない書き込みによるレビュー数の水増しといったレビュースパムが近年問題となっている。フェイクレビュー・レビュースパムを自動検出できれば、多くの消費者にとって有益である。

評判情報分析では、書き手の意見が肯定的か否定的かを判定する極性判定の技術を中心に研究が進められている。過去の極性判定の対象は、文書全体、文、評価対象物 (製品やサービスなど)、評価対象物の属性といったように多岐にわたり、それぞれについて多くの手法が提案されている。また、極性判定のためにどういった素性をどのように利用するのか、といった観点からも様々な手法が提案されている。評価対象物の属性を対象とした極性判定に限っても、属性の抽出、極性判定のための辞書・コーパスの作成といったタスクが存在する。属性抽出の方法に関しても、係り受け構造の解析に基づく手法、教師あり・教師なしの機械学習による手法など多岐にわたる。そのため、極性判定に関する研究の現状を把握することは容易ではない。

このような状況を踏まえ、本課題研究は、極性判定の研究動向を調査することを目的とする。極性判定に関わる技術を (1) 極性判定手法、(2) 属性抽出手法、(3) 評価語辞書作成手法、(4) スパム・フェイクレビュー判定手法、(5) 属性が評価対象のものであるかの判定手法、という 5 つのグループに分け、文献を調査し、最新の研究成果を俯瞰できるようにする。現在の極性判定技術を用いるとどういった結果が得られるのかを整理し、極性判定研究の今後の指針を検討する上で意義のある調査結果を得ることを目指す。

極性判定に関連する論文を網羅的に収集するため、文献検索エンジンと論文データベースを利用した。前者は Google Scholar と SemanticSholar を用い、後者は ACL Anthology と言語処理学会論文集を利用した。極性判定に関するキーワードを用いて論文を検索し、308 件の論文を収集した。これらのうち、論文の内容が調査対象とした 5 つのカテゴリのいずれかと合致し、かつ Google Scholar での他の論文からの参照数が 100 以上の論文を選別した。ただし、極性判定手法に関する論文は非常に多かったため、各年毎に 2 本程度という追加基準を設けた。最終的に 40 件の論文を調査対象とした。

(1) 極性判定手法に関する研究は、極性判定の対象が、文書、文、属性と属性値というように、次第に小さい範囲のテキストに移っていく傾向が見られた。ただし、極性判定のために利用される情報としては、品詞、N グラム、構文解析結果、極性コーパス、単語頻度などが一貫して用いられてきた。しかし、ここ数年は深層学習の手法を適用し、これら

の情報を明示的に与えない手法が提案されている。ただし、従来手法を大きく上回ってはいないため、極性判定研究の主流が深層学習にシフトするかは不明である。

(2) 評価対象物の属性を抽出する手法としては、トピックモデルを改良したモデルや、品詞と構文解析結果を手がかりとしたルールに基づく手法が提案されている。ここ数年は、深層学習を利用し、品詞などの素性を明示的に与えない手法も提案されている。深層学習に基づく手法は、F 値は従来手法に比べて大きく上回るが、精度と再現率の両方ともが良いというわけではない。このため、今後も属性抽出に深層学習を用いる傾向が続くとは断言できない。

(3) 評価語辞書作成手法として、例えば、数語からなる初期単語セットを与え、文書や他の言語資源から評価語を自動抽出し、その極性のタグを付与した上で辞書に追加する手法がいくつか提案されている。また、調査対象とした全ての論文において、品詞情報と構文情報が利用されていた。評価語の中には、対象ドメインによって極性が変わる語もあるため、ドメイン毎に評価語辞書を自動構築することの意義は大きい。

(4) スパム・フェイクレビュー判定手法の多くは、与えられた文がスパム・フェイクかどうかの 2 値分類を行なっている。ただし、テキストから得られる情報のみでスパム・フェイクを判定するのは一般には難しい。そのため、ユーザーのレビュー行動、例えばレーティングパターンからスパムを書くレビュアーを判定する、などの手法が提案されている。ユーザーの行動分析結果を機械学習の素性の 1 つとする手法も提案されているが、ユーザーの行動を分析できる環境が常に整っているわけではないという問題もある。

(5) ユーザが意見を述べているとして自動抽出された属性が評価対象の製品の属性と一致しているかを判定する研究については、今回の調査では見つからなかった。関連研究として、Jindal と Liu による研究事例があるが、彼らはレビュー文書全体が評価対象であるかを判定している。属性が評価対象のものであるかを判定する技術は、評価対象以外の意見文を除外するための有効な手段であり、今後の重要な研究課題と言える。

目 次

第 1 章　序論 **1**

第 2 章　調査方法 **3**
2.1　文献の収集 3
2.2　文献の分類 5

第 3 章　評判分析の研究動向 **8**
3.1　極性判定手法 8
3.1.1　極性判定手法の精査 8
3.1.2　極性判定手法の研究動向 29
3.2　属性抽出手法 30
3.2.1　属性抽出手法の精査 31
3.2.2　属性抽出手法の研究動向 43
3.3　評価語辞書作成 43
3.3.1　評価語辞書作成手法の精査 44
3.3.2　評価語辞書作成手法の研究動向 53
3.4　スパム・フェイクレビュー判定手法 54
3.4.1　スパム・フェイクレビュー判定手法の精査 54
3.4.2　スパム・フェイクレビュー判定手法の研究動向 61
3.5　属性が評価対象のものであるかの判定手法 62

第 4 章　結論 **64**

第1章　序論

インターネット通販や旅行・ホテル予約サービスにおいて、ユーザーによるレビューテキストの分析は、消費者・製品／サービス提供者の双方にとって有益な情報をもたらす重要な技術である。平成 28 年版情報通信白書によれば、インターネットで商品を購入する際、6 割強の消費者がレビューを参考にしている (図 1.1)。

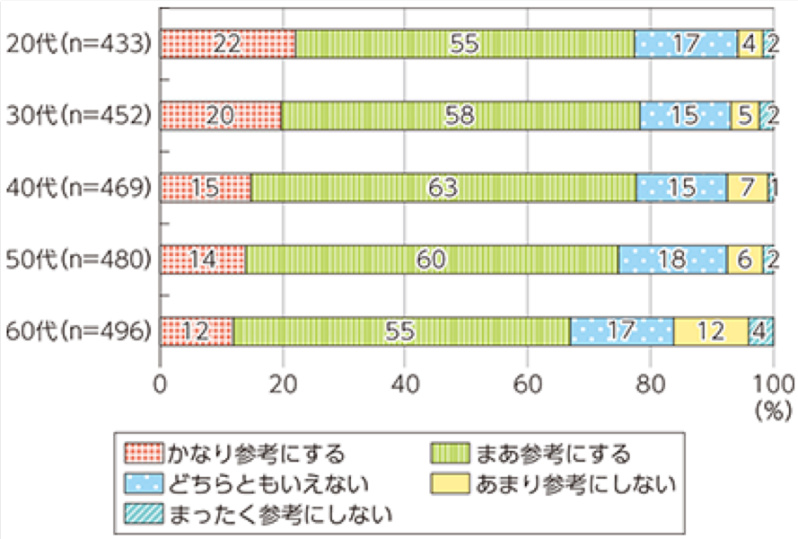


図 1.1: 消費者がレビューをどの程度参考にするかの調査 (総務省情報通信白書 図表 1-4-2-14)

レビューテキストには、(1) パソコン、スマートフォン、家電のような製品、(2) ホテル、レストランのような施設、(3) 携帯電話、接客業のようなサービスに対して、それを購入・利用したユーザの感想や意見が書かれている。これらのテキストを解析し、製品・施設・サービスといった評価対象に対する人々の評判を明らかにする技術は、「評判分析」あるいは「評判情報分析」と呼ばれている。その中でも、製品に対する意見が好意的かそうでないかを判定することを極性判定と呼ぶ。また、ユーザーの意見の評価対象の属性を検出する技術を属性抽出と呼ぶ。例えば、評価対象が PC のとき、「メモリ」や「キーボード」など、PC の属性を表す単語を抽出する。一方、自社製品への肯定的あるいはライバル商品への否定的なレビューの書き込みといったフェイクレビュー、もしくは商品と直接関係のない書き込みによるレビュー数の水増しといったレビュースパムが近年問題となっている。

極性判定の対象は、文書全体、文、評価対象物 (製品やサービスなど)、評価対象物の属性といったように多岐にわたる。文書全体の判定では、文書中で単一の評価対象物や属性について言及している場合には問題ないが、複数の対象物に対して異なる極性の意見を述べている場合を想定していない。このため、より細かい文単位への判定へと研究の主眼が移ってきた。このとき、文が主観的なものなのか、客観的なものなのかを判定し、主観的な文が意見を持つとしてその極性を判定するアプローチが一般的である。しかしながら、主観的な文が必ずしも極性を表しているとは限らないし、客観的な文中に意見を暗示させる含みのある表現が入っていないとも限らない。例えば、「先月、新しいスマートフォンを購入した。しかしバッテリーが 6 時間しか持たない。」という二文は、一見すると客観的な事実のみからなっているが、否定的な意見を暗示している。この問題の他に、文書単位での極性判定と同様に、文単位で極性を判定する場合でも、1 つの文中で複数の対象物に対して異なる極性の意見が言及されている場合を想定していないという問題点もある。このため、実際に消費者が好意的あるいは好意的でない意見を持っている対象は何かを把握するために、評価対象物もしくは評価対象物の属性が極性判定の対象となってきており、それぞれについて多くの手法が提案されている。また、極性判定のためにどういった素性をどのように利用するのかといった観点からも様々な手法が提案されている。評価対象物の属性を対象とした極性判定に限っても、属性の抽出、極性判定のための辞書の作成といったタスクが存在し、やはり多くの手法が提案されている。属性抽出の方法に関しても、係り受け構造の解析に基づく手法、教師あり・教師なしの機械学習による手法など多岐にわたる。

極性判定に関する先行研究が数多く存在するという状況を踏まえ、本課題研究は、極性判定に関わる技術を 5 つのグループに分けて文献を調査し、最新の研究成果を俯瞰できるように整理することを目的とする。現在の極性判定技術を用いるとどういった結果が得られるのかを整理し、極性判定研究の今後の指針を検討する上で意義のある調査結果を得ることを目指す。また、フェイクレビュー・レビュースパムの自動検出に関する先行研究や、極性判定技術の応用によるスパムの自動検出の実現可能性も調査対象とする。なお、5 つのグループ分けの詳細は第 2章で述べる。評判情報分析における極性判定の現状の課題や問題点を把握し、どのような技術が今後の極性判定技術の向上につながる可能性があるかを明らかにすることは、今後の研究指針を決めるための重要な資料となり、多くの研究者にとって有益な情報を提供できるという点で有意義である。

第2章　調査方法

本章では、第1章で述べた評判情報分析に関する研究動向の調査のために、どのように関連論文を収集し、整理し、精査したかについて述べる。

2.1　文献の収集

評判情報分析に関する研究動向を調査するために、まず関連する論文を網羅的に収集した。主に以下の2つの方法を用いた。

- 文献検索エンジンによる論文収集

  学術論文を検索できる検索エンジンを利用し、適切なキーワードをクエリとして検索することで、関連研究の論文を収集した。本課題研究で利用した文献検索エンジンは以下の2つである。

  - Google Scholar[1]
  - SemanticScholar[2]

  検索クエリとして用いたキーワードの一覧を表2.1に示す。

表 2.1: 検索キーワードの一覧

sentiment analysis, polarity identification, aspect identification aspect extraction, aspect based sentiment analysis, fake review, spam review, 感情分析, 極性分析, 評判分析, 意見分析, 評判情報, 意見情報, 極性判定, 極性抽出, 評価 属性, 評価 アスペクト, スパム レビュー , フェイク レビュー

- 論文データベースからの論文収集

  国内および海外のいくつかの自然言語処理関連の学会では、自然言語処理に関連する論文のデータベースを公開している。これらの論文データベースから、評判情報

[1] https://scholar.google.com
[2] https://www.semanticscholar.org

分析に関する論文を選別、収集した。本課題研究で利用した論文データベースは以下の 2 つである。

- ACL Anthology[3]
  アメリカの学会 The Association for Computational Linguistics が公開している論文データベースである。同学会が発行しているジャーナル Computational Linguistics の他、以下の自然言語処理に関連する主要国際会議の論文を閲覧できる。
  - ACL(Annual Meeting of the Association for Computational Linguistics)
  - EACL(Conference of the European Chapter of the Association for Computational Linguistics)
  - NAACL(Annual Conference of the North American Chapter of the Association for Computational Linguistics)
  - CONLL(Conference on Computational Natural Language Learning)
  - EMNLP(Conference on Empirical Methods in Natural Language Processing)
  - SemEval(Lexical and Computational Semantics and Semantic Evaluation)
- 言語処理学会発表論文集[4]
  国内の言語処理学会が毎年開催している「言語処理学会年次大会」で発表された論文が収録されている。同学会のウェブサイトで公開され、自由に閲覧できる。

ACL Anthology では、上記の主要国際会議の 2012 年から 2017 年までの 5 年分の論文を取得した。また、言語処理学会発表論文集についても同様に、2012 年から 2017 年までの論文を取得した。次に、PDF リーダーで論文の内容を確認し、文献検索エンジンによる論文収集の際に用いた表 2.1と同様のキーワードで検索し、調査対象とする論文を選択した。言語処理学会発表論文集の場合は、アーカイブに同梱されている HTML ファイルにより論文タイトルが一覧できるため、まずはタイトルを基準に検索し、内容を確認した上で選択した。なお、2012 年から 2017 年までの 5 年分を中心に収集しているが、これより以前のものでも、他の研究に大きな影響を与えたと考えられる研究については調査対象としている。

文献検索エンジンならびに論文データベースから網羅的に収集した論文を一次調査対象論文と呼ぶ。一次調査対象論文の中から、本課題研究で内容を精査する論文を選別した。選別する際に 2 つの基準を設けた。まず、評判情報分析には様々なタスク設定 (問題設定) があるが、一次調査対象論文を概観し、調査対象とするタスク設定を 5 つに絞るこ

[3] https://aclanthology.info/
[4] http://www.anlp.jp/guide/nenji.html

とにした。調査対象とした5つのタスク設定の詳細は2.2節で述べる。1つ目の基準は、5つのタスク設定のいずれかに関する手法について述べている論文を選別するという基準である。2つ目は、ある程度著名な論文を選別するという基準である。具体的には、Google Scholar における他の論文からの参照数が100以上の論文を選別した。ただし、5つのグループの中の一つは極性判定手法に関する論文であるが、参照数が100以上の論文数が非常に多かったため、年毎に参照数が多い論文を2本程度選ぶという追加基準を設けた。一方、極性判定手法以外では、収集できた論文数が少なかったため、参照数が100以下のものであっても選択している。一次調査対象論文の数は308であったが、上記の基準により、最終的に調査対象とする文献を40件に絞り込んだ。

調査対象として選別した文献は、文献管理サービス／ソフトウェアのMendeley、Zoteroを用いて分類・整理を行った。両者とも、クラウドに文献を保存する機能、会議名や著者名などのメタデータを追加する機能、フォルダによる分類機能、注釈を記述する機能、タグ付け機能、検索機能などを提供している。ブラウザで利用するだけでなく、デスクトップアプリケーションとモバイルアプリケーションも提供されている[5]。無料の基本プランと有料のプランがあり、Mendeleyでは2GBまで、Zoteroでは300MBまでの保存領域が無料プランで利用できる。有料プランでは、前者は共同作業が行えるグループを作成できたり、保存領域を5GB、10GBあるいは無制限まで拡張できる。後者の場合は、保存領域の容量変更のみが行え、2GB、6GBまたは無制限まで拡張できる。

2.2 文献の分類

評判情報分析といっても、その具体的な目的(タスク設定)は様々である。収集した文献を概観し、評判情報分析として具体的にどのようなタスク設定が多くなされているのかを調べた。さらに、タスク設定の違いによって、収集した論文を以下の5つのグループに分類した。

1. 極性判定

   極性判定とは、文章、文、製品の属性などに対し、それが書き手の肯定的な意見を表すのか、否定的な意見を表すのかを分類するタスクである。極性判定の手法を提案している文献を集め、その技術動向を調査した。

2. 属性抽出

   属性抽出とは、与えられたテキストから、評価対象物の属性を抽出するタスクである。一般に、対象物の極性を判定するだけでは不十分で、対象物の属性に対する極性を判定することがしばしば求められる。例えば、A社のパソコンに対して肯定的か否定的かだけではなく、A社のパソコンのうち価格に対しては肯定的であるがメ

[5]Zoteroの場合、公式のものは無く、サードパーティから提供されている。

モリに関しては否定的である、といった分析を行うことが求められる。評価対象物の属性抽出手法を提案している文献をまとめ、その技術動向を調査した。

3. 評価語辞書作成

   評価語とは、書き手の何らかの評価を表す単語である。評価語の例としては、「よい」「すばらしい」「悪い」「ひどい」などがある。評価語辞書は評価語のデータベースである。評価語に対して、それが肯定的が否定的かを表す極性情報が付与されていることも多い。評価語辞書には、人手によって作成されたものと自動構築されたものがある。評価語辞書の自動構築、あるいは既存の評価語辞書を自動拡張する手法を提案している文献を集め、その技術動向を調査した。

4. スパム・フェイクレビュー判定

   スパム・フェイクレビュー判定とは、与えられた文書や文が、スパム・フェイクレビューかを判定するタスクである。スパム・フェイクレビューの代表的なものには、ある商品・サービスに対して、全く同じ文面の好意的・非好意的なレビューが複数の消費者から投稿されているもの、ほぼ同じ内容の文面の好意的・非好意的なレビューがいろいろな商品・サービスに対して投稿されているもの、質問、広告、ブランドや小売店・サービス提供者など評価対象物以外に対するレビュー、といったものがあり、これらを判定することが求められる。スパム・フェイクレビュー判定の手法を提案している文献をまとめ、その技術動向を調査した。

5. 属性が製品について言及しているかの判定

   レビューテキストでは、評価対象以外の属性に対して意見や感想が述べられることがある。例えば、パソコンのレビューの中で、宅配業者の遅延に関するクレームが書かれることがある。クレーム自体は否定的な意見であるが、評価対象であるパソコンに対する意見ではない。評価対象に関する評判を正確に把握するためには、評価対象以外の属性に関する意見を除外する必要がある。

   上記を踏まえ、2の手法によって自動的に抽出された属性が、実際に評価対象物、すなわち製品について言及しているかを判定する手法をまとめ、その研究動向を調査した。

上記の 5 つのグループについて、収集した文献数とリストを表 2.2 に示す。極性判定に関する文献が一番多いが、初期の収集段階ではさらに多い 188 本あった。これは、極性判定というタスクは評判情報分析の初期の段階から研究されてきており、その歴史が長いからと考えられる。2 番目に多い属性抽出に関しては、特に 2013 年以降増えている。これは、SemEval-2014[1] において、タスクとして Aspect Based Sentiment Analysis、すなわち属性ベースの極性判定タスクが実施されたことが影響していると言える。評価語辞書作成に関しての文献数は想定よりも少なかった。本課題研究では、極性判定と属性抽出の研究動向の調査に主眼を置いており、これらのタスクに利用する評価語辞書を用意するた

めにどのような手法があるかを調べることを目的にこのグループを設けている。したがって、極性判定や属性抽出をキーワードとして含むような論文を選別しており、このことが評価語辞書作成に関する論文数が少ないことの一因になっている可能性がある。スパム判定に関しては、選択した文献数は 8 本と少なかったが、初期の収集段階では 25 本あった。しかし、スパムやフェイクレビューの性質に関する研究の調査報告や、スパム・フェイクレビューが与える社会的・経済的な影響、スパムレビューの書き手グループの判定など、スパムやフェイクを自動判定する研究とは直接関係のないものも多かったため、選択した本数が少なくなった。製品属性判定に関する文献は、調査段階では見つけることができなかったが、収集した論文の内容を確認している過程で見つけることができた。ただ、10 年近く前の文献であり、インターネット通販などの進化を考えると、この手法がどれだけ有効なのは疑問が残る。

表 2.2: グループ毎の文献数と文献リスト

グループ	文献数	文献リスト
1 極性判定	16	[2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17]
2 属性抽出	10	[18] [19] [20] [21] [22] [23] [24] [25] [26] [27]
3 評価語辞書作成	6	[28] [29] [30] [31] [32] [33]
4 スパム判定	7	[34] [35] [36] [37] [38] [39] [40]
5 製品属性判定	1	[34]

第 3章では、上記の 5 つのグループについて実施した研究動向の調査結果を報告する。

第3章　評判分析の研究動向

3.1　極性判定手法

“極性判定” とは、ユーザが表明した意見が肯定的か否定的かを判定する技術である。極性判定は、極性分析、感情分析、意見分析などといった様々な名称で呼ばれている。本課題研究報告書では、これらを統一して極性判定と呼ぶ。

3.1.1　極性判定手法の精査

本項では、極性判定手法に関する個々の論文の内容を概観する。

Turney の研究

Turney は、同氏が 2001 年に提案した PMI-IR(Pointwise Mutual Information and Information Retrieval) アルゴリズム [41] を用いて、句を対象とした極性判定を行う手法を提案している [2]。この手法は、1) 入力文の品詞タグ付け、2) 主観表現抽出ルールにマッチした句の抽出、3) 抽出した句の極性判定、という 3 つのステップから構成される。

ステップ 2) では、表 3.1に示す抽出ルールに合致する単語列を主観表現として抽出する。抽出ルールにおいて、1 番目か 2 番目の単語のどちらかは形容詞 (JJ) もしくは副詞 (RB,RBR,RBS) となっている。また、3 番目の単語はパターンマッチには用いられるが、主観表現としては抽出されない。形容詞や副詞が主観的な意見を表すというのは、Hatzivassiloglou らや Wiebe らによる報告 [42, 43, 44] をもとにしている。ただし、単に形容詞や副詞を抽出するだけでは不十分だとし、前記のような抽出ルールを定義して主観表現となる句を抽出している。

表 3.1: Turney による主観表現抽出ルールの定義 [2]

	First Word	Second Word	Third Word (Not Extracted)
1.	JJ	NN or NNS	anything
2.	RB, RBR, or RBS	JJ	not NN nor NNS
3.	JJ	JJ	not NN nor NNS
4.	NN or NNS	JJ	not NN nor NNS
5.	RB, RBR, or RBS	VB, VBD, VBN, or VBG	anything

ステップ 3) の極性判定には PMI-IR アルゴリズムを用いる。これは、Church と Hanks が提案した PMI(自己相互情報量) による手法 [45, 46] をもとにしたもので、抽出した句と極性を表す単語の共起確率を検索エンジンを利用して推定する手法である。PMI は、2 つの単語の共起関係の強さを測る指標で、式 (3.1) により算出される。この PMI を基に、極性 (SO:Semantic Orientaton) スコアを式 (3.2) のように定義する。主観表現の句 *phrase* がポジティブな句を判別するキーワード “excellent” と強い共起関係を持てば SO スコアは正となり、ネガティブな句を判別するキーワード “poor” との共起関係が強ければ負となる。

$$\mathrm{PMI}(word_1, word_2) = \log_2 \frac{\mathrm{p}(word_1, word_2)}{\mathrm{p}(word_1)\mathrm{p}(word_2)} \tag{3.1}$$

$$\mathrm{SO}(phrase) = \mathrm{PMI}(phrase, \text{“excellent”}) - \mathrm{PMI}(phrase, \text{“poor”}) \tag{3.2}$$

式 (3.3) は SO スコアの実際の計算式である。この式は式 (3.2) の対数となっている。また、$p(word_1, word_2)$ を hits($phrase$ NEAR X) で、$p(word)$ を hits(X) で近似している (X は “excelent” もしくは “poor” を表す)。hits(C) は C をクエリとしたときのウェブ検索エンジンのヒット件数を表し、“A NEAR B” は「A と B が近傍に出現する」ことを条件としたクエリを表す。Turney は、AND クエリ (2 つの単語が同一ウェブページ上に存在するときにヒットする) よりも NEAR クエリの方が単語の関連性の強さを測るのに適していると報告している [41]。

$$\mathrm{SO}(phrase) = \log_2 \frac{\mathrm{hits}(phrase \text{ NEAR “excellent”})\mathrm{hits}(\text{“poor”})}{\mathrm{hits}(phrase \text{ NEAR “poor”})\mathrm{hits}(\text{“excellent”})} \tag{3.3}$$

提案手法では、検索エンジンに AltaVista を利用している[1]。この理由は、AltaVista が検索クエリで NEAR をサポートしていたためである。なお、AltaVista の NEAR クエリでは、指定された単語が 10 文字以内にある場合にヒットする。なお、Turney は、PMI に基づく SO スコアによる極性判定手法と、Landauer と Dutnais による LSA を利用した極性判定手法 [47] を比較し、前者の方が正解率が高かったと報告している [41]。

表 3.2は、提案手法による極性判定の正解率 (Accuracy) と、SO スコアとレビュアーによって与えられた 1 から 5 までの星の数の相関 (Correlation) を示している。この結果から、Turney は、SO スコアとレビュアーによる星の数との相関度が強い、と主張している。しかし、レビュー対象のドメインによる差が大きく、銀行 (Banks) を除くと相関係数は 0.5 以下である。このため、判定の正解率は高いと言えるが、星の数との相関度に関してはそれほど高いとは言えない。

[1] 2003 年に米 Yahoo!社に買収され、サービスは終了している。

表 3.2: Turney の手法の実験結果 [2]

Domain of Review	Accuracy	Correlation
Automobiles	84.00 %	0.4618
Honda Accord	83.78 %	0.2721
Volkswagen Jetta	84.21 %	0.6299
Banks	80.00 %	0.6167
Bank of America	78.33 %	0.6423
Washington Mutual	81.67 %	0.5896
Movies	65.83 %	0.3608
The Matrix	66.67 %	0.3811
Pearl Harbor	65.00 %	0.2907
Travel Destinations	70.53 %	0.4155
Cancun	64.41 %	0.4194
Puerto Vallarta	80.56 %	0.1447
All	74.39 %	0.5174

Pang らの研究

Pang らは、単純ベイズ分類器、最大エントロピー分類器、サポートベクターマシンの 3 つの教師あり機械学習手法と、ユニグラムやバイグラムなどの異なる素性の組み合わせについて、極性判定の正解率を実験的に比較し、最適な機械学習手法と素性の組み合わせを調査している [3]。極性判定モデルの学習と評価には、正・負のラベルが付けられた映画レビューの公開データを利用している[2]。実験結果を表 3.3に示す。“Features” は利用した素性を、“# of features” は素性数を表す。“NB”、“ME”、“SVM” は、それぞれ機械学習アルゴリズムとして単純ベイズ、最大エントロピー、サポートベクターマシンを用いたときの極性判定の正解率を表す。太字はこれら 3 つの中で正解率が一番高いモデルを示している。

表 3.3: Pang らによる機械学習手法と素性の組み合わせの比較 [2]

	Features	# of features	frequency or presence?	NB	ME	SVM
(1)	unigrams	16165	freq.	**78.7**	N/A	72.8
(2)	unigrams	”	pres.	81.0	80.4	**82.9**
(3)	unigrams+bigrams	32330	pres.	80.6	80.8	**82.7**
(4)	bigrams	16165	pres.	77.3	**77.4**	77.1
(5)	unigrams+POS	16695	pres.	81.5	80.4	**81.9**
(6)	adjectives	2633	pres.	77.0	**77.7**	75.1
(7)	top 2633 unigrams	2633	pres.	80.3	81.0	**81.4**
(8)	unigrams+position	22430	pres.	81.0	80.1	**81.6**

実験では、まず第 1 に、ユニグラム (単語) を素性としたとき、素性の頻度を学習に用いることの効果を検証している。表 3.3の “unigrams” は、少なくとも 4 回コーパス中に出現した単語を素性として利用することを表す。該当する素性数は 16,165 であった。“frequency or presence?” の列は、素性の重みとして、単語の出現頻度 (freq.) を用いるか、1 または 0

[2] http://www.cs.cornell.edu/people/pabo/movie-review-data/

すなわち存在しているか否か (pres.) を用いるかを表す。ただし、最大エントロピー法では単語の出現頻度を考慮することはできない。(1) と (2) の結果を比較すると、単語の有無を素性の重みとした方が出現頻度を重みとしたときよりも正解率が高い。特に SVM では 10 ポイント以上の差が見られる。一方、McCallum と Nigam は、機械学習によるテキスト分類では、頻度が重要な役割を持っていると報告している [48]。今回の極性判定の実験では反対の結果が得られたが、この理由として、テキスト分類では同じ単語が繰り返し使われることはテキストのトピック判定の有効な手がかりとなるため、頻度を学習に利用した方が結果がよいと考察している。この実験結果を踏まえ、以降の実験では、素性の重みを 1 または 0 と設定するモデル (pres.) のみを検証する。

第 2 に、素性として単語バイグラムを用いることの効果を検証した。表 3.3の “bigrams” は、7 回以上出現したバイグラムのうち出現頻度が上位のものを素性として用いることを示す。バイグラムの素性数はユニグラムと同じ数とした。(3)unigrams+bigrams や (4)bigrams の正解率は (2)unigrams と比べて同等もしくは低いため、バイグラムはそれほど有効な素性ではないことがわかった。

第 3 に、品詞の情報を素性として用いることの効果を検証した。(5)unigrams+POS は、ユニグラムとその品詞の情報を素性として用いることを示す。これと (2)unigrams を比較すると、NB はわずかに改善したが、ME は変わらず、SVM は逆に低下した。また、Hatzivassiloglou と Wiebe の研究 [49] や Turney の研究 [2] では、極性を示す句を抽出する手がかりとして形容詞が重要であることが報告されている。このため、(6)adjectives では形容詞のみを素性したモデルの正解率を調べている。比較のため、出現頻度の大きいユニグラムを形容詞と同じ数 (2633) だけ用いたとき ((7)top 2633 unigrams) の正解率も求めた。(6) の正解率は (2) と比べて大きく劣ることから、形容詞だけを素性とすることは有効な方法ではないことがわかる。さらに、(6) と (7) の比較により、同じ数の素性を使うなら、品詞 (形容詞) で素性選択するよりも出現頻度で素性選択した方がよいことがわかる。

(8)unigrams+position の “position” は、文書内の前半・後半の 1/4 に出現する、中程に出現する、といった素性の出現位置の情報を素性として付加することを表す。(2) との比較から、場所の素性を利用することによる正解率の改善は見られなかった。

3 つの機械学習アルゴリズムのうち、全般的には SVM の正解率が高い。最高の正解率が得られたのは、素性として (2)unigrams を用いて学習した SVM であり、その正解率は 82.9%であった。

Pang らは、判定誤りの主な原因として、レビュー文における “対照的な意見の表明” を指摘している。つまり、肯定的な意見から始まり、“However” や “But” などの逆接の接続詞以降で否定的な意見を表明するようなレビューである。こういったレビューは bag-of-features による分類器では文脈がわからないことから、判定が難しいと推測している。そして、次の重要なステップは、文がトピックに言及していること、あるいは言及していないことを示唆する素性を見つけ出すことだとしている。

Wilson らの研究

Wilson らは、教師あり機械学習によって句単位で極性分類を行なう手法を提案している [6]。ここでの問題設定は、レビュー中の句の極性がポジティブ、ネガティブ、両方、中立のいずれかであるかを分類することである。また、単語が持つ潜在的な極性を判定するのではなく、文脈に応じた句の極性を判定することを目的とする。Wilson らは以下の文を例として挙げている。

> Philip Clapp, president of the National Environ-ment Trust, sums up well the general thrust of the reaction of environmental movements: "There is no reason at all to believe that the polluters are suddenly going to become reasonable."

この文における下線のついた単語は、極性辞書ではポジティブな極性を持っているが、これらは文全体がポジティブな意見を表明していることを示唆しない。"reason" は否定されることでネガティブな極性を示しているし、"no reason at all to believe" はそれに続く句の極性が反対であることを示唆している。この例のように、Wilson らは、特定の文脈に出現する句の極性を正しく判定することを目指している。

提案手法は以下の 2 つのステップからなる。(1) 句が極性表現を含むか含まないか (中立か否か) を分類する "中立性分類" と、(2) 先のステップで極性表現を含むとされた句の極性 (ポジティブ、ネガティブ、両方、中立) を判定する "極性分類" である。句が中立か否かはステップ (1) で判定されるが、中立の句が誤って極性ありと分類される場合も多いため、ステップ (2) でもう一度中立かどうかを判定する。中立性分類も極性分類も、機械学習手法として AdaBoost アルゴリズム [50] を用い、Schapire と Singer によって開発された BoosTexter AdaBoost.HM[51] を分類器として利用している。

ステップ (1) の中立性判定では、表 3.4に示す 28 種類の素性を用いている。"Word Features" は、句に出現する単語や潜在極性辞書から得られる情報を素性とする。潜在極性辞書 (Prior-Polarity Subjectivity Lexicon) とは、評価語のリストであり、評価語自体が持つ潜在的な極性や、評価語の主観性の強さ (強い主観性 (strongsubj) もしくは弱い主観性 (weaksubj)) の情報を含む。"Modification Features" は、分類対象の句と係り受け関係ある語を素性とする。"Sentence Features" は、強い主観性を持つ語の数や形容詞の数など、分類対象の句が含まれる文の情報を素性とする。"Structure Features" は、分類対象の句の構文情報を素性とする。"Document Feature" は、分類対象の句が含まれるテキストのトピックを素性とする。

表 3.4: Wilson らによる中立性分類手法で利用されている素性の一覧 [6]

Word Features	Sentence Features	Structure Features
word token	strongsubj clues in current sentence: count	in subject: binary
word part-of-speech	strongsubj clues in previous sentence: count	in copular: binary
word context	strongsubj clues in next sentence: count	in passive: binary
prior polarity: positive, negative, both, neutral	weaksubj clues in current sentence: count	
reliability class: strongsubj or weaksubj	weaksubj clues in previous sentence: count	
Modification Features	weaksubj clues in next sentence: count	**Document Feature**
preceeded by adjective: binary	adjectives in sentence: count	document topic
preceeded by adverb (other than not): binary	adverbs in sentence (other than not): count	
preceeded by intensifier: binary	cardinal number in sentence: binary	
is intensifier: binary	pronoun in sentence: binary	
modifies strongsubj: binary	modal in sentence (other than will): binary	
modifies weaksubj: binary		
modified by strongsubj: binary		
modified by weaksubj: binary		

中立性分類の実験結果を表 3.5に示す。“Acc” は中立性分類の正解率、“Rec”、“Prec”、“F” はそれぞれ極性ありクラス (Polar) と中立クラス (Neut) の再現率、精度、F 値を示している。また、素性セットとして、単語のみを素性とする “word token”、単語と潜在極性辞書の情報を素性とする “word+priorpol”、全ての素性を用いる “28 features” の 3 つを比較している。実験結果から、提案する全ての素性を用いた分類器の性能が最も良いことがわかる。ただし、Polar クラスの再現率が顕著に低いが、その原因については考察されていない。

表 3.5: Wilson らによる中立性分類の実験結果 [6]

	Acc	Polar Rec	Polar Prec	Polar F	Neut Rec	Neut Prec	Neut F
word token	73.6	45.3	72.2	55.7	89.9	74.0	81.2
word+priorpol	74.2	54.3	68.6	60.6	85.7	76.4	80.7
28 features	75.9	56.8	71.6	63.4	87.0	77.7	82.1

ステップ (2) の極性分類では、表 3.6に示す 10 種類の素性を用いている。“Word Features” は中立性分類で用いられた素性と同じである。“Polarity Features” は、否定表現の有無 (negated) や、極性の反転を示唆する表現の有無 (general polarity shifter) など、文脈に依存する極性を判定する手がかりとなる情報である。

表 3.6: Wilson らによる極性分類手法で利用されている素性の一覧 [6]

Word Features
word token
word prior polarity: positive, negative, both, neutral
Polarity Features
negated: binary
negated subject: binary
modifies polarity: positive, negative, neutral, both, notmod
modified by polarity: positive, negative, neutral, both, notmod
conj polarity: positive, negative, neutral, both, notmod
general polarity shifter: binary
negative polarity shifter: binary
positive polarity shifter: binary

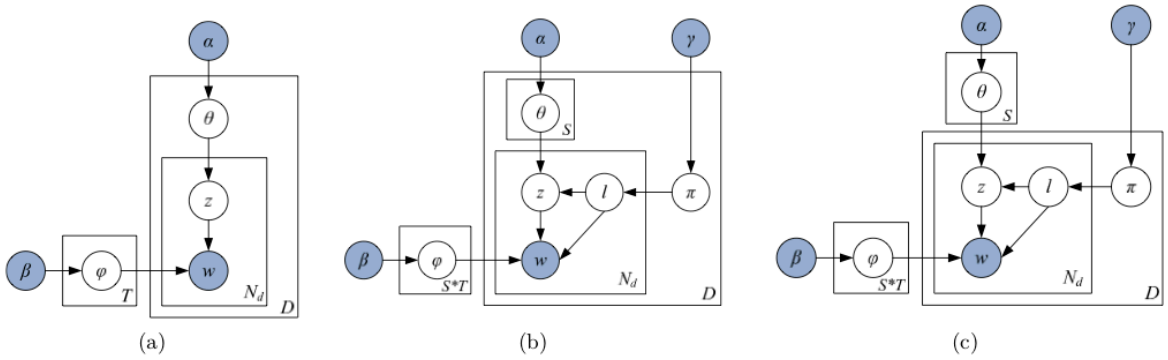
極性分類の評価結果を表3.7に示す。中立性分類と同様に、“word token” や “word+priorpol” と比べて、Polarity Features を含む全ての素性を用いた分類器 “10 features” の性能が最も良い。ただし、“word token”(ベースライン) と “10 features”(提案手法) を比較すると、提案手法は、ポジティブクラス (Positive) については再現率が大きく上回るのに対し、ネガティブクラス (Negative) については精度が大きく上回るという違いが見られる。また、両方クラス (Both) の F 値が他のクラスと比べて低いのは、訓練データにおいて極性を両方持つ句の数が少ないためと考えられる。

表 3.7: Wilson らによる極性分類手法の評価結果 [6]

		Positive			Negative			Both			Neutral		
	Acc	Rec	Prec	F	Rec	Prec	F	Rec	Prec	F	Rec	Prec	F
word token	61.7	59.3	63.4	61.2	83.9	64.7	73.1	9.2	35.2	14.6	30.2	50.1	37.7
word+priorpol	63.0	69.4	55.3	61.6	80.4	71.2	75.5	9.2	35.2	14.6	33.5	51.8	40.7
10 features	65.7	67.1	63.3	65.1	82.1	72.9	77.2	11.2	28.4	16.1	41.4	52.4	46.2

Lin と He の手法

Lin と He は Joint Sentiment/Topic Model（JST）と呼ぶ Latent Dirichlet Allocation（LDA、潜在ディリクレ分配法）を改良したモデルにより極性とトピックを同時に抽出する手法を提案し、サポートベクターマシンを利用した既存手法 [52] と同程度の精度を達成したと報告している [10]。図 3.1は、LDA と JST および JST の派生型として提案している Tying-JST モデルのグラフィカルモデル（プレート表現）である。



(a)LDA モデル　(b)JST モデル　(c)Tying-JST モデル

図 3.1: LDA, JST, Tying-JST のグラフィカルモデル [10]

まず簡単に LDA について説明したあと、提案手法である JST について説明する。LDA はトピックモデルの手法の 1 つで、確率的生成モデルとして提案された。文書には複数の潜在トピックが存在すると仮定し、その分布を離散分布としてモデル化する。単語の出現頻度の違いがトピックの特徴を表していると仮定している。単語がどの潜在トピックによって生成されたかを示す潜在変数を用い、この値が同じ単語は同じトピックに属する、

というモデル化を行う。図 3.1 (a) において、D は文書集合を表す。この時、文書 d に出現する単語数（文書長）を N_d、文書中の 1 単語を w とし、対応する潜在変数を z とする。θ は文書におけるトピックの出現確率を表し、$\theta_{d,t}$ は文書 d でトピック t が出現する確率（文書 d でのトピック t の構成比率）となる。φ はトピックにおける単語の出現確率で、$\varphi_{t,v}$ はトピック t における単語 v の出現確率となる。θ や φ は確率ベクトルであり、式 (3.4) のように Dirichlet 分布による生成を仮定する。

$$\left.\begin{array}{ll} \theta_d \sim \mathrm{Dir}(\alpha) & (d = 1, \cdots, D) \\ \varphi_t \sim \mathrm{Dir}(\beta) & (t = 1, \cdots, T) \end{array}\right\} \tag{3.4}$$

α はトピックの出現確率の偏りを、β は単語の出現確率の偏りを表す。

JST ではこのモデルを拡張し、図 3.1 (b) に示すように、極性ラベル集合 S、文書 d における極性ラベル l の出現確率 π_d、極性ラベルの出現確率の偏りを表す γ を導入している。

- 各文書 d における極性ラベルの出現確率 $\pi_d \sim \mathrm{Dir}(\alpha)$
- 文書 d における極性ラベル l の出現確率 $\theta_{d,l} \sim \mathrm{Dir}(\alpha)$
- 文書 d における単語 w_i の生成過程

  極性ラベルを選択 $l_i \sim \pi_d$

  トピックを選択 $z_i \sim \theta_{d,l}$

  単語 w_i をトピック z_i と極性ラベル l_i の出現確率 φ_{zi}^{li} から選択

上記のような処理により極性とトピックを同時に抽出する[3]。一方、図 3.1 (c) の Tying-JST モデルは JST のバリエーションである。JST が θ_d により全ての文書のトピックの出現確率を保持するのに対し、Tying-JST の場合は文書集合全てにおいて 1 つの θ によってトピックの出現確率を保持する。

評価実験では実験データとして映画のレビューデータを用い、2 つの観点で提案手法を評価している。第 1 に、JST と Tying-JST は教師なし機械学習手法であるが、いくつかの種類の事前知識を与えて半教師ありでモデルを学習させ、その品質がどれだけ向上するかを検証する。第 2 に、4 つの先行研究と比較する。実験結果を表 3.8に示す。まず、表 3.8における 1 列目の “Prior Information” は、与える事前情報（教師データ）の種類を表す。事前情報の詳細を表 3.9 に示す。ただし、下から 4 行は比較した先行研究の文献中での名称となる。先行研究の詳細を表 3.10 に示す。次に、表 3.8の 2 列目は事前情報として与えた評価語の数を表す。3 列目、4 列目は JST と Tying-JST を用いて文書の極性判定をしたときの正解率である。ポジティブ、ネガティブ、全体の 3 つの正解率が掲載されている。

[3]正確には単語が極性ラベルならびにトピックに属する確率を推定する。

[4]http://mpqa.cs.pitt.edu

[5]http://www.cs.cornell.edu/people/pabo/movie-review-data/

[6]http://alias-i.com/lingpipe/demos/tutorial/sentiment/read-me.html

表 3.8: JST と Tying-JST の評価 [10]

Prior information	# of polarity words (pos./neg.)	JST (%)			Tying-JST (%)		
		pos.	neg.	overall	pos.	neg.	overall
Without prior information	0/0	63	56.6	59.8	59.2	53.8	56.5
Paradigm words	21/21	70.8	77.5	74.2	74.2	71.3	73.1
Paradigm words + MI	41/41	76.6	82.3	79.5	78	73.1	75.6
Full subjectivity lexicon	1335/2214	74.1	66.7	70.4	77.6	69	73.3
Filtered subjectivity lexicon	374/675	84.2	81.5	82.8	84.6	73.1	78.9
Filtered subjectivity lexicon (subjective MR)	374/675	96.2	73	84.6	89.2	74.8	82
Pang *et al.* (2002) [16]	N/A	Classifier used: SVMs			Best accuracy: 82.9%		
Pang and Lee (2004) [15] (subjective MR)	N/A	Classifier used: SVMs			Best accuracy: 87.2%		
Whitelaw *et al.* (2005) [24]	1597 appraisal groups	Classifier used: SVMs			Best accuracy: 90.2%		
Kennedy and Inkpen (2006) [10]	1955/2398	Classifier used: SVMs			Best accuracy: 86.2%		

表 3.9: 表 3.8における Prior Information の詳細

Without prior information	事前情報なし。つまり教師なし学習。
Paradigm words	少量のポジティブ・ネガティブな単語のリスト。
Paradigm words + MI	上記のリストに、ポジティブ・ネガティブクラスとの相関が高い単語を加えたリスト。相関は相互情報量 (MI) で測る。ただし、文献では手順の詳細は説明されていない。
Full subjectivity lexicon	MPQA 主観表現辞書[4] のうち、実験に使った映画レビューデータに出現した単語全て。
Filtered subjectivity lexicon	上記のリストのうち、映画レビューデータでの出現頻度が 50 未満の単語、語幹抽出により極性が変わる単語を削除したもの。
Filtered subjectivity lexicon (subjective MR)	上記と同じ事前知識を与え、かつ前処理として主観的な文をまず選別し、その後極性判定を行う方法 [53] による実験結果。subjective MR は主観的な文のみを抽出した映画レビューデータを表す。主観性の判定は、Subjectivity v1.0 dataset[5] を用いて、LingPipe パッケージ[6] によって学習した分類器を用いる。

表 3.10: 表 3.8における先行研究の詳細

Pang et al. (2002)	サポートベクターマシン (SVM) による分類手法 [3]
Pang and Lee (2004)	文をグラフ構造と見なした上でグラフカットにより極性分類を行う手法 [53]
Whitelaw et al. (2005)	Adjectival Appraisal Groups と呼ぶ形容詞的な動きをする極性語辞書と SVM による分類手法 [54]
Kennedy and Inkpen (2006)	Valence Shifters と呼ばれる否定形などの極性を変化させる語を考慮した極性辞書と SVM による分類手法 [55]

表 3.8 では、JST を利用した “Filtered subjectivity lexicon (subjective MR)” が提案手法の中では全体 (overall) で最も高い正解率を示した。ただし、Lin と He は、先行研究の手法に対する JST の強みとして完全な教師なし学習手法であることを主張しており、一方 “(subjective MR)” では教師あり学習による分類器を利用しているため、この値は参考値とみなすべきである。純粋な教師なし学習は “Without prior information” であるが、わずかな事前知識を与えることで先行研究の教師あり機械学習手法に近い正解率が得られることがわかった。また、JST と Tying-JST との比較では、軒並み JST の方が高い正解率を示した。

図 3.2は、JST モデルにおいて、抽出するトピック数と極性分類の正解率との関係を示している。(a) から (f) は、表 3.8の “Prior Information” に対応している。

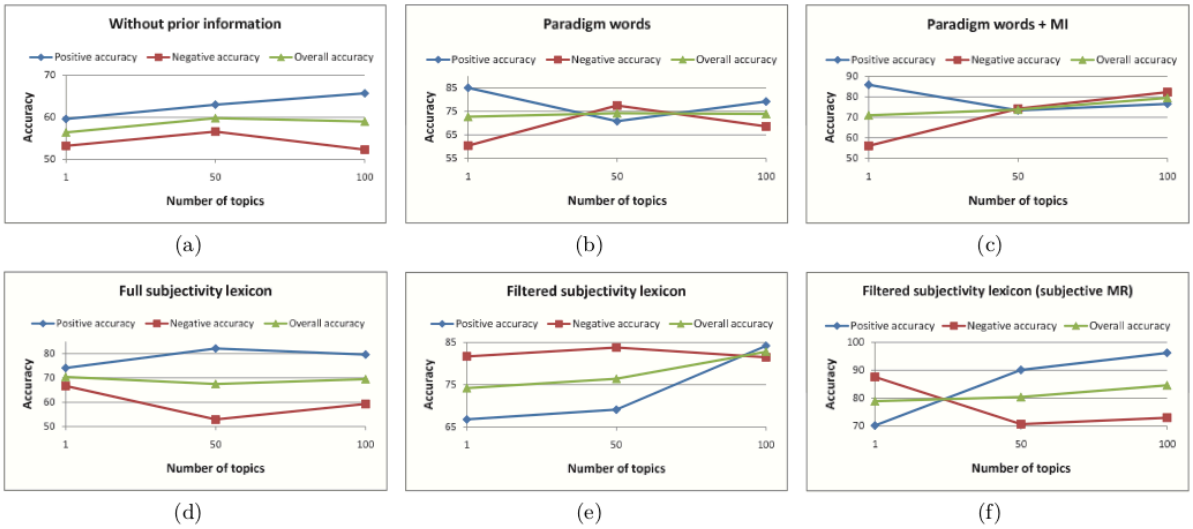


図 3.2: JST による抽出トピック数と極性分類の正解率の関係 [10]

トピック数を 1 とした場合、図 3.1(b) の S(極性) のみが抽出されることになる。つまり、極性ラベルとトピックとの関連性は無視されることになる。Overall の正解率に関しては、(d) を除き、トピック数を増やした方が、すなわち極性ラベルとトピックとの関連性を考慮した方が正解率が高くなる。表 3.8の結果でも言えることだが、(d) の “Full subjectivity lexicon” の結果が低調なことから、事前知識として与える極性辞書のサイズを大きくすることは必ずしも効果的ではないことがわかる。

Kim の研究

Kim は、Mikolov らによって事前に作成された単語分散表現 [56] を入力とし、畳み込みニューラルネットワーク (Convolutional Neural Network, CNN) を用いて文レベルでの極性分類を行う手法を提案している [14]。図 3.3は提案手法のネットワークモデルを表す。ネットワークは 1 つの畳み込み層、1 つのプーリング層（Max プーリング）、1 つの全結

合層からなる。左側の入力層の $n \times k$ 行列が 2 重になっているのは 2 つのチャンネルを表す。また入力文字列に対しては、必要であればパディングを行い長さを合わせる。すなわち、文の最大長 n をあらかじめ決めておき、入力文の単語数がそれより小さい場合には 0 ベクトルを埋める。

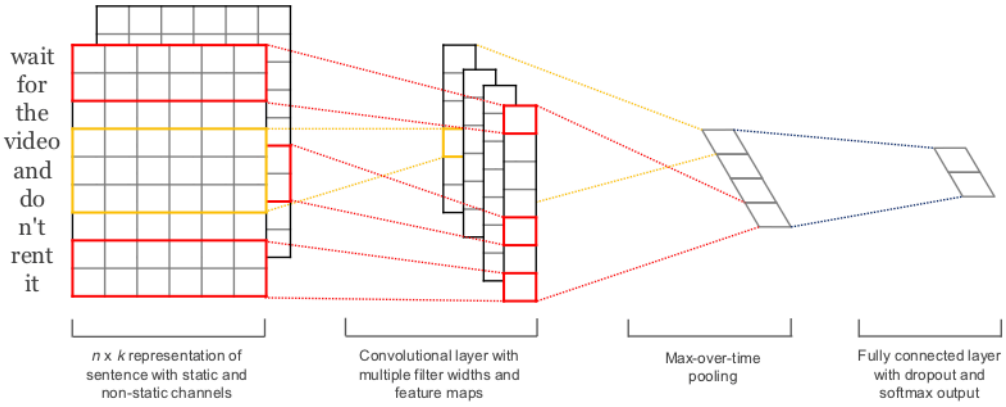


図 3.3: Kim による提案手法のネットワークモデル [14]

ある文における i 番目の語に対する k 次元の単語分散表現を $\mathbf{X}_i \in \mathbb{R}^k$ とする。長さ n の入力単語列を式 (3.5) のように表す。

$$\mathbf{x}_{1:n} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \ldots \oplus \mathbf{x}_n \tag{3.5}$$

ここで $\oplus$ は単語ベクトルの結合を表す演算子である。$\mathbf{x}_{i:i+j}$ は、$\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+j}$ の結合であり、入力文の部分単語列を表す。次に、部分単語列に対しウィンドウ幅 h のフィルター $\mathbf{w} \in \mathbb{R}^{hk}$ を適用する。$x_{i:i+h-1}$ から c_i を生成する場合を式 (3.6) に示す。

$$c_i = f\left(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b\right) \tag{3.6}$$

$b \in \mathbb{R}$ はバイアス、f は双曲線正接関数のような非線形な関数である。ウィンドウを一定間隔でスライドさせながらフィルターを適用し、特徴マップを生成する。$\{\mathbf{x}_{1:h}, \mathbf{x}_{2:h+1}, \ldots, \mathbf{x}_{n-h+1:n}\}$ は長さ n の入力文に対してウィンドウ幅 h のフィルタをずらしたときに得られる行列の列であり、これから式 (3.7) のような特徴マップ $\mathbf{c} \in \mathbb{R}^{n-h+1}$ が生成される。

$$\mathbf{c} = [c_1, c_2, \ldots, c_{n-h+1}] \tag{3.7}$$

この特徴マップに対して Max プーリング [57] を適用することにより最大値 $\hat{c} = \max\{\mathbf{c}\}$ を取得し、フィルターの出力とする。これにより、各特徴マップの中で最も重要な特徴を

取得できる。最後の全結合層 (Fully connected layer) は、特徴マップを入力とし、極性クラスを出力とするフィードフォワードネットワークである。

表 3.11は Kim らによる評価実験の結果である。2 行目から 5 行目は提案手法のバリエーションである。各手法の概略を以下に述べる。

CNN-rand ベースラインとなるモデル。入力単語のベクトルをランダムに初期化し、CNN 学習の際に勾配を逆伝搬させることでベクトルの値も更新する。

CNN-static word2vec による単語分散表現を入力とする。単語ベクトルの値は更新しない。

CNN-non-static word2vec による単語分散表現を入力とし、CNN 学習の際に単語ベクトルの値も更新する。

CNN-multichannel "static" と "non-static" の 2 つのチャンネルを利用する。

表 3.11の 6 行目以降は先行研究の結果を示す。概要を表 3.12に示す。実験に利用したデータセットは、**MR**(映画レビュー)[58]、**SST-1**(Stanford Sentiment Treebank データ)[13]、**SST-2**(SST-1 から中立のレビューを削除したデータ)、**Subj**(主観的/客観的のラベルが付与された文)[53]、**TREC**(TREC 質問データセット)[59]、**CR**(商品のレビュー記事データ)[60]、**MPQA**(極性判定のためのデータセット)[61] の 7 つである。

提案手法は畳み込み層が 1 つのシンプルなネットワーク構造ながら高い正解率を示すことが確認された。CBOW モデルによる単語の分散表現は他の研究でも広く使われ、比較的良好な成果が得られることが多いため、これを入力とする "CNN-rand" 以外のモデルの正解率が高いことは理解できる。しかし、これを入力としない "CNN-rand" の SST-1、SST-2 データセットにおける正解率が先行研究と同じ程度であることは予想外であった。ただ、より現実に近いデータであると考えられる "MR"、"CR" では、CBOW モデルによる単語分散表現を入力としたモデルの方が正解率が顕著に高い。

表 3.13は、"CNN-multichannel" における Static/Non-static チャンネルで入力/学習した単語の分散表現によって算出された類似度の高い単語の例を示している。学習データは SST-2 を用いている。Static Channel(CBOW モデルであらかじめ学習された単語分散表現) では、"bad" に似ている単語として、"terrible" などの意味が近い単語が得られているが、"good" のような統語的には似ているが意味的には似ていない単語も得られる。一方、Non-static Channel では "good" との類似度が高くないことから、CNN の学習の際に単語ベクトルの値を同時に更新することで極性判定に適した分散表現が得られている。

表 3.11: Kim による提案手法の評価実験結果 [14]

Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	**89.6**
CNN-non-static	**81.5**	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	**88.1**	93.2	92.2	**85.0**	89.4
RAE (Socher et al., 2011)	77.7	43.2	82.4	–	–	–	86.4
MV-RNN (Socher et al., 2012)	79.0	44.4	82.9	–	–	–	–
RNTN (Socher et al., 2013)	–	45.7	85.4	–	–	–	–
DCNN (Kalchbrenner et al., 2014)	–	48.5	86.8	–	93.0	–	–
Paragraph-Vec (Le and Mikolov, 2014)	–	**48.7**	87.8	–	–	–	–
CCAE (Hermann and Blunsom, 2013)	77.8	–	–	–	–	–	87.2
Sent-Parser (Dong et al., 2014)	79.5	–	–	–	–	–	86.3
NBSVM (Wang and Manning, 2012)	79.4	–	–	93.2	–	81.8	86.3
MNB (Wang and Manning, 2012)	79.0	–	–	**93.6**	–	80.0	86.3
G-Dropout (Wang and Manning, 2013)	79.0	–	–	93.4	–	82.1	86.1
F-Dropout (Wang and Manning, 2013)	79.1	–	–	**93.6**	–	81.9	86.3
Tree-CRF (Nakagawa et al., 2010)	77.3	–	–	–	–	81.4	86.1
CRF-PR (Yang and Cardie, 2014)	–	–	–	–	–	82.7	–
SVM_S (Silva et al., 2011)	–	–	–	–	**95.0**	–	–

表 3.12: 表 3.11における先行研究の概要

RAE [62]	Autoencoder を再帰的に適用したモデル。句の分散表現を単語分散表現の合成により求めている。
MV-RNN [63]	再帰的ニューラルネットワークによる極性分類手法。単語と句をベクトルと行列の両方で表現する。単語 (句) の行列は近傍の単語ベクトルを含む。
RNTN [13]	再帰的ニューラルテンソルネットワークモデル。子ノードの単語ベクトルから再帰的に親ノード (句または文) のベクトルを求める。
DCNN [64]	プーリング時に k 個の最大値を取得する Dynamic k-Max プーリング層を導入した手法。
Paragraph-Vec [65]	Paragraph vector を利用した手法。bag-of-words モデル、bag-of-n-grams モデルと異なり、単語の順序を保持する。
CCAE [66]	CCG パーサーにより得た構文情報を入力とする再帰的な Autoencoder。
Sent-Parser [67]	Sentiment Grammar の定義とそのためのパーサーによる手法。Sentiment Grammar は、文脈自由文法 [68] を基に定義された極性分類のための文法である。
NBSVM, MNB [69]	単語ユニグラム・バイグラムを素性とし、ナイーブベイズ、SVM、多項分布によるナイーブベイズ分類器を組み合わせたモデル。
G-Dropout, F-Dropout [70]	Dropout によって学習速度の低下を抑える手法。ガウス分布を利用した近似 (G-) と閉形式の近似 (F-) による手法を比較している。
Tree-CRF [71]	入力文の構文木をサブツリーに分け、それらの極性を判定した上で、CRF を適用する手法。
CRF-PR [72]	CRF の学習に Posteriro Regularaization(PR) と呼ばれる制約を利用する手法。
SVM_S [73]	単語ユニグラム・バイグラム・トライグラムを素性とした SVM。

表 3.13: CNN-multichannel モデルで得られる類似度の高い単語 [14]

	Most Similar Words for	
	Static Channel	Non-static Channel
bad	*good* *terrible* *horrible* *lousy*	*terrible* *horrible* *lousy* *stupid*
good	*great* *bad* *terrific* *decent*	*nice* *decent* *solid* *terrific*
n't	*os* *ca* *ireland* *wo*	*not* *never* *nothing* *neither*
!	*2,500* *entire* *jez* *changer*	*2,500* *lush* *beautiful* *terrific*
,	*decasia* *abysmally* *demise* *valiant*	*but* *dragon* *a* *and*

Wang らの研究

Wang らは、アテンションを利用した LSTM(Long Short-Term Memory) による属性の極性判定手法を提案している [15]。RNN(Recurrent Neural Network, 再帰型ニューラルネットワーク) は、従来のニューラルネットワークを時系列データを取り扱えるように拡張したモデルである。文 (単語列) は時系列とみなせることから、RNN は自然言語処理によく適用されるが、勾配消失・勾配爆発問題により遠く離れた単語間の依存関係を学習できないという問題点がある。この問題に対処するために LSTM が提案された。一般的な LSTM の構造を図 3.4に示す。

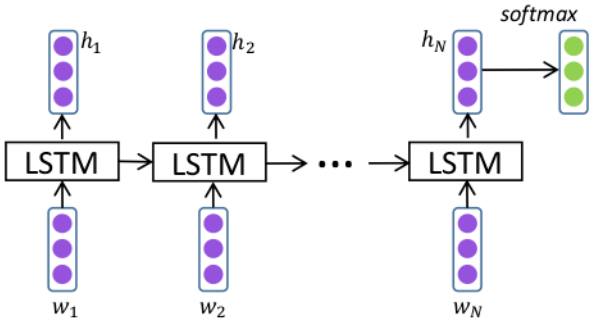


図 3.4: LSTM の構造 ([15] より抜粋)

$\{w_1, w_2, \ldots, w_N\}$ は長さ N の単語ベクトルの列であり、$\{h_1, h_2, \ldots, h_N\}$ は隠れ状態ベクトルを表す。LSTM における各メモリセルは式 (3.8)〜(3.13) により算出される。

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \tag{3.8}$$

$$f_t = \sigma\left(W_f \cdot X + b_f\right) \tag{3.9}$$

$$i_t = \sigma\left(W_i \cdot X + b_i\right) \tag{3.10}$$

$$o_t = \sigma\left(W_o \cdot X + b_o\right) \tag{3.11}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh\left(W_c \cdot X + b_c\right) \tag{3.12}$$

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{3.13}$$

$W_i, W_f, W_o \in \mathbb{R}^{d \times 2d}$ は重み付き行列、$b_i, b_f, b_o \in \mathbb{R}^d$ はバイアスで、両者ともそれぞれ、input ゲートの出力 i_t、forget ゲートの出力 f_t、output ゲートの出力 o_t のパラメータとなる。σ はシグモイド関数、$\odot$ はベクトルの要素毎の積を示す。h_t は隠れ層のベクトルである。x_t は LSTM の記憶セルへの入力であり、図 3.4の w_t に相当する。図中の最後の隠れ状態ベクトル h_N は文全体の意味内容を表すベクトルであり、これが *softmax* 層へ渡される。提案手法では positive, negative, neutral の 3 つのクラス、もしくは positive, negative の 2 つのクラスへ分類される。

図 3.5は提案手法の 1 つであるアテンションを利用した LSTM (Attention-based LSTM: AT-LSTM) モデルの構造を表す。アテンションとは、ここでは文中の各単語が属性の極性判定にどの程度大きく寄与するかを表すパラメタである。v_a は極性判定の対象とする属性の分散表現を、α はアテンションの重みを、r はアテンションを考慮した文の分散表現を表す。r は次の式で推定される。

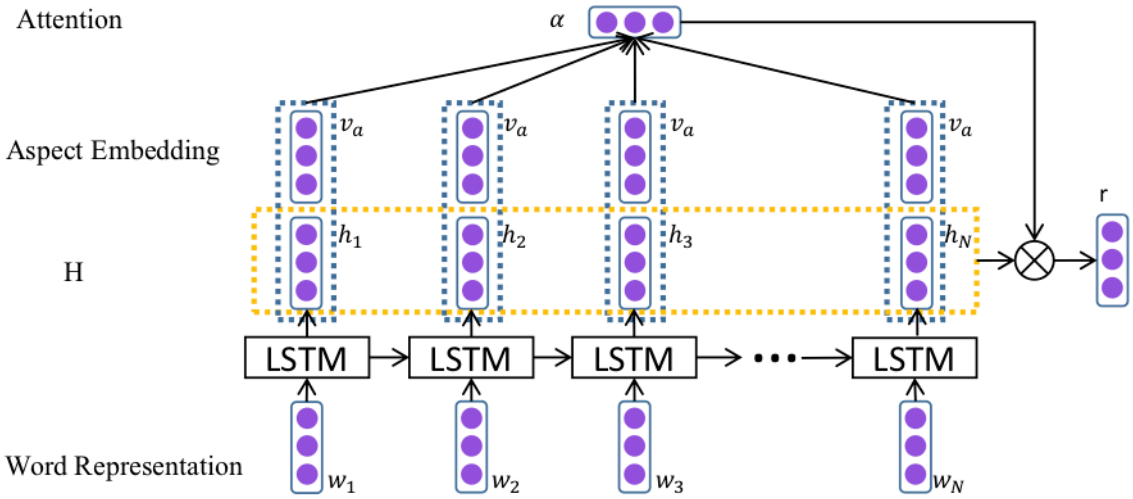


図 3.5: AT-LSTM の構造 [15]

$$M = \tanh\left(\begin{bmatrix} W_h H \\ W_v v_a \otimes e_N \end{bmatrix}\right) \tag{3.14}$$

$$\alpha = \mathrm{softmax}\left(w^T M\right) \tag{3.15}$$

$$r = H\alpha^T \tag{3.16}$$

$H \in \mathbb{R}^{d \times N}$ は隠れ層のベクトル h_i を連結した行列である (d は隠れ層のノード数、N は入力単語数)。一方、$M \in \mathbb{R}^{(d+d_a)\times N}, \alpha \in \mathbb{R}^N, r \in \mathbb{R}^d$、$W_h \in \mathbb{R}^{d\times d}, W_v \in \mathbb{R}^{d_a \times d_a}, w \in \mathbb{R}^{d+d_a}$ は学習パラメタである。式 (3.14) の M は隠れ層のベクトルと属性のベクトルをまとめたものであり、これをもとに α が式 (3.15) によって計算される。α はアテンションの重み、つまり各単語の重要度を表す。r は、式 (3.16) に示すように隠れ層のベクトル h_i の α を重みとする線形和であり、これが属性の情報を含んだ文の分散表現となる。

$h^* \in \mathbb{R}^d$ は最終的な文の分散表現であり、式 (3.17) のように r と h_N から求められる。

$$h^* = \tanh\left(W_p r + W_x h_N\right) \tag{3.17}$$

最後に、h^* が *softmax* 層の入力として与えられ、出力 y が得られる (式 (3.18))。

$$y = \mathrm{softmax}\left(W_s h^* + b_s\right) \tag{3.18}$$

さらに、Wang らは、属性の分散表現を入力とし、アテンションを利用した LSTM (Attention-based LSTM with Aspect Embedding: ATAE-LSTM) モデルを提案している。その構造を図 3.6に示す。図 3.5の AT-LSTM との違いは、入力として単語の分散表現 w_i と属性の分散表現 v_a を連結したベクトルを与えている点にある。

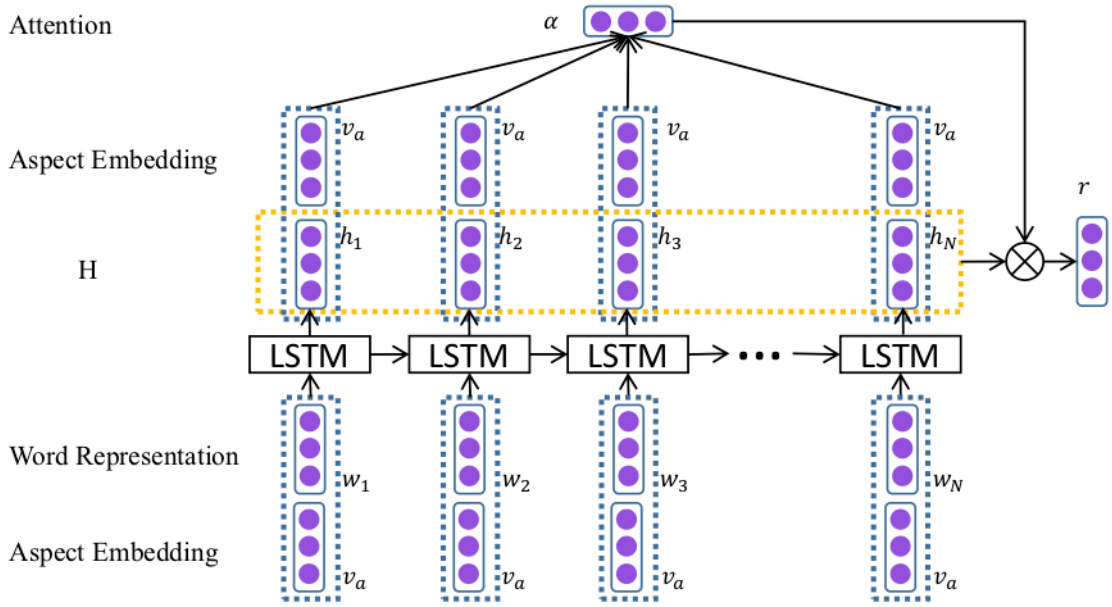


図 3.6: ATAE-LSTM の構造 [15]

表 3.14は評価実験における 3 つの提案手法および先行研究の極性判定の正解率を示している。“Three-way” は positive, negative, neutral の 3 つのクラス分類を示し、“Pos./Neg.” はポジティブかネガティブかの二値分類の結果である。先行研究のモデルのうち、LSTM は一般的な LSTM であり、TD-LSTM と TC-LSTM は Tang らによって提案された Target-Dependent LSTM (TD-LSTM) と Target-Connection LSTM (TC-LSTM)[74] である。TD-LSTM は、ターゲットとなる属性（“battery life” など）の左右で文を分割し、それぞれについて LSTM で単語列の意味表現を得て、これらを合成したベクトルを *softmax* 層への入力とする手法である。TC-LSTM は、TD-LSTM を各語の埋め込み、ベクトルを合成するように拡張した手法である。一方、AE-LSTM は提案手法のバリエーションであり、LSTM に属性の分散表現を入力として与える。

表 3.14: Wang らの手法の評価実験結果 [15]

Models	Three-way	Pos./Neg.
LSTM	82.0	88.3
TD-LSTM	82.6	89.1
TC-LSTM	81.9	89.2
AE-LSTM	82.5	88.9
AT-LSTM	83.1	89.6
ATAE-LSTM	**84.0**	**89.9**

実験結果より、ATAE-LSTM の性能の高さがわかる。ベースラインとなる一般的な LSTM でも “Pos./Neg.” の正解率は高いが、ATAE-LSTM はそれを上回る。さらに “Three-way”

では、LSTM だけでなく TD-LSTM、TC-LSTM よりも 2 ポイント程度高い正解率を示している。

図 3.7は提案手法でのアテンションの効果を視覚化したものである。色が濃いほどアテンションベクトル α での重要度が大きいことを示す。例文 (a)、(b) において判定の対象とした属性は service と food であり、それぞれの属性に関連の深い単語の重要度が高いと判定されていることがわかる。

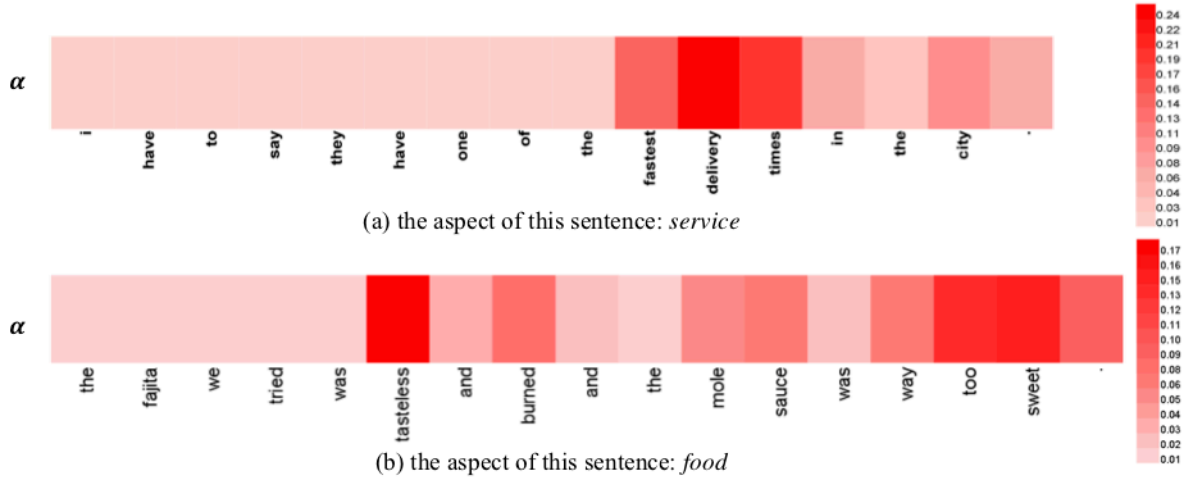


図 3.7: Wang らの手法におけるアテンションの効果を視覚化した結果 [15]

極性判定の例を図 3.8に示す。(a) では 2 つの属性 “food” と “service” の極性が正しく判定されている。(b) では属性 “food” の極性が直前の “not” の影響を受けずに正しく判定されている。(c) では複雑に入り組んだ構造の文からも属性の極性が正しく判定できていることがわかる。

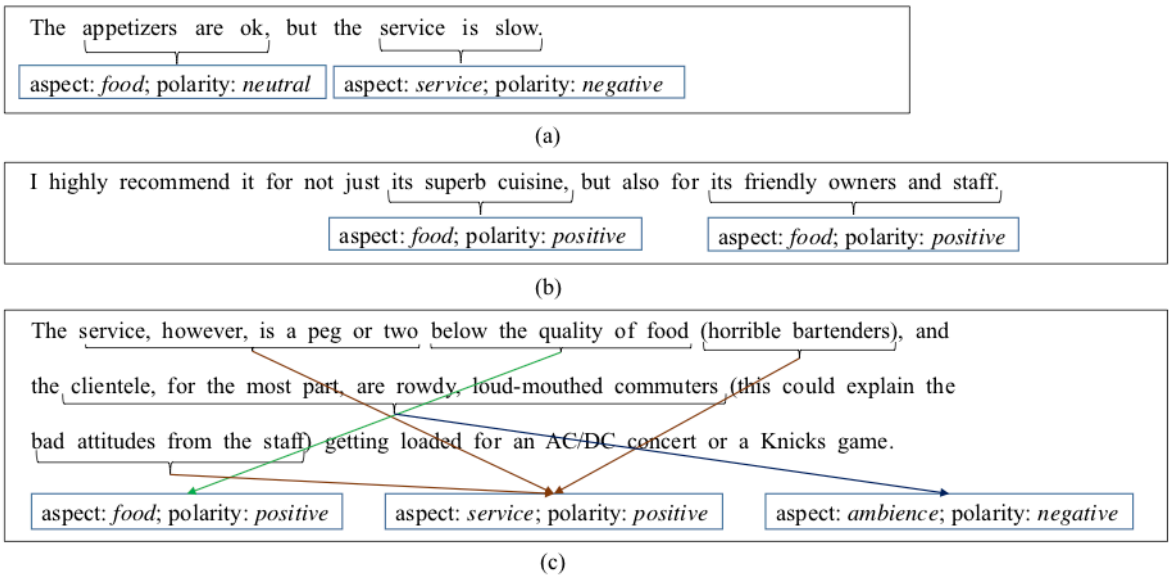


図 3.8: Wang らの手法による極性分類の例 [15]

その他の研究

Yu と Hatzivassiloglou は、文が意見を述べたものかそうでないかの判定を行う分類器と、意見文として判定した文の極性を判定する分類器をそれぞれ実装し、文レベルでの極性判定を行う手法を提案している [4]。意見文の判定には 3 種類の手法を実装し比較している。1 つ目は SIMFINDER[75] によって計算された文の近似度による手法、2 つ目はナイーブベイズ分類器による手法、3 つ目は複数のナイーブベイズ分類器を用いる手法である。3 番目の手法は異なる素性セットで複数の分類器を学習するもので、単語、バイグラム、トライグラム、品詞、極性情報の 5 つの素性セットを用いる。一方、極性判定では、ポジティブ、ネガティブ、中立の 3 つのクラスに分類している。Hatzivassiloglou と McKeown によって選択されたポジティブ・ネガティブな形容詞 [42] との共起の頻度によって文の極性を判定する。評価実験では、ナイーブベイズ分類器による意見文判定で 90% 程度、極性判定でも最大 90% という高い正解率を示した。

Hu と Liu は収集した製品レビューの要約を行うことを目的とし、要点となる極性を抽出することを試みている [5]。極性抽出には品詞、N グラム、極性辞書による単語の極性などの素性を利用する。まず、相関ルールマイニングにより出現頻度の高い語を全て抽出し、ヒューリスティクによって属性となる語を選別する。次に、形容詞の近くに頻度の高い語があった場合、その形容詞を意見語とみなす。また頻度の低い属性を抽出するため、既知の意見語の近くにある名詞・名詞句を属性とみなす。文の極性を判定する際には、文中に出現する複数の意見語の極性をブートストラップ法と WordNet[76] で決め、その多数決により決める。評価実験では、提案手法の再現率が 0.80、精度が 0.72 と、高水準を示した。

Kim らはサポートベクター回帰による教師あり機械学習によって、商品レビューの有

用性をモデル化し判定することを試みている [7]。判定のための素性として、構造的、語彙的、統語的、意味的、メタデータの 5 種類の素性を利用している。例として、レビューのテキスト長や HTML タグなどの文章構造の情報、ユニグラム・バイグラムなどの語彙的情報、名詞・形容詞の割合のような統語情報、製品属性・極性語辞書のような意味的情報、レビューの星の数のようなメタデータ、などがある。評価実験では、ユーザーが “役に立った” とするレビュー記事と、これらの素性から計算されるレビュー記事のランクとの一致度を測る。Amazon.com の MP3 プレーヤー、デジタルカメラに関するレビュー記事において、スピアマンの順位相関係数はそれぞれ 0.656 と 0.60 となった。

Blitzer らは、自身が提案した Structural Correspondence Learning(SCL) [77] による極性分類手法を改良した手法を提案している [8]。ここでの目標は、ソースドメインとターゲットドメインという 2 つの異なる分野のテキストデータがあり、ソースドメインのみ正解のラベルが付与されているという状況において、ターゲットドメインのデータの極性判定の正解率を向上させることにある。まず、ソースとターゲットの両方のドメインで頻度の高い単語が “pivot” と呼ばれる素性として抽出される。そして、この “pivot” とそれ以外の素性との相関関係から、“pivot” でない素性を選択する。評価実験では、上記の SCL と、“pivot” 選択のときに頻度だけでなく相互情報量も利用する SCL-ML を比較している。学習したドメインと同じドメインが対象であれば 80.4%-87.7%、異なるドメインを対象にした場合でも概ね 70%以上の精度を示した。

Titov と McDonald は、MG-LDA（Multi-grain LDA）と呼ばれる教師なし機械学習のトピックモデルを用いた極性判定手法を提案している [9]。MG-LDA は、LDA を拡張したもので、文書集合全体と文書ごとの局所的なトピックを個別に抽出する。局所的なトピック抽出は、文書内で隣接する文を仮想的な文書とみなすことで行う。文書集合全体のトピックをレビュー対象となる商品の属性とみなす。一方、文書ごとの局所的なトピックは属性値とみなす。評価実験では、MP3 プレーヤーのレビュー記事を対象とし、トピック抽出と極性判定の 2 つのタスクを評価する。トピック抽出の評価では、局所的なトピックとして “sound”,“quality”,“headphones”,“volume”,“bass”, 全体的なトピックとして “ipod”,“music”,“apple”,“songs”,“use”,“mini” が抽出された。極性判定の実験では、MG-LDA は LDA と比べてランキング損失が 0.03 程度改善した。

Pak と Paroubek は Twitter のツイートデータから極性タグ付きコーパスを作成し、これを用いて極性を判定する分類器を学習した [11]。30 万ツイートを収集し、ポジティブな表現を含むもの、ネガティブな表現を含むもの、感情表現の含まれないもの、という 3 つに分類する。ポジティブ・ネガティブの判定には顔文字を利用している。例えば、:-), :), =), :D はポジティブ、:-(, :(, =(, ;(はネガティブなツイートを表す顔文字である。一方、客観的なツイートは New York Times、Washington Posts などのニュース社の公式ツイートを利用している。作成した極性タグ付きコーパスでは、URL の削除、“@” などの特殊記号の削除、品詞タグ付け、ストップワードの削除などの前処理が行われている。このコーパスを訓練データとし、単語 N グラムを素性としてナイーブベイズ分類器を学習する。評価実験では、ユニグラム、バイグラム、トライグラム、品詞、否定を付加した N

グラムなどを素性とし、これらの組み合わせを変えて分類器をいくつか学習し、それらの性能を比較している。ただし、他の手法との比較は行なっていない。実験の結果、バイグラムと品詞タグを素性として利用した場合に正解率が最も高かった。

Taboada らは、極性とその強さ、否定の情報も含む極性辞書を利用した極性分類手法 Semantic Orientation CALculator (SO-CAL) を提案している [12]。極性辞書の語には 5 から −5 までの Semantic Orientation(SO) 値が与えられている。この値はその語の極性と強さを表している。文における各語の SO 値の合計がその文の SO 値となる。また、SO 値の正負が文の極性 (ポジティブかネガティブか) を表す。評価実験では、ネガティブレビューの判定の正解率が 89.37%、ポジティブレビューでは 63.2%、全体では 76.37% となった。同じデータセットを利用した Pang と Lee の手法 [53] の極性分類の正解率が 87.15% であったことから、ネガティブレビューに関しては同程度と言えるが、全体では他の手法に比べてそれほど優れているとは言えない。

Socher らは、Stanford Sentiment Treebank コーパスと Recursive Neural Tensor Network（RNTN）との組み合わせにより極性判定を行う手法を提案している [13]。Sentiment Treebank とは、構文木のサブツリーに対して極性ラベルを与えたコーパスである。文の構造を考慮して極性を判定する点に特徴がある。RNTN では、入力文は 2 分木にパースされ、単語が葉となる。親のベクトルは、子のベクトルからボトムアップに計算される。評価実験では、RNN、同じ著者による先行研究の MV-RNN[63] などと提案手法を比較している。文の極性判定の正解率は、MV-RNN が 79.0% であるのに対し、提案手法は 85.4% となり、大幅に向上した。また、“X but Y” のような否定を表す接続詞を含む文のみを対象としたとき、正解率は 41% となった。RNN、MV-RNN の正解率は 30% 後半であったため、こちらも正解率の向上が確認できた。さらにポジティブな文の否定、ネガティブな文の否定を対象とした実験を行った。前者では文の極性がネガティブに変化するが、後者の場合はネガティブの強さは減るが必ずしもポジティブに変わるわけではないことから、極性判定が難しいタスクと言える。ポジティブな文の否定では、RNN、MV-RNN、RNTN の正解率はそれぞれ 33.3%、52.4%、71.4% となり、ネガティブな文の否定ではそれぞれ 45.5%、54.6%、81.8% となった。両者ともに提案手法は既存の手法を大きく上回ることが確認できた。以上の実験結果は、Sentiment Treebank コーパスと RNTN に基づく提案手法は否定を含む文の極性を正しく評価することができるようになったことを示している。

芥子らは人手によって構築した単語意味ベクトル辞書を用いて単語拡張したコーパスからパラグラフベクトルを学習し、それを素性として極性判定の分類器を学習する手法を提案している [16]。文長が短いツイートに対して単語拡張をすることで、文脈情報を追加する。例えば、「新しい製品 B を推す」というツイートであれば、基本単語「新しい」「推す」に共通して関連する特徴単語「流行・人気」「価値・質」「優良」「肯定的」「強力」が新たな情報として追加される。拡張後のツイートに対し、語順の情報を利用する PV-DM モデルと、語順の情報を利用しない PV-DBOW モデルによりパラグラフベクトルを生成する。最後に、パラグラフベクトルを素性として、文の極性を判定するモデルをサポートベクターマシンにより学習する。評価実験では、提案手法の F 値は、Le と Mikolov によ

るパラグラフベクトルを用いた手法 [65] よりも、3 クラス分類、2 クラス分類ともに 9-10 ポイント上回った。

Chen らは、双方向 LSTM(Bidirectional LSTM,BLSTM) に再帰的な Attention 機構を追加することで複数の属性の極性を考慮できる手法を提案している [17]。文中の属性毎にメモリを用意し、属性に合わせて利用するメモリを変えてアテンションスコアを計算し、GPU への入力とする。メモリ上で数回のアテンションを適用し、最終的なエピソードを素性として利用する。本提案手法では品詞、極性辞書などの言語情報による素性は使っていない。入力は CBOW モデルによる単語分散表現を利用した。評価実験では、サポートベクターマシンや Tang らによる TD-LSTM[74] などと比較し、正解率、F 値共に提案手法の方が高いことを確認した。

3.1.2 極性判定手法の研究動向

表 3.16は、使用されている情報や素性によって、3.1.1項で紹介した各論文を分類した一覧表である。なお、表の各列は、以下の事柄を示す。また、“-” は素性・手法の利用のないもの、“✓” は利用のあるものを示している。

表 3.15: 文献一覧表における記法の説明

判定対象	極性判定の対象を示す
形態素解析	形態素解析器あるいは品詞タガーによる品詞付け、語幹抽出などの結果を素性として利用していることを示す
N グラム	N グラムを素性として利用していることを示す
構文解析	構文解析器・係り受け解析の結果を利用していることを示す
単語頻度	単語頻度などの統計情報を利用していることを示す
外部言語資源	極性辞書、コーパスなど、外部で作成された言語資源を使用していることを示す
ルールベース	品詞、係り受け解析結果を基にしたルールによる極性判定であることを示す
教師あり機械学習	教師あり機械学習による極性判定であることを示す
教師なし機械学習	教師なし機械学習による極性判定であることを示す
トピックモデル	LDA などの潜在的トピックを推定する手法による極性判定であることを示す
深層学習	深層学習に基づく手法による極性判定であることを示す

表 3.16: 利用されている素性・情報による極性判定手法の分類

	文献	判定対象	形態素解析 (品詞タグ)	N グラム	構文解析	単語頻度	外部言語資源	ルールベース	教師あり機械学習	教師なし機械学習	トピックモデル	深層学習
1.	[2]	句	✓	-	-	-	-	✓	-	-	-	-
2.	[3]	文書	-	✓	-	✓	-	-	✓	-	-	-
3.	[4]	文	✓	✓	✓	✓	-	-	✓	-	-	-
4.	[5]	文	✓	-	-	✓	✓	✓	-	-	-	-
5.	[6]	句	✓	-	-	-	✓	-	✓	-	-	-
6.	[7]	文書	-	✓	✓	✓	-	-	✓	-	-	-
7.	[8]	文書	-	✓	-	-	-	-	✓	-	-	-
8.	[9]	属性と属性値	-	✓	-	-	-	-	-	✓	✓	-
9.	[10]	文書	-	-	-	-	-	-	-	✓	✓	-
10.	[11]	文書	✓	✓	-	-	-	-	-	✓	-	-
11.	[12]	文・単語	✓	-	-	-	✓	✓	-	-	-	-
12.	[13]	文	-	✓	-	-	✓	-	✓	-	-	✓
13.	[14]	文	-	-	-	-	-	-	-	✓	-	✓
14.	[15]	属性と属性値	-	-	-	-	-	-	✓	-	-	✓
15.	[16]	文書	✓	-	-	-	✓	-	✓	-	-	-
16.	[17]	属性と属性値	-	-	-	-	-	-	✓	-	-	✓

極性判定に関しては、1979 年の Carbonell による「ある物事に対する主観的な理解の仕方についての研究」[78] が始まりとも言われているが、極性判定の論文が増え始めたのは 2001 年以降であり、言語処理における機械学習の普及と、そのためのデータセットが充実し始めたことによる影響と考えられる。Pak と Paruoubek の研究 [11] のような Twitter などのマイクロブログと呼ばれる短文の場合を除けば、判定対象が文書、文、属性と属性値といったように、次第に小さい範囲のテキストに移っていく傾向が見られる。一時期はトピックモデルなどの教師なし機械学習に研究の中心が移ったが、その後深層学習が利用され始めてから、大規模な学習データを利用した深層学習による手法が多く提案されるようになっている。2007 年頃までの機械学習手法と違い、品詞、単語 N-gram、係り受け関係といった素性を明示的に与えずにモデルを学習するのが深層学習の特徴である。しかし、深層学習のモデルと、形態素解析 (品詞タグ付けを含む) 結果、構文解析結果、極性辞書などから得られる素性を組み合わせて用いる手法もある。今後、深層学習のように素性設計を必要としない手法が主流になるのか、それとも素性設計に頼った手法が主流になるのかは、はっきりしない。

3.2 属性抽出手法

本節では、製品の属性抽出に関する研究動向の調査結果を報告する。

3.2.1 属性抽出手法の精査

Jakob と Gurevych の研究

Jakob と Gurevych は、Conditional Random Fields(CRF)[79] による属性抽出手法を提案し、また学習データとテストデータとで異なる分野 (ドメイン) のコーパスを用いたときの属性抽出の正解率の変化を実験的に調査している [18]。表 3.17は CRF を学習する際に用いた素性を示している。「対象単語」とは、CRF においてラベル (B、I、O のいずれか) を決定する対象となる単語を指す。

表 3.17: Jakob と Gurevych の手法で用いられている学習素性 (括弧内は論文内での呼称)

単語 (tk)	対象単語そのもの
品詞 (pos)	対象単語の品詞タグ
依存構造上の短いパス (dLn)	対象単語と直接の係り受け関係がある単語
単語間距離 (wDs)	対象単語の一番近くにある名詞句
評価文 (sSn)	評価語のある文に出現する単語

実験では、映画、ウェブサービス、自動車、カメラという 4 分野に関するレビューデータを使用した。単語 (tk)、品詞 (pos) は常に使用し、それ以外の素性の使用の有無の組み合わせを変え、F 値が一番良くなる素性セットを調査した。実験結果を表 3.18に示す。この実験では、Zhuang らによる品詞の組み合わせ、係り受け関係のルールによる抽出手法 [80] と比較している。表 3.18の最下行がその結果である。

表 3.18: Jakob と Gurevych の手法の実験結果 [79]

	movies			**web-services**			**cars**			**cameras**		
Features	**Prec**	**Rec**	**F-Me**	**Prec**	**Rec**	**F-Me**	**Prec**	**Rec**	**F-Me**	**Prec**	**Rec**	**F-Me**
tk, pos	0.639	0.133	0.220	0.500	0.051	0.093	0.438	0.110	0.175	0.300	0.085	0.127
tk, pos, wDs	0.542	0.181	0.271	0.451	0.272	0.339	0.570	0.354	0.436	0.549	0.375	0.446
tk, pos, dLn	0.777	0.481	0.595	0.634	0.380	0.475	0.603	0.372	0.460	0.569	0.376	0.453
tk, pos, sSn	0.673	0.637	0.653	0.604	0.397	0.476	0.453	0.180	0.257	0.398	0.172	0.238
tk, pos, dLn, wDs	**0.792**	0.481	0.598	0.620	0.354	0.450	0.603	0.389	0.473	0.596	0.425	0.496
tk, pos, sSn, wDs	0.662	0.656	0.659	0.664	0.461	0.544	0.564	0.370	0.446	0.544	0.381	0.447
tk, pos, sSn, dLn	0.791	0.477	0.594	0.654	0.501	0.568	0.598	0.384	0.467	0.586	0.391	0.468
tk, pos, sSn, dLn, wDs	0.749	**0.661**	**0.702**	**0.722**	**0.526**	**0.609**	**0.622**	**0.414**	**0.497**	0.614	**0.423**	**0.500**
pos, sSn, dLn, wDs	0.672	0.441	0.532	0.612	0.322	0.422	0.612	0.369	0.460	**0.674**	0.398	0.500

評価実験で F 値（F-Me）が最も高くなったパターンは、全ての素性を利用したものである。このときの結果は、カメラ分野の精度を除いて、Zhuang らの手法による結果を上回った。ただ、全体的に再現率（Rec）が低いように思われる。

表 3.19: Jakob と Gurevych の手法の実験結果 – 学習とテストで分野が異なる場合 [79]

Training	Testing	web-services			movies		
		Pre	**Rec**	**F-Me**	**Pre**	**Rec**	**F-Me**
	web-services	-	-	-	0.560	0.339	0.422
	movies	**0.565**	0.219	0.316	-	-	-
	cars	0.538	0.248	0.340	0.642	0.382	0.479
	cameras	0.529	0.256	0.345	0.642	0.408	0.499
	movies + cars	0.554	0.249	0.344	-	-	-
	movies + cameras	0.530	**0.273**	**0.360**	-	-	-
	movies + cars + cameras	0.562	0.250	0.346	-	-	-
	cars + cameras	0.538	0.254	0.345	0.641	0.395	0.489
	web-services + cars	-	-	-	**0.651**	0.396	0.492
	web-services + cameras	-	-	-	0.642	**0.435**	**0.518**
	web-services + cars + cameras	-	-	-	0.639	0.405	0.496
		cars			**cameras**		
		Pre	**Rec**	**F-Me**	**Pre**	**Rec**	**F-Me**
	web-services	0.391	0.277	0.324	0.505	0.330	0.399
	movies	0.512	0.307	0.384	0.550	0.303	0.391
	cars	-	-	-	0.665	0.369	0.475
	cameras	**0.589**	0.384	**0.465**	-	-	-
	cameras + movies	0.567	**0.394**	**0.465**	-	-	-
	cameras + web-services	0.572	0.381	0.457	-	-	-
	movies + web-services	0.489	0.327	0.392	0.553	0.339	0.421
	movies + cars	-	-	-	0.634	0.376	0.472
	web-services + cars	-	-	-	**0.678**	0.376	**0.483**
	web-services + movies + cars	-	-	-	0.635	**0.378**	0.474
	movies + web-services + cameras	0.549	0.381	0.450	-	-	-

表 3.19は、訓練データの分野とテストデータの分野が異なるときの実験結果を示している。一般に、訓練データとテストデータで分野が異なると、機械学習の性能が落ちることが知られている。精度（Pre）は大きく悪化することはなかったが、再現率（Rec）、F 値（F-Me）は実用上問題があるように見える。分野に依存しない属性抽出については、まだ研究の余地があると言える。

Mukherjee と Liu の研究

Mukherjee と Liu は、は 2 つの半教師あり機械学習に基づく属性抽出手法を提案している [19]。2 つの手法は共に、シードと呼ぶ初期単語セットを基に、この単語と頻繁に共起する単語を属性あるいは極性語として抽出する。属性か極性語かの判定は階層型サンプリングにより行う。抽出された候補単語がシードリストの単語であった場合は属性とし、そうでなかった場合は崩壊ギブスサンプリングにより属性か極性語かを推測する。このモデルを “Seeded Aspect and Sentiment model (SAS)” と呼んでいる。もう 1 つのモデルは崩壊ギブスサンプリングの代わりに最大エントロピー法を属性と極性語の選別に利用する。最大エントロピーのパラメータ推論は、Hu と Liu が論文内で構築した極性辞書 [60] を用い、自動的に学習データを生成して行う。この手法を “Maximum Entropy (Max-Ent) SAS ”(ME-SAS) と呼んでいる。両手法とも、シードの単語をもとにその語と関係のある

語がクラスタリングされるため、ユーザー自身が知りたい属性を入力として与えることができる。

表 3.20: Mukherjee と Liu の手法の実験結果 [19]

Aspect (seeds)	ME-SAS		SAS		ME-LDA		DF-LDA
	Aspect	Sentiment	Aspect	Sentiment	Aspect	Sentiment	Topic
Staff (staff service waiter hospitality upkeep)	attendant	friendly	attendant	friendly	staff	friendly	staff
	manager	attentive	waiter	nice	maintenance	nice	***friendly***
	waitress	polite	waitress	dirty	**room**	courteous	***helpful***
	maintenance	nice	manager	**comfortable**	upkeep	**extremely**	**beds**
	bartender	**clean**	maintenance	nice	**linens**	nice	**front**
	waiters	pleasant	**helpful**	**clean**	room-service	**clean**	room
	housekeeping	**slow**	waiters	polite	receptionist	polite	**comfortable**
	receptionist	courteous	housekeeping	**extremely**	**wait**	**little**	**large**
	waitstaff	rude	receptionist	courteous	**pillow**	helpful	receptionist
	janitor	professional	**polite**	efficient	waiters	**better**	housekeeping
Cleanliness (curtains restroom floor beds cleanliness)	carpets	clean	hall	clean	cleanliness	clean	clean
	hall	dirty	carpets	dirty	floor	good	pool
	towels	comfortable	towels	fresh	carpets	dirty	**beach**
	bathtub	fresh	pillow	old	bed	**hot**	carpets
	couch	wet	stain	nice	**lobby**	**large**	**parking**
	mattress	filthy	mattress	good	bathroom	nice	bed
	linens	extra	**filthy**	**enough**	**staff**	**fresh**	bathroom
	wardrobe	stain	linens	new	closet	**thin**	**nice**
	spa	**front**	**interior**	**front**	spa	new	comfortable
	pillow	worn	bathtub	**friendly**	**décor**	**little**	suite
Comfort (comfort mattress furniture couch pillows)	bedding	comfortable	bed	nice	bed	great	bed
	bedcover	clean	linens	dirty	mattress	clean	mattress
	sofa	soft	sofa	comfortable	suites	awesome	**nice**
	linens	nice	bedcover	large	furniture	dirty	***stay***
	bedroom	uncomfortable	**hard**	clean	**lighting**	best	**lighting**
	suites	spacious	bedroom	**best**	**décor**	comfortable	lobby
	décor	hard	**privacy**	spacious	room	soft	comfort
	comforter	comfy	**double**	**only**	bedroom	nice	room
	blanket	dirty	**comfy**	big	**hallway**	**only**	**dirty**
	futon	quiet	futon	**extra**	**carpet**	**extra**	sofa

表 3.20 は、SAS と ME-SAS、ならびに他の手法との比較実験の結果を示している。1 列目は属性と初期単語セットであり、2 列目以降は各手法で自動獲得された上位 10 件の属性およびそれに関する極性語である。赤字の単語は不適切と判定された語である。ME-LDA は、Zhao らが提案した、最大エントロピー法による教師あり機械学習により属性と極性語を選別できるように LDA を改良した手法である [81]。DF-LDA は、Andrzejewski らが提案した、“must-link” と “cannot-link” という単語抽出における制約を与えられるように改良したモデルである [82]。must-link で与えられた 2 つの単語は必ず同じトピックに現れなければいけない、cannot-link で与えられた 2 つの単語は同じトピックになることはない、という制約である。DF-LDA は極性語を抽出できないため、Topic（本実験では属性として考えていい）のみの結果が示されている。この実験結果を見る限りでは、ME-SAS の結果が一番良いように思われる。さらに、定量的評価として、9 種類の属性について、各手法の P@10、P@20、P@30 を調べている。P@N(N=10,20,30) とは、上位 N 件の属性を出力したときの精度 (正解率) である。実験の結果、ME-SAS が他の手法を上回ることを確認している。例えば, ME-SAS, SAS, ME-LDA, DF-LDA の P@10 はそれぞれ 0.88, 0.72, 0.67, 0.52 である。

Chen らの研究

Chen らは、MC-LDA (LDA with m-set and c-set) という LDA を改良した手法を提案している [20]。MC-LDA の MC は、“m-set”(must set) と “c-set”(cannot set) を表す。それぞれ、必ず同じトピックに含まれなければいけない単語と、異なるトピックに含まれなければならない単語を定義する。これらの制約を与えることで、潜在的トピック学習による単語のクラスタリングの性能を向上させることを狙う。このアイデアは、“must-link” と “cannot-link” による制約を与えていた Andrzejewski らによる DF-LDA[82] と似ている。大きな違いは、DF-LDA では、“must-link” と “cannot-link” の間では矛盾せずに単語が定義されていることが前提となっていたことである。つまり、ある単語 w_1 が別の単語 w_2 の “cannot-link” であった時、w_1 の ‘must-link” と定義されている単語全てが w_2 の “cannot-link” となっている。一方、MC-LDA では、このような制約の伝播は考慮されていない。

Mukherjee と Liu の文献 [19] と同様の属性抽出実験を行っている。提案手法と比較しているのは、m-set のみを使った MC-LDA である M-LDA、および LDA[83] である。“Amazon”、“Price”、“Battery” という 3 つの属性カテゴリについて抽出を試みたところ、MC-LDA のみ全てのカテゴリについて属性を抽出できた。また、適切でない語の抽出も少なく、他の手法と比べて優れていると言える。

Chen らは、AKL(Automated Knowledge LDA) という手法も提案している [21]。MC-LDA が m-set、c-set という事前知識を人手で与えて LDA によるトピック推定を改善しているのに対し、AKL では事前知識をコーパスから自動的に獲得する。まず、分野別のレビューデータのコーパスを複数用意する。ステップ 1 では、個々の分野のデータから LDA を用いて潜在的トピックを推定し、属性の集合を抽出する。こうして得られた分野別のトピックをひとつにまとめる。ステップ 2 では、まとめられたトピック集合に対してクラスタリングを行う。クラスタリング手法として k-medoids 法 [84] が利用されている。ステップ 3 では、それぞれのクラスタに出現する単語群に対して、頻出パターンマイニング (FPM)[85] を適用し、属するトピックが似ている単語対を発見する。最終的に、似ている単語対の集合がクラスタ毎に事前知識として得られる。これは MC-LDA における m-set のように同じトピックに属するべき単語対であるとみなせる。さらに、こうして得られた事前知識を直接的に利用するために、潜在的トピックを推定する独自のアルゴリズム AKL を提案している。Chen らは、これら一連の手続きを複数回繰り返すことで潜在的トピックの品質を高める工夫もしている。上記の手続きのステップ 1 において、最初は LDA によって学習されたトピックを用いるが、2 回目以降は AKL によって学習されたトピックを用いる。

実験では、36 種類の異なる製品タイプのレビューを集めたデータを用いている。提案手法である AKL と比較しているのは、MC-LDA [20]、GK-LDA [86]、通常の LDA [83] である。MC-LDA と GK-LDA は同じ著者らによって過去に提案された手法である。ただし、事前知識は人手で与えるのではなく、この論文の方法で獲得された事前知識を与えている。MC-LDA では、must-link のみ与え、cannot-link は与えない。評価指標として Mimno ら

が提案した “Topic Coherence” [87] を用いている。図 3.9 は各手法の Topic Coherence を示している。横軸は潜在的トピックの学習回数である。この実験結果から、AKL は他の手法よりも良い結果が得られている。また、AKL は学習を繰り返すことで Topic Coherence が向上する傾向が見られる。しかし、属性抽出の精度、再現率、F 値といった直観的にわかりやすい指標では評価されていないため、この差がどれほど大きいのかは解釈が難しい。

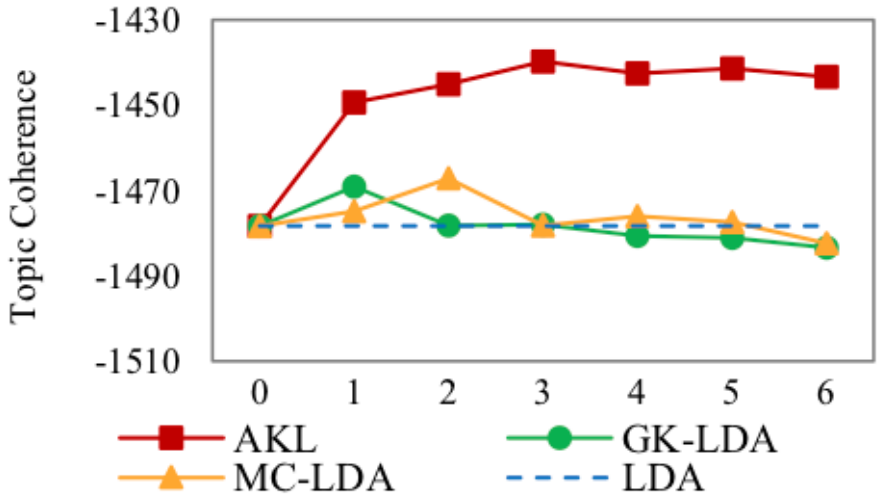


図 3.9: AKL の評価実験の結果 [28]

Poria らの研究

Poria らは明示的属性と暗黙的属性の両方を抽出するルールベースの手法を提案している [22]。ここで暗黙的属性 (論文では Implicit Aspect Clue (IAC) と呼ばれている) とは、“This camera is sleek and very affordable.” のような例文において、“sleek” は見た目を、“affordable” は値段を指しているというように、間接的に属性を表す表現である。IAC は単語だけでなく “easy to manipulate” や “user friendly” のような複数語からなる表現を含む。

属性抽出のためのルールは、極性辞書と、構文解析の結果得られる係り受け関係の結果を利用して定義されている。極性辞書として SenticNet 3[88]、構文解析ツールとして Stanford Dependency Parser[7]が用いられている。例えば、“ある単語が副詞的あるいは形容詞的修飾語を持ち、その語が SenticNet に存在する場合は、属性となる” といったルールを複数定義している。そのため、提案手法のルールによる属性抽出の正解率は、構文解析の正確さに強く依存する。

これとは別に、IAC の辞書の整備を試みている。Cruz らは、IAC およびその属性カテゴリをタグ付けしたコーパスを構築した [89]。属性カテゴリとは、機能、重量、値段、見た

[7] http://nlp.stanford.edu:8080/parser/

目など、属性の大まかな分類を表す。Cruz らのコーパスから、タグ付けされた IAC を抽出し、属性カテゴリ毎にまとめる。結果として、属性カテゴリ毎に IAC のリストが定義された辞書が構築される。さらに、既知の IAC に対し、その類義語と反意語を WordNet[76] から求め、それを IAC の辞書に追加している。

評価実験として、SemEval 2014 のデータセットを用いて、ラップトップとレストランのレビューから属性を抽出したときの精度と再現率を報告している[8]。ラップトップでは精度と再現率はそれぞれ 82.15%、84.32%、レストランでは 85.21%、88.15%であった。ただし、SemEval 2014 のデータセットでは暗黙的属性はタグ付けされていないため、ここでは明示的属性の抽出のみを評価している。実験では比較的高い精度と再現率が得られるものの、構文解析器の性能に強く依存するという問題点もある。SNS 上のテキストなどくだけた表現が多いテキストでは、既存の構文解析ツールの性能が落ちることが知られている。また、属性抽出の正解率を向上させるためには属性抽出のためのルールを追加することが考えられるが、ルールの数が増えたとき、それらに矛盾が生じないように管理するのが難しくなるという問題点もある。

中野らの研究

中野らは、文節が属性を含むかどうかを判定する分類器、意見（属性値）を含むかどうかを判定する分類器を作成し、それぞれの分類器で属性または意見を含むと判定された文節間に特定の係り受けがあるときに、属性意見ペアを抽出する手法を提案している [23]。属性を含む文節の分類器の学習には、1) 注目文節の末尾の助詞、2) 注目文節がストップワードを含むか、3) 係り先文節の主辞の品詞、4) 直近係り元文節の主辞の品詞、の 4 つを素性として用いる。これらは、存在する／しない、含む／含まないのバイナリ (0 または 1) で表現される。属性を含むと判定された文節から、その中に出現する名詞を属性として抽出する。意見を含む文節の分類器の学習では、1) 注目文節が文末か、2) 注目文節が評価極性辞書の語を含むか、3) 係り元文節の末尾の助詞、4) 直近係り元文節の主辞の品詞、5) 係り元に属性を含むか、という 5 つの素性が 2 値で表現され利用される。5 番目の素性は属性抽出の結果を利用している。つまり、最初に属性を抽出し、その結果を利用して意見抽出の分類器を学習している。属性と同様に、意見を含むと判定された文節内の名詞が意見として抽出される。属性と意見の対は、「属性を含む文節の係り先文節または係り元文節が意見を含む場合」、「属性を含む文節が並列助詞を含み、その文節の係り先の係り先文節が意見を含む場合」に抽出される。

実験では、Amazon.co.jp から取得した 338 件の「掃除機」のレビューと 70 件の「掛け時計」のレビューを用いている。これらのレビューに人手で正解の属性と意見をタグ付けした。属性抽出の精度、再現率、F 値はそれぞれ 0.569、0.519、0.534 であった。意見抽出の実験結果を表 3.21に示す。この表では提案手法とベースラインを比較している。ここでのベースラインは、「意見の候補を含む文節のうち、属性を含む文節と係り受け関係に

[8] http://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools

あり、主辞の形態素が評価極性辞書に含まれる文節を意見を含む文節として抽出」する手法である。また、機械学習の際に属性の素性を用いるとき、提案手法で抽出した属性を使用する場合と人手で抽出した属性を使用する場合も比較している。提案手法はベースラインと比べて適合率 (精度) は低いが、再現率と F 値は高い。また、人手で抽出した属性を用いると各指標が大幅に改善することがわかる。前述のように提案手法による属性抽出の F 値は 0.534 と低いため、これを改善する必要があることがわかる。表 3.22 は属性意見ペアの抽出結果を示している。ここでは掃除機のデータを分類器の学習に用いている。このため表 3.22における「掃除機」の結果は訓練データとテストデータが同じという設定である。両ドメインとも適合率 (精度) は比較的良好だが、再現率が低いと考察している。

表 3.21: 中野らの手法による意見抽出の実験結果 [23]

属性の抽出	意見の抽出	適合率	再現率	F 値
提案手法	ベースライン	**0.679**	0.350	0.462
	提案手法	0.575	**0.502**	**0.535**
手作業	ベースライン	**1.000**	0.459	0.629
	提案手法	0.922	**0.703**	**0.795**
使用せず	提案手法	0.554	0.525	0.538

表 3.22: 中野らの手法による属性意見ペア抽出の実験結果 [23]

	適合率	再現率	F 値
掃除機	0.663	0.392	0.493
掛け時計	0.789	0.170	0.280

Liu らの研究

Liu らは、統語論的アプローチによる属性抽出手法を提案している [24]。品詞・単語の接続ルール、係り受け関係の制約ルールなどを利用するという意味では Poria らによる手法 [22] と似ているが、人手で (比較的少量の) ルールを用意するのではなく、最初にルールの候補を作成し、その中から有効なルールを自動的に選択している点が異なる。提案手法におけるルールは以下の 3 つに大別される。

1. ルール 1: 既知の極性語を手がかりに新しい属性を抽出する
   初期極性語リストをあらかじめ用意する。

2. ルール 2: 既知の属性を手がかりに新しい属性を抽出する
   例えば、以下のようなルールが定義できる。

$IF\ depends\,(conj, A_i, A_j) \wedge\ pos\,(A_i, nn) \wedge\ aspect\,(A_j)\ \ THEN\ aspect\,(A_i)$

このルールは、A_i と A_j は "A_i and A_j" や "A_i, A_j" のような接続関係 (conj) にあり、A_i は名詞 (nn) で、かつ A_j が既知の属性であれば、A_i を属性として抽出する。

3. ルール 3: 既知の属性と極性語、またはそのどちらか一方を手がかりに新しい極性語を抽出する
以下のようなルールが定義できる。
$IF\ depends\,(amod, A, O) \wedge\ pos\,(O, jj) \wedge\ aspect\,(A)\ \ THEN\ opinionword\,(O)$

このルールは、O が A を修飾 (amod) していて、O が形容詞 (jj) で、A が既知の属性であれば、O は極性語として抽出する。

3 つのグループのルールを用意した後、それぞれのルールの性能を評価する。具体的には、属性にラベル付けされた学習コーパスに対してルールを適用し、属性抽出の精度と再現率を測る。次に、ルールのランク付けを行う。ルールを精度が高い順に、精度が同じ場合は再現率が高い順にランク付けする。最後に、最終的に採用するルール集合を決める。最初のルール集合を空集合とし、ランクが高いルールをひとつ加える。ルールを追加する前と後で、ルール集合を学習コーパスに適用したときの属性抽出の F 値を測る。ルールを追加することで F 値が改善すれば、次のルールを加え、改善しなければルールの獲得を止める。

評価実験では、2 つのルールセット、2 つの属性抽出手法の組み合わせを比較している。ルールセットとしては、Qiu らよって提案された 8 つのルール [32, 90] を用いた場合 (ここでは「基本ルールセット」と呼ぶ) と、これらルールの中で使われている統語的関係を他の統語関係に置き換えて作成した新しいルールも用いた場合 (以下「拡張ルールセット」と呼ぶ) を比較している[9]。属性抽出手法としては、提案手法によって選別したルールを用いる手法と、Qiu らが提案した Double Propagation による手法を比較している。参考のため、ルールベースではない教師あり機械学習に基づく手法として、Jakob と Gurevych による CRF を用いた手法 [18] とも比較する。分野が異なる 8 つのコーパスから属性を抽出し、精度、再現率、F 値の平均を評価指標としている。実験の結果、拡張ルールセットかつ提案手法を用いた場合の F 値が最も高かった。ただし、再現率については、拡張ルールセットかつ Qiu らの属性抽出手法を用いた方が良かった。

Poria ら (2016) の研究

Poria らは、畳み込みニューラルネットワーク (Convlutional Neural Network; CNN) と、著者らが言語パターンと読んでいるルールベースの手法を組み合わせた属性抽出手法を提案している [25]。CNN のネットワークは、1 つの入力層、2 つの畳み込み層、2 つの最大

[9]Qiu らの手法の詳細については 3.3.1節で延べる

プーリング層、そして softmax 関数によって出力を決める全結合層という構成になっている。CNN の入力として与えるのは、属性であるかをどうかを決める単語 (対象単語) の文脈を表す行列である。この行列は、対象単語とその前後 2 単語、合計 5 単語の単語ベクトルを連結したものである。単語ベクトルとして、CBOW による単語の分散表現 [91] を用いる。単語の分散表現は、Mikolov らによってあらかじめ学習されウェブ上で公開されているもの [91] か、McAuley と Leskovec が構築した Amazon.com の製品レビューコーパス [92] から著者らが学習したものを用いる。いずれも単語ベクトルの次元数は 300 である。さらに、6 種類の品詞を単語ベクトルに追加する[10]。与えられた文に対し、上記の CNN に基づく手法とルールベースの手法を適用し、どちらか一方によって属性であると判定された単語を抽出する。

SemEval 2014 データセットを使って提案手法を評価する実験を行っている。対象ドメインはラップトップとレストランである。実験結果を表 3.23 に示す。“ZW” は Toh と Wang による CRF に基づく属性抽出手法 [93] であり、“CNN+LP(our)” がこの論文の提案手法である。どちらのドメインでも、再現率、精度、F 値いずれも提案手法は Toh と Wang の手法を上回る。特に再現率が大きく改善していることがわかる。また、これとは別に、CNN のみ、LP のみ、CNN と LP の組み合わせを比較した実験も行っている。CNN のみでもラップドップドメインの再現率は 76.32%と高いことが報告されている。

表 3.23: Poria らの手法の実験結果 [25]

Domain	Framework	Recall	Precision	F-Score
Laptop	ZW	66.51%	84.80%	74.55%
Laptop	CNN+LP (our)	**78.35%**	**86.72%**	**82.32%**
Restaurant	ZW	82.72%	85.35%	84.01%
Restaurant	CNN+LP (our)	**86.10%**	**88.27%**	**87.17%**

He らの研究

He らは、アテンション機構を用いた属性抽出手法 (Attention-based Aspect Extraction; ABAE) を提案している [26]。図 3.10 に ABAE の概要を示す。

[10] したがって、最終的には単語ベクトルの次元数は 306 となる。

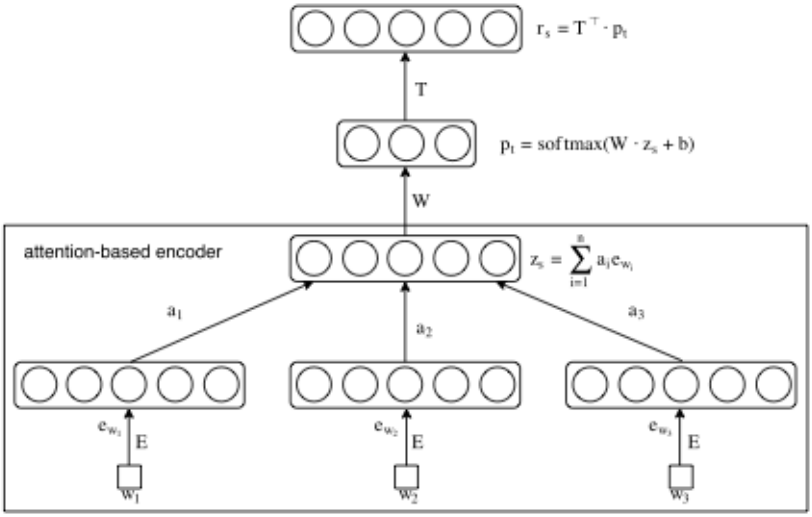


図 3.10: ABAE の概要

ABAE はエンコーダー・デコーダーモデルである。エンコーダでは、解析対象とする文の分散表現 (ベクトル)Z_s を生成する。Z_s は、文中に出現する単語 w_i の分散表現 e_{w_i} の重み付き和である。重みは a_i で表され、アテンションと呼ばれる。デコーダーでは、まず文の分散表現 Z_s を、ベクトルの次元を縮退したベクトル p_t に変換する。p_t は文の抽象的な表現とみなせる。さらに、p_t から元の文の分散表現 r_s を復元する。教師信号としての r_s は Z_t と同じベクトルを与えればよいので、学習にはラベル付きデータを必要としない。図 3.10 のモデル全体を学習すると、アテンション a_i が自動的に推定される。直観的には、属性に相当する単語 w_i に対しては、a_i も高くなるように推定される。

実験では、レストランドメインを対象とし、“Food”、“Staff”、“Ambience” という 3 つの属性カテゴリについて属性を抽出している。提案手法と以下の手法を比較している。

LocLDA Brody と Samuel の手法 [94]。文を文書として扱う Local LDA と呼ぶ手法を属性抽出に適用している。

ME-LDA Zhao らの手法 [81]。最大エントロピー法による教師あり機械学習により属性と極性語を選別できるように LDA を改良している。

SAS 32ページで述べた Mukherjee らの手法 [19]。

BTM Yan らの手法 [95]。短い文書向けに提案されているトピックモデルの BTM(Biterm Topic Model) を属性抽出に適用している。

SERBM Wang らの手法 [96]。制限付きボルツマンマシンによる属性抽出手法である。

k-means 文のベクトル (文中に含まれる単語の単語ベクトルの平均) と最も近いベクトルを持つ単語を属性として抽出する手法。

実験結果を表3.24に示す。著者らが主張する通り、ABAEは他の手法と比べてもトップレベルの性能を示している。“Food” カテゴリではABAEよりもSERBMの方が再現率とF値が高いが、SERBMは “Staff” や “Ambience” ではそれほどF値が高くないことを考えると、ABAEはどのような属性カテゴリに対しても安定して属性を抽出することができると言える。

表 3.24: Heらの手法の実験結果 [26]

Aspect	Method	Precision	Recall	F_1
Food	LocLDA	0.898	0.648	0.753
	ME-LDA	0.874	0.787	0.828
	SAS	0.867	0.772	0.817
	BTM	0.933	0.745	0.816
	SERBM	0.891	**0.854**	**0.872**
	k-means[3]	0.931	0.647	0.755
	ABAE	**0.953**	0.741	0.828
Staff	LocLDA	0.804	0.585	0.677
	ME-LDA	0.779	0.540	0.638
	SAS	0.774	0.556	0.647
	BTM	**0.828**	0.579	0.677
	SERBM	0.819	0.582	0.680
	k-means	0.789	0.685	0.659
	ABAE	0.802	**0.728**	**0.757**
Ambience	LocLDA	0.603	0.677	0.638
	ME-LDA	0.773	0.558	0.648
	SAS	0.780	0.542	0.640
	BTM	0.813	0.599	0.685
	SERBM	0.805	0.592	0.682
	k-means	0.730	0.637	0.677
	ABAE	**0.815**	**0.698**	**0.740**

Wangらの研究

Wangらは、多層結合アテンション機構を利用して属性と極性語を同時に抽出する手法を提案している [27]。図3.11は提案手法の概要を図示している。このモデルは2つのアテンション機構を結合している。それぞれ属性の抽出、極性語の抽出に用いる。図3.11中の実線は属性、破線は極性語のアテンションを表す。2つのアテンションは同じ隠れ層のベクトル h_i を共有している。これにより、属性抽出に極性語の情報を、極性語抽出に属性の情報を利用できる。e_t^a と e_t^p はアテンション・スコアのベクトルであり、このスコアの高い単語が属性もしくは極性語として抽出される。一方、h_i から e_t^a,e_t^p を得るネットワークを一つの層とし、これを重ねて多層構造を形成している。多層ネットワークを導入しているのは、単語間の直接の修飾関係だけでなく、間接的な係り受け関係も考慮したいからである。このモデルの入力として与えられるのは、解析したい文に対応する単語の分散表現の列である。このため、前処理として形態素解析や構文解析を必要とせず、また極性辞書などの外部言語資源も必要としない。

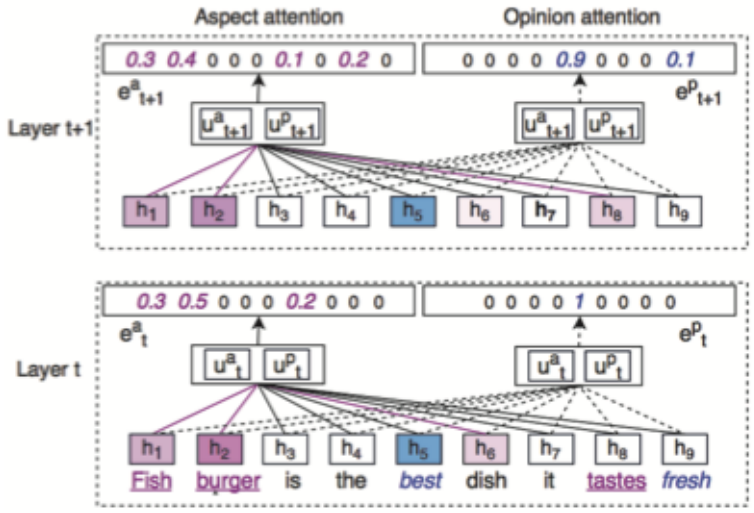


図 3.11: 多層結合アテンション機構による属性と極性語の抽出手法 [27]

表 3.25は提案手法の評価実験の結果である。S1、S2 は、それぞれ SemEval Challenge 2014 [1] のレストランとラップトップのデータ、S3 は SemEval Challenge 2015 [97] のレストランデータを表す。また、AS は属性抽出、OP は極性語抽出タスクの F 値を表す。1 列目がモデル名で、最下行の CMLA が提案手法である。DLIREC、IHS_RD は、SemEval Challenge 2014 において、それぞれレストラン、ラップトップドメインで最上位の成績をあげた手法である。EiXa は、SemEval Challenge 2015 のレストランドメインにおけるベストのシステムである。LSTM は単語分散表現と LSTM を利用した手法 [98]、WDEmb は単語と係り受け解析結果を CRF の入力として利用する手法 [99]、RNCRF は CRF と再帰的ニューラルネットによる手法 [100] を表している。また、WDEmb*と RNCRF*は、それぞれ WDEmb と RNCRF に言語的な素性を追加したモデルである。

表 3.25: Wang らの手法の実験結果 [27]

	S1		S2		S3	
Model	AS	OP	AS	OP	AS	OP
DLIREC	84.01	-	73.78	-	-	-
IHS_RD	79.62	-	74.55	-	-	-
EliXa	-	-	-	-	70.04	-
LSTM	81.15	80.22	72.73	74.98	64.30	66.43
WDEmb	84.31	-	74.68	-	69.12	-
WDEmb*	84.97	-	75.16	-	69.73	-
RNCRF	84.05	80.93	76.83	76.76	67.06	66.90
RNCRF*	84.93	**84.11**	**78.42**	79.44	67.74	67.62
CMLA	**85.29**	83.18	77.80	**80.17**	**70.73**	**73.68**

CMLA は他の手法と比べて全般的に F 値が高い。特に S3 のデータセットについては他の手法を大きく上回る。論文では S3 だけ特に F 値が高い理由は考察されていなかった

が、この理由が解明されれば、今後の属性抽出、極性語抽出の研究の指針となるかも知れない。

3.2.2 属性抽出手法の研究動向

表 3.26は、使用されている情報や素性によって、3.2.1項で紹介した各論文を分類した一覧表である。各列は分類の観点を表し、これらは表 3.15と同じである。“-” は利用のないもの、“✓” は利用のあるものを示している。

表 3.26: 利用されている素性・情報による属性抽出手法の分類

文献	形態素解析 (品詞タグ)	N グラム	構文解析	外部言語資源	ルールベース	教師あり機械学習	教師なし機械学習	トピックモデル	深層学習
1. [18]	✓	-	-	✓	-	✓	-	-	-
2. [19]	✓	-	-	-	-	(✓)	-	✓	-
3. [20]	✓	-	-	-	-	(✓)	-	✓	-
4. [21]	✓	-	-	-	-	-	✓	✓	-
5. [22]	✓	-	✓	✓	✓	-	-	-	-
6. [23]	✓	-	✓	-	-	✓	-	-	-
7. [24]	✓	-	✓	-	-	(✓)	-	-	-
8. [25]	✓	-	-	-	✓	✓	-	-	✓
9. [26]	-	-	-	-	-	-	✓	-	✓
10. [27]	-	-	-	-	-	-	✓	-	✓

* (✓) は半教師あり学習を表す。

属性抽出については、極性判定とは異なり、トピックモデルを用いた教師なし、半教師あり機械学習手法が比較的多いことがわかった。とはいえ、これらの一連の研究は同じ研究グループによるものである。前処理として形態素解析がよく使われるのは当然だが、それ以外の要素技術については、特に頻繁に用いられる情報や素性は見つからなかった。また、2015 年頃から深層学習を用いた研究が増えていることがわかった。これらには、品詞、ルールと深層学習を組み合わせたもの、単語の分散表現のみを利用した手法が含まれる。深層学習による手法以外は、品詞や構文解析を前処理とし、これから得られる情報を素性として利用することが多いが、深層学習では素性を明示的に抽出することを必要としない、あるいはそれを目指しているという点に特徴があると言える。

3.3 評価語辞書作成

本節では、評価語辞書を自動的に作成する手法の調査結果を報告する。評価語辞書だけでなく、属性表現をコーパスから抽出する手法も、極性判定や属性抽出に必要な知識を自動構築するという点は評価語辞書作成と共通しているため、この節で取り上げる。

3.3.1 評価語辞書作成手法の精査

小林らの研究

小林らは、“評価対象”、“評価属性”、“評価値” の 3 つの組からなる主観的な記述を極性表現であると定義し、これをコーパスから抽出することで、最終的に属性表現と評価値表現を網羅的に収集する手法を提案している [28]。ここでは、事前に定義した “< 対象 > の < 属性 > は < 評価値 >” のような文型を共起パターンとし、パターンにマッチした単語の組を極性表現として抽出する。抽出パターンはあらかじめ人手で用意する。また、誤抽出を防ぐための制約として、既にわかっている評価対象、属性、評価値表現が共起パターンのスロットのいずれかを満たした場合のみ、極性表現を抽出する。このとき、既にわかっているもの以外のスロットに当てはまるものが新たな表現として獲得される。さらに、共起パターンに合致しても属性や評価値表現とはならない表現を排除するために、1) 品詞、2) スコア、3) 既知の表現を条件としたルールによるフィルタリングを行なっている。提案手法を評価するために、提案手法によって自動的に属性表現と評価値表現を収集した場合と、人手で収集した場合とで、収集できた表現数と収集に要する時間を比較した。実験では、車とゲーム関連のレビューサイトから収集した記事を使用した。記事数は車 15,000 記事 (230,000 文)、ゲームでは 9,700 記事 (90,000 文) である。図 3.12に車分野の属性表現・評価値表現の収集結果を示す。提案手法では少量の評価対象、属性表現、評価値表現をシードとして与えるため、図 3.12では半自動と記している。人手に比べて圧倒的に早く属性表現・評価値表現が収集できていることがわかる。

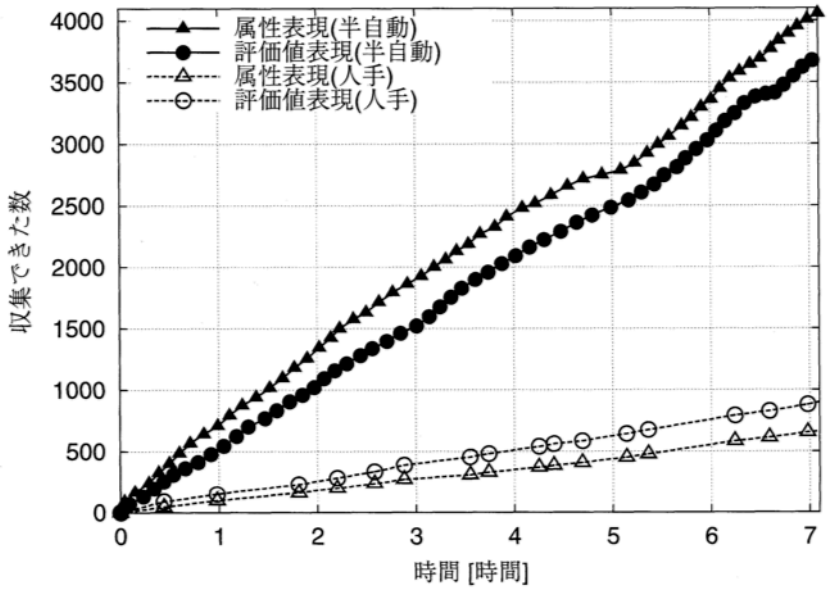


図 3.12: 小林らによる属性表現・評価値表現の収集結果 [28]

高村らの研究

高村らはスピン系のモデルを応用することで、単語の極性を自動的に判定する手法を提案している [29]。各電子がスピンと呼ばれる方向（上向きあるいは下向きをとる）を持つように、各単語は感情極性と呼ばれる方向（ポジティブあるいはネガティブ）を持つと想定する。まず、ある単語とその単語の語釈文内の各単語を連結した語彙ネットワークを構築する。単語間を結ぶリンクの集合を、SL（same-orientationlinks，同極性リンク集合)、DL（diffeerent-orientationlinks，逆極性リンク集合）という 2 つのグループに分ける。これは、同極性あるいは逆極性を持ちやすいと予測される単語対の集合とみなせる。シソーラスで反義語として登録されている単語対を結ぶリンクは DL へ、それ以外は SL へ属するとする。さらに、コーパスに出現する文において、形容詞が “and” で結ばれているときは SL へ、“but” で結ばれているときは DL へ属するとする。こうして、同じ極性を持つと予測される語彙ネットワークが作成される。スピン系モデルでは、各ノード (単語) はエネルギー関数を持ち、その正負によってスピンの方向 (極性) が決まる。ラベル付き初期極性単語集合を与え、隣接する単語のエネルギー関数を書き換える操作を繰り返すことで、初期単語の極性を他の単語に伝播させる。この計算により、最終的にエネルギー関数の平均値が正だった単語は感情極性がポジティブと判定し、負だった単語はネガティブであると判定する。

実験では、語釈文データとして WordNet [76] を用い、シソーラスとして WordNet の持つ反義語を利用している。また、PennTreeBank[101] の Wall Street Journal Corpus と Brown Corpus から、“and” または “but” で結ばれた形容詞の組を 804 個抽出し、利用している。表 3.27は、提案手法による極性判定の正解率を示している。比較のため、Hu と Liu によるブートストラップ法に基づく手法 [60] の結果も載せている。ただし、高村らのモデルはどのような単語でも極性を推定できるが、Hu と Liu の手法は初期単語と同じ品詞の語しか極性を予測できないという制約がある。そのため、表 3.27 は形容詞のみを対象とした結果である。また、表 3.27 の 1 列目の “seeds” は初期単語の数を表す。例えば、2 の場合は {good, bad} が初期単語として与えられている。評価実験の結果、提案手法が Hui らの手法と比べて正解率が大幅に上回ることが確認されている。

表 3.27: 高村らの提案手法による単語の極性判定の正解率 [29]

seeds	提案手法	Hu らの手法
14	83.6 (0.8)	72.8
4	82.3 (0.9)	73.2
2	83.5 (0.7)	71.1

Kanayama と Nasukawa の研究

Kanayama と Nasukawa は、分野毎に評価語辞書を人手で作成することには限界があるとし、分野依存の評価語をアノテーションなしの文書から収集する手法を提案してい

る [30]。図 3.13 は彼らの手法の処理の流れを示している。

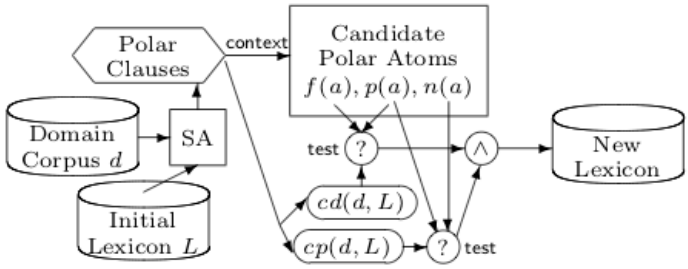


図 3.13: Kanayama と Nasukawa による評価語辞書の自動拡張手法 [30]

まず、ドメインコーパス d と初期の評価語辞書 L から、SA(Sentiment Analysis) モジュールで極性表現を含む文節を抽出する。SA モジュールでは、1) 文書の文への分割、2) 極性を表す表現を含む文節の抽出、3) その文節に対する極性の推定、の 3 つのステップから構成される。この処理により、極性を持つ文節 (図 3.13の Polar Clauses) をドメインコーパス d から抽出する。さらに、極性を持つ文節の集合から、「動詞」「形容詞」「動詞←名詞-助詞」「形容詞←名詞-助詞」を評価語の候補 (図 3.13の Candidate Polar Atoms) として抽出する。ここで、「動詞←名詞-助詞」とは、「ボディが小さい」のように名詞+助詞が動詞にかかる句を表す。評価語の候補を a とし、それが出現する文節の数 $f(a)$、それが出現するポジティブな文節の数 $p(a)$、それが出現するネガティブな文節の数 $n(a)$ を得る。

提案手法では、極性を持つ 2 つの文節が近くに出現したとき、原則として、逆接表現が存在する場合には 2 つの文節の極性は反対となり、それ以外の場合には一致すると仮定する。極性を持つ 2 つの文節が上記の条件を満たすとき、それらを Coherent と呼び、満たさないときは Conflict と呼ぶ。さらに、ドメインコーパスが上記の基準をどの程度満たすかを測る指標として、Coherent Precision $cp(d, L)$ と Coherent Density $cd(d, L)$ を式 (3.19) と式 (3.20) のように定義する。

$$cp(d, L) = \frac{\#(\text{ Coherent })}{\#(\text{ Coherent }) + \#(\text{ Conflict })} \tag{3.19}$$

$$cd(d, L) = \frac{\#(\text{ Coherent })}{\#(\text{ Polar })} \tag{3.20}$$

式 (3.19) において、#(Coherent) と #(Conflict) は、それぞれ極性に矛盾がない、矛盾がある文節の組の数であり、$cp(d, L)$ はコーパスの中で 2 つの極性を持つ文節がどの程度の割合で Coherent の条件を満たすかを示している。一方、式 (3.20) において、#(Polar) は極性を持つ文節の数であり、$cd(d, L)$ は Coherent の条件を満たす極性を持つ文節が全体の中でどれだけ出現するかを表している。$cp(d, L)$ も $cd(d, L)$ も高い値を取ることが仮定

されていること、また両方ともドメインコーパス d と初期の極性辞書 L に対して計算されることに注意していただきたい。

個々の評価表現の候補 a に着目すると、$\frac{p(a)}{p(a)+n(a)}$ は a についての Coherent Precision に相当し、それはコーパス全体の $cp(d,L)$ よりも大きくなることが期待される。つまり、$\frac{p(a)}{p(a)+n(a)}$ が $cp(d,L)$ よりも大きいなら、a は評価語として正しい可能性が高い。この手法では、$\frac{p(a)}{p(a)+n(a)}$ が $cp(d,L)$ よりも大きいかを統計的に検定し、信頼度 90% で有意差があるかをチェックする。一方、$\frac{p(a)}{f(a)}$ は a についての Coherent Density に相当し、それはコーパズ全体の $cd(d,L)$ よりも大きくなることが期待される。もし、$\frac{p(a)}{f(a)}$ が $cd(d,L)$ よりも大きいなら、やはり a は評価語として正しい可能性が高く、前者が後者よりも信頼度 90% で有意に大きいことを統計的検定で確認する。この 2 つの統計的検定をパスした a が新しい評価語として辞書に追加される。上記の説明はポジティブな評価語を獲得する手続きだが、ネガティブな評価語も $p(a)$ と $n(a)$ を入れ換えることで同様に獲得できる。

表 3.28: Kanayama と Nasukawa による評価語辞書の自動拡張の実験結果 [30]

Domain	#	Type Prec.	Token Prec.	Relative Recall
digital cameras	708	65%	96.5%	1.28
movies	462	75%	94.4%	1.19
mobile phones	228	54%	92.1%	1.13
cars	487	68%	91.5%	1.18

表 3.28は提案手法による実験結果を示している。“Domain” は実験に用いたコーパスの分野を、“#” は獲得できた評価語の数を、“Type Prec.” と “Token Prec.” は自動獲得された評価語の属性が人手による極性分類と一致した割合である。“Type Prec.” は極性判定の正解率であるのに対し、“Token Prec.” は新しく獲得した評価語の出現頻度に応じて重みをかけた正解率である。“Relative Recall” は、自動獲得された評価語のうち、新たに獲得できたものの数と初期の極性辞書にもともと含まれているものの割合である。“Token Prec.” は 90%以上と十分に高い。それに比べると “Type Prec.” は全般的に低く、またドメインによって 54%から 75%とばらつきが見られる。

Kaji と Kitsuregawa の研究

Kaji と Kitsuregawa は、大量の HTML 文書から評価語辞書を作成する手法を提案している [31]。まず最初に、HTML 文書から肯定的または否定的な意見を表す文 (極性付き文) を収集する。極性付き文は、(1) レイアウト構造と (2) 言語構造を参照したパターンマッチによって取得する。いずれのパターンも、基本的には、再現率は低いが精度が十分に高い、つまりパターンにマッチしたときにはほぼ確実の極性付き文が取得できるように設計する。

レイアウト構造によるパターンマッチでは、HTML 文書における箇条書きとテーブルの 2 つのレイアウト情報を利用する。箇条書きの場合、“Cue Words” が出現したとき、その下位の箇条書きに出現する文を極性付き文として抽出する。テーブルの場合、図 3.14 にしたがい、C_+(ポジティブな Cue Word) と C_-(ネガティブな Cue Word) に隣接するセル C に出現する文を極性付き文として抽出する。ここで “Cue Word” とは、極性付き文が近くにあることを示唆するキーワードで、“利点”、“欠点”、“プラス”、“マイナス” などがある。

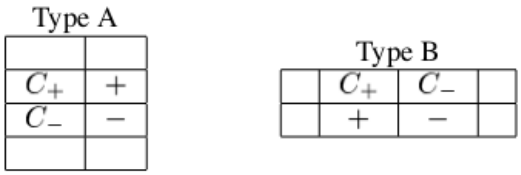


図 3.14: レイアウト構造のパターンマッチによる極性付き文の抽出 [31]

一方、言語構造によるパターンマッチでは、極性付き文でよく出現する係り受け構造を語彙構造パターンとして用意する。図 3.15 は語彙構造パターンを図示したものである。HTML 文書に出現する文を構文解析し、このパターンにマッチしたとき、(POLAR) の部分木に対応する文が極性付き文として抽出される。例えば、「このソフトウェアの利点は早く動くことです」という文は図 3.15のパターンにマッチし、「早く動く」が極性付き文として抽出される。「利点」に相当する単語としては、レイアウト構造によるパターンマッチのときに用いた Cue Word と同じキーワードを用いる。

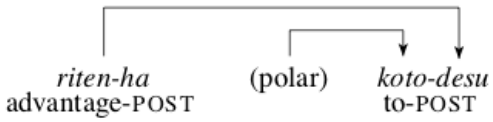


図 3.15: 言語構造のパターンマッチによる極性付き文の抽出 [31]

上記の手法を 10 億件の HTML 文書に対して適用し、509,471 の極性付き文を取得した。そのうちの 220,716 がポジティブ、それ以外がネガティブなものであった。これを極性付き文コーパスと呼ぶ。

次に、極性付き文コーパスから、評価表現の候補を抽出する。ここでの評価表現の候補とは、“名詞 + 助詞 + 形容詞” からなる形容詞句と定義する。抽出した評価表現の候補に対して、その出現頻度、ポジティブな文での出現頻度、ネガティブな文での出現頻度をカウントしておく。

最後に、評価表現の候補の中から辞書に追加するべき評価表現を選別する。評価表現と、極性付き文コーパスにおける極性クラスとの相関関係を統計的に測る。統計的指標として、χ^2 値と PMI(自己相互情報量)[45] のいずれかを用いる。これらの統計的指標から、

評価表現の候補 c の極性の強さを表すスコア $PV(c)$ を計算する。$PV(c)$ が閾値 θ よりも大きいときには c をポジティブな評価表現として獲得し、$-\theta$ よりも小さいときにはネガティブな評価表現として獲得する。それ以外は c は中立であるとみなし、評価語辞書に追加しない。

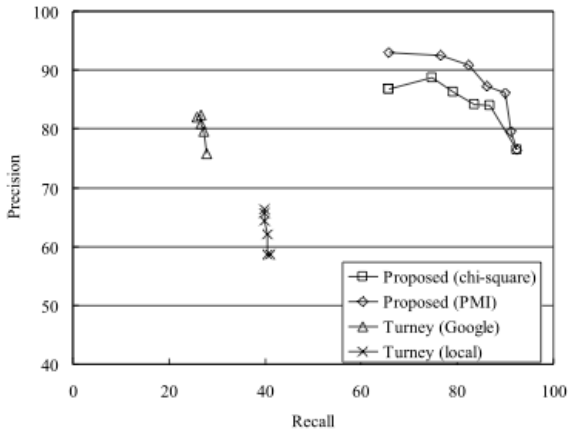


図 3.16: Kaji と Kitsuregawa による手法の実験結果 [31]

実験では、無作為に抽出した 405 の評価表現 (形容詞句) に対して人手で極性タグを付与したデータを用いる。極性タグの内訳は、ポジティブな句が 158、ネガティブな句が 150、中立な句が 97 となる。これらの評価表現をコーパスから抽出できるかというタスクを設定し、その精度と再現率で提案手法を評価する。図 3.16 は、ポジティブな評価表現を抽出するタスクにおける Kaji と Kitsuregawa の手法の精度-再現率曲線を示している。精度、再現率ともに高いことがわかる。また、χ^2 値よりも PMI を用いた方がよい結果が得られている。比較として、Turney による PMI-IR 手法 [2] の結果も載せている。Turney の手法では、ウェブ検索エンジンで “A NEAR B” というクエリで検索し、そのヒット件数で A と B の相関を測っている。実験では、Google 検索エンジンを用いた手法 (Turny(Google))[11] と著者らが開発した検索エンジンを使う手法 (Turny(local)) を評価している。Kaji と Kitsuregawa の手法はこれらの手法を大きく上回ることが確認されている。

Qiu らの研究

Qiu らは、ドメインに固有の評価語ならびに製品の属性を表す語 (以下、単に「属性」と呼ぶ) を自動的に獲得する手法を提案している [32]。この手法では、評価語と属性の間には何らかの統語的関係が成立する可能性が高いことを仮定している。例を以下に挙げる。

(1) The Phone has a good screen.

[11] Google 検索エンジンでは NEAR (2 つのクエリが近傍に出現するときにヒットする) をサポートしていないため、AND を用いている。

(2) The camera is amazing and easy to use.

例文 (1) では、評価語 “good” と属性 “screen” の間には修飾関係が成立する。例文 (2) では、2 つの評価語 “amazing” と “easy” の間には等位接続関係が成立する。この仮定を踏まえ、まず少量の極性単語 (極性が付与された評価語) のリストをシードとして用意する。既知の評価語または属性と特定の統語関係にある語を新しい評価語または属性として抽出する。この操作は「伝播」と呼ばれ、以下の 4 つのケースがある。

1. 既知の評価語と特定の統語関係にある語を新しい評価語として抽出する。
2. 既知の評価語と特定の統語関係にある語を新しい属性として抽出する。
3. 既知の属性と特定の統語関係にある語を新しい評価語として抽出する。
4. 既知の属性と特定の統語関係にある語を新しい属性として抽出する。

「伝播」を新しい評価語もしくは属性が獲得されなくなるまで繰り返す。

評価語と属性の間に成立する統語関係は表 3.29のように定義されている。これは新しい評価語または属性を抽出するためのルールともみなせる。この表で使われている記号の意味は表 3.30の通りである。例えば、ルール $R1_1$ は、$S_{j(i)}$ が既知の評価語であり、それと $S_{i(j)}$ の間に接続関係 (CONJ) があり、かつ $S_{i(j)}$ の品詞が形容詞 (JJ) のとき、$S_{i(j)}$ を新しい評価語として抽出する。このルールにより、“amazing” が既知の評価語のとき、上記の例文 (2) から “easy” が新しい評価語として獲得される。

次に、抽出した評価語に対し、その極性を決める。この際、以下の仮定をおく。

1. ユーザーは、1 つの属性に対して違う極性を持たない
2. あるドメインに固有の評価語は、そのドメインでは極性は変わらない (肯定的にも否定的に使われることはない)

これをもとに、以下のルールに従って評価語の極性を決定する。

- ルール 1
  評価語が既知の属性からの伝播で抽出されたとき、あるいは属性が既知の評価語からの伝播で抽出されたとき、既知の語と同じ極性を与える。属性には極性はないが、既知の属性の極性はその伝播元の評価語の属性と同じものとする。

- ルール 2
  評価語が既知の評価語からの伝播で抽出されとき、あるいは属性が既知の属性からの伝播で抽出されたとき、両者の間に逆接表現がない限り、既知の語と同じ極性を与える。

- ルール 3
  他のレビューから抽出された属性からの伝播で新しく抽出された評価語の場合、極性を与えない。

表 3.29: Qiu らの手法における評価語と属性抽出ルール [32]

	Observations	Constraints	Outputs
$\mathbf{R1}_1$	$S_{i(j)} \rightarrow S_{i(j)}\text{-}Dep \rightarrow S_{j(i)}$	$S_{j(i)} \in \{S\}$, $S_{i(j)}\text{-}Dep \in \{CONJ\}$, $POS(S_{i(j)}) \in \{JJ\}$	$s = S_{i(j)}$
$\mathbf{R1}_2$	$S_i \rightarrow S_i\text{-}Dep \rightarrow H \leftarrow S_j\text{-}Dep \leftarrow S_j$	$S_i \in \{S\}$, $S_i\text{-}Dep == S_j\text{-}Dep$, $POS(S_j) \in \{JJ\}$	$s = S_j$
$\mathbf{R2}_1$	$S \rightarrow S\text{-}Dep \rightarrow F$	$F \in \{F\}$, $S\text{-}Dep \in \{MR\}$, $POS(S) \in \{JJ\}$	$s = S$
$\mathbf{R2}_2$	$S \rightarrow S\text{-}Dep \rightarrow H \leftarrow F\text{-}Dep \leftarrow F$	$F \in \{F\}$, $S/F\text{-}Dep \in \{MR\}$, $POS(S) \in \{JJ\}$	$s = S$
$\mathbf{R3}_1$	$S \rightarrow S\text{-}Dep \rightarrow F$	$S \in \{S\}$, $S\text{-}Dep \in \{MR\}$, $POS(F) \in \{NN\}$	$f = F$
$\mathbf{R3}_2$	$S \rightarrow S\text{-}Dep \rightarrow H \leftarrow F\text{-}Dep \leftarrow F$	$S \in \{S\}$, $S/F\text{-}Dep \in \{MR\}$, $POS(F) \in \{NN\}$	$f = F$
$\mathbf{R4}_1$	$F_{i(j)} \rightarrow F_{i(j)}\text{-}Dep \rightarrow F_{j(i)}$	$F_{j(i)} \in \{F\}$, $F_{i(j)}\text{-}Dep \in \{CONJ\}$, $POS(F_{i(j)}) \in \{NN\}$	$f = F_{i(j)}$
$\mathbf{R4}_2$	$F_i \rightarrow F_i\text{-}Dep \rightarrow H \leftarrow F_j\text{-}Dep \leftarrow F_j$	$F_i \in \{F\}$, $F_i\text{-}Dep == F_j\text{-}Dep$, $POS(F_j) \in \{NN\}$	$f = F_j$

表 3.30: 表 3.29における記号の説明 [32]

記号	記号の意味
s または f	新しく抽出する評価語または属性
$\{S\}$ または $\{F\}$	既知の評価語または属性の集合
$S\text{-}Dep$ または $F\text{-}Dep$	一方を既知の評価語または属性としたときの二単語間の統語的関係
H	任意の単語
$POS(S)$ または $POS(F)$	S または F の品詞
$\{JJ\}$ または $\{NN\}$	形容詞または名詞を表す品詞の集合。形容詞は基本形、比較級、最上級の品詞を含む。名詞は単数形、複数形の品詞を含む。
$\{MR\}$	特定の係り受け関係の集合。修飾-被修飾関係、主語-述語関係、目的語-述語関係など
$\{CONJ\}$	接続関係 (and,or,but などで結ばれた関係)

表 3.31: Qiu らの手法の実験結果 [32]

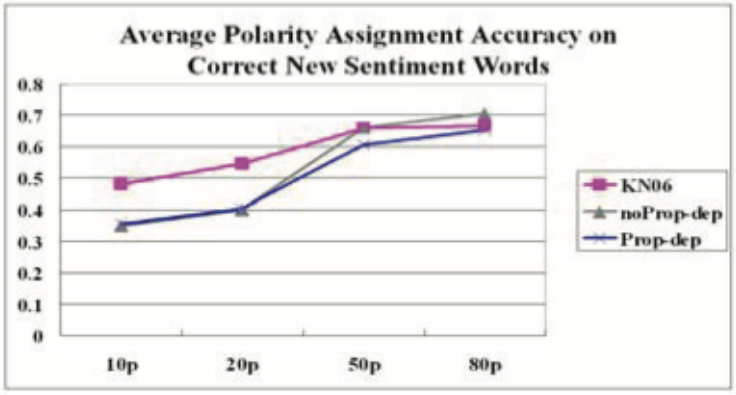


表 3.31は評価実験の結果を示している。横軸は初期の極性単語のリストの数を、縦軸は新たに獲得された評価語の極性の正解率である。Prop-dep はこの論文の提案手法を、noProp-dep は (伝播の仕組みを繰り返し使わずに) 初期の極性単語リストから表 3.29のルールを 1 回適用して新しい評価語と属性を獲得する手法を表す。一方、KN06 は Kanayama と Nasukawa による手法 [30] を表す。この手法では伝播の仕組みを用いていないため、提案手法と同じ条件で比較するために noProp-dep の実験結果が示されている。提案手法は初期の極性単語数が少ない場合には KNP06 よりも正解率が低いが、多い場合 (80 個) には KNP06 を上回る。初期の極性単語数が少ないときに正解率を向上させることを今後の課題としている。また、表 3.31とは別に、提案手法が KN06 と比べて再現率が高いことが示されている。これは伝播を繰り返すことでドメインに固有の評価語や属性が網羅的に獲得されたことを示している。

Liu と白井の研究

Liu と白井は、製品属性の異表記辞書を自動的に構築する手法を提案している [33]。一般に、製品の属性は様々な表現で書き表されるが、異表記辞書は同じ実体 (属性) を指す異なる表現を収録した辞書である。提案手法では処理を 2 つに分けている。まず最初に製品製造元のウェブページにある仕様表から属性を抽出し、初期の製品属性異表記辞書を作成する。次に、製品レビュー文から製品属性の異表記を抽出し、製品属性異表記辞書へ追加する。

最初の属性抽出では、ウェブページの表を解析し、属性と属性値の組を抽出する。次に、属性を属性値を要素とするベクトルで表現し、クラスタリングを行うことで、同じ実体を指す属性を 1 つにまとめる。次の段階のレビュー文からの属性抽出においては、属性を抽出するパターンを自動生成する。パターンの候補を生成し、その信頼度を表すスコアを算出し、それが高いものを採用する。パターンは以下のように定義される。$[A]$ は初期の製品属性異表記辞書 D_t に含まれる属性、$[E]$ は既存の評価辞書に登録されている評価語である。w_i は属性と評価語の間に出現する単語を表す。l は 1,2,3 のいずれかとしている。

$[A]\, w_1 ... w_l\, [E]$　　または　　$[E]\, w_1 ... w_l\, [A]$

このパターンにマッチする単語列を抽出パターンの候補とする。パターンの候補 P_i に対し、(3.21) 式によりパターン候補のスコア $S(P_i)$ を計算する。

$$S\left(P_i\right) = \frac{\text{パターン } P_i\text{にマッチしかつ } D_t\text{中の属性が抽出される文の数}}{\text{パターン } P_i\text{にマッチする文の数}} \tag{3.21}$$

マッチする文の数が 3 以上で、かつ $S\left(P_i\right)$ が 0.5 以上のパターンを製品属性抽出パターンとして獲得する。そして、このパターンをレビュー文に適用し、新しい製品属性を抽出する。

実験では、「PC」「カメラ」「テレビ」を対象に、属性の異表記辞書構築を試みた。最初のステップの仕様表からの属性抽出に関しては、精度 0.89、再現率 0.82 と十分高い性能が得られた。一方、2 番目のステップの自動獲得した抽出パターンによる属性抽出に関しては、表 3.32のように十分ではない結果となった。

表 3.32: Liu と白井によるレビュー文からの製品属性抽出の実験結果 [33]

	PC	カメラ	テレビ
パターン数	27	16	5
抽出属性数	108	64	65
D_t に含まれない属性数	69	45	58
異表記とみなせる属性数	26	3	7
抽出精度	0.34	0.067	0.12

3.3.2　評価語辞書作成手法の研究動向

表 3.33は、使用されている情報や素性によって、3.3.1項で紹介した各論文を分類した一覧表である。各列は分類の観点を表し、これらは表 3.15と同じである。表中における “-” は利用のないもの、“✓” は利用のあるものを示している。

表 3.33: 利用されている素性・情報による評価語辞書作成手法の分類

文献	形態素解析 (品詞タグ)	N グラム	構文解析	単語頻度	外部言語資源	ルールベース	教師あり機械学習	教師なし機械学習
1. [28]	✓	-	✓	✓	✓	✓	-	-
2. [29]	✓	-	✓	-	✓	✓	-	-
3. [30]	✓	-	-	-	✓	✓	-	-
4. [31]	✓	-	✓	-	-	✓	-	-
5. [32]	✓	-	✓	-	✓	✓	-	-
6. [33]	-	✓	-	-	✓	✓	-	-

極性判定や属性抽出とは異なり、教師あり機械学習や教師なし機械学習を用いた手法は存在しなかった。評価語辞書作成は、コーパスから新しい評価語や属性を自動獲得することが目標であり、機械学習で扱いやすい分類問題とは異なる問題設定であるため、当然の結果と考えられる。機械学習の手法を使わないとなれば、ルールベースの手法を採用するのが自然であり、全ての研究において評価語や属性を抽出するためのルールが使用されている。このルールは人手で作成されることもあるし、コーパスから自動獲得されることもある。前処理となる自然言語解析に着目すると、形態素解析の結果得られる品詞の情報がよく使われるのは当然だが、構文解析を使う研究が多いのは大きな特徴と言える。

3.1節で調査した極性判定の研究、3.2節で調査した属性抽出の研究の多くで評価語辞書が利用されている。評価語辞書は、これらの解析をする上で重要な知識であり、これをどのように作成するのか、あるいはどのように拡張するのかは重要な研究である。

3.4　スパム・フェイクレビュー判定手法

本節では、スパムレビューやフェイクレビューの自動判定に関連する研究動向の調査結果を報告する。

3.4.1　スパム・フェイクレビュー判定手法の精査

Jindal と Liu の研究

そもそも何を持ってスパムというのかは、米連邦取引委員会が 2009 年に出したガイダンス [102] を基準と考えることが多いとされている。Jindal と Liu の研究 [34] は、そのガイダンスが発表される以前の、レビュースパムに関する最初期の研究の 1 つとされている。論文の中で、彼らは独自に表 3.34のように 3 種類のスパムを定義している。

表 3.34: Jindal と Liu によるスパムレビューの分類と定義 [34]

タイプ 1　偽りの意見
タイプ 2　(対象製品ではなく) ブランドへの意見
タイプ 3　(広告や、質問とその回答のような) レビューでないもの

彼らは、580 万レビュー、214 万レビュアー、670 万商品の情報を Amazon.com から収集し、スパムレビューの特徴を解析した。その結果、スパムとして複製レビューあるいは一部の文言を変えた複製に近いレビューが多いことを発見した。このため、まず初めに、判定対象の文書が別の文書の複製であるかの判定を行い、複製と判定されたレビューに対してスパム判定を行っている。複製かどうかの判定は、Broder による Shingle（ドキュメント・ハッシュ）手法 [103] を利用している。この手法は、文書から連続する単語 (Shingle) を抜き出し、2 つの文書間でどれだけ共通する Shingle が存在するかで文書の同一性を判定

する。例えば、文 “This is a spam” から 2-shingle で抽出すると、“This is”, “is a”, “a spam” が抽出される。Jindal と Liu の論文では “2-gram” と記述されているが、これは 2-shingle を指すと思われる。二文書の Shingle の集合間の Jaccard 類似度係数が 90%以上のときに複製とみなしている。複製と判定された後のスパム判定では、タイプ 1 とそれ以外で異なる手法を提案している。タイプ 2 とタイプ 3 に対しては、人手によりラベル付けした 470 件のレビューから、タイプ 2 またはタイプ 3 か否かを判定するロジスティック回帰モデルを学習している。詳細は 3.5節で後述する。タイプ 1 に関しては、人手でラベル付けを行うことは難しいという判断から、“複製と判定されたスパムレビューを正例”、“それ以外のレビューを負例” とみなすことで、教師あり機械学習による分類を行なった。コンピュータや家電製品などの “mProducts” というカテゴリーに絞り、複製スパムレビューが 4488 件、他のレビューが 218514 件の訓練データを得た。また、ユーザーからのフィードバックの数、販売価格、レビュー投稿時期など、テキスト以外の情報の素性も利用されている。ただし、この実験は、厳密にはタイプ 1 のスパムを判定するものではない。論文でも、タイプ 1 の判定に関しては研究の初期段階であるとし、今後の研究課題としている。

Ott らの研究

Ott らは、シカゴ地域にある 5 つ星の評価を受けているホテル 20 件を選び、これらのホテルに対するスパムレビューを負例、スパムではないレビューを正例とし、これらを統計的手法により分類できるかを調査した [36]。レビューは旅行レビューサイトの TripAdvisor[12] から収集した。負例とするスパムレビューは、アマゾン・メカニカル・ターク (AMT) を用いて作成した[13]。AMT は、クラウドソーシングのための売買市場を提供するウェブサービスである。作業を発注する側は、不特定多数の作業者に対して作業内容の指示と金額の提示を行い、これに応じた作業者からの成果物を受け取る。著者らは、1 件あたり 1 ドルでスパムレビューの作成を発注している。また、作業者の条件として、アメリカ在住であり、過去の作業の承認率が 90%以上であること、としている。一方、正例となるスパムではないレビューとして、TripAdvisor に投稿されたレビューを用いる。但し、表 3.35に示した条件を満たすレビューは正例から除外する。

表 3.35: スパムではないレビューが満たすべき条件 [36]

1.	5 つ星のないレビュー	3130 件
2.	英語以外で記述されたもの	41 件
3.	150 文字以下のレビュー	150 件
4.	初めてのレビュー投稿者によるもの	1607 件

(件数は実際に削除されたレビューの件数)

表 3.35の 3. の条件は、負例とするスパムレビューを作成する際に、その長さを 150 文字以上としており、正例の長さもこれに合わせるために設定されている。4. の条件は、1 つ

[12] https://www.tripadvisor.com
[13] https://www.mturk.com

しか投稿していないレビュアーはスパマーである可能性が高いという Wu らの調査結果 [104] による。最終的に用意した正例の数は 6977 件である。

実験では、機械学習のための素性として品詞 (POS)、LIWC(The Linguistic Inquiry and Word Count)、N グラムを、機械学習アルゴリズムとしてナイーブベイズ (NB) とサポートベクターマシン (SVM) を用いて、スパムか否かを判定する分類器を学習する。また、素性と機械学習アルゴリズムの組合せを変えて分類器を学習し、それらを比較する。LIWC(The Linguistic Inquiry and Word Count) とは、Pennebaker らにより開発されたテキスト解析ソフトウェア [105] の出力である。このソフトウェアは、入力されたテキストに対し、文字数などのテキスト特性や単語のカテゴリ情報などを含む 80 次元のデータを出力する。実験結果を表 3.36に示す。

表 3.36: Ott らによるスパム判定実験の結果 [36]

			TRUTHFUL			DECEPTIVE		
Approach	**Features**	**Accuracy**	**P**	**R**	**F**	**P**	**R**	**F**
GENRE IDENTIFICATION	POS_{SVM}	73.0%	75.3	68.5	71.7	71.1	77.5	74.2
PSYCHOLINGUISTIC DECEPTION DETECTION	LIWC_{SVM}	76.8%	77.2	76.0	76.6	76.4	77.5	76.9
TEXT CATEGORIZATION	$\text{UNIGRAMS}_{\text{SVM}}$	88.4%	89.9	86.5	88.2	87.0	90.3	88.6
	$\text{BIGRAMS}^{+}_{\text{SVM}}$	89.6%	90.1	89.0	89.6	89.1	90.3	89.7
	$\text{LIWC+BIGRAMS}^{+}_{\text{SVM}}$	**89.8%**	89.8	**89.8**	**89.8**	**89.8**	89.8	**89.8**
	$\text{TRIGRAMS}^{+}_{\text{SVM}}$	89.0%	89.0	89.0	89.0	89.0	89.0	89.0
	$\text{UNIGRAMS}_{\text{NB}}$	88.4%	**92.5**	83.5	87.8	85.0	**93.3**	88.9
	$\text{BIGRAMS}^{+}_{\text{NB}}$	88.9%	89.8	87.8	88.7	88.0	90.0	89.0
	$\text{TRIGRAMS}^{+}_{\text{NB}}$	87.6%	87.7	87.5	87.6	87.5	87.8	87.6
HUMAN / META	JUDGE 1	**61.9%**	57.9	87.5	**69.7**	74.4	36.3	48.7
	JUDGE 2	56.9%	53.9	**95.0**	68.8	**78.9**	18.8	30.3
	SKEPTIC	60.6%	**60.8**	60.0	60.4	60.5	**61.3**	**60.9**

著者らは、大きく分けて 3 つの手法を比較している。1) ジャンル判定による手法 (GENRE IDENTIFICATION)[14]、2) 心理言語学による手法 (PSYCHOLINGUISTIC DECEPTION DETECTION)、3) テキスト分類による手法 (TEXT CATEGORIZATION) である。なお、人手による判定 (HUMAN / META) を基準値 (自動判定の正解率の上限) として示している。

表 3.36の 2 列目 “Features” における UNIGRAMS、BIGRAMS+、TRIGRAMS+は、それぞれ素性としてユニグラム、バイグラム、トライグラムを利用したことを表す。入力テキストは小文字に正規化され、語幹抽出はされていない。また、BIGRAMS+、TRIGRAMS+と “+” がついているのは、ユニグラム、バイグラムの素性も含むからである。各素性の下付き文字として表示されている SVM と NB は、それぞれサポートベクターマシン、ナイーブベイズを分類器に使用したことを表す。“TRUTHFUL” はスパムではないレビューを検出するタスク、“DECEPTIVE” はスパムレビューを検出するタスクの精度 (P)、再現率 (R)、F 値 (F) を示している。この結果から、LIWC とバイグラムを素性とした SVM に

[14] “ジャンル判定” と呼んでいるのは、Ott らは、スパム・フェイクレビューは創作物の副ジャンル、スパム・フェイクでないレビューは情報記事の副ジャンル、という見方をしているためである。

よる分類が最高の性能を示している。ただし、バイグラムと SVM の組み合わせでも同程度の性能が得られている。

Ott らは、2011 年に作成したスパムレビューデータセット [36] を拡張し、スパムレビューを自動判定する実験を行っている [38]。彼らは、スパムレビューには、1) 製品・サービスを持ち上げることを意図したポジティブなもの、2) 競合製品・サービスを批判・中傷するためのネガティブなもの、という 2 種類があるとしている。文献 [36] で作成したデータセットは 1) に対応したポジティブなスパムレビューであったので、2) に対応するネガティブなものを追加することを試みた。ネガティブスパムレビューの作成には、前回同様にアマゾン・メカニカル・タークを利用し、作業者選定条件なども同じ条件で、前回と同じシカゴ周辺にある 20 件のホテルに対して、各ホテルあたり 20 件、合計 400 件のスパムレビューを作成している。作成されたデータセットは、現在も公開されている[15]。一方、スパムではないレビューデータは、Expedia、Hotels.com、Orbitz、Priceline、TripAdvisor、Yelp といった 6 つの代表的な旅行サイトから収集している。具体的には、評価対象とした 20 のホテルに対して、1 つあるいは 2 つ星をつけたレビュー記事を収集した。作成したデータセットを用いて、ユニグラムとバイグラムを素性として、サポートベクターマシン (SVM) によって分類器を学習する。実験結果を表 3.37に示す。

表 3.37: Ott らによるスパム判定実験の結果 [38]

			TRUTHFUL			DECEPTIVE		
Train Sentiment	**Test Sentiment**	**Accuracy**	**P**	**R**	**F**	**P**	**R**	**F**
POSITIVE (800 reviews)	POSITIVE (800 reviews, Cross Val.)	89.3%	89.6	88.8	89.2	88.9	89.8	89.3
	NEGATIVE (800 reviews, Held Out)	75.1%	69.0	91.3	78.6	87.1	59.0	70.3
NEGATIVE (800 reviews)	POSITIVE (800 reviews, Held Out)	81.4%	76.3	91.0	83.0	88.9	71.8	79.4
	NEGATIVE (800 reviews, Cross Val.)	86.0%	86.4	85.5	85.9	85.6	86.5	86.1
COMBINED (1600 reviews)	POSITIVE (800 reviews, Cross Val.)	88.4%	87.7	89.3	88.5	89.1	87.5	88.3
	NEGATIVE (800 reviews, Cross Val.)	86.0%	85.3	87.0	86.1	86.7	85.0	85.9

表3.37の 1 列目の “POSITIVE”、“NEGATIVE”、“COMBINED” は、学習に利用したデータセットである。それぞれ、文献 [36] のポジティブなレビュースパムデータセット、この論文で作成されたネガティブなレビュースパムデータセット、この 2 つを結合したものを表す。2 列目の “Cross Val.”、“Held Out” は、それぞれ交差検定、ホールドアウト法を指す。ここでのホールドアウト法は、学習に用いたデータセットとテストに用いたデータセットの極性が異なること、つまりポジティブスパムレビュー (ネガティブスパムレビュー) を訓練データとしたときにはネガティブスパムレビュー (ポジティブスパムレビュー) をテストデータとすることを表す。“TRUTHFUL” はスパムではないレビューを検出するタスク、“DECEPTIVE” はスパムレビューを検出するタスクの精度 (P)、再現率 (R)、F 値 (F) を示している。実験の結果、ポジティブスパムもネガティブスパムもほぼ同じ正解率で検出できることがわかる。また、ホールドアウトのときの方が結果が悪くなる傾向が見られる。さらに、人手で作成したスパムレビューとウェブから収集したスパムでないレビュー

[15] http://myleott.com/op-spam.html

の違いについて調査している。スパムレビューには、1) 部屋の広さ、フロア、ホテルの場所などといったことを表す単語が少ない、2) 動詞が多い、3)1 人称単数形の単語がスパムでないレビューの 2 倍くらい出現する、という違いがある。

Mukherjee らの研究

Mukherjee らは、スパマーグループを検知する手法を提案している [37]。彼らが提案する GSRank(Group Spam Rank) は、頻出アイテム集合マイニングにより、レビューアーのグループをスパマーグループの可能性が高い順にランク付けする。まず、GSRank を構築するためのラベル付きデータセットを作成する。Amazon.com から、53469 レビュアー、109518 レビュー、39392 製品の情報を収集する。このデータは、Jindal と Liu が収集したデータセット [34] に新しいデータを加えて拡張したものである。ここからスパマーグループの候補を抽出し、それを 8 人の専門家が “スパマー”、“スパマーでない”、“どちらとも言えない” に分類する。8 週間で 2431 グループに対してラベル付けを行った。次に、GSRank の計算を行う基礎的な情報として、グループおよび個人がスパマーかどうかを判断するためのパラメータを、グループ向けに 8 つ、個人向けに 4 つ定義している。例えば、スパマーグループは同時期に同一製品に対してスパムレビューを書くことから、どれだけ短い時間内に書き込みを行っているかを表すパラメータを設定する。また、グループ内でレビューの複製を使いまわすことから、同一製品に対するグループメンバーのレビューが似る傾向にあるため、レビューの類似度をパラメータに設定する。個人向けのパラメータとしては、個人が書き込んだレビュー間の類似度、製品に与える星の数の傾向などが定義されている。これらのパラメータから、グループとターゲット製品、グループのメンバーとターゲット製品、グループとグループのメンバーといった 3 つの二項関係モデルにより、GSRank 値を計算し、スパマーグループかどうかを判定する。

上記の研究とは別に、Mukherjee らは、米大手レビューサイトの Yelp[16]が実装しているスパムフィルターのアルゴリズムを明らかにするための実験を行っている [39]。Yelp はスパムレビューフィルターを実装し、既にサービスとして提供していることが知られている。このスパムレビューフィルターはうまく機能しているように思われるが、そのアルゴリズムは公開されていない[17]。Yelp から収集したレビューデータを用いて、先行研究の手法を用いてスパムレビューを判定する実験を行い、その結果を分析することで、Yelp がどのようにスパムフィルターを実装しているのか、どのような素性が利用されているのかを考察する。

実験に用いるデータは、Yelp のウェブサイトから、シカゴ近郊の 130 件のレストランと 85 件のホテルのレビューを、Yelp のフィルターありとフィルターなしの両方の条件で取得する。フィルターなしの条件のみで取得されたレビューをスパムとみなす。データの偏りを防ぐため、レビュー件数の多い施設に限定せず、レビュー件数の少ない施設も含める。ホテルについては 5678 件のレビューと 5124 人のレビュアー、レストランについては

[16] https://www.yelp.com

[17] Why Yelp Has A Review Filter https://www.yelpblog.com/2009/10/why-yelp-has-a-review-filter

58517 件のレビューと 35593 人のレビュアーの情報を収集した。

表 3.38: Yelp データセットによるスパム判定実験結果 [39]

Features	P	R	F1	A
Word unigrams (WU)	62.9	76.6	68.9	65.6
WU + IG (top 1%)	61.7	76.4	68.4	64.4
WU + IG (top 2%)	62.4	76.7	68.8	64.9
Word-Bigrams (WB)	61.1	79.9	69.2	64.4
WB+LIWC	61.6	69.1	69.1	64.4
POS Unigrams	56.0	69.8	62.1	57.2
WB + POS Bigrams	63.2	73.4	67.9	64.6
WB + Deep Syntax	62.3	74.1	67.7	64.1
WB + POS Seq. Pat.	63.4	74.5	68.5	64.5

(a): Hotel

P	R	F1	A
64.3	76.3	69.7	66.9
64.0	75.9	69.4	66.2
64.1	76.1	69.5	66.5
64.5	79.3	71.1	67.8
64.6	79.4	71.0	67.8
59.5	70.3	64.5	55.6
65.1	72.4	68.6	68.1
65.8	73.8	69.6	67.6
66.2	74.2	69.9	67.7

(b): Restaurant

表 3.38 は実験結果を示している。機械学習の素性として、ユニグラム (WU)、バイグラム (WB)、品詞 (POS) などの他、Ott らの実験 [36] と比較するために LIWC も利用している。表中の IG は、情報利得を指標として素性選択を行ったことを表す。“P”、“R”、“F”、“A” は、それぞれ精度、再現率、F 値、正解率を示す。Ott らは、表 3.36にあるように 90%近い正解率を達成したと報告している [36] が、この実験での同条件のモデルの正解率は、ホテルのデータで 64.4%、レストランのデータで 67.8%であった。2 つの実験で正解率に大きな差がある要因を以下のように考察している。Ott らの実験データではアマゾン・メカニカル・タークで人為的に作成されたスパムレビューを用いているが、スパムではないレビューと比べて単語の頻度分布に大きな違いがあり、このことがスパム判定を比較的容易にしていると考えられる。一方、この実験で収集したスパムレビューにおける単語の頻度分布は、スパムでないレビューとそれほど大きく変わらないため、スパムの判定が難しくなっている。別の見方では、人為的に作成したスパムは実際のスパムと内容に大きな乖離があり、前者を利用した実験ではスパム判定手法の性能を過大に見積る可能性がある。

上記の考察を踏まえ、言語処理によってテキストから得られる素性を利用するだけでなく、レビュアーの行動分析に基づく “Behavioral Features (BF)” という素性を利用することを提案している。BF として、1 日あたりのレビュー投稿数、レビュー投稿数に占めるポジティブレビューの比率、レビューのテキスト長 (約 80%のスパマーの平均的なテキスト長は 135 文字程度となっている)、レビュアーがつけた星の数の平均からの逸脱度、以前に投稿したレビューとのテキスト類似度、が提案されている。各指標は、0 から 1 の値を取り、かつ 0 に近いほどスパムの可能性が高くなるように設定されている。閾値を 0.5、0.25 など設定し、これ以下のレビューをスパムと判定する。

表 3.39: レビューアーの行動分析を素性としたモデルの実験結果 [39]

Feature Setting	P	R	F1	A
Unigrams	62.9	76.6	68.9	65.6
Bigrams	61.1	79.9	69.2	64.4
Behavior Feat.(BF)	81.9	84.6	**83.2**	**83.2**
Unigrams + BF	83.2	80.6	81.9	83.6
Bigrams + BF	86.7	82.5	**84.5**	**84.8**

(a): Hotel

P	R	F1	A
64.3	76.3	69.7	66.9
64.5	79.3	71.1	67.8
82.1	87.9	**84.9**	**82.8**
83.4	87.1	85.2	84.1
84.1	87.3	**85.7**	**86.1**

(b): Restaurant

表 3.39は、表 3.38と同じデータセットを用いて、BF を素性として使うモデルと使わないモデルを比較した実験結果である。BF を導入することで正解率が大きく改善されていることがわかる。これにより、Yelp のフィルタも同様の手法を用いているのではないかと考察している。

その他の研究

Lim らは、レーティングの行動パターンからスパマーを判定する手法を提案している [35]。レーティングとは、ユーザーが製品などの評価対象に対して点数で評価を示す行動であり、通常は星の数で評価の高さを表す。彼らの手法では、判定対象ユーザと他のユーザのレビュー記事との類似度と、判定対象ユーザと他のユーザのレーティング行動との類似度を計算し、これらの 2 つのスコアの合計値を基にスパマーを判定する。テキスト間の類似度は、バイグラムを特徴、その TFIDF を値とするベクトルでテキストを表現し、そのベクトル間の類似度で測る。IDF を利用することで、出現頻度の高いバイグラムの影響を抑えている。一方、レーティング行動の類似度を測る素性として、1) レビュー投稿日時、2) 特定の製品グループに対するレーティングパターン、を利用している。1) は、スパムは製品がリリースされてから短時間で投稿される傾向があることから、その製品に対するレビューの平均投稿時間と比べて短い場合を検出するための素性である。2) は、特定の製品グループに対して、ある期間に集中して非常に高いレーティングをつける、あるいは低いレーティングをつける行動を検知するための素性である。ここで、特定の製品グループとは同じような属性を持つ製品群を指す。

Amazon.com から、Jindal と Liu らと同様 [34] に “MProducts” カテゴリーの製品とレビューデータを収集した。前処理として、無名のレビュー投稿記事の削除、二重に存在する同一商品の削除、レビュー投稿数が 3 以下のユーザーの削除、レビュー数が 3 以下の製品の削除を行い、さらにブランド名の類義語処理を人手により行っている。50 人のレビューアーに対してスパマーか否かの判定を行い、その結果を 3 人の評価者により評価した。その結果、“Unhelpfulness score” の数でスパマーかを判定するベースラインよりも良い正解率が得られたと報告している。Amazon.com では、レビューが役に立たなかった場合に unhelpness review として報告できる機能があるが、“Unhelpfulness score” とはその報告数である。

Rayana と Akoglu は、SpEagle(Spam Eagle) と呼ばれるスパムレビュー・レビュアー判定のための統合フレームワークを提案している [40]。SpEagle では、レビューのテキス

ト、星の数、投稿日時などの情報と、レビュアーの関係をグラフ化した情報を用いて判定を行う。テキストの類似度や単語の頻度分布だけでなく、レビュアーの行動データを利用する点は、Mukherjee らの手法 [37, 39] と近いとも言える。彼らの手法では、ユーザー-レビュー-製品のグラフネットワークを構築する。各レビューノードは対応するユーザーと製品ノードへ接続されている。もしレビューテキストを含むメタデータの内部スコアがスパムであると疑わしい数値を示した場合、グラフ上で接続するユーザー、レビューあるいは製品も疑わしいと考える。スパムかどうかを判定する素性としては、1 日あたりの投稿数、ポジティブ・ネガティブレビューの投稿数に占める比率、サイトのメンバーになってからの日数、レビューのテキスト長、他の投稿との類似度などを利用している。

3.4.2 スパム・フェイクレビュー判定手法の研究動向

表 3.40は、使用されている情報や素性によって、3.4.1項で紹介した各論文を分類した一覧表である。各列は分類の観点を表し、これらは表 3.15と同じである。

表 3.40: 利用されている素性・情報によるスパム・フェイクレビュー判定手法の分類

文献	形態素解析 (品詞タグ)	N グラム	構文解析	単語頻度	外部言語資源	ルールベース	教師あり機械学習	教師なし機械学習
1. [34]	-	✓	-	-	✓	-	✓	-
2. [35]	-	✓	-	✓	-	-	-	-
3. [36]	✓	✓	-	-	-	✓	-	-
4. [37]	-	-	-	-	-	✓	-	-
5. [38]	-	✓	-	✓	✓	-	✓	-
6. [39]	-	-	-	-	-	-	-	-
7. [40]	-	✓	-	✓	✓	✓	-	-

スパム・フェイクレビュー判定には、N グラムがよく手がかりとして使われていることがわかった。教師あり機械学習の手法が少ないのは、スパムを正解ラベルとして付与したデータセットの整備が進んでいないためと考えられる。この表には載せていないが、テキストから得られる素性だけではなく、ユーザーの行動パターンやスパマーの傾向など、テキスト以外の情報も素性とする研究も多い。調査の結果、テキストの分析だけではスパムやスパマーを判定することには限界があることがわかった。

スパム・フェイクレビューには、下記のように様々なケースがある。

1. 評価対象の製品について言及していないもの
2. 製品について言及しているものの、業者・関係者による意図を持ったもの
3. 誤解・間違いによるもの

4. 製品ではなく、その製品の製造元、販売者に対するもの

調査するまでは、上記のような分類を行う研究があるものと期待していたが、実際には単にスパムか否かを判定する研究しか存在しなかった。3.2節で報告した属性抽出手法を応用することで、スパムを上記のリストのようなカテゴリに分類することができるのではないかと考えている。

3.5 属性が評価対象のものであるかの判定手法

本課題研究では、3.2節で述べた手法によって抽出された属性が、実際に評価対象となっている製品の属性と一致しているかを判定する手法、あるいはそれに近い関連研究があることを期待して文献調査を行った。結果として、そのような先行研究は見つからなかった。ただし、評価対象となる製品の属性ではないが、評価対象となる製品そのものについてレビューが言及しているかを判定する先行研究が1件見つかった。この Jindal と Lin の論文 [34] で得られた知見は、意見文が指す属性が真に評価対象の属性であるかを判定するのに役立つと考えられるため、本節で紹介する。文献は1件しかないが、これまでの書き方に合わせて、使用されている要素技術を表3.41にまとめる。表中における “-” は素性の利用がないことを、“✓” は利用があることを示している。

表 3.41: 属性が評価対象のものであるかの判定手法で使われている要素技術

文献	形態素解析 (品詞タグ)	N グラム	構文解析	単語頻度	外部 言語資源	教師あり 機械学習	教師なし 機械学習
1. [34]	-	✓	-	-	✓	✓	-

3.4.1項でも述べたように、Jindal と Liu は、“(対象製品ではなく) ブランドへの意見” と “(広告や質問とその回答のような) レビューでないもの” の判別を、教師あり機械学習手法の一つであるロジスティック回帰により、非常に高い精度で実現している。

機械学習には36種類の素性を利用している。これらの素性は “Review Centric Features”、“Reviewer Centric Features”、“Product Centric Features” の3つに分類される。さらに、それぞれのカテゴリに分類される素性は、テキストから得られる素性とテキスト以外から得られる素性に細分される。テキストから得られる素性の例として、レビューに占めるポジティブ／ネガティブな極性を表す単語の比率、レビューと製品属性のコサイン類似度、レビューの全単語数に占めるブランド名・数値表現・大文字のみからなる単語の比率といったものがある。最初の素性を得るための極性辞書として、Hu と Liu が作成した極性辞書 [60] を拡張したものを利用している。しかし、具体的にどのような単語を極性辞書に追加したのかなど、極性辞書の拡張の詳細は不明である。一方、テキスト以外から得られる素性としては、レビューに対するフィードバックがある。ここでのフィードバックとは、レビュー記事に対する他のユーザからの「役に立った」「役に立たなかった」の

フィードバックを指す。フィードバックに関する素性として、1) フィードバックの数、2)「役に立った」というフィードバックの数、3) その比率、の 3 つを用いている。これらは “Review Centric Features” に分類される。

実験では、人手によって Type 2 もしくは Type 3 であるかを判定した 470 件のスパムレビュー記事のセットを用いた。10 分割交差検定を行い、AUC の平均値を評価指標とした。実験結果を表 3.42に示す。

表 3.42: Type 2 と Type 3 のスパム判定実験の結果 [34]

Spam Type	Num reviews	AUC	AUC – text features only	AUC – w/o feedbacks
Types 2 & 3	470	98.7%	90%	98%
Type 2 only	221	98.5%	88%	98%
Type 3 only	249	99.0%	92%	98%

3 列目の “AUC” は 36 種類の素性全てを用いたとき、4 列目の “AUC - text features only” は “Review Centric Features” におけるテキストに関する素性のみを利用したとき、5 列目の “AUC - w/o feedbacks” はレビューに対するフィードバックの素性を使わなかったときの結果である。全素性を用いたときとテキスト素性だけを用いたときの AUC の差が大きいことから、スパムの判定にテキスト以外の素性を利用することの有効性が示されている。一方、フィードバック素性を除いても正解率はほとんど変わらなかったことから、フィードバック素性はスパム判定に有効でないことがわかった。この理由として、フィードバックもまたスパムの対象となっているから、と推測されている。

Jindal と Liu は比較的良好な結果を報告しているが、懸念されることもある。Mukherjee らが、Ott らによる実験結果 [36] を再現できなかったと報告 [39] している際の分析では、実験のためにクラウドソーシングによって作成したフェイクレビューデータと、現実のデータに大きな差があることが指摘されている。Jindal と Liu の研究でも人工的に作成したスパムレビューの記事を実験に利用しているため、より現実のスパムレビューに近いデータを用いた再現実験をした方が良いと言える。また、表 3.42の実験結果とは別に、“嘘の意見” を判定する場合、行動分析による素性を加味しても AUC で 78%、これを除くと 63% という結果が得られている。テキストのみの分析によるスパム判定という点では改善の余地が残されているように思われる。

本節の冒頭で述べたように、製品の属性に対して意見を述べている文があったとき、言及されている属性が評価対象の属性であるかを判定する手法を見つけることはできなかった。このため、これを実現する手法を探究することは今後の重要な研究課題と言える。3.1節、3.2節、3.3節で紹介した研究の成果を活用し、レビューの評価対象属性が実際の製品の属性と一致しているのものであるかを判定できれば、評価対象に対する評判をより正確に把握できると同時に、少なくとも製品について言及していないようなスパムレビュー・フェイクレビューを自動判別できるようになると考える。

第4章　結論

本課題研究では、極性判定・属性抽出に関する技術を 5 つのグループに分け、過去 10 年ほどの研究動向を調査した。各グループ毎に得られた知見を以下にまとめる。

1. 極性判定

   判定対象は文書から製品の属性へとより小さい範囲のテキストに移っていく傾向が確認できた。単語 N グラム、品詞、構文情報、極性語 (極性辞書に登録されている語) などを素性する教師あり機械学習から、トピックモデルによる手法、さらに深層学習を利用したものへと極性判定手法の主流が移り変わってきた。Bag-of-Words のような語順を保持しないモデルでは否定語、否定的な接続語による極性反転によって判定誤りが生じ、正解率を落とす原因となる。極性辞書を利用する場合、同一の単語でもドメインによって極性の違い、極性の強さの違いがあるため、ドメイン毎に最適な辞書が必要となる。ここ数年多くなってきた深層学習を用いた手法では、品詞、極性辞書などの言語資源を用いず、CBOW などの単語分散表現を入力としてある程度の成果を挙げている。

2. 属性抽出

   極性辞書を利用する場合、1と同様に極性語がドメインに依存するという問題点があり、このことが属性抽出の再現率低下の原因となっている。対象ドメイン毎に辞書が必要となる。深層学習による手法が多くなるまでは、多くの手法で品詞が素性として利用されていた。最近の深層学習による手法では品詞などは利用せず単語分散表現と組み合わせた手法が多い。

3. 評価語辞書作成

   1、2とは違い、多くの手法が、まず構文解析を行い、その結果を利用したルールベースの手法となっている。また WordNet などの外部言語資源を利用している手法も多い。機械学習、深層学習によるものはなかった。製品属性の異表記辞書、ドメイン依存の辞書をいかに効率的に構築するか、あるいは既存辞書をいかに拡張するか、といった提案があるのも特徴的である。

4. スパム・フェイクレビュー判定

スパムか否かといった二値分類問題へ落とし込む手法が多い。テキスト分析だけでスパム・フェイクを判定するのは難しいことから、レビューアーの行動分析によってスパマー・スパマーグループを判定する方向へと移り変わっている。

5. 属性が製品について言及しているかの判定

   3.5節でも言及したが、レビュー文が真に評価対象に対する意見を述べているかを判定する研究があることを期待したが、見つからなかった。フェイク・スパム判定も、レビュアーの行動分析を素性として活用する方向になりつつあり、期待していたものとは違った。今後の重要な研究課題と言える。

極性判定・属性抽出については非常に多くの研究・提案がされてきており、本報告書ではその一部を紹介しただけに留まる。とはいえ、他論文からの参照件数の多い主要な論文は網羅しており、極性判定、属性抽出、その関連研究の研究動向を把握することができた。今後、日本語での研究事例についてさらに調査したい。

また、3.4.1項で述べたように、Mukherjee らは、Ott らの実験結果 [36] を再現できなかったとし、その原因として、Ott らは人工的なデータを使っていたのに対し Mukherjee らは現実のスパムレビューのデータを用いたためと報告している [39]。現実に即したデータを使用しての研究が重要だと痛感した。特に深層学習を利用する場合には、どれだけ大規模なデータがあるかが決め手となるため、現実的なデータをどのように大量に集めるかが重要な課題になると考えられる。

今後の課題として、5の調査結果を踏まえ、属性が製品について言及しているかの判定手法についての研究に取り組むことが挙げられる。考えられる手法としては、与えられたレビュー文からの属性抽出・極性判定と、その意見が評価対象に向けられたものかどうかの判定が必要となる。このため、調査対象とした論文の成果から参考にできる手法がいくつかある。また、この研究により評価対象に対する正確な意見を消費者やメーカーに提供することが可能になり、社会的な面からも意義のある研究と言える。

参考文献

[1] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 27–35, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.

[2] Peter D Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 417–424, 2002.

[3] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86, 2002.

[4] Hong Yu and Vasileios Hatzivassiloglou. Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pp. 129–136, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[5] Minqing Hu and Bing Liu. Mining Opinion Features in Customer Reviews. In *Proceedings of AAAI*, p. 6, 2004.

[6] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pp. 347–354, 2005.

[7] Soo-Min Kim, Patrick Pantel, Tim Chklovski, and Marco Pennacchiotti. Automatically assessing review helpfulness. p. 423. Association for Computational Linguistics, 2006.

[8] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, Bollywood, Boomboxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 440–447, Prague, Czech Republic, 2007. Association for Computational Linguistics.

[9] Ivan Titov and Ryan McDonald. Modeling Online Reviews with Multi-grain Topic Models. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pp. 111–120, New York, NY, USA, 2008. ACM.

[10] Chenghua Lin and Yulan He. Joint Sentiment/Topic Model for Sentiment Analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pp. 375–384, New York, NY, USA, 2009. ACM.

[11] Alexander Pak and Patrick Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *LREC*, Vol. 10, pp. 1320–1326, 2010.

[12] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-Based Methods for Sentiment Analysis. *Computational linguistics*, Vol. 37, No. 2, pp. 267–307, 2011.

[13] Richard Socher, Alex Perelygin, and Jy Wu. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, pp. 1631–1642, 2013.

[14] Yoon Kim. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[15] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based LSTM for Aspect-level Sentiment Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 606–615, Austin, Texas, November 2016. Association for Computational Linguistics.

[16] 芥子育雄, 鈴木優, 吉野幸一郎, グラムニュービッグ, 大原一人, 向井理朗, 中村哲. 単語意味ベクトル辞書を用いたTwitterからの評判情報抽出. データ工学と情報マネジメント論文特集, No. 4, pp. 530–543, 2017.

[17] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. Recurrent Attention Network on Memory for Aspect Sentiment Analysis. pp. 452–461. Association for Computational Linguistics, 2017.

[18] Niklas Jakob and Iryna Gurevych. Extracting Opinion Targets in a Single- and Cross-domain Setting with Conditional Random Fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pp. 1035–1045, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[19] Arjun Mukherjee and Bing Liu. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, pp. 339–348, 2012.

[20] Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Exploiting Domain Knowledge in Aspect Extraction. In *Proc. of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP '13*, pp. 1655–1667, 2013.

[21] Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. Aspect Extraction with Automated Prior Knowledge Learning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 347–358, 2014.

[22] Soujanya Poria, Erik Cambria, Lun-Wei Ku, Chen Gui, and Alexander Gelbukh. A Rule-Based Approach to Aspect Extraction from Product Reviews. In *second workshop on natural language processing for social media (SocialNLP)*, pp. 28–37, 2014.

[23] 中野裕介, 湯本高行, 新居学, 上浦尚武. 機械学習による商品レビューの属性-意見ペアの抽出. 情報処理学会研究報告, Vol. 2015, No. 14, pp. 1–8, 2015.

[24] Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. Automated Rule Selection for Aspect Extraction in Opinion Mining. In *IJCAI*, 2015.

[25] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. Aspect Extraction for Opinion Mining with a Deep Convolutional Neural Network. *Knowledge-Based Systems*, Vol. 108, pp. 42–49, 2016.

[26] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An Unsupervised Neural Attention Model for Aspect Extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 388–397, 2017.

[27] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Coupled Multi-Layer Attentions for Co-Extraction of Aspect and Opinion Terms. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, p. 7, 2017.

[28] 小林 のぞみ, 乾 健太郎, 松本 裕治, 立石 健二, 福島 俊一. 意見抽出のための評価表現の収集. 自然言語処理, Vol. 12, No. 3, pp. 203–222, July 2005.

[29] 高村 大也, 乾 孝司, 奥村 学. スピンモデルによる単語の感情極性抽出. 情報処理学会論文誌, Vol. 47, No. 2, pp. 627–637, 2006.

[30] Hiroshi Kanayama and Tetsuya Nasukawa. Fully Automatic Lexicon Expansion for Domain-oriented Sentiment Analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pp. 355–363, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[31] Nobuhiro Kaji and Masaru Kitsuregawa. Building Lexicon for Sentiment Analysis from Massive Collection of HTML Documents. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 1075–1083, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[32] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Expanding Domain Sentiment Lexicon through Double Propagation. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*, pp. 1199–1204, 2009.

[33] LIU Chaoyu, 白井清昭. 評判情報分析のための製品属性の異表記辞書の自動構築. 言語処理学会第 22 回年次大会, 2016.

[34] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of the international conference on web search and web data mining 2008*, pp. 219–230, 2008.

[35] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. Detecting Product Review Spammers Using Rating Behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pp. 939–948, New York, NY, USA, 2010. ACM.

[36] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 309–319, 2011.

[37] Arjun Mukherjee, Bing Liu, and N Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st international conference on World Wide Web*, pp. 191–200, 2012.

[38] Myle Ott, Claire Cardie, and Jeffrey T Hancock. Negative Deceptive Opinion Spam. In *NAACL-HLT*, pp. 497–501, 2013.

[39] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and N S Glance. What yelp fake review filter might be doing? In *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*, pp. 409–418, 2013.

[40] Shebuti Rayana and Leman Akoglu. Collective Opinion Spam Detection: Bridging Review Networks and Metadata. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '15*, pp. 985–994, 2015.

[41] Peter D. Turney. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In G. Goos, J. Hartmanis, J. van Leeuwen, Luc De Raedt, and Peter Flach, editors, *European Conference on Machine Learning*, Vol. 2167, pp. 491–502, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[42] Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the Semantic Orientation of Adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, pp. 174–181, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.

[43] Janyce Wiebe. Learning Subjective Adjectives from Corpora. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 735–740. AAAI Press, 2000.

[44] Janyce Wiebe, Rebecca Bruce, Matthew Bell, Melanie Martin, and Theresa Wilson. A corpus study of evaluative and speculative language. Vol. 16, pp. 1–10. Association for Computational Linguistics, 2001.

[45] Kenneth Ward Church and Patrick Hanks. Word Association Norms, Mutual Information, and Lexicography. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pp. 76–83, Vancouver, British Columbia, Canada, June 1989. Association for Computational Linguistics.

[46] Kenneth Church, William Gale, Patrick Hanks, and Donald Hindle. Using Statistics in Lexical Analysis. In *Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon*, pp. 115–164. 1991.

[47] Thomas K Landauer and Susan T Dutnais. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. In *Psychological Review*, pp. 211–240, 1997.

[48] Andrew McCallum and Kamal Nigam. A comparison of event models for Naive Bayes text classification. In *In Aaai-98 Workshop on Learning for Text Categorization*, pp. 41–48. AAAI Press, 1998.

[49] Vasileios Hatzivassiloglou and Janyce M. Wiebe. Effects of Adjective Orientation and Gradability on Sentence Subjectivity. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, COLING '00, pp. 299–305, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[50] Yoav Freund and Robert E Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.*, Vol. 55, No. 1, pp. 119–139, August 1997.

[51] Robert E. Schapire and Yoram Singer. BoosTexter: A Boosting-based Systemfor Text Categorization. *Mach. Learn.*, Vol. 39, No. 2-3, pp. 135–168, May 2000.

[52] V. N. Phu and P. T. Tuoi. Sentiment classification using Enhanced Contextual Valence Shifters. In *2014 International Conference on Asian Language Processing (IALP)*, pp. 224–229, October 2014.

[53] Bo Pang and Lillian Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04*, pp. 271–es, Barcelona, Spain, 2004. Association for Computational Linguistics.

[54] Casey Whitelaw, Navendu Garg, and Shlomo Argamon. Using appraisal groups for sentiment analysis. p. 625. ACM Press, 2005.

[55] Alistair Kennedy and Diana Inkpen. SENTIMENT CLASSIFICATION of MOVIE REVIEWS USING CONTEXTUAL VALENCE SHIFTERS. *Computational Intelligence*, Vol. 22, No. 2, pp. 110–125, May 2006.

[56] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. p. 9, 2013.

[57] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning*, Vol. 12, pp. 2493–2537, 2011.

[58] Bo Pang and Lillian Lee. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Proceedings of the 43rd*

Annual Meeting on Association for Computational Linguistics, ACL '05, pp. 115–124, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[59] Xin Li and Dan Roth. Learning Question Classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pp. 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[60] Minqing Hu and Bing Liu. Mining and Summarizing Customer Reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177, 2004.

[61] Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating Expressions of Opinions and Emotions in Language. *Language Resources and Evaluation*, Vol. 39, No. 2-3, pp. 165–210, May 2005.

[62] Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 151–161, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[63] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic Compositionality Through Recursive Matrix-vector Spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pp. 1201–1211, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[64] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 655–665, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[65] Quoc Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pp. II–1188–II–1196, Beijing, China, 2014. JMLR.org.

[66] Karl Moritz Hermann and Phil Blunsom. The Role of Syntax in Vector Space Models of Compositional Semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 894–904, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[67] Li Dong, Furu Wei, Shujie Liu, Ming Zhou, and Ke Xu. A Statistical Parsing Framework for Sentiment Classification. *Comput. Linguist.*, Vol. 41, No. 2, pp. 293–336, June 2015.

[68] N. Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, Vol. 2, No. 3, pp. 113–124, September 1956.

[69] Sida Wang and Christopher Manning. Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 90–94, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[70] Sida I Wang and Christopher D Manning. Fast dropout training. In *Proceedings of ICML 2013*, p. 9, 2013.

[71] Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. Dependency Tree-based Sentiment Classification using CRFs with Hidden Variables. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 786–794, Los Angeles, California, June 2010. Association for Computational Linguistics.

[72] Bishan Yang and Claire Cardie. Context-aware Learning for Sentence-level Sentiment Analysis with Posterior Regularization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 325–335, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[73] João Silva, Luísa Coheur, Ana Cristina Mendes, and Andreas Wichert. From Symbolic to Sub-symbolic Information in Question Classification. *Artif. Intell. Rev.*, Vol. 35, No. 2, pp. 137–154, February 2011.

[74] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective LSTMs for Target-Dependent Sentiment Classification. *arXiv:1512.01100 [cs]*, December 2015. arXiv: 1512.01100.

[75] Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min-yen Kan, and Kathleen R. Mckeown. Simfinder: A flexible clustering tool for summarization. In *In Proceedings of the NAACL Workshop on Automatic Summarization*, pp. 41–49, 2001.

[76] George Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[77] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain Adaptation with Structural Correspondence Learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pp. 120–128, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[78] Jaime Guillermo Carbonell. *Subjective Understanding: Computer Models of Belief Systems.* January 1979.

[79] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[80] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie Review Mining and Summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, CIKM '06, pp. 43–50, New York, NY, USA, 2006. ACM.

[81] Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. Jointly Modeling Aspects and Opinions with a MaxEnt-LDA Hybrid. *Computational Linguistics*, Vol. 16, No. October, pp. 56–65, 2010.

[82] David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating Domain Knowledge into Topic Modeling via Dirichlet Forest Priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 25–32, New York, NY, USA, 2009. ACM.

[83] David M Blei. Latent Dirichlet Allocation. *Latent Dirichlet Allocation*, Vol. 3, pp. 993–1022, 2003.

[84] Leonard Kaufman and Peter Rousseeuw. *Finding Groups in Data: An Introduction To Cluster Analysis.* 1990.

[85] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, Vol. 15, No. 1, pp. 55–86, July 2007.

[86] Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Discovering Coherent Topics Using General Knowledge. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pp. 209–218, New York, NY, USA, 2013. ACM.

[87] David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processin*, pp. 262–272, 2011.

[88] Erik Cambria, Daniel Olsher, and Dheeraj Rajagopal. SenticNet 3: A Common and Common-Sense Knowledge Base for Cognition-Driven Sentiment Analysis. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, June 2014.

[89] Ivan Cruz, Alexander Gelbukh, and Grigori Sidorov. Implicit Aspect Indicator Extraction for Aspect-based Opinion Mining. *International Journal of Computational Linguistics and Application*, Vol. 5, No. 2, pp. 135–152, 2014.

[90] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Opinion Word Expansion and Target Extraction through Double Propagation. *Computational linguistics*, Vol. 37, No. 1, pp. 9–27, 2011.

[91] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*, January 2013. arXiv: 1301.3781.

[92] Julian McAuley and Jure Leskovec. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pp. 165–172, New York, NY, USA, 2013. ACM.

[93] Zhiqiang Toh and Wenting Wang. DLIREC: Aspect Term Extraction and Term Polarity Classification System. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 235–240, 2014.

[94] Samuel Brody. An Unsupervised Aspect-Sentiment Model for Online Reviews. *Computational Linguistics*, No. June, pp. 804–812, 2010.

[95] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web - WWW '13*, pp. 1445–1456, New York, New York, USA, 2013. ACM Press.

[96] Linlin Wang, Kang Liu, Zhu Cao, Jun Zhao, and Gerard de Melo. Sentiment-Aspect Extraction based on Restricted Boltzmann Machines. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 616–625, 2015.

[97] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. SemEval-2015 Task 12: Aspect Based Sentiment Analysis. p. 10, 2015.

[98] Pengfei Liu, Shafiq Joty, and Helen Meng. Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1433–1443, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[99] Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. Unsupervised Word and Dependency Path Embeddings for Aspect Term Extraction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, 2016.

[100] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 616–626, Austin, Texas, November 2016. Association for Computational Linguistics.

[101] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguist.*, Vol. 19, No. 2, pp. 313–330, June 1993.

[102] FTC. Guides Concerning the Use of Endorsements and Testimonials in Advertising, 2009.

[103] A.Z. Broder. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, pp. 21–29, Salerno, Italy, 1997. IEEE Comput. Soc.

[104] Guangyu Wu, Derek Greene, Barry Smyth, and Pádraig Cunningham. Distortion As a Validation Criterion in the Identification of Suspicious Reviews. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pp. 10–13, New York, NY, USA, 2010. ACM.

[105] James W Pennebaker, Cindy K Chung, Molly Ireland, Amy Gonzales, and Roger J Booth. The Development and Psychometric Properties of LIWC2007. p. 22, 2007.