修士論文

感情ラベル付きデータ不足問題解決に向けた他データセット使用の有用性

崔 井源
奈良先端科学技術大学院大学
先端科学技術研究科
情報理工学プログラム

主指導教員: 松本 裕治 教授
自然言語処理研究室（情報科学領域）

令和 02 年 2 月 22 日提出

本論文は奈良先端科学技術大学院大学先端科学技術研究科に
修士 (工学) 授与の要件として提出した修士論文である。

崔 井源

審査委員：

松本 裕治 教授　(主指導教員，情報科学領域)
中村 哲 教授　(副指導教員，情報科学領域)
新保 仁 准教授　(副指導教員，情報科学領域)
進藤 裕之 助教　(副指導教員，情報科学領域)

感情ラベル付きデータ不足問題解決に向けた他データセット使用の有用性*

崔 井源

内容梗概

レビューテキストの極性やその根拠を理解することは，製品に対するカスタマーの意見を分析するため欠かせない．感情分析のための代表的データセットとしてAmazon データセットや SemEval データセットがある．しかし，Amazon データセットの場合，大量だがスコア以外のラベルは付けられておらず，正しく対象語や根拠表現などの感情表現を探し出せたのか評価できない．一方，SemEval データセットは文毎に極性や根拠などのラベルがつけられており，感情表現抽出を学習するには便利だが，量が少ないため実際 SemEval データセットのみを活用するには限界がある．データ量やラベル有無の違いはあるが，感情分析のため作られた異なるデータセットを共に学習に使用することで，お互いに必要な情報の埋め合わせが可能になるかもしれない．以上の仮説を確認するため，文毎に極性と根拠となるカテゴリラベルが付与されている SemEval データセットを用いてテキスト分類タスクに取り組む際，近いドメインに属しているがレビュー毎スコアラベルだけが付与されている Amazon データセットを使用し，僅かだが性能の向上を確認することができた．また，本研究の目的は state-of-the-art を記録するのではなく，感情分析においてデータ不足問題解決のための手法として，大量の他データセットを共に学習に使用することの有用性を確認することである．

キーワード

マルチテスク学習， テキスト分類， 感情分析， ニューラルネットワーク，データ不足問題

*奈良先端科学技術大学院大学 先端科学技術研究科 修士論文, 令和 02 年 2 月 22 日.

Usefulness of the use of the other dataset to alleviate sentiment labeled data sparsity*

Jungwon Choi

Abstract

Understanding the polarity and aspect of the review text is crucial task for analyzing the consumer's point of view regarding a product. Amazon dataset and SemEval dataset are well used in sentiment analysis tasks. Amazon has large amount of review data but there are no labels but score label, so we can't evaluate whether extracted sentiment representations are proper results. On the other hand, SemEval's review data is helpful when we evaluate extracted sentiment representations, but lacks practicality because amout of data is too small to utilize in the industry. If we could supplement these two different sets of data with each other, maybe we can solve data sparsity. We tested the above hypothesis by studying the degree of performance improvements in utilizing Amazon's large amount of unlabeled review data while classifying text through SemEval's small amount of review data in a similar domain, which includes sentences that have been labeled with polarity and aspect categories. The goal of this study is to ensure usefulness of the use of the other dataset to alleviate sentiment labeled data sparsity, not to outperform the current state-of-the-art.

Keywords:

multi-task learning, text classification, sentiment analysis, neural network, data sparsity

*Master's Thesis, Graduate School of Science and Technology, Nara Institute of Science and Technology, February 22, 2020.

目 次

図 目 次

表 目 次

1. はじめに

1.1 研究背景

近年，インターネットの普及に伴い， 消費者のオンライン商品に対する個人的意見や購入傾向などをデータ化し分析することが可能になった．中でも， レビューデータの感情分析は，消費者と販売者のウィンウィン関係が構築できるという観点から，非常に大事である．消費者のニーズをもとにした販売戦略変更は，消費者の満足度向上や販売者の収益創出といった好循環をもたらすからである．感情分析のタスクにおいて， 極性・対象・根拠を見つけることは問題解決の鍵となっている． 感情分析タスクに最もよく使用される SemEval データセットは，このような感情情報（極性・対象・根拠） を文毎にアノテーションしており，本論文ではそれらを感情ラベルと称す．

The **display** *is* *vibrant* *and* *clear*.

The **keyboard** *is* *well* *designed and* *pleasant* *to type on.*

例えば，上の文は “POSITIVE” という極性ラベルと共に，それぞれ “DISPLAY/QUALITY”， “KEYBOARD/USABILITY” のように対象と根拠の組み合わせをラベルとして持っている．極性は “vibrant”， “clear”， “well”， “pleasant” などの根拠表現から，対象は “display”， “keyboard” から分かる．つまり，上の文を作成した消費者は，鮮やかで綺麗な映像を流す画面（DISPLAY）の質（QUALITY）に満足して（POSITIVE）おり，キーボード（KEYBOARD）が良く設計されキータッチが快適であるその使用性（USABILITY）に満足して（POSITIVE）いることがわかる．

しかし， 消費者が生成する生のレビューデータの場合，新しいレビューの作成につれデータも増えるので，その量は膨大と言えるが，製品に対するスコアラベル以外，以上のような感情情報はアノテーションされておらず， 企業側が最初からアノテーションするには， 多大なコストと時間が必要となる．

従来のニューラルネットワークより多くのデータが高速に処理できる深層学習の発展に伴い，ネット上のあらゆるデータを学習資源として用いることが可能に

なった．ただし，少量のデータで学習を行う場合，過学習の恐れがあるため，大量のデータで学習を行う必要がある．だが，ニューラルネットワークのモデルを学習させるために作られた SemEval のような感情ラベル付きデータセットの場合，データ量が少量の場合が多く，実際の産業の場で活用するのは難しい．

まとめると，以上で述べた 2 種類のデータセットにはそれぞれ長所と短所が存在する．感情情報が豊かだがデータ量が少ないデータセットは，正解として抽出した感情表現（e.g. 対象語・根拠表現など）の評価はしやすいが，過学習の恐れがあり，実際のビジネスに活用するのは困難である．一方，スコア情報のみであるがデータ量は多いデータセットは，正しく対象語や根拠表現などが探索できたのか判断しがたい．だが，深層学習の手法を用いて学習を行うには少量のデータセットより有用である．したがって，これらの特徴を生かし，相互補完すべきである．

1.2 研究目的

本研究では，大量のデータセットに新たに感情情報をアノテーションするなどの手法は一切用いない．その代わり，上で言及した 2 種類のデータセットのそれぞれの特徴を活かし，相互補完しながら学習する際， 少量のデータセットのみを使用する場合より過学習を防ぎ，ニューラルネットワークモデルの精度に肯定的影響を及ぼすか確認することを目的とする．お互いの長所を十分活用するため，マルチタスク学習 (Multi Task Learning: MTL) を使用し，同時学習を行う．また，データの影響力を確認することが目的なので，モデルは自然言語処理の分野において，最も普遍的に使用されるニューラルネットワークをシンプルに構築したものを用いる．

1.3 本論文の構成

本論文の構成は以下のとおりである．2 章では，感情テキスト分類タスク・マルチタスク学習・データ不足問題を解決するための手法に関して関連研究を説明

する．3 章では，提案手法の説明をした後，4 章で実験の流れとその結果を示す．最後に 5 章では，本研究の結論を述べ，今後の課題について説明する．

2. 関連研究

2.1 テキスト分類

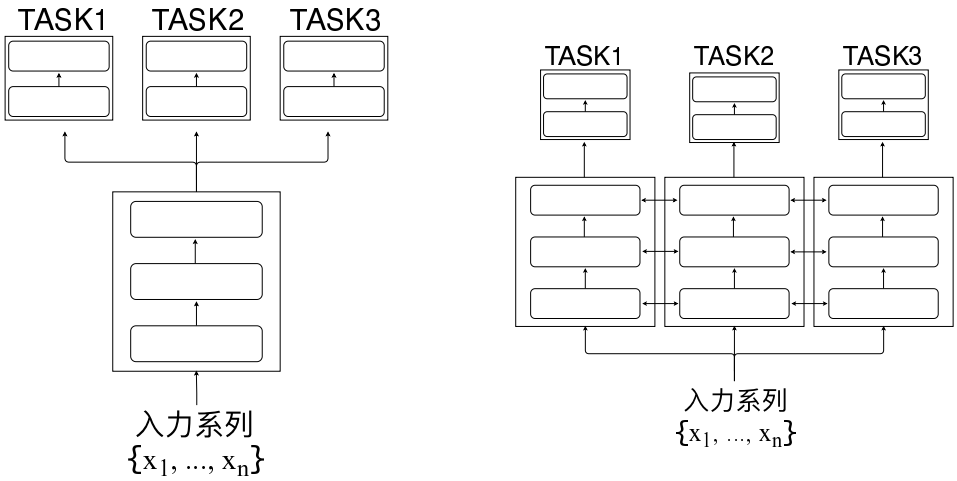
テキスト分類は，自然言語処理の研究において最も基礎的なタスクである．文に与えられた単数もしくは複数のラベルを当てるよう学習させることにより，文の内容を計算機に判断できるようにすることができる．テキスト分類タスクは，文の構造化・スパムメールのフィルタリング・フェイクニューズの分別・感情分類などのタスクに拡張できる．テキストを正確に分類するため，テキスト表現が重要な手がかりとして使われる．従来は，bag-of-words や n-grams などといった人手で作成したベクトル表現が用いられたが，近年，深層学習を用いた手法が優れた結果を出すことにより，多く使われるようになった．例えば，自然言語処理分野で最もよく使われる LSTM (Long Short Term Memory) [9]・GRU (Gated Recurrent Units) [4] に代表される再帰型ニューラルネットワーク (Recurrent Neural Network: RNN) [6] や畳み込みニューラルネットワーク (Convolutional Neural Network: CNN) [13]，注意機構 (Attention mechanism) [29] などが挙げられる． RNN や CNN は，系列を入力値として受け取り文を処理するため，文脈を考慮した特徴量抽出が可能である．RNN の場合，可変長のテキストを一連の順序を持つパターンとして認識できるので，計算機が前後の文脈から単語の意味を判断し文を理解できるようになる．一方，CNN の場合，位置普遍性からニューラルネットワークモデルの中で特徴抽出機として使用される [11, 31, 33]．最後に，注意機構は，RNN の最初の方に出現した情報が，系列の最後の情報を処理する際には薄れる問題を，注意すべき表現に重みを集中させることで解決する．本研究では，以上のアーキテクチャーの中から，自然言語処理の研究で多く使用される CNN と LSTM ＋注意機構の組み合わせの 2 つを用いて，同時学習のための新たなモデルを構築しテキスト分類に取り組む．

2.2 単語埋め込み

テキストデータを分散表現に埋め込むのは自然言語処理分野において必須になってきている．分散表現は，単語や文を固定長のベクトルにしたものである．単語埋め込みのため最もよく使用されるモデルとして，word2vec[22] や GloVe[25] などがある．これらは，単語の意味は周りの共起表現により決まるという分布仮説に基づいた教師無し学習の手法である．しかし，これらのモデルでは，単語の意味を決める際文全体の意味は考慮されておらず，多義語などを処理し文脈を完璧に読み取ることが困難である．近年，以上の問題を解決するため開発されたモデルとして ELMo[26] や BERT[5] などがある．ELMo は，従来のモデルとは違い，2 層の双方向 LSTM で構築した言語モデルを訓練し用いている．これにより，入力文の両側から文脈を考慮することができる．ELMo は発表当時，質問応答・感情分析・固有表現抽出などの 6 つのタスクで評価し，state-of-the-art を記録した．BERT は，ELMo などの手法で使用される言語モデルは単方向にしか情報処理ができないことを問題として指摘し，マスクされた言語モデル (Masked Language Model: MLM) を用いて事前学習を行い，ELMo を超えた新たな state-of-the-art を記録した．しかし，BERT は多くの GPU 資源を必要とするなど，実装環境に制限があるため，本研究では，予め訓練された ELMo を用いて単語埋め込みを行う．

2.3 マルチタスク学習

情報遷移を用いた手法は過学習を防ぐため用いられ，特に自然言語処理の分野では，解決したいタスクのデータが十分ではない際，有用であることが知られている．情報を遷移させ学習する手法として，遷移学習法 (Transfer Learning: TL) とマルチタスク学習法 (Multi-Task Learning: MTL) があり，共有する部分と，タスク特定部分に分かれているのが特徴である．TL は，解決したいタスク T_T の精度向上のため T_S を訓練させ，その情報を遷移する．テキストの各単語は，イメージのピクセルとは違い，意味情報を持っているため，どの情報をどれぐらい遷移するのかにより，精度向上に多大な影響が及ぶ．



(a) ハードパラメーター共有　　(b) ソフトパラメーター共有

図 2.1: マルチタスク学習の 2 つの手法

一方，MTL は TL とは違い，タスク $\mathcal{T}_1, \cdots, \mathcal{T}_n$ の同時学習を進めながら，タスク $\mathcal{T}_1, \cdots, \mathcal{T}_n$ の精度向上を図る．関連する複数のタスクの同時学習を行う際，共通する隠れ状態ベクトルが学習でき，タスクの予測精度の向上やモデルの汎化が期待できる．MTL には 2 種類の手法があり，まず，ハードパラメータ共有モデル（図 2.1（a））は，複数のタスクを同時に学習し，多くのパラメーターを共有する．これにより，過学習の可能性が低くなる [1]．一方，ソフトパラメータ共有モデル（図 2.1（b））は，共有するパラメーターはあるけれど，タスク毎に独立したモデルとパラメーターも存在する．この場合，お互いのパラメーターを揃えるため正規化する必要がある．感情分析の場合，MTL を利用することで，極性表現や対象語・根拠表現となる表現を同時に分類・抽出できるのが利点である．このような特徴から，近年，自然言語処理分野でも多く利用されている [8, 17, 18, 20]．He ら [8] は，トークン単位の対象語抽出と対象語の極性分類を行う際，MTL を利用した．訓練時は，ドキュメント単位の分類タスクも同時に学習し，評価時にはこの情報を遷移しトークン単位のタスクを解いた．特に，同時学習を行うと同時に，対象語抽出タスクの予測情報を極性分類タスクの入力系列に渡し結合したり，共有ネットワークから得られた特徴量に，前回 epoch で各タスクから得られた出

力分布や特徴量を結合させ，次回 epoch の共有入力系列として使用するといったメッセージパッシング (Message Passing) を活用した手法を提案した．Liu ら [17] もマルチタスク学習法で同時学習を行いながら，グラフニューラルネットワークモデルの各ノードがお互いにメッセージパッシングするようにした手法を提案した．Liu ら [18] は，レビューテキストを分類するタスクに，敵対的ネットワーク (Adversarial Network) を利用し，タスク関係なく出現する表現とタスク特定表現を分離して同時学習を行った．例えば，“infantile”という単語が幼児用品のレビューでは中立的単語たが，映画レビューでは否定的であることを別々で学習しながらも，感情判断に必要なキーワードということを同時に学習することができた．以上から，関連のあるタスクを同時に学習する際，お互いを補完できる情報を遷移・伝達することが学習に有用であることがわかる．本研究では，性質が異なるデータ間の同時学習を通じた情報遷移と，極性ラベル分類タスクの予測情報と対象・根拠ラベル分類タスクの予測情報との情報伝達を試みる．

2.4 データ不足への対応

1.1 節でも述べたとおり，研究のための感情ラベル付きデータセットは存在するが，実用システムに活用できるほどの量ではない．近年，自然言語処理分野の中でも特にテキスト分類タスクに関して，データ不足問題を解決するための手法は多く提案されている [3, 8, 19, 34, 35, 36]．Chen ら [3] と Liu ら [19] は，ドメイン分類を行う際，ドメインラベルが不足している問題を解決するための手法を提案した．Chen ら [3] は多項敵対的ネットワーク (Multinomial Adversarial Network: MAN) を利用し，Liu ら [19] は MTL を利用し，異なる複数のドメインラベル付きデータセットから共通表現を抽出し，未知のドメインを分類するタスクに取り組んだ．Yong ら [34] は，トピック分類タスクにおいて，異なるトピックラベル付きデータセットを使用し，2 つの間の関係や共通した特徴を学習することで，未知のトピックを推論できる手法を提案した．そして，Zhu ら [36] は，感動分類タスクにおいて，分類体系が異なる 2 つのコーパスを用いてコーパス間の関係性を捕らえる手法を提案した．しかし，これらの手法は，異なるドメイン・クラスに属している形の似た少量のデータセットを複数個使用しており，実際の産業の場で

活用できる手法なのかは確認できていない．また，実世界で得られる大量のデータセットの有用性に関しても確認していない．一方，He ら [8] は，少量のデータセットの限界を解消するため，同じドメインを持つ大量のデータセットを使用した MTL モデルを提案した．訓練時は，大量のドキュメント単位のデータを用いて少量のデータセットを使用した他分類タスクと共に学習を行った．対象語抽出情報を対象語の極性分類タスクのネットワークに遷移させることで，少量のデータセットの限界を克服することができた．だが，実際メッセージパッシングや MTL に関しての有効性は確認できたものの，大量のデータセットがどのように少量のデータセットの学習に影響したかは確認されてないままである．以上の理由から，本研究では，感情ラベル（極性・対象・根拠）がアノテーションされている少量のデータセットと，スコアラベルのみを持っている大量のデータセットを用いて，データ量・ドメイン関連性・文長[1]の違いなどが与える影響力を確認すると同時に，データ不足問題の解決のための有用性をも確認する．

[1] 1 文に含まれている単語数

3. 提案手法

3.1 データ活用

表 3.1: 感情情報がアノテーションされているデータの例

カテゴリラベル 対象#根拠	テキスト	極性ラベル
LAPTOP#GENERAL	Well worth the extra cost.	POSITIVE
DISPLAY#GENERAL	The Retina display is fantastic.	POSITIVE
OS#PERFORMANCE	Windows runs very poorly.	NEGATIVE

自然言語処理の研究において，感情分析の技術は多くの発展を成し遂げてきた．しかし，多くの手法は用いるデータ量に限界がある．研究のため作られた感情ラベル付きデータセットを用いて行われている．さらに，深層学習でそれらのデータを分析する提案手法の場合，実際の産業に適用するには未だ限界が存在している．例えば，企業が分析したい消費者のスコア付きレビューデータは無限に生成されている反面，極性・対象・根拠などの感情情報がアノテーションされているレビューデータは少なく，新たにアノテーションするにも時間やコストがかかる．また，ニューラルネットワークや深層学習を利用したくても，感情情報が欠如しているため，訓練が困難である．本研究では，これらの問題を解決するため，少量の感情ラベル付きデータセットと共に大量のスコアラベル付きデータセットを訓練に使用し，その有用性を確認する．

本研究で使用するデータセットは，自然言語処理分野で感情分析の研究に多く使用される SemEval2015[28]・SemEval2016[27] の Laptop レビューデータセットと，膨大なデータ量の Amazon[21] の Electronics レビューデータセットである．SemEval のレビューデータは，レビューを文単位に分け，文毎に感情ラベルがアノテーションされている（表 3.1）．一方，Amazon のレビューデータは，レビュー作成時に付けるスコア（1~5）のみがラベルとして付けられている．表 3.2 から

表 3.2: データ数比較

データセット	データ数	
	訓練データ	評価データ
SemEval Laptop15	1,739	761
SemEval Laptop16	2,500	808
Amazon Electronics	7,824,482	

データ量の差を確認できる．本研究では，大量のデータセットのデータ量・ドメイン・文長などが，少量のデータセットの対象・根拠分類タスクと極性分類タスクにそれぞれどのような影響を及ぼすか確認する．

3.2 マルチタスク学習: 共有ネットワーク

本研究では，ハードパラメータ共有モデル（図 2.1（a））を利用し MTL を行う．共有モデルを利用し各データセットのデータを同時に処理することから，感情分析に必要な共通した特徴量が得られる．

同時学習するタスクは，SemEval の対象・根拠ラベルの分類と極性ラベルの分類 3 つのタスクに，Amazon レビューのスコアを分類する 1 つのタスクを加えた 4 つである．SemEval のデータは表 3.1 のように感情ラベルがアノテーションされており，根拠ラベルは対象ラベルの特性であるため，親と子の関係にあると言える．対象（親）の分類タスクを T^{OP}，根拠（子）の分類タスクを T^{OC} と記し実験に取り組む．SemEval の文を `POSITIVE`・`NEUTRAL`・`NEGATIVE` 3 つの極性に分けるタスクを T^{SP} とし，Amazon のレビューをスコアに沿って `SUPER NEGATIVE`（1 点）から `SUPER POSITIVE`（5 点）まで 5 つに分けるタスクを T^{DS} とする．

入力される SemEval の文を $\mathbf{x}^S = \{x_1^S, \ldots, x_n^S\}$，Amazon のレビュー[2]を $\mathbf{x}^D = \{x_1^D, \ldots, x_m^D\}$ とする．ここで，x_i は i 番目の単語を指す．共有モデルでは，これら

[2]レビューは複数の文を含むドキュメントではなく，1 個の長い文を持つセンテンスとして扱う．

の入力系列を単語埋め込みにより分散表現として表現し，LSTM・CNN 各モデルを利用し特徴量を抽出する．得られた分散表現は，それぞれ，$\mathbf{E}^S = \{\mathbf{x'}^S_1, \ldots, \mathbf{x'}^S_n\}$ と $\mathbf{E}^D = \{\mathbf{x'}^D_1, \ldots, \mathbf{x'}^D_m\}$ で表現し，単語埋め込みテーブル $\mathbf{E}_{\mathcal{X}} \in \mathbb{R}^{d \times |V|}$ とする．LSTM や CNN から得られた最終特徴量は，$\mathbf{Z}^O = \{\mathbf{z}^O_1, \ldots, \mathbf{z}^O_i\}$ とし，ここで O は全体のタスクを表す文字として使用する．

本研究で用いる LSTM（図 3.1）・CNN（図 3.2）モデルは，実験のため新しく構築したものであり，共通特徴量抽出機として使用される LSTM モデルに関しては 3.2.1 節で，CNN モデルに関しては 3.2.2 節で，タスク毎の共通特徴量処理に関しては 3.3 節で詳しく述べる．

3.2.1 LSTM モデル

LSTM ネットワークは，RNN の一種であり，RNN の長期的依存性を忘れる問題 [24] を解決するために提案された手法である．つまり，単語間距離が遠くてもその依存関係を探すことができる．本研究では，双方向 LSTM (Bidirectional LSTM: bi-LSTM) を用いる（図 3.1）．bi-LSTM は，入力系列を前後から処理することで，文脈をより細かに捉えることができる．

各タイムステップ t で，順方向 LSTM は，隠れ状態 $\overrightarrow{\mathbf{h}}^{(l)}_t$ のみではなく，$\overrightarrow{\mathbf{c}}^{(l)}_t$ の中に記憶を保存している．タイムステップ t で行われるアップデートを以下に示す：

$$\overrightarrow{\mathbf{i}}^{(l)}_t = \sigma\left(\overrightarrow{\mathbf{W}}_i \mathbf{x}'_t + \overrightarrow{\mathbf{U}}_i \overrightarrow{\mathbf{h}}^{(l-1)}_{t-1} + \overrightarrow{b}_i\right) \tag{1}$$

$$\overrightarrow{\mathbf{f}}^{(l)}_t = \sigma\left(\overrightarrow{\mathbf{W}}_f \mathbf{x}'_t + \overrightarrow{\mathbf{U}}_f \overrightarrow{\mathbf{h}}^{(l-1)}_{t-1} + \overrightarrow{b}_f\right) \tag{2}$$

$$\overrightarrow{\mathbf{o}}^{(l)}_t = \sigma\left(\overrightarrow{\mathbf{W}}_o \mathbf{x}'_t + \overrightarrow{\mathbf{U}}_o \overrightarrow{\mathbf{h}}^{(l-1)}_{t-1} + \overrightarrow{b}_o\right) \tag{3}$$

$$\overrightarrow{\mathbf{u}}^{(l)}_t = \tanh\left(\overrightarrow{\mathbf{W}}_u \mathbf{x}'_t + \overrightarrow{\mathbf{U}}_u \overrightarrow{\mathbf{h}}^{(l-1)}_{t-1} + \overrightarrow{b}_u\right) \tag{4}$$

$$\overrightarrow{\mathbf{c}}^{(l)}_t = \overrightarrow{\mathbf{f}}^{(l)}_t \odot \overrightarrow{\mathbf{c}}^{(1)}_{t-1} + \overrightarrow{\mathbf{i}}^{(l)}_t \odot \overrightarrow{\mathbf{u}}^{(l)}_t \tag{5}$$

$$\overrightarrow{\mathbf{h}}^{(l)}_t = \overrightarrow{\mathbf{o}}^{(l)}_t \odot \tanh \overrightarrow{\mathbf{c}}^{(l)}_t \tag{6}$$

σ と tanh は，それぞれシグモイド関数と双曲線正接関数を表す．$\overrightarrow{\mathbf{f}}^{(l)}_t$ は，過去の情報を忘れるためのゲートであり，シグモイド関数の計算結果が 0 なら記憶を忘

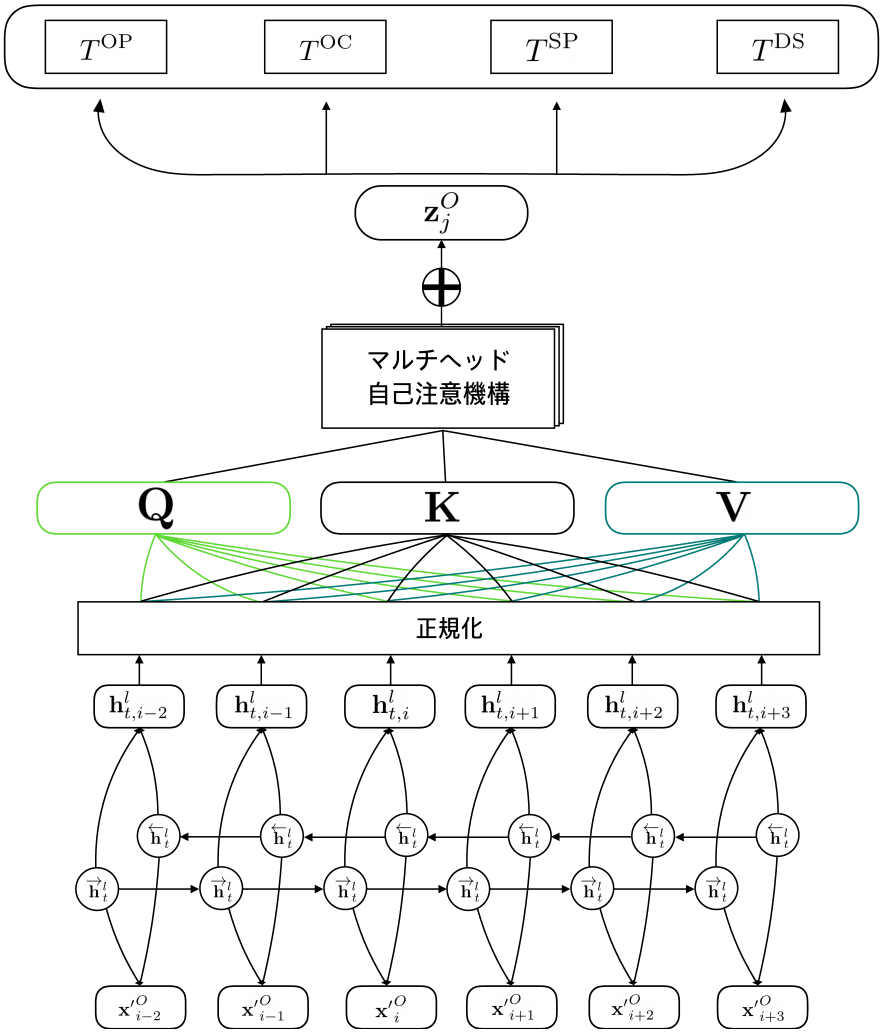


図 3.1: LSTM モデルの概要

れ，1 なら記憶を保存する．$\overrightarrow{\mathbf{i}}_t^{(l)}$ は，現在の記憶を保存するためのゲートである．重み $\overrightarrow{\mathbf{W}}_i, \overrightarrow{\mathbf{W}}_f, \overrightarrow{\mathbf{W}}_o, \overrightarrow{\mathbf{W}}_u \in \mathbb{R}^{\overrightarrow{d}_i \times \overrightarrow{d}_{l-1}}$ は入力重みであり，$\overrightarrow{\mathbf{U}}_i, \overrightarrow{\mathbf{U}}_f, \overrightarrow{\mathbf{U}}_o, \overrightarrow{\mathbf{U}}_u \in \mathbb{R}^{\overrightarrow{U}_i \times \overrightarrow{d}_l}$ は，再帰重みである．$\overrightarrow{d}_l$ は，順方向の隠れセルの数を表す．逆方向 LSTM も順方向 LSTM と同じ仕組みであり，l 個の LSTM 層の分だけ情報が統合された特徴量 $\mathbf{h}_t = \overrightarrow{\mathbf{h}}_t^l + \overleftarrow{\mathbf{h}}_t^l \in \mathbb{R}^{\overrightarrow{d}_l + \overleftarrow{d}_l}$ が得られる．また，各層のニューロン間正規化 (Layer Normalization: LN) を行う（式（7））．

$$y = \frac{x - \mu_x}{\sqrt{\sigma_x + \epsilon}} \times \gamma + \beta \tag{7}$$

より正確に感情を分類するため，関連のある表現のみに集中すべきである．例えば，表 3.1 の最初の文は，“worth”によりノートパソコンに関して肯定的な意見を書いていることがわかる．同じく 2 番目の文は “fantastic”から，3 番目の分は “poorly”から極性が判断できる．また，2 番目の文はその対象が “Retina display”であり，3 番目の文の極性が指す対象は “Windows”であることがわかる．しかし，bi-LSTM ネットワークから得られた $\mathbf{h}_t$ は，系列全てに関しての情報を持っているため，さらにキーワードだけに注目し学習を進める必要がある．近年，以上の問題を解決するため，注意機構を LSTM ネットワークと共に利用し，高い精度を記録していることが証明されている [7, 16, 30, 32]．

本研究ではマルチヘッド自己注意機構 (Multi-Head Self Attention: MHA) を用いる．

$$\text{Attention}\left(\mathbf{Q}, \mathbf{K}, \mathbf{V}\right) = \text{SoftMax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \tag{8}$$

式（8）で $\mathbf{Q}$ はクエリを，$\mathbf{K}$ はキーを，$\mathbf{V}$ はバリューを表す．d_k は K の次元とする．自己注意機構では，前の層から得た特徴量を $\mathbf{Q}$，$\mathbf{K}$，$\mathbf{V}$ として扱う．つまり，自分自身をヒントとして注目すべき表現を探索する仕組みになっている．また，各ヘッドでは，入力系列からそれぞれ表見を計算する．式（9）と式（10）の $\mathbf{W}_i^Q$，$\mathbf{W}_i^K$，$\mathbf{W}_i^V$，$\mathbf{W}^A$ はパラメーターの重み行列である．

$$\mathbf{H}_i = \text{Attention}\left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right) \tag{9}$$

1 つの文には必ずしも 1 つの対象と 1 つの根拠があると断言できるわけではない．1.1 節で述べた様に，複数の注目すべき表現が存在する場合もある．マルチヘッド注意機構は，N_h 個のヘッドの数だけ入力系列を計算し，その情報を統合する．つまり，複数の人が各々文を読み，意見を合わせ結論を出すのと同様である．1 つのヘッドが行う計算は，式（8）と式（9）の様であり，式（10）で N_h 個のヘッドから計算された特徴量を統合する．式（8）の SoftMax() は，必ず計算された全ての重みの和を 1 にする．そして注意機構が注目した LSTM の隠れ状態 $\mathbf{h}_t$ の重みの和を計算し，最終的に表現 $\mathbf{z}_j^O$ を得る．

$$\text{MultiHead}\left(\mathbf{Q}, \mathbf{K}, \mathbf{V}\right) = \left[\mathbf{H}_1, \ldots, \mathbf{H}_i\right] \mathbf{W}^A \tag{10}$$

3.2.2 CNN モデル

オッカムの剃刀 (Occam's razor principle) [2] に従い，シンプルな CNN モデルを組むため，LeCun ら [12] の CNN アーキテクチャーを用いる．畳み込み演算から ReLU 演算までを一つの畳み込み層とする．LeCun ら [12] の CNN アーキテクチャーは，系列ラベリングタスクのため提案された手法であり，LSTM の各セルが順序に依存関係を持つという問題が解決できた．各セルが順序に依存関係を持つことで，逆伝播の際，全ての情報を考慮しないといけないことから，モデルの訓練などが遅くなる恐れが生じるのである．一方，CNN は畳み込み演算により入力系列の各要素を独立的に計算する．

本研究では CNN のモデルでマックスプーリングは使用しない．代表すると思われる局所情報を抽出しまとめるため，畳み込み層が重なればなるほど情報を失うからである．3.2.1 節でも述べた様に，テキスト中の感情情報は一個だけだとは限らない．また，対象・根拠は極性情報と関わりを持ち，逆もそうであるため，プーリングせず情報を処理すべきである．

提案モデル（図 3.2）は，l 個の畳み込み層を持ち，各層は入力系列に対して畳み込み演算・バッチ正規化 (Batch Normalization)（式（7））・ReLU 演算（式（11））

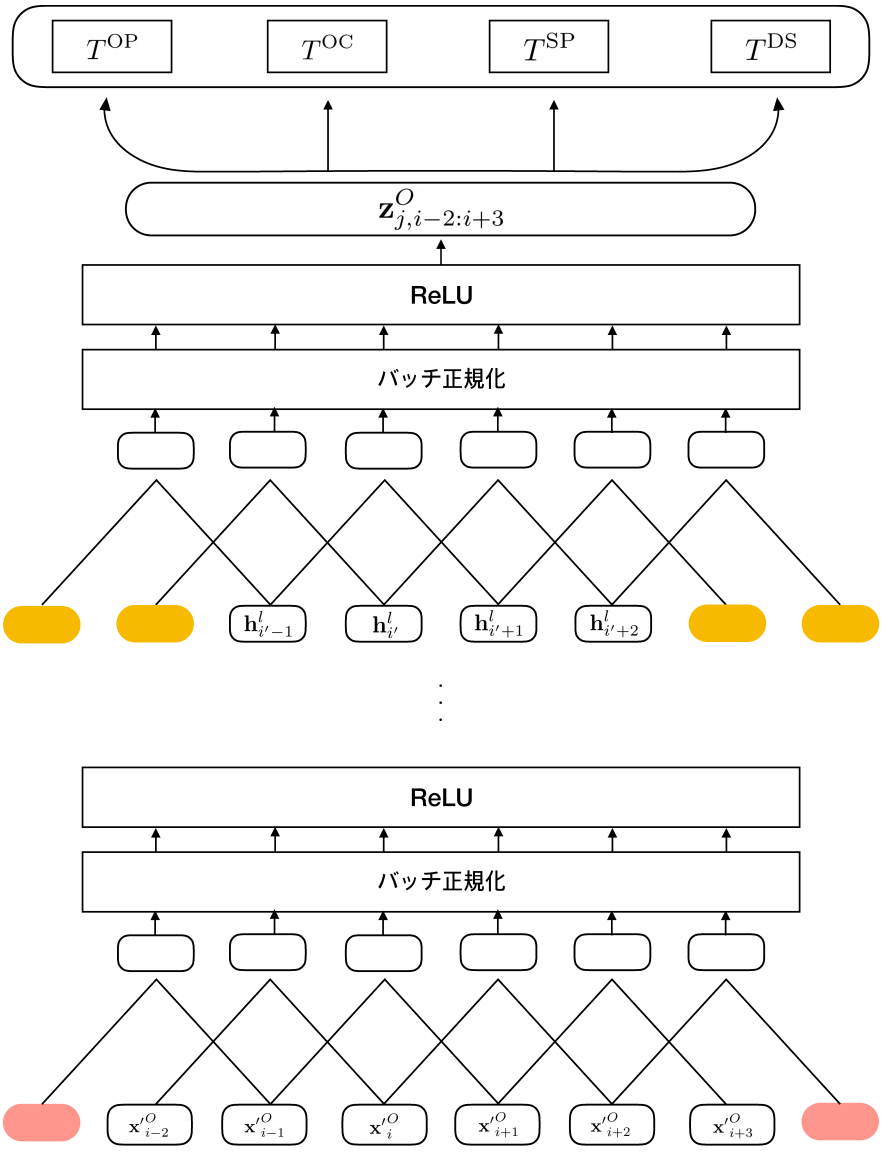


図 3.2: CNN モデルの概要

を行う．$\mathbf{W}_i^{C(l)}$ は，l 番目畳み込み層のパラメーターの重み行列である．

$$\mathbf{z}_i^O = \max\left(0, \mathbf{W}_i^{C(l)}\mathbf{x}_i'^O + b_i^{C(l)}\right) \tag{11}$$

バッチ正規化は，モデルの訓練速度を向上させると共に，モデルを安定化できることが知られている [10]．深層学習は，前の層の重みの演算をも更新しながら学習を進めていくが，バッチ正規化により隠れ層の分布をガウス分布に変換させ，学習内容に大きく差が出ることを防ぐ．

各畳み込み層は異なるサイズのフィルターを持っており，そのサイズは $k = 2c-1$ である[3]．しかし，フィルターにより系列情報が文の長さほど保たない場合がある．それを防ぐため，パディング数を $p = k/2$ 個設定する．CNN モデルから得られた特徴量 $\mathbf{z}_j^O$ は，個別ネットワークにてタスク別の計算を行う．ここで i は入力系列の中の単語の順番を意味し，j はバッチ中の文の順番を意味する．

3.3 マルチタスク学習: 個別ネットワーク

共有モデルから得られた $\mathbf{z}^O$ は，個別ネットワーク（図 3.3）に渡され，タスク毎に計算を行う．タスク毎の流れは全体的に一緒である．だが，SemEval データセットを用いるタスクの間では情報伝達が行われる．これは，3.3.1 節で述べる．各タスクはまず文 $\mathbf{z}_j^O$ から最も可能性のある i 番目の単語の特徴量のみを抽出（式 (12)）する（MP^O 層）:

$$\mathbf{z}'^O = \underset{i}{\operatorname{argmax}}\, \mathbf{z}_j^O \tag{12}$$

全結合 (Fully Connected: FC) 層では $\mathbf{z}'^O$ を入力系列として受け取り，アフィン写像を行った後，出力層でソフトマックスを用いて確率を計算する (式 (13)) ．式 (13) の $\mathbf{W}^F$ はパラメーターの重み行列である．

$$\hat{y}^O = \mathrm{SoftMax}\left(\mathbf{z}'^O\mathbf{W}^F + b^O\right) \tag{13}$$

[3]奇数個のフィルターを用いることで，計算後，長さが $n-k+1$ になったベクトルに対し，両側に同じ数のパディングが設定できる．これにより，入力系列と同じ長さの特徴量が得られる．

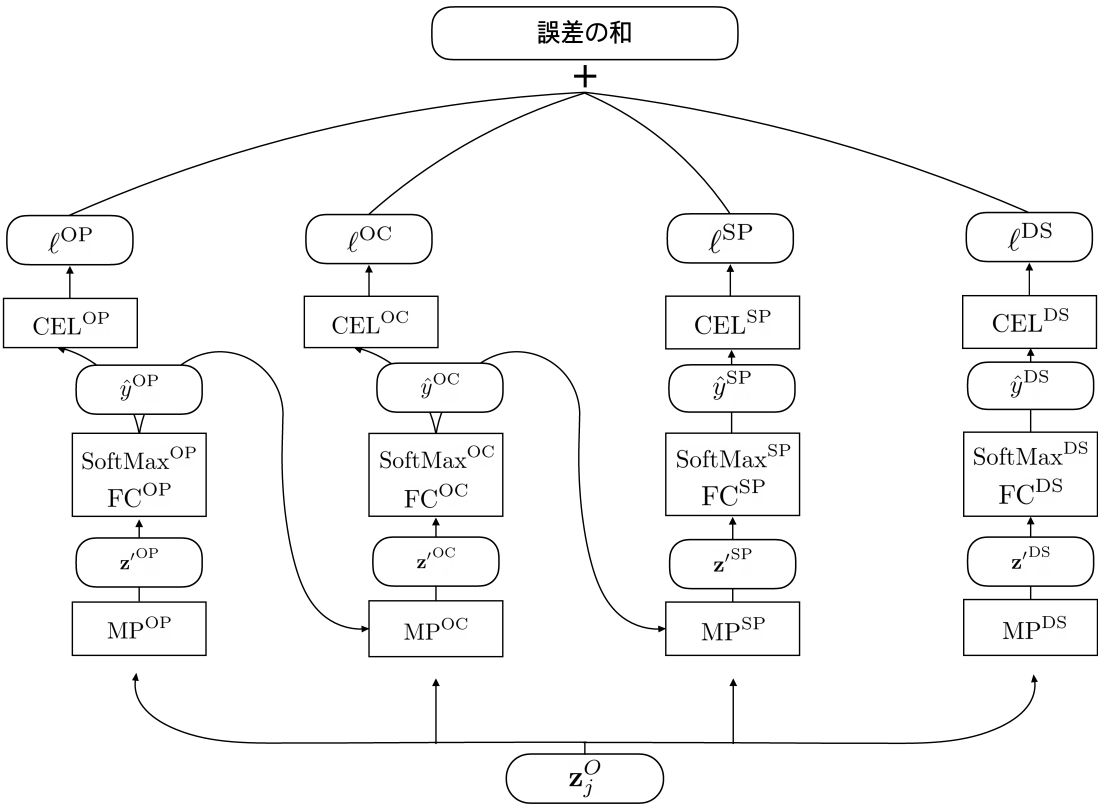


図 3.3: タスク別のネットワークの概要

誤差計算のため，損失関数として交差エントロピー (Cross Entropy Loss: CEL)（式（14））を計算する（CEL^O 層）．N_O はタスク O の訓練インスタンスの数を表す．T^{DS} は，評価時には考慮しない．

$$\ell = -\frac{1}{N_O}\left(\sum_{j=1}^{N_O} y^O \log \hat{y}^O\right) \tag{14}$$

各タスクのネットワークから得られた誤差は，式（15）[23] により総合誤差を計算する．ℓ_S は SemEval データセットの誤差関数を意味し，ℓ_D は Amazon データセットの誤差関数を意味する．

$$\ell = \lambda\ell_S + (l - \lambda)\ell_D \tag{15}$$

Amazon のレビューデータの量を SemEval の文単位データに比べ，N_d 倍多く設定し訓練を行うため，学習の均衡を保つためのハイパーパラメーター $\lambda \in (0, 1)$ を取り入れる．

3.3.1 情報伝達

3.2 節でも述べた通り，対象と根拠は親と子の関係を持つ．例えば，表 3.1 のカテゴリラベルを見ると，最初の文はノートパソコンの一般的な部分を褒め，2 番目の文もディスプレイの一般的部分を褒め，最後の文はオペレーティングシステムの性能に関して貶していることがわかる．これら以外にも SemEval のデータセットには 73 個のカテゴリの組み合わせがある．これらは全て親と子の関係にあり，親カテゴリの情報は子カテゴリを予測するのに役立つ可能性が高い．また，対象語と根拠表現は近い範囲内で共起することが知られている [14, 15]．

そこで本研究では，T^{OP} の出力層で計算された結果を T^{OC} へ伝達する．さらに，対象ラベルの確率分布を受け取り計算した根拠ラベルの予測結果を，T^{SP} へ伝達することで，極性ラベルの分類を行う際，対象語や根拠表現に重みが加算され，正しい極性ラベルが予測できるようになる．情報伝達された T^{OC}，T^{SP} の確

率は以下である：

$$\hat{\mathbf{y}}^{\mathrm{OC}} = Pr\left(\mathbf{z}_j^{OC} \mid \hat{\mathbf{y}}^{\mathrm{OP}}\right) \tag{16}$$

$$\hat{\mathbf{y}}^{\mathrm{SP}} = Pr\left(\mathbf{z}_j^{SP} \mid \hat{\mathbf{y}}^{\mathrm{OC}}\right) \tag{17}$$

情報伝達のため，T^{OP} の予測確率分布 $\hat{y}^{\mathrm{OC}}$ を MP^{O} 層に渡される前の T^{OC} の特徴量 $\mathbf{z}_j^{OC}$ と結合する．T^{OC} から T^{SP} への情報伝達も同様に行われる．

$$\mathbf{z}_j^{\mathrm{OC}} = \left[\mathbf{z}_j^{\mathrm{OC}}, \hat{\mathbf{y}}^{\mathrm{OP}}\right] \tag{18}$$

$$\mathbf{z}_j^{\mathrm{SP}} = \left[\mathbf{z}_j^{\mathrm{SP}}, \hat{\mathbf{y}}^{\mathrm{OC}}\right] \tag{19}$$

対象・根拠の情報が一方的に極性表現探索の手がかりになるとは限らないので，逆方向の情報伝達も試みる．ただし，子は親に属する関係であることから，親から子への情報伝達の向きは固定する．

$$\mathbf{z}_j^{\mathrm{OP}} = \left[\mathbf{z}_j^{\mathrm{OP}}, \hat{\mathbf{y}}^{\mathrm{SP}}\right] \tag{20}$$

$$\mathbf{z}_j^{\mathrm{OC}} = \left[\mathbf{z}_j^{\mathrm{OC}}, \hat{\mathbf{y}}^{\mathrm{OP}}\right] \tag{21}$$

4. 実験・結果

4.1 実験設定

本章では，3 章で述べた提案手法の効果を確認するため行った諸実験の結果を説明する．ベイスラインとなるのは L1516 のみを使用した際の結果であり，訓練時に AE・ALC を共に用いた際の実験結果を，データ量・ドメイン関連度・文長の 3 要素にフォーカスを当て比較を行う．

4.1.1 データセット

表 4.1: 実験に使用するデータ数

データセット	訓練データ数
SemEval Laptop15 + Laptop16（L1516）	3,261
Amazon Electronics（AE）	5,040
	10,500
Amazon Laptop + Computer（ALC）	5,040
	10,500

本研究では，表 4.1 に示したデータセットを使用する．表 4.1 の AE は Amazon データセットの Electronics カテゴリに属するデータ全ての中から無作為に抽出したデータである．ALC は，Electronics カテゴリのレビュー全体の中でさらに Laptop や Computer に関したレビューのみに絞り作ったデータセットであるが，AE から抽出したものではない．訓練の際，データ量の影響を確認するため，Amazon のレビューデータは L1516 の文単位データの $N_d \in \{1.5, 3\}$ 倍用いる[4]．SemEval データセットの場合，既に評価データが用意されているため，検証データのみ訓練デー

[4]ラベルの比率を合わせるため，正確に N_d 倍ではない場合がある．また，N_d は 2 つのデータ間バッチサイズの均衡を保てる数字を選択した．

タから無作為に抽出した．Amazon データセットの場合，評価データのため訓練データの 20%の量のデータを使用した．訓練データ・検証データ・評価データは重複しないよう Amazon データセットの全体から無作為に抽出した．

表 4.2: レビューデータの文長比較

データセット	最大長	平均長
L1516	74	13
AE	6,465	87

本研究では Amazon のレビューをドキュメントとして扱うのではなく，SemEval と同様に長い一つのセンテンスとして扱う．大量データの数の影響力だけではなく，文長の影響力も確認する必要がある．ここで文長とは，一文に含まれている単語数を意味する．2 つのデータセット間，文長の差を表 4.2 に示す．そして，文長が学習に影響を及ぼすか確認するため，本実験では Amazon のレビューの文長 $N_w \in \{90, 150\}$（表 4.3）を調整しながら実験を行う[5]．

表 4.3: 調整後の文長比較

データセット	最大長	平均長
L1516	74	13
AE / ALC	90	
	150	

4.1.2 ハイパーパラメータ

ハイパーパラメーターを決めるため，L1516 の検証データは訓練データの 10%を抽出し使用した．共有ネットワークには，予め訓練された ELMo を使用し単語

[5]90 は L1516 の最大文長に最も近い最低限の数字であり，150 は L1516 の最大文長の約 2 倍になる数字である．

埋め込みを行なった分散表現を渡す．埋め込み次元は 1,024 に設定した．LSTM モデルの場合，隠れ層の次元を 300，層の数を 1，ドロップアウトを 0.5 に設定した．MHA のヘッド数は 8，ヘッドの次元は 64 に設定した．CNN モデルの場合，3 層に構築しフィルターを $k \in \{1, 3, 5\}$，ドロップアウトを 0.2 に設定した．誤差の和を求めるための λ は 0.01 から 0.09 までを試し 0.02 に決定した．Optimizer は Adam を使用し，学習率は 0.0001 に設定した．

4.2 評価方法

評価時，T^{OP}，T^{OC}，T^{SP} ではそれぞれ $\hat{\mathbf{y}}^{OP}$ を基に対象ラベルを，$\hat{\mathbf{y}}^{OC}$ を基に根拠ラベルを，$\hat{\mathbf{y}}^{SP}$ を基に極性ラベルを予測した．本研究では，評価指標として micro-F1 を用いた．$\mathrm{F1}^{\mathrm{CAT}}$ は，SemEval2016 の Slot1 のように T^{OP} と T^{OC} を合わせて (対象#根拠) 精度を計算したものであり，$\mathrm{F1}^{\mathrm{SP}}$ は T^{SP} の精度を計算したものである．

4.3 実験結果

`L1516` のみを使用した際の実験結果（表 4.4）を評価基準とする．4.3.1 節では，$N_w = 150$ の場合の実験結果（表 4.5）を基に用いる大量データセッドのデータ量に焦点を当て，4.3.2 節では `AE` と `ALC` をそれぞれ用いた結果を比較し，4.3.3 節では $N_w = 90$ の場合の結果（表 4.6）を $N_w = 150$ の場合の結果と比較しながら分析を行う．4.3.1 節と 4.3.2 節では $N_w = 150$ に設定した結果を持って分析を行い，4.3.3 節ではこれらと $N_w = 90$ に設定した場合の結果とを比較・分析することで，精度向上にどの要素が最も影響を及ぼしたかを調べる．また，全ての実験結果において P 値が 0.05 以下になることを確認した．

本章の表や図に記している（SP $\rightarrow$ CAT）は T^{SP} から T^{OP} へ，さらに T^{OC} へと情報を伝達したことを意味する．（CAT $\rightarrow$ SP）は T^{OP} から T^{OC} へ情報伝達し得た確率分布を T^{SP} へと伝達したことを意味する．

表 4.4: L1516 のみを使用した際の実験結果

	L1516	
	$F1^{CAT}$	$F1^{SP}$
LSTM	69.06	74.47
CNN	70.84	73.84
LSTM (SP → CAT)	69.42	74.51
CNN (SP → CAT)	70.97	74.18
LSTM (CAT → SP)	68.87	74.67
CNN (CAT → SP)	70.62	74.00

表 4.5: 文長が 150 の場合の実験結果

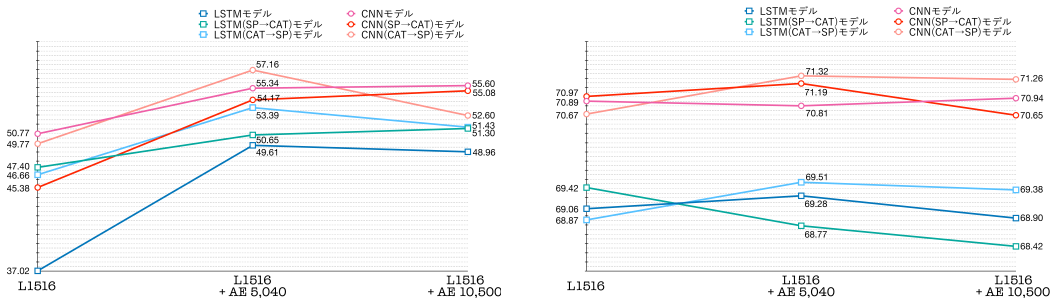
	L1516 +AE5,040		L1516 +ALC5,040		L1516 +AE10,500		L1516 +ALC10,500	
	$F1^{CAT}$	$F1^{SP}$	$F1^{CAT}$	$F1^{SP}$	$F1^{CAT}$	$F1^{SP}$	$F1^{CAT}$	$F1^{SP}$
LSTM	69.28	74.47	69.92	74.63	68.90	74.41	69.60	74.31
CNN	70.81	74.63	71.22	74.73	70.94	73.93	70.40	73.49
LSTM (SP → CAT)	68.77	74.63	69.02	74.72	68.42	74.35	69.22	75.19
CNN (SP → CAT)	71.19	74.95	71.45	74.89	70.65	74.09	70.87	74.36
LSTM (CAT → SP)	69.51	74.51	68.87	75.27	69.38	74.98	68.80	75.27
CNN (CAT → SP)	71.32	74.16	70.79	74.25	71.26	74.06	70.68	74.09

表 4.6: 文長が 90 の場合の実験結果

	L1516 +AE5,040		L1516 +ALC5,040		L1516 +AE10,500		L1516 +ALC10,500	
	$F1^{CAT}$	$F1^{SP}$	$F1^{CAT}$	$F1^{SP}$	$F1^{CAT}$	$F1^{SP}$	$F1^{CAT}$	$F1^{SP}$
LSTM	68.83	74.09	69.25	74.86	68.99	74.38	68.80	74.70
CNN	71.64	73.96	70.65	74.09	70.59	74.60	70.91	73.80
LSTM (SP → CAT)	69.31	75.08	69.90	74.63	68.02	74.60	68.99	74.76
CNN (SP → CAT)	70.59	74.03	70.46	73.87	69.82	74.06	70.20	74.12
LSTM (CAT → SP)	69.38	74.67	69.34	74.92	69.47	74.54	69.57	74.76
CNN (CAT → SP)	71.32	73.87	70.87	73.26	70.46	74.03	71.21	73.49

4.3.1 データ量の影響

本実験では，検証データを使用した場合と評価データを使用した場合とで精度の差が大きく異なっていたので，図 4.1 と図 4.2 に両方の精度変化の差を比較した．

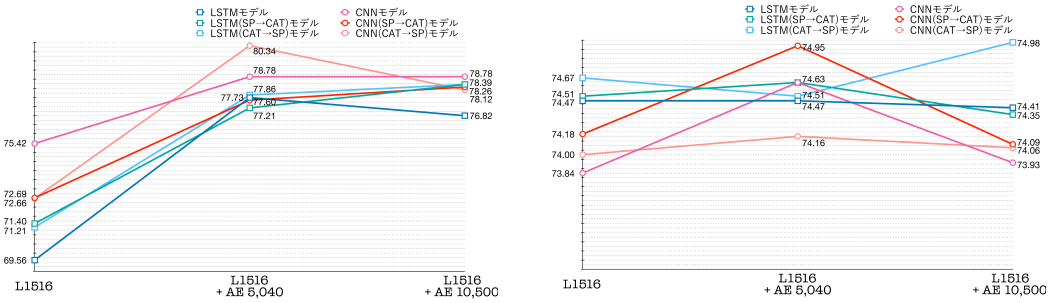


(a) 検証データ　　(b) 評価データ

図 4.1: データ量変化による対象・根拠分類機の精度変化

検証データを使用した場合（図 4.1（a））の $\mathrm{F1}^{\mathrm{CAT}}$ は，L1516 のみを使用した場合より AE を共に使用した場合，精度が 6.65%向上したことから，共有ネットワークで抽出された T^{DP} の情報が T^{OP} と T^{OC} に肯定的影響を及ぼしたことがわかる．LSTM（SP → CAT）・CNN（SP → CAT）モデルの場合，N_d が大きくなるにつれ精度も向上した．一方，LSTM（CAT → SP）・CNN（CAT → SP）モデルの場合，$N_d = 1.5$ の場合最高値を記録し，$N_d = 3$ の場合精度が少し下がった．これは，共有ネットワークで抽出された T^{DP} の特徴量が，対象や根拠を分類する際より極性分類の際に役立ったと解釈できる．

評価データを使用し $\mathrm{F1}^{\mathrm{CAT}}$ を測定した場合（図 4.1（b））， L1516 のみ使用したのに $\mathrm{F1}^{\mathrm{CAT}}$ が 20.20%向上し，$N_d = 1.5$ の場合と 0.35%しか差のない数値を記録した．だが，LSTM（SP → CAT）・CNN モデルを除いた 4 つのモデルは，AE を学習に用いた場合 $\mathrm{F1}^{\mathrm{CAT}}$ が向上しており，CNN（CAT → SP）モデルが最高値 71.32%を記録したことから，AE は T^{OP} と T^{OC} において肯定的影響を及ぼしていることがわかる．しかし，$N_d = 3$ の場合，CNN モデルの $\mathrm{F1}^{\mathrm{CAT}}$ が 0.13%向上したことを除いては，むしろ精度が下がってしまったため，必ずしも外部データの量が増加することに比例し精度も向上するとは言い難い．



(a) 検証データ　　(b) 評価データ

図 4.2: データ量変化による極性分類機の精度変化

検証データを使用した場合（図 4.2（a））, $F1^{CAT}$ と同様に L1516 のみを学習に使用した場合より $N_d = 1.5$ の時 $F1^{SP}$ が 4.92%向上し，80.34%を記録した．この結果から AE は T^{SP} に肯定的な影響を及ぼしたと考えられる．$N_d = 3$ の場合も L1516 のみを使用した場合より $F1^{SP}$ が 3.36%向上したため，学習に効果的だったとは言える．だが，$N_d = 1.5$ の時より精度が下がったため，データを増やしすぎるのは逆効果をもたらすことも推測できる．

一方，L1516 の評価データで $F1^{SP}$ を計算した場合（図 4.2（b））, LSTM･LSTM（CAT → SP）モデル以外のモデルにおいて，$N_d = 1.5$ の方が，$N_d = 3$ の時より $F1^{SP}$ が高く，検証データの場合と似た様相になっている．だが，$N_d = 1.5$ の場合，逆に $F1^{SP}$ が 0.16%下がり，$N_d = 3$ の場合 0.47%上がった．$F1^{CAT}$ とは異なり，L1516 のみを使用した場合の $F1^{SP}$ は 73〜75%にとどまった．しかし，AE を共に用いた場合 74.98%を記録し，L1516 のみ用いた場合より $F1^{SP}$ が 0.31%上がったため，微々たるが大量のデータとの学習は肯定的な影響を及ぼしていることを確認できた．

極性分類タスクにおいては，外部の大量データを約 1.5 倍以上用いることで精度向上に肯定的影響を及ぼし，特に LSTM（SP → CAT）･ CNN ･ CNN（SP → CAT）モデルが効果的ということが確認できた．LSTM（CAT → SP）･ CNN（CAT → SP）モデルがあまり差を見せなかったのは，T^{OP} ･ T^{OC} の予測分布は T^{SP} には役立たなかったためだと推測できる．一方，3 倍以上多くデータを用い

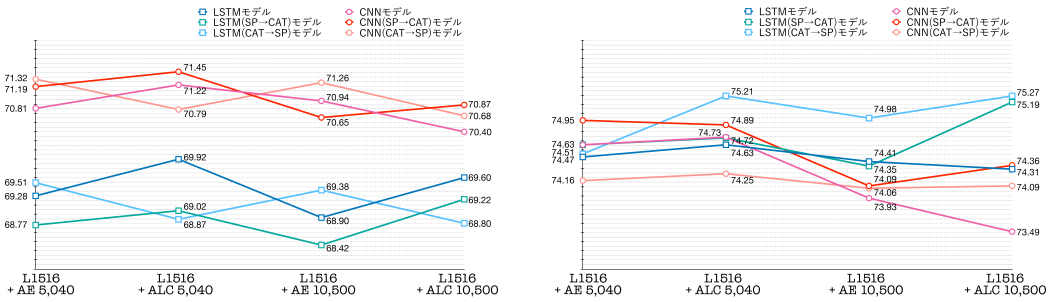
る際は，LSTM モデルが効果的であった．

4.3.2 ドメイン関連性の影響

AE の場合，家電製品全般のレビューを含んでいるため，L1516 と属するドメインは他ドメイン（e.g. レストラン，映画など）より関連度が高いかもしれないが，以下の例文のように異なる話も多数ある．以下の文は AE のレビューデータの一文である：

Overall, this is a good product to help keep the **glove box** *tidy.*

以上の文は “Car & Vehicle Electronics” カテゴリに属している．“glove box” を褒めているが，SemEval データセットのドメイン “Laptop” とは全く無関係であり，ノイズにしかならない可能性がある．従って，実験の際，カテゴリ名に “Laptop” と “Computer” という表現を含んでいるレビューデータだけを追加で抽出（ALC）し，精度への影響力を比較した（図 4.3）．



(a) T^{OP}・T^{OC}　　(b) T^{SP}

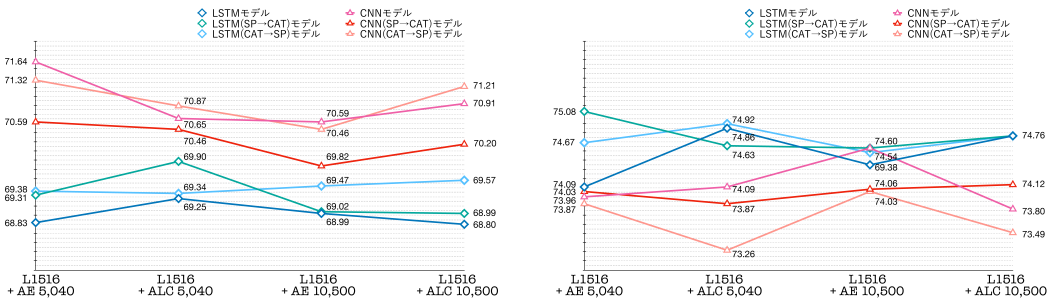
図 4.3: データ間ドメイン関連度変化による精度変化

まず，$\mathrm{F1}^{\mathrm{CAT}}$（図 4.3（a））は，LSTM（CAT → SP）・CNN（CAT → SP）モデルを除外した全てのモデルにおいて $N_d \in \{1.5, 3\}$ の場合，AE を使用した際より ALC を使用した際 $\mathrm{F1}^{\mathrm{CAT}}$ が向上した．最も高い精度を出したのは，$N_d = 1.5$ の場合の CNN（SP → CAT）モデルであり，最高値 71.45%を記録した．L1516 の

み使用した場合の最高値 70.97%より 0.58%上回った．(SP → CAT) 向き情報伝達を行った全てのモデルにおいて $\mathrm{F1}^{\mathrm{CAT}}$ が上がったことから，共有ネットワークで T^{DP} から得られた特徴量が T^{SP} 解決に役立ち，T^{OP} と T^{OC} へ肯定的影響を及ぼしたと考えられる．ただ，この場合もまた，各モデルにおいて $N_d = 1.5$ の場合最高値を記録しているため，外部のデータの量の増加に比例し精度が無条件に良くなるということではないことがわかった．

T^{SP} においては，精度変化が $\mathrm{F1}^{\mathrm{CAT}}$ とは異なる様相になっている（図 4.3 (b)）．T^{OP} と T^{OC} において CNN モデルは必ず LSTM モデルより優れていた反面，T^{SP} においては LSTM モデルが CNN モデルより高い精度を記録している．さて，LSTM (SP → CAT)・LSTM (CAT → SP) モデルは `AE` を使用した際より `ALC` を使用した場合 $\mathrm{F1}^{\mathrm{SP}}$ が向上し，N_d が上がるにつれ精度も上がっていった．特に，LSTM (SP → CAT) モデルの場合，75.27%を記録し，`L1516` のみ使用した場合より 0.60%上回った．$\mathrm{F1}^{CAT}$ と同様，共有ネットワークから得られた T^{DP} の特徴量が T^{SP} の予測に役立ち，全てのタスクへ肯定的影響を及ぼし，結果的に $\mathrm{F1}^{SP}$ が向上できたと推測できる．

4.3.3 文長の影響



(a) T^{OP}・T^{OC}　　(b) T^{SP}

図 4.4: 文長変化による精度変化

図 4.4 に，$N_w = 90$ に設定し行った実験の結果を N_d の変化と共に示す．まず，`AE` を $N_d = 1.5$ にした場合 $\mathrm{F1}^{\mathrm{CAT}}$（図 4.4 (a)）が 71.64%を記録し，`L1516` のみを

使用した場合より 0.67%上回り，$N_w = 150$ の場合よりは 0.19%上回った．LSTM・LSTM（SP $\rightarrow$ CAT）モデルは，ALC を $N_d = 1.5$ 倍使用した場合，AE を $N_d = 1.5$ 倍使用した場合より $\mathrm{F1}^{\mathrm{CAT}}$ が高くなり，より関連のあるドメインのみを使用することで精度向上に役立つことは確認できたものの，L1516 のみ用いた場合の最高値 70.97%には及ばなかった．また，LSTM（CAT $\rightarrow$ SP）モデルを除いた全てのモデルが Amazon データ量の増加と精度が反比例しており，外部データが多すぎることはかえって妨げになることがわかった．

$N_w = 90$ の場合 $\mathrm{F1}^{SP}$ は（図 4.4（b））， 同条件の $\mathrm{F1}^{\mathrm{CAT}}$ とは逆の様相を見せた．CNN を利用したモデルより LSTM を利用したモデルの方が優れた性能を見せている．$\mathrm{F1}^{CAT}$ と同様に，$\mathrm{F1}^{\mathrm{SP}}$ の最高値は $N_d = 1.5$ の時 ALC を使用した場合であり，75.08%を記録し L1516 のみ使用した場合より 0.41%上回った．しかし，$N_w = 150$ の時の 75.27%には及ばなかった．$N_d = 1.5$ の場合，（SP $\rightarrow$ CAT）向き情報伝達を行ったモデルに限りドメインを絞ることで精度が下がった原因として，N_w が小さくなり T^{SP} へ有用な情報が十分渡らなくなったのが，T^{OP} と T^{OC} へ否定的影響が及んだと推測できる．

5. 結論

本研究では，自然言語処理分野の感情分類タスクにおいて，感情ラベル付きデータの数量的限界を解決するための手法として多く使用されている，関連度の高い他データセット使用に関し，実際の影響力を確認すべく諸実験を行った．実験では，LSTM と CNN を利用しシンプルに構築したモデルを使用し，データ量・ドメイン関連性・文長などを変化させながら性能向上の可能性を確認した．

まず，データ量に関して述べると，他データセットの量を少量のデータセットの量の 1.5 倍にした場合，最も高い精度を記録した．それ以上他データセットのデータ量を増やす場合，むしろ精度は下がる傾向を見せた．この結果から，少量のデータセットのデータ量の約 1．5 倍の外部のデータを用いることが，データ不足問題を解消し，精度向上に繋がることがわかった．

次に，ドメインの関連性に関して，最大文長が 150 であった場合を述べると，T^{OP}・T^{OC} において，（CAT $\rightarrow$ SP）向き情報伝達を行ったモデル以外，用いる他データセットの量に関係なく ALC を用いる場合精度が向上した．一方，T^{SP} においては LSTM（SP $\rightarrow$ CAT）・LSTM（CAT $\rightarrow$ SP）モデル以外の全てのモデルの精度が下がった．ドメインを絞ることで，T^{OP}・T^{OC} 解決の鍵となる対象語や根拠表現が多く出現するようになり，返って極性分類の妨げになったと考えられる．そして最大文長が 90 であった場合，LSTM を利用したモデルを除いた全てのモデルで，どのタスクにおいても AE を用いた方が精度向上に役立つ様相になった．この場合，一文に含まれている単語数が減少し，そもそもどのタスクのための表現も出現しなくなったことが主な原因として挙げられる．したがって，ドメインを絞る際，豊富な情報が含まれていることが望ましいため，外部データの最大文長を少量のデータの文長より 2 倍以上に設定する必要がある．

実験から各タスクに有利なネットワークが分かれることも確認できた．評価データを用いて得た実験結果において，T^{OP}・T^{OC} の場合，CNN モデルが LSTM モデルより優れた性能を見せた．一方，T^{SP} の場合，LSTM モデルが CNN モデルより高い精度を記録した．そのため，感情分類タスクに取り組む際，対象語や根拠表現を探索するためなら CNN を特徴量抽出機として利用し，極性表現探索なら LSTM を特徴量抽出機として利用する方が精度向上に有利であると判断される．

本研究の特異点として，対象語・根拠表現分類タスクにおいて，検証データを使用した場合と評価データを使用した場合とで精度に大きく差が生じたことが挙げられる．特に，外部データなしで訓練したモデルを評価した際，検証データを用いた際の結果より約 20%精度が向上したが，これは訓練時検証データが役割を果たせなかったと考えられる．原因として考えられるのは，L1516 の検証データが L1516 の訓練データから抽出されたものである一方，AE や ALC の検証データはそれらの訓練データと一切重複しないよう抽出したものであることである．

今後の課題として，以上で述べた検証データ問題の他，Electronics とは全く性質の異なる Restaurant や Movie などのドメインにおいても同一の実験を行い，大量データの肯定的影響を一般化する必要がある．

謝辞

他分野から進学し，研究に詰まることも多々ありましたが，諦めず優しくご指導下さった指導教員の松本裕治教授に心から感謝申し上げます．同じく有意義なご指導ご助言下さった指導教員の新保仁准教授，進藤裕之助教授にも感謝致します．秘書の北川祐子さんには，留学生活に有用な様々な情報を頂き，大変感謝致します．また，研究会や勉強会など色々な場面で研究生活，留学生活を支えて下さった研究室の皆様にも深く感謝致します．松本研で過ごした二年間，多様な知識や技術は勿論，人生を生きていく上で必要な知恵も沢山学ぶことができました．これからの社会人生活も，松本研で学んだことを肝に銘じ精進致します．末筆ながら，他国での留学・研究生活を支えて頂きました家族に深く感謝致します．

参考文献

[1] Jonathan Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning*, 28(1):7–39, 1997.

[2] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occam's razor. *Information processing letters*, 24(6):377–380, 1987.

[3] Xilun Chen and Claire Cardie. Multinomial adversarial networks for multi-domain text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1226–1240, 2018.

[4] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[6] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[7] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org, 2017.

[8] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 504–515, 2019.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[11] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, 2014.

[12] Y LE CUN. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, pages 255–258, 1995.

[13] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. 1995.

[14] Xin Li, Lidong Bing, Piji Li, and Wai Lam. A unified model for opinion target extraction and target sentiment prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6714–6721, 2019.

[15] Xin Li and Wai Lam. Deep multi-task learning for aspect term extraction with memory interaction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2886–2892, 2017.

[16] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[17] Pengfei Liu, Jie Fu, Yue Dong, Xipeng Qiu, and Jackie Chi Kit Cheung. Learning multi-task communication with message passing for sequence learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4360–4367, 2019.

[18] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10, 2017.

[19] Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 912–921, 2015.

[20] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*, 2015.

[21] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015.

[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[23] Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. How transferable are neural networks in nlp applications? In *Proceedings of*

the 2016 Conference on Empirical Methods in Natural Language Processing, pages 479–489, 2016.

[24] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

[25] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[26] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, 2018.

[27] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30, 2016.

[28] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, 2015.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[30] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[31] Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. Double embeddings and cnn-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 592–598, 2018.

[32] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[33] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377, 2019.

[34] Jiawei Yong. A cross-topic method for supervised relevance classification. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 147–152, 2019.

[35] Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. Integrating semantic knowledge to tackle zero-shot text classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1031–1040, 2019.

[36] Suyang Zhu, Shoushan Li, Ying Chen, and Guodong Zhou. Corpus fusion for emotion classification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages

3287–3297, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.