Aspect-Level Sentiment Analysis based on Gradual Machine Learning

Yanyan Wang, Qun Chen, Jiquan Shen, Boyi Hou, Murtadha Ahmed, Zhanhuai Li

School of Computer Science, Northwestern Polytechnical University

Key Laboratory of Big Data Storage and Management, Northwestern Polytechnical University, Ministry of Industry and Information Technology, Xi'an Shaanxi, P.R.China

Abstract

The state-of-the-art solutions for Aspect-Level Sentiment Analysis (ALSA) were built on a variety of Deep Neural Networks (DNN), whose efficacy depends on large quantities of accurately labeled training data. Unfortunately, high-quality labeled training data usually require expensive manual work, thus may not be readily available in real scenarios. In this paper, we propose a novel approach for aspect-level sentiment analysis based on the recently proposed paradigm of Gradual Machine Learning (GML), which can enable accurate machine labeling without the requirement for manual labeling effort. It begins with some easy instances in a task, which can be automatically labeled by the machine with high accuracy, and then gradually labels the more challenging instances by iterative factor graph inference. In the process of gradual machine learning, the hard instances are gradually labeled in small stages based on the estimated evidential certainty provided by the labeled easier instances. Our extensive experiments on the benchmark datasets have shown that the performance of the proposed solution is considerably better than its unsupervised alternatives, and also highly competitive compared with the state-of-the-art supervised DNN models.

Keywords: Gradual machine learning, Factor graph inference, Aspect-level sentiment analysis

1. Introduction

Aspect-Level Sentiment Analysis (ALSA) [1], a fine-grained classification task, is highly valuable to both consumers and companies because it can provide them with detailed opinions expressed towards certain aspects of an entity. The task of ALSA has been classified into two finer subtasks, Aspect-Term Sentiment Analysis (ATSA) and Aspect-Category Sentiment Analysis (ACSA) [2]. ATSA aims to predict the sentiment polarity associated with an explicit aspect term appearing in the text. ACSA instead deals with both explicit and implicit aspects. It needs to predict the sentiment polarities of all the pre-specified aspects in a review, even though an aspect term may not explicitly appear in the text. For instance, consider the running example shown in Table 1, in which r_i and s_{ij} denote the review and sentence identifiers respectively. The review r_2 expresses the opinions about a laptop from two aspects, *battery* and *performance*. The goal of ATSA is to predict the sentiment polarity toward the explicit aspect *battery*; while ACSA has to identify the aspect polarities of both *battery* and *performance* even though the aspect term of *performance* does not appear in the text. In this paper, we target both ATSA and ACSA.

The state-of-the-art techniques for aspect-level sentiment analysis have been built on a variety of DNN models [2, 3, 4]. Compared with previous learning models [5, 6], the DNN models can effectively improve classification accuracy by automatically learning multiple levels of representation from data. However, their efficacy depends heavily on large quantities of accurately labeled training data. Unfortunately, high-quality labeled data usually require expensive manual work, thus may not be readily available in real scenarios. To address this limitation, this paper presents a novel solution based on the recently proposed paradigm of Gradual Machine Learning (GML) [7, 8], which can

Email addresses: wangyanyan@mail.nwpu.edu.cn (Yanyan Wang), chenbenben@nwpu.edu.cn (Qun Chen), shenjiquan@mail.nwpu.edu.cn (Jiquan Shen), ntoskrnl@mail.nwpu.edu.cn (Boyi Hou), a.murtadha@mail.nwpu.edu.cn (Murtadha Ahmed), lizhh@nwpu.edu.cn (Zhanhuai Li)

Table 1: A running example from laptop reviews.

r_i	s_{ij}	**Text**
r_1	s_{11}	I **like** the battery that can last **long** time.
	s_{12}	However, the keyboard sits a little **far** back for me.
r_2	s_{21}	The laptop has a **long** battery life.
	s_{22}	It also can run my games **smoothly**.

enable accurate machine labeling without the requirement for manual labeling effort. Inspired by the gradual nature of human learning, which is adept at solving problems with increasing hardness, GML begins with some easy instances in a task, which can be automatically labeled by the machine with high accuracy, and then gradually reasons about the labels of the more challenging instances based on the observations provided by the labeled easier instances. The general paradigm of GML consists of three steps: easy instance labeling, feature extraction and influence modeling, and finally gradual inference. GML has been successfully applied to the problem of entity resolution [8]. It has been empirically shown that GML performs considerably better than its unsupervised alternatives; its performance is even highly competitive compared with the state-of-the-art DNN solution.

As pointed out in [7, 8], even though there already exist many learning paradigms, including transfer learning [9], lifelong learning [10], curriculum learning [11], and self-training learning [12] to name a few, GML is fundamentally distinct from them due to its following two properties:

- Distribution misalignment between easy and hard instances in a task. The scenario of gradual machine learning does not satisfy the i.i.d (independent and identically distributed) assumption underlying most existing machine learning models. In GML, the labeled easy instances are not representative of the unlabeled hard instances. The distribution misalignment between the labeled and unlabeled instances renders most existing learning models unfit for gradual machine learning.

- Gradual learning by small stages in a task. Gradual machine learning proceeds in small stages. At each stage, it typically labels only one instance based on the evidential certainty provided by the labeled easier instances. The process of iterative labeling can be performed in an unsupervised manner without any human intervention.

We summarize the major contributions of this paper as follows:

- We propose a novel approach for aspect-level sentiment analysis based on the paradigm of gradual machine learning. It can achieve accurate machine labeling without the requirement for manual labeling effort.

- We present a package solution to enable effective gradual learning on ALSA, which include the techniques for easy instance labeling, feature extraction and influence modeling, and scalable gradual inference.

- We have empirically evaluated the performance of the proposed solution by a comparative study on benchmark data. Our extensive experiments have shown that its performance is considerably better than its unsupervised alternatives, and also highly competitive compared with the state-of-the-art supervised DNN models. Moreover, the GML solution is robust in that its performance is, to a large extent, insensitive to algorithmic parameters.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 defines the task of ALSA and provides a paradigm overview of gradual machine learning. Section 4 describes the technical solution for ALSA. Section 5 presents the scalable solution of gradual inference. Section 6 empirically evaluates the performance of the proposed solution. Finally, we conclude this paper with Section 7.

2. Related work

In this section, we first review the existing machine learning paradigms, and then discuss the existing work on sentiment analysis with the focus on aspect-level sentiment analysis.

2.1. Machine Learning Paradigms.

We first proposed the paradigm of gradual machine learning and applied it to the task of entity resolution in [7, 8]. There exist many other machine learning paradigms proposed for a wide variety of classification tasks. Here we will briefly review those closely related to GML and discuss their difference from GML.

Traditional machine learning algorithms make predictions on the future data using the statistical models that are trained on previously collected labeled or unlabeled training data [13, 14, 15, 16]. In many real scenarios, the labeled data may be too few to build a good classifier. Semi-supervised learning [17, 18] addresses this problem by making use of a large amount of unlabeled data and a small amount of labeled data. Nevertheless, the efficacy of both supervised and semi-supervised learning paradigms depends on the i.i.d assumption. Therefore, they can not be applied to the scenario of gradual machine learning.

In contrast, transfer learning [9], allows the distributions of the data used in training and testing to be different. It focused on using the labeled training data in a domain to help learning in another target domain. The other learning techniques closely related to transfer learning include lifelong learning [10] and multi-task learning [19]. Lifelong learning is similar to transfer learning in that it also focused on leveraging the experience gained on the past tasks for the current task. However, different from transfer learning, it usually assumes that the current task has good training data, and aims to further improve the learning using both the target domain training data and the knowledge gained in past learning. Multi-task learning instead tries to learn multiple tasks simultaneously even when they are different. A typical approach for multi-task learning is to uncover the pivot features shared among multiple tasks. However, all these learning paradigms can not be applied to the scenario of gradual machine learning. Firstly, focusing on unsupervised learning within a task, gradual machine learning does not enjoy the access to good labeled training data or a well-trained classifier to kick-start learning. Secondly, the existing techniques transfer instances or knowledge between tasks in a batch manner. As a result, they do not support gradual learning by small stages within a task.

The other related machine learning paradigms include curriculum learning (CL) [11] and self-paced learning (SPL) [20]. Both of them are, to some extent, similar to gradual machine learning in that they were also inspired by the learning principle underlying the cognitive process in humans, which generally start with learning easier aspects of a task, and then gradually takes more complex examples into consideration. However, both of them depend on a curriculum, which is a sequence of training samples essentially corresponding to a list of samples ranked in ascending order of learning difficulty. A major disparity between them lies in the derivation of the curriculum. In CL, the curriculum is assumed to be given by an oracle beforehand, and remains fixed thereafter. In SPL, the curriculum is instead dynamically generated by the learner itself, according to what the learner has already learned. Based on the traditional learning models, both CL and SPL depend on the i.i.d assumption and require good-coverage training examples for their efficacy. However, the scenario of gradual machine learning does not satisfy the i.i.d assumption. GML actually aims to eliminate the dependency on good-coverage training data.

Online learning [21] and incremental learning [22] have also been proposed for the scenarios where training data only becomes available gradually over time or its size is out of system memory limit. Built on the traditional learning models, both of them depend on high-quality training data for their efficacy. Therefore, they can not be applied for gradual learning either.

2.2. Sentiment Analysis

In general, sentiment analysis involves various tasks, such as polarity classification, subjectivity or objectivity identification, and multimodal fusion [23]. In this paper, we focus on the essential task of polarity classification. Sentiment analysis at different granularity levels, including document, sentence, and aspect, has been extensively studied in the literature [24]. At the document (resp. sentence) level, its goal is to detect the polarity of the entire document (resp. sentence) without regard to the mentioned aspects. The state-of-the-art solutions have been built based on deep neural networks (e.g. CNN and RNN), which include Character-level Convolutional Networks [25], Deep Pyramid Convolutional Neural Networks [26] and Linguistically Regularized LSTM [27]. There also exist some semi-supervised approaches for the particular problem of social data analysis [28], for which the acquisition of labeled data often requires a costly process that involves skilled experts whereas the acquisition of unlabeled ones is relatively inexpensive. Specifically, Silva [29] proposed a semi-supervised learning framework that can effectively leverage the unsupervised information captured by a similarity matrix, which is constructed based on unlabeled data, in classifier training. Hussain [30] also presented a combined model of random projection and support vector machine.

Unfortunately, all these proposals can not be directly applied to aspect-level sentiment analysis because a sentence may hold different opinions on different aspects. Moreover, their efficacy depends on the availability of large quantities of labeled training data.

Since deep neural networks can automatically learn high-quality features or representations, most recent work attempted to adapt such models for aspect-level sentiment analysis. We discuss the work for ATSA and ACSA separately. For the ATSA task, Dong [31] proposed an Adaptive Recursive Neural Network (AdaRNN) that employed a novel multi-compositionality layer to propagate the sentiments of words towards the target. Noticing that the models based on recursive neural network heavily rely on external syntactic parser, which may result in inferior performance, many researchers subsequently focused on recurrent neural networks. Tang [32] proposed a target-dependent LSTM (TD-LSTM) model to capture the connection between target words and their contexts. The alternative solutions include the memory networks and the convolutional neural networks. Wang [33] proposed a Target-sensitive Memory Network that aimed to capture the sentiment interaction between targets and contexts. Li [34] presented a Transformation Network that employed a CNN layer to extract salient features from the transformed word representations originated from a bi-directional RNN layer. Due to the great success of attention mechanism in image recognition [35], speech recognition [36], machine translation [37, 38] and question answering [39], many attention-based models have also been proposed for ATSA. These models, including Hierarchical Attention Network [40], Segmentation Attention Network [41], Interactive Attention Networks [42], Recurrent Attention Network [43], Attention-over-Attention Neural Networks [44], Effective Attention Modeling [45], Content Attention Model [46], Multi-grained Attention Network [47] and Sentic LSTM [48], employed different attention mechanisms to output the aspect-specific sentiment features. It is noteworthy that Sentic LSTM [48] specifically focused on leveraging commonsense knowledge in the deep neural sequential model. More recently, some researchers investigated how to leverage the BERT model for ATSA. Song [49] proposed an Attentional Encoder Network (AEN) which employed the pretrained BERT and the attention-based encoders. Zeng [50] presented a Local Context Focus (LCF) mechanism based on Multi-head Self-Attention (MHSA), and adopted a BERT-shared layer in its LCF design.

In comparison, there exist fewer proposals for ACSA because implicit aspects make polarity detection more challenging. Ruder [4] proposed a hierarchical bidirectional LSTM for ACSA by modeling the inter-dependencies of sentences in a review. Wang [3] presented an attention-based LSTM that employed an aspect-to-sentence attention mechanism to concentrate on the key part of a sentence given an aspect. Xue [2] introduced a convolutional neural network augmented with gating mechanisms, which was empirically shown to be more accurate and efficient compared with its previous alternatives. It is worthy to point out that the DNN models proposed for ACSA can also be used for ATSA, but the models proposed for ATSA are usually not applicable to ACSA because they employ specific mechanisms to model an explicit aspect term along with its relative context. However, the efficacy of the existing DNN-based approaches for ATSA and ACSA depends heavily on good-coverage training data, which may not be readily available in real scenarios.

There also exist some work for semi-supervised aspect-level sentiment classification. Cheng [51] employed the Variational Autoencoder based on Transformer to effectively improve the performance of supervised DNN models for ATSA. Wang [52, 53] presented a joint framework, SenHint, which can seamlessly integrate the output of deep neural networks and the implications of linguistic hints in a unified model of factor graph. Similar to GML, SenHint also identifies easy instances, and leverages the extracted features to improve the accuracy of polarity reasoning. However, SenHint labels all the hard instances simultaneously in a single iteration, and its efficacy depends heavily on the output of DNN models. In comparison, GML gradually labels hard instances in small stages based on the evidential certainty provided by labeled easier instances. Typically, GML runs in many iterations, and in each iteration, it labels one and only one hard instance. Moreover, the GML solution proposed in this paper does not require any manually labeled data.

Finally, it is worthy to point out that word polarity disambiguation is an important problem in sentiment analysis, whose main challenge is to resolve the polarity of the sentiment-ambiguous words in different contexts. Recently, Xia [54] proposed a Bayesian model for opinion-level features to solve the problem of polarity disambiguation. Rafeek [55] presented a new approach of Bayesian Network Based Contextual Polarity Disambiguation (BbNCPD) to resolve the polarities of context-dependent opinion words. Their empirical studies have shown that these solutions can effectively improve the accuracy of polarity detection.

Table 2: Frequently used notations.

Notation	Description
r_j	a review
s_k	a sentence
a_l	an aspect category or aspect term
$t_i = (r_j, s_k, a_l)$	an aspect unit
$T = \{t_i\}$	a set of aspect units
v_i	a boolean variable indicating the polarity of the aspect unit t_i
$V = \{v_i\}$	a set of aspect polarity variables

3. Preliminaries

In this section, we first define the task of aspect-level sentiment analysis, and then provide a paradigm overview of gradual machine learning.

3.1. Task Statement

For presentation simplicity, we have summarized the frequently used notations in Table 2. Formally, we formulate the task of aspect-level sentiment analysis as follows:

Definition 1. [Aspect-level Sentiment Analysis] *Let $t_i = (r_j, s_k, a_l)$ be an aspect unit, where r_j denotes a review, s_k denotes a sentence in the review r_j, and a_l denotes an aspect associated with the sentence s_k. Note that the aspect a_l can be an aspect category or aspect term, and a sentence may express opinions towards multiple aspects. Given a corpus of reviews, R, the goal of the task is to predict the sentiment polarity of each aspect unit t_i in R.*

In this paper, we suppose that an aspect polarity is either positive or negative.

3.2. Paradigm Overview

As presented in [7, 8], the paradigm of gradual machine learning, which has been shown in Figure 1, consists of the following three steps:

- **Easy Instance Labeling.** Given a classification task, it is usually very challenging to accurately label all the instances in the task without good-coverage training examples. However, the work can become much easier if we only need to automatically label some easy instances in the task. In real scenarios, easy instance labeling can be performed based on the simple user-specified rules or the existing unsupervised learning techniques. For instance, in unsupervised clustering, an instance close to a cluster center in the feature space can usually be considered as an easy instance, because it has only a remote chance to be misclassified. Gradual machine learning begins with the label observations of easy instances. Therefore, high accuracy of automatic easy instance labeling is critical for GML's ultimate performance.

- **Feature Extraction and Influence Modeling.** In GML, feature serves as the medium to convey the knowledge obtained from labeled easy instances to unlabeled harder ones. This step extracts the common features shared by the labeled and unlabeled instances. To facilitate effective knowledge conveyance, it is desirable that a wide variety of features are extracted to capture as much information as possible. For each extracted feature, this step also needs to model its influence over the labels of its relevant instances.

- **Gradual Inference.** This step gradually labels the instances with increasing hardness in a task. Since the scenario of gradual learning does not satisfy the i.i.d assumption, gradual learning is fulfilled from the perspective of evidential certainty. As shown in Figure 1, it constructs a factor graph, which consists of the labeled and unlabeled instances and their common features. Gradual learning is conducted over the factor graph by iterative inference. At each iteration, it chooses to label the unlabeled instance with the highest degree of evidential

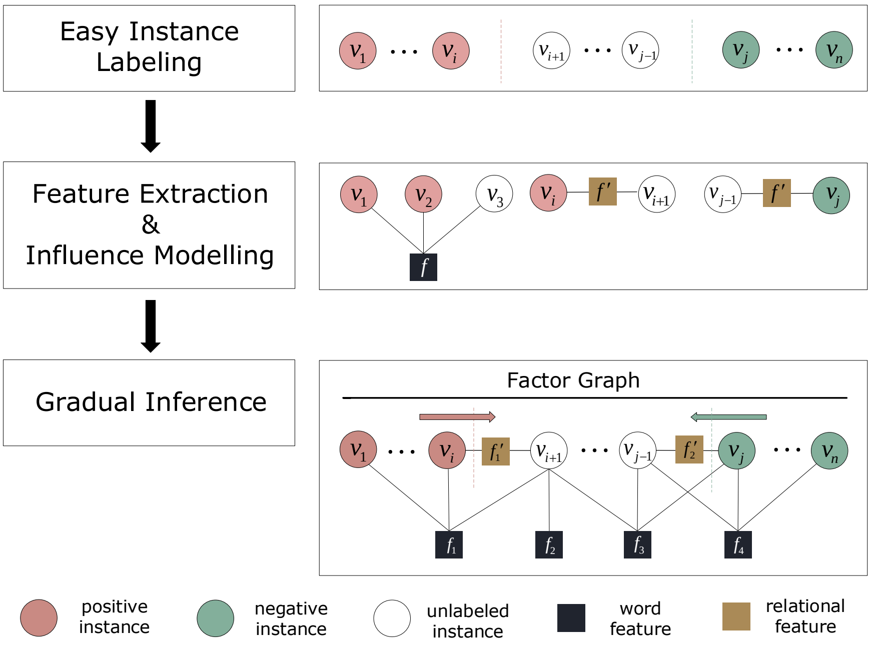


Figure 1: Learning paradigm overview.

certainty. The iteration is repeatedly invoked until all the instances in a task are labeled. In gradual inference, a newly labeled instance at the current iteration would serve as an evidence observation in the following iterations.

Formally, we denote the model of factor graph corresponding to a classification workload by G. Suppose that G consists of a set of evidence variables $\mathbf{\Lambda}$, whose labels are known, a set of inference variables $\mathbf{V_I}$, whose labels are unknown, and a group of factor functions of variables to indicate the probabilistic relations among the variables, denoted by $\mathbf{F}_\theta(V_i) : V_i \rightarrow P_\theta(V_i)$, in which V_i denotes a set of variables and $V_i \in \text{PowerSet}(\mathbf{\Lambda} \cup \mathbf{V_I})$. In each iteration, GML generally chooses to label the inference variable in $\mathbf{V_I}$ with the highest degree of evidential certainty. Suppose that the total number of label types, denoted by $\{T_1, T_2, \ldots, T_t\}$, is t. Given an inference variable v, GML measures its evidential certainty by the inverse of entropy as follows

$$E(v) = \frac{1}{H(v)} = \frac{1}{-\sum\limits_{1 \leq i \leq t} P_i(v) \cdot \log_2 P_i(v)}, \tag{1}$$

in which $E(v)$ and $H(v)$ denote the evidential certainty and entropy of v respectively, and $P_i(v)$ denotes the inferred probability of v having the label of T_i.

4. Solution for ALSA

This section presents the solution of gradual machine learning for ALSA, which include the corresponding techniques for the three steps laid out in Subsection 3.2.

4.1. Easy Instance Labeling

The existing lexicon-based approaches [56] essentially reason about polarity by summing up the polarity scores of all the sentiment words in a sentence. The score of a sentiment word indicates its intensity of sentiment, which is supposed to increase with the absolute value of score. Since a negation word can effectively reverse the polarity of a sentiment word, they usually perform negation detection for each sentiment word by examining whether there is any negation in its neighboring words [57].

Unfortunately, the lexicon-based approaches are prone to making mistakes under some ambiguous circumstances. Firstly, the presence of contrast (e.g. *but* and *although*), hypothetical (e.g. *if*) or condition (e.g. *unless*) connectives could significantly complicate polarity detection. For instance, the sentence, "would be a very nice laptop if the mousepad worked properly", contains only the positive sentiment words "nice" and "properly", but it holds negative attitude due to the presence of the hypothetical connective "if". Secondly, the negation words involving long-distance dependency could also make polarity detection challenging. For instance, in the sentence, "I don't really think the laptop has a good battery life", the negation word "don't" reverses the polarity, but it is far away from the sentiment word "good". Finally, a sentence may contain multiple sentiment words that hold conflicting polarities; in this case, its true polarity is not easily detectable based on sentiment word scoring.

Therefore, as originally proposed in [53], we identify easy instances by excluding the aforementioned ambiguous circumstances as follows:

Definition 2. [Easy Instance] *We consider an aspect polarity, $t_i = (r_j, s_k, a_l)$, as an easy instance if and only if the sentence expressing opinions about the aspect, s_k, simultaneously satisfies the following three conditions:*

- *It contains at least one sentiment word, but does not simultaneously contain any sentiment word holding a conflicting polarity;*
- *It does not contain any contrast, hypothetical or condition connective;*
- *It does not contain any negation word involving long-distance dependency.*

The polarity of an easy instance is simply determined by the polarity of its sentiment words. Moreover, a negation word is supposed to involve long-distance dependency if and only if it is not in the 3-gram preceding any sentiment word. We illustrate the difference between the easy and challenging instances by Example 1.

Example 1. [Easy Instances] *In a phone review, the sentence, "the screen is not good for carrying around in your bare hands", which expresses opinion about "screen", is an easy instance because the sentiment word "good" associated with the local negation cue "not" strongly indicates the negative sentiment. In contrast, the sentence, "I don't know why anyone would want to write a great review about this battery", which expresses opinion about "battery", is not an easy instance. Even though it contains the sentiment word "great", it also includes the negation word "don't" involving long-distance dependency. Similarly, the sentence, "I like this laptop, the only problem is that it can not last long time", is not an easy instance because it contains both positive and negative words, i.e. "like" and "problem" respectively.*

4.2. Feature Extraction and Influence Modeling

We extract two types of features for influence modeling: word feature and relational feature.

Word Feature. Sentiment polarity is usually determined by sentiment words. Therefore, we extract sentiment words, which have been specified in the open-source lexicons, from sentences and consider them as the features of aspect polarities. To capture more information shared among aspect instances, besides the single sentiment words, we also extract k-grams ($k \geq 2$) as word features. Since word context (e.g. negation and subjunctive) can effectively alter the polarity of an opinion word, we thus perform context detection for each word feature by examining whether there is any negation or subjunctive word in its neighborhood.

In ALSA, a sentence may express opinions towards multiple aspects. Therefore, we need to associate each sentiment feature with its target aspect. For this purpose, we first use the technique proposed in [58] to extract opinion phrases. It leverages the patterns based on dependency relations [59] for phrase extraction. The typical patterns include: 1) adjective modifier: $amod(N,A) \rightarrow< N,A >$ (e.g. in the sentence "great camera", we have $amod(camera,great) \rightarrow< camera,great >$); 2) joint clausal complement and nominal subject: $acomp(N,A) + nsubj(V,N) \rightarrow< N,A >$(e.g. in the sentence "the camera looks beautiful", we have $acomp(camera,beautiful) + nsubj(looks,camera) \rightarrow< camera,beautiful >$); 3) joint direct object and nominal subject: $dobj(V,N) + nsubj(V,N') \rightarrow< N,V >$(e.g. in the sentence "i like this keyboard", we have $dobj(like, keyboard) + nsubj(like, I) \rightarrow< keyboard, like >$). Please refer to [58] for more details.

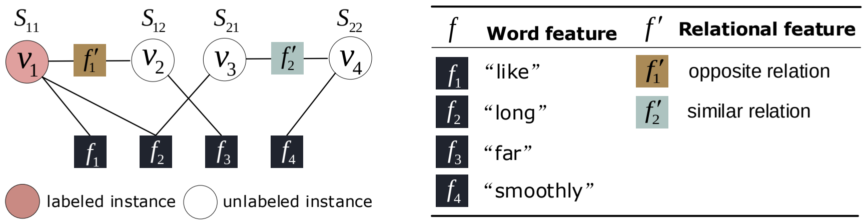


Figure 2: The factor graph constructed for the running example.

Next, we associate sentiment features with their target aspects based on the extracted opinion phrases. In the case of ATSA, it is easy to correlate a opinion word (corresponding to a sentiment feature) in a sentence with its target aspect because the aspect term explicitly appears in the text. In the case of ACSA, we assign an opinion word to an aspect if and only if either its opinion target or the opinion word itself is close to the aspect term in the vector space (namely, their similarity exceeds a threshold (e.g. 0.5 in our implementation)).

Relational Feature. Modeling sentences independently, the existing DNN models for aspect-level sentiment analysis have very limited capability in capturing the contextual information at sentence level. However, the sentences in a review build upon each other. There often exist some discourse relations between clauses or sentences, which can provide valuable hints for polarity reasoning. Specifically, it can be observed that two sentences connected with a shift word usually have opposite polarities. In contrast, two neighboring sentences without any shift word between them usually have similar polarities. In the running example shown in Table 1, the polarities of s_{11} and s_{12} are opposite because they are connected by the shift word of "but", while the polarities of s_{21} and s_{22} are similar due to the absence of any shift word between them.

Therefore, as originally proposed in [53], we use the rules to extract the similar or opposite relation between two aspect units based on their sentence context. Given two aspect units $t_i = \{r_i, s_i, a_i\}$ and $t_j = \{r_j, s_j, a_j\}$ that are opinioned in the same review (namely $r_i = r_j$), the rules for polarity relation extraction are specified as follows:

1. If the sentences s_i and s_j are identical (s_i=s_j) or adjacent and neither of them contains any shift word, t_i and t_j are supposed to hold similar polarities;
2. If two adjacent sentences s_i and s_j are connected by a shift word and neither of them contains any inner-sentence shift word, t_i and t_j are supposed to hold opposite polarities;
3. If the sentences s_i and s_j are identical and the opinion clauses associated with them are connected by an inner-sentence shift word, t_i and t_j are supposed to hold opposite polarities.

Given an ATSA task, it is easy to correlate an opinion clause with its target aspect because the aspect term explicitly appears in the text. Therefore, the condition specified in the 3rd rule can be easily checked in the scenario of ATSA. The scenario of ACSA is instead more challenging. Our solution first uses the dependency-based parse tree to extract all the opinion phrases, and then associates an opinion clause with a specific aspect if either its opinion target or opinion word is close to the aspect in the vector space.

4.3. Gradual Inference

As usual, we construct a factor graph, G, in which the labeled easy instances are represented by the *evidence variables*, the unlabeled hard instances by the *inference variables*, and the features by the *factors*. The value of each variable represents its corresponding polarity. An evidence variable has the constant value of 0 or 1, which indicate the polarity of *negative* and *positive* respectively. The values of the evidence variables remain unchanged during the inference process. The values of the inference variables should instead be inferred based on G. The factor graph constructed for the running example has been shown in Figure 2.

Gradual machine learning is attained by iterative factor graph inference on G. In G, we define the probability distribution over its variables V by

$$P_w(V) = \frac{1}{Z_w} \prod_{v \in V} \prod_{f \in F_v} \phi_f(v) \prod_{f' \in F'} \phi_{f'}(v_i, v_j), \tag{2}$$

where F_v denotes the set of word features associated with the variable v, F' denotes the set of relational features, $\phi_f(v)$ denotes the factor associated with v and f, and $\phi_{f'}(v_i, v_j)$ denotes the factor associated with the relational feature f'. In Eq. 2, the factor of a word feature f is defined by

$$\phi_f(v) = \begin{cases} 1 & v = 0; \\ e^{w_f} & v = 1; \end{cases} \tag{3}$$

where v denotes a variable having the feature f, and w_f denotes the weight of f. Similarly, the factor of a relational feature f' is defined by

$$\phi_{f'}(v_i, v_j) = \begin{cases} e^{w_{f'}} & if\ v_i = v_j; \\ 1 & otherwise; \end{cases} \tag{4}$$

where v_i and v_j denote the two variables sharing the feature f', and $w_{f'}$ denotes the weight of f'. Note that the weight of a word factor can be positive or negative, while the weight of a *similar* relational factor is positive and the weight of an *opposite* relational factor is negative. In our implementation, the weights of all the *similar* relational factors are set to be the same; the weights of all the *opposite* relational factors are also set to be the same.

As in [7], given a factor graph with some labeled evidence variables, we reason about the factor weights by minimizing the negative log marginal likelihood of

$$\hat{w} = arg \min_{w} -log \sum_{V_I} P_w(\Lambda, V_I), \tag{5}$$

where Λ denotes the observed labels of evidence variables and V_I denotes the set of inference variables. The objective function effectively learns the factor weights most consistent with the label observations of evidence variables. In our implementation, we have used the Numbskull library [1] to optimize this objective function by interleaving stochastic gradient descent steps with Gibbs sampling ones, similar to contrastive divergence.

As usual, gradual inference proceeds in small stages. At each stage, it chooses to label the unlabeled variable with the highest degree of evidential certainty in G. The iteration is repeatedly invoked until all the inference variables are labeled. In gradual inference, evidential certainty is measured by the inverse of entropy. In the case of ALSA, entropy is formally defined by

$$H(v) = -(P(v) \cdot \ln P(v) + (1 - P(v)) \cdot \ln(1 - P(v))), \tag{6}$$

in which $H(v)$ denotes the entropy of a variable v, and $P(v)$ denotes the inferred probability of v.

In gradual inference, only the inference variables receiving considerable evidential support from labeled instances need to be considered for labeling. In the next section, we will present a scalable solution for gradual inference on ALSA.

5. Scalable Gradual Inference

We have built the scalable solution based on the framework proposed in [8], which consists of three steps, measurement of evidential support, approximate ranking of entropy and construction of inference subgraph. The process of scalable gradual inference is sketched in Algorithm 1. Given a factor graph G, it first selects the top-m unlabeled variables with the most evidential support in G as the candidates for probability inference. To reduce the invocation frequency of factor graph inference, it then approximates entropy estimation by an efficient algorithm on the m candidates and selects only the top-k most promising variables among them for factor graph inference. Finally, it infers the probabilities of the chosen k variables in G. For each variable, its probability is not inferred over the entire graph of G, but over a potentially much smaller subgraph.

[1] https://github.com/HazyResearch/numbskull

```
Algorithm 1: Scalable Gradual Inference
while there exists any unlabeled variable in $G$ do
    $V' \leftarrow$ all the unlabeled variables in $G$;
    for $v \in V'$ do
        Measure the evidential support of $v$ in $G$;
    Select top-$m$ unlabeled variables with the most evidential support (denoted by $V_m$) ;
    for $v \in V_m$ do
        Approximately rank the entropy of $v$ in $V_m$;
    Select top-$k$ most promising variables in terms of entropy in $V_m$ (denoted by $V_k$) ;
    for $v \in V_k$ do
        Compute the probability of $v$ in $G$ by factor graph inference over a subgraph of $G$;
    Label the variable with the minimal entropy in $V_k$;
```

5.1. Measurement of Evidential Support

Here we first introduce the Dempster–Shafer (D-S) theory [60], the classical framework for evidential support estimation, and then describe how to leverage it for evidential support measurement.

The D-S theory, also known as evidence theory, is a general framework for reasoning with uncertainty. It allows one to combine the beliefs from different evidence sources and arrive at a degree of belief that takes into account all the available evidences. The basic concepts involved in the D-S theory are described as follows:

- **Propositions.** Denoted by X, it represents all possible states of a situation under consideration.

- **Power set of propositions.** Denoted by 2^X, it includes all of the subsets of the propositions.

- **Belief function.** Denoted by $m(\cdot)$, it assigns a degree of belief (or mass) to each element E of the power set 2^X. The masses of elements satisfy $\sum_{E\in 2^X} m(E) = 1$ and $m(\emptyset) = 0$. In case that only singleton propositions are assigned degrees of belief, a belief function reduces to a classical probability function.

- **Belief combining rules.** It aims to combine degrees of belief indicated by independent evidence sources with various fusion operators. The popular fusion operator is Dempster's rule of combination, which derives common shared belief between multiple sources and ignores all the conflicting (non-shared) belief through a normalization factor.

Given an inference variable v, the purpose is to estimate its overall evidential support in terms of labeling. It can be observed that 1) various features (e.g. word features and relational features) can be considered as different evidence sources providing hints for labeling; 2) each feature has some inherent uncertainty when indicating label status.

We first define two propositions: "label the instance", denoted by L, and "unlabel the instance", denoted by U. With $X = \{L, U\}$, the power set of X can be represented by $2^X = \{\emptyset, L, U, X\}$. Then, we define different belief functions for various evidences (namely word features and relational features). Given an inference variable v and its word feature f, we estimate the evidential support that v receives from f by the belief function

$$m_f(E) = \begin{cases} (1-d_f)\cdot max\{P(f), 1-P(f)\} & E = \{L\}, \\ (1-d_f)\cdot min\{P(f), 1-P(f)\} & E = \{U\}, \\ d_f & E = \{L, U\}, \end{cases} \tag{7}$$

where d_f denotes the degree of uncertainty of f, and $P(f)$ denotes the proportion of positive instances among all labeled instances having the feature f. According to Eq. 7, the belief assigned to the element of $\{L\}$ increases as the value of $P(f)$ becomes more extreme (i.e. close to 0 or 1). The underlying intuition is that the more extreme the value of $P(f)$ is, the more evidential support the element of $\{L\}$ should receive from the feature f.

Similarly, given an inference variable v and its relational feature f', we estimate the evidential support that v receives from f' by the belief function

$$m_{f'}(E) = \begin{cases} (1-d_{f'}) \cdot R(f') & E = \{L\}, \\ (1-d_{f'}) \cdot (1-R(f')) & E = \{U\}, \\ d_{f'} & E = \{L, U\}, \end{cases} \tag{8}$$

where $d_{f'}$ denotes the degree of uncertainty of f', and $R(f')$ denotes the accuracy of the relation f'. In Eq. 8, $R(f')$ can be considered as the statistical accuracy of the extracted relations, which can be estimated based on labeled instances; the evidential support that the element of $\{L\}$ receives from f' thus increases with the estimated accuracy.

We are now ready to describe how to measure the aggregate evidential support provided by multiple features. Suppose that an inference variable v has i word features, $\{f_1,\ldots,f_i\}$, and j relational features, $\{f'_1,\ldots,f'_j\}$. Given the element of $E = \{L\}$, we estimate its aggregate evident support by combining the estimated beliefs as follows

$$m(E) = m_{f_1}(E) \oplus \cdots \oplus m_{f_i}(E) \oplus m_{f'_1}(E) \oplus \cdots \oplus m_{f'_j}(E), \tag{9}$$

where $m(E)$ denotes the total amount of evidential support that v receives, and the combination is calculated from the two sets of mass functions, $m_{f_1}(E)$ and $m_{f_2}(E)$, as follows

$$m_{f_1}(E) \oplus m_{f_2}(E) = \frac{1}{1-K} \sum_{E' \cap E'' = E} m_{f_1}(E') \cdot m_{f_2}(E''), \tag{10}$$

where E' and E'' denote the elements of the power set, and

$$K = \sum_{E' \cap E'' = \emptyset} m_{f_1}(E') \cdot m_{f_2}(E''), \tag{11}$$

which is a measure of the amount of conflict between E' and E''.

Note that the degree of uncertainty, denoted by d_f and $d_{f'}$ in Eq. 7 and 8, indicates how much impact a feature has on the whole degree of belief in terms of evidential support measurement. The lower the value, the greater the impact. It can be observed that relational features usually provide more reliable information than word features. Therefore, in practical implementation, we suggest that $d_{f'}$ is set to be smaller than d_f (e.g., $d_f = 0.4$ and $d_{f'} = 0.1$). Our empirical evaluation in Subsection 6.4 has shown that the performance of gradual machine learning is, to a large extent, insensitive to the parameter setting of d_f and $d_{f'}$.

On time complexity, each iteration takes $O(n \times n_f)$ time, in which n denotes the total number of instances in a task, and n_f denotes the total number of extracted features. Therefore, the time complexity of evidential support measurement can be represented by $O(n^2 \times n_f)$.

5.2. Approximate Ranking of Entropy

Since more evidential conflict means more status uncertainty, we approximate the entropy ranking of inference variables by measuring their evidential conflict. Specifically, we define two propositions: "label it as *positive*", denoted by L^+, and "label it as *negative*", denoted by L^-. Given an inference variable v and its word feature f, we approximate v's evidential certainty w.r.t f with the belief function

$$m^*_f(E) = \begin{cases} (1-d^*_f) \cdot P(f) & E = \{L^+\}, \\ (1-d^*_f) \cdot (1-P(f)) & E = \{L^-\}, \\ d^*_f & E = \{L^+, L^-\}, \end{cases} \tag{12}$$

where d^*_f denotes the degree of uncertainty of f, and $P(f)$ denotes the proportion of positive instances among all the labeled instances having the feature f.

Similarly, given an inference variable v and its relational feature f', we approximate v's evidential certainty w.r.t f' with the belief function

$$m^*_{f'}(E) = \begin{cases} (1-d^*_{f'}) \cdot P(f') & E = \{L^+\}, \\ (1-d^*_{f'}) \cdot (1-P(f')) & E = \{L^-\}, \\ d^*_{f'} & E = \{L^+, L^-\}, \end{cases} \tag{13}$$

where $d^*_{f'}$ denotes the degree of uncertainty of f', and $P(f')$ denotes the probability of v being positive if only the evidence f' is considered for labeling v. If the labeled variable on the other side of the relation f' is positive, we set $P(f') = \frac{e^{w_{f'}}}{1+e^{w_{f'}}}$, in which $w_{f'}$ denotes the weight of f'; otherwise (i.e., it is negative), we set $P(f') = \frac{1}{1+e^{w_{f'}}}$.

Finally, we measure the amount of conflict between the multiple pieces of evidence using the generalized expression of K as specified in Eq. 11. Similar to the case of evidential support measurement, we suggest that $d^*_{f'}$ is set to be a lower value than d^*_f. Our empirical evaluation in Subsection 6.4 has shown that the performance of gradual machine learning is, to a large extent, insensitive to the parameter setting of d^*_f and $d^*_{f'}$.

On time complexity, each iteration takes $O(n_f \times m)$ time, in which n_f denotes the total number of extracted features and m denotes the number of candidate variables selected for approximate entropy estimation as specified in Algorithm 1. Therefore, the time complexity of approximate entropy estimation can be represented by $O(n \times n_f \times m)$.

5.3. Construction of Inference Subgraph

It has been empirically shown [61] that given a variable v in G, its probability inference can be effectively approximated by considering the subgraph consisting of v and its r-hop neighboring variables, and even with a small value of r (e.g. 2 and 3), the approximation can be sufficiently accurate in many real scenarios. Therefore, given a target inference variable v in G, we extract all its 2-hop neighbors reachable by the relational factors and include them in the subgraph. For each word feature of v, all the labeled and unlabeled instances sharing the feature with v are also included in the constructed subgraph because potentially, their labels can significantly influence the label of v.

On time complexity, each iteration of subgraph construction takes $O(n_f \times k)$ time, in which n_f denotes the total number of extracted features and k denotes the number of candidate variables selected for factor graph inference as specified in Algorithm 1. Therefore, the time complexity of inference subgraph construction can be represented by $O(n \times n_f \times k)$.

6. Empirical Evaluation

In this section, we empirically evaluate the performance of the proposed solution by a comparative study. We have compared GML with the state-of-the-art techniques proposed for both ACSA and ATSA. Note that the DNN models proposed for ACSA can also be used for ATSA, but the models proposed for ATSA are usually not applicable to ACSA because they employ specific mechanisms to model an explicit aspect-term along with its relative context.

For the ACSA task, the compared techniques include:

- **LEX-SYN [62].** It is an unsupervised approach built on lexicons and syntactic dependency analysis;
- **VADER [57].** It is a rule-based method proposed for sentence-level sentiment analysis. We have adapted it for the task of ALSA. Given a sentence with multiple aspects, the solution identifies the sentiment polarity of an aspect by analyzing its opinioned clause, whose extraction has been explained in Subsection 4.2.
- **H-LSTM [4].** It is an enhanced DNN model. It models the inter-dependencies of sentences in a review using a hierarchical bidirectional LSTM;
- **AT-LSTM [3].** Referring to the Attention-based LSTM, it employs an attention mechanism to concentrate on the key parts of a sentence given an aspect, where the aspect embeddings are used to determine the attention weights;
- **ATAE-LSTM [3].** Referring to the Attention-based LSTM with Aspect Embedding, it is supposed to be an improvement over AT-LSTM. It extends AT-LSTM by appending the input aspect embedding to each word's input vector;
- **GCAE [2].** It uses convolutional neural networks and gating mechanisms to predict the sentiment polarity of a given aspect. Compared with the LSTM and attention mechanisms, it can be more accurate and efficient.

For the ATSA task, besides LEX-SYN, VADER, AT-LSTM, ATAE-LSTM and GCAE, the compared techniques also include:

- **IAN [42].** Referring to the interactive attention network, it models targets and contexts separately and learn their own representations via interactive learning. By modeling targets and contexts separately, it can pay close attention to the important parts in the target and context;

- **RAM [43].** It is a multiple-attention network where the features from multiple attentions are non-linearly combined with a recurrent neural network. It can effectively capture sentiment features separated by a long distance, and is usually more robust against irrelevant information;

- **AOA [44].** Referring to the attention-over-attention network, it models aspect and sentence in a joint way and benefits from modeling the interaction among word-pairs between sentences and targets;

- **TNet [34].** It is a target-specific transformation network that employs a CNN layer to extract salient features from the transformed word representations originated from a RNN layer. It avoids using attention for feature extraction so as to alleviate the attended noise.

- **ASVAET [51]**. It is a semi-supervised model, which can induce the underlying sentiment prediction for unlabeled data by disentangling the latent representation into the aspect-specific sentiment and the lexical context. Since the model is classifier-agnostic, it can be built upon various DNN models(e.g. IAN, RAM, AOA, TNet). In our empirical evaluation, we only report the results of ASVAET(IAN), which integrates IAN into ASVAET; the results on other DNNs are quite similar, thus omitted in the paper.

Besides the aforementioned approaches, we have also compared GML with SenHint [53]. The original SenHint depends on the output of DNN models. For fair comparison, we have implemented a trimmed version of SenHint without requiring DNN outputs. The trimmed SenHint first identifies some easy instances using the same technique proposed for GML, and then leverages the extracted shared features for polarity reasoning. Note that unlike GML, SenHint labels all the hard instances simultaneously in a single iteration. The rest of this section is organized as follows: Subsection 6.1 describes the experimental setup. Subsection 6.2 presents the comparative evaluation results. Subsection 6.3 evaluates the performance of easy instance labeling. Subsection 6.4 evaluates the performance sensitivity of the proposed solution w.r.t various parameters. Finally, Subsection 6.5 evaluates the scalability of the proposed solution.

6.1. Experimental Setup

In the empirical evaluation, we have used six benchmark datasets in four domains (phone, camera, laptop and restaurant) and two languages (English and Chinese) from the SemEval 2015 task 12 [63] and 2016 task 5 [64]. In all the experiments, we perform 2-class classification to label an aspect polarity as *positive* or *negative*. Note that the datasets of LAP16, RES16, LAP15 and RES15 contain some neutral instances, which are simply ignored in our experiments. The statistics of the benchmark datasets are presented in Table 3, in which #N(ACSA) and #N(ATSA) denote the numbers of aspect category units and aspect term units respectively. Note that aspect terms are not specified on these benchmark datasets. We have manually identified the aspect terms in LAP16, RES16, LAP15 and RES15 according to the annotation guideline [2].

For DNN models, we used the Glove embeddings [3] for English data, and the word embeddings from Baidu [4] for Chinese data. We employed jieba [5] to tokenize Chinese sentences. In easy instance labeling and feature extraction for GML, we used the open-source Opinion Lexicon [6] for English data, and the EmotionOntology [7] and BosonNLP [8] lexicons for Chinese data.

For DNN model training, we used the default ratio of train and test data provided in the benchmark. GML has been instead directly run on the test data without leveraging any labeled training data. For easy instance identification,

[2] http://alt.qcri.org/semeval2014/task4/data/uploads/semeval14_absa_annotationguidelines.pdf

[3] https://nlp.stanford.edu/projects/glove/

[4] http://pan.baidu.com/s/1jIb3yr8

[5] https://github.com/fxsjy/jieba

[6] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

[7] http://ir.dlut.edu.cn/EmotionOntologyDownload

[8] https://bosonnlp.com/dev/resource

Table 3: Data statistics of benchmark datasets.

Data	**Language**	**#N(ACSA)**		**#N(ATSA)**	
		Train	**Test**	**Train**	**Test**
PHO16	Chinese	1333	529	—	—
CAM16	Chinese	1259	481	—	—
LAP16	English	2715	751	1702	479
RES16	English	2134	693	1711	592
LAP15	English	1864	868	1168	533
RES15	English	1410	725	1191	521

Table 4: Accuracy comparison for ACSA on benchmark datasets.

Model	**PHO16**	**CAM16**	**LAP16**	**RES16**	**LAP15**	**RES15**
LEX-SYN	68.43%	78.17%	69.64%	76.77%	75.81%	75.31%
VADER	–	–	68.31%	75.18%	74.31%	75.59%
H-LSTM	(73.30 ± 0.19) %	(78.80± 0.60)%	(77.68 ± 0.65)%	(81.44± 0.39)%	(79.03± 0.48)%	(73.13± 1.26) %
AT-LSTM	(73.27± 1.21)%	**(82.49± 0.58)%**	(76.32 ± 0.74) %	(83.00± 0.43)%	(79.03 ± 1.00)%	(76.52 ± 1.51)%
ATAE-LSTM	(72.40± 0.82)%	(81.12± 0.70)%	(77.90 ± 1.10) %	(83.81± 1.08)%	(79.88 ± 0.79)%	(79.42 ± 1.07)%
GCAE	**(76.94± 0.48)%**	(82.12± 0.53)%	**(81.94 ± 0.40) %**	**(86.44± 0.61)%**	(82.21 ± 0.51)%	**(79.81 ± 0.70)%**
GML	(76.14± 0.63)%	(81.41± 0.25)%	(79.84 ± 0.34) %	(85.31 ± 0.06)%	**(83.94 ± 0.28)%**	(78.57 ± 0.48)%

the scores for the sentiment words in the Chinese lexicons are normalized into the range of [-4, 4], and we use the sentiment words whose scores are at least 1. In our implementation of GML, the initial weights of word features, similar relational features and opposite relational features are set to 0, 2 and -2 respectively. In the process of scalable gradual inference, if none of the unlabeled instances receives any evidential support from the labeled easier instances, GML employs the existing unsupervised method of LEX-SYN [62] to label the remaining instances.

In the comparative study, we report the average and standard deviation of accuracy over ten runs. Our implementation codes of GML have been made open-source available [9].

6.2. Comparative Evaluation

In the comparative study, we set m=20, k=3, d_f=d_f^*=0.4, and $d_{f'}$=$d_{f'}^*$=0.1 for GML. Our sensitivity evaluation in Subsection 6.4 has shown that the performance of GML is, to a large extent, insensitive to the parameter setting.

The detailed evaluation results for ACSA and ATSA are presented in Table 4 and 5 respectively. Note that VADER can not be directly applied on Chinese data, therefore we only report its performance on English data. The

Table 5: Accuracy comparison for ATSA on benchmark datasets.

Model	**LAP16**	**RES16**	**LAP15**	**RES15**
LEX-SYN	67.01%	79.90%	76.55%	75.82%
VADER	67.22%	79.39%	77.11%	78.50%
AT-LSTM	(75.49 ± 1.22) %	(88.11± 0.39)%	(80.08 ± 0.87)%	(76.99 ± 1.53)%
ATAE-LSTM	(76.91 ± 0.50) %	(85.44± 0.95)%	(78.65 ± 0.64)%	(74.98 ± 0.98)%
GCAE	**(80.21 ± 0.83)%**	**(90.07± 0.47)%**	(80.26 ± 0.82)%	(78.31 ± 0.64)%
IAN	(78.04 ± 0.43) %	(87.50± 0.44)%	(79.43 ± 0.81)%	(78.34 ± 1.02)%
RAM	(80.21 ± 1.26) %	(87.74 ± 0.45)%	(80.49 ± 0.88)%	(77.61 ± 1.04)%
AOA	(78.04 ± 0.74) %	(87.80 ± 0.47)%	(81.13 ± 0.40)%	(78.88 ± 0.58)%
TNet	(79.16 ± 1.10) %	(86.99± 0.49)%	(79.06 ± 0.79)%	(76.37 ± 0.89)%
ASVAET(IAN)	(78.71 ±0.56) %	(89.36 ±0.32) %	(81.99 ± 0.64) %	(78.96 ± 0.62) %
GML	80.13 ± 0.31 %	(85.54 ± 0.64)%	**(82.48 ± 0.44)%**	**(80.58± 0.22)%**

[9] https://chenbenben.org/gml.html

Table 6: Performance comparison between GML and SenHint: polarity detection accuracy on hard instances.

	ACSA						ATSA			
	PHO16	CAM16	LAP16	RES16	LAP15	RES15	LAP16	RES16	LAP15	RES15
GML	**62.65 %**	**68.54 %**	**75.95%**	**77.57 %**	**78.63 %**	**73.70 %**	**80.57 %**	**86.34 %**	**76.27 %**	**80.87 %**
SenHint	60.82 %	66.75 %	73.77 %	76.76 %	75.38 %	72.17 %	76.14 %	83.06 %	70.17 %	76.39 %

best result achieved on each dataset is also highlighted in the table. We can see that GML consistently outperforms the unsupervised alternatives, LEX-SYN and VADER (their performance difference is less than 1% in most cases), by considerable margins on all the test datasets. For ACSA, the improvement margins on *PHO16*, *RES16* and *LAP15* are around 7-9%; the margins on *LAP16* is even larger at more than 10%. For ATSA, it achieves the improvements of more than 10% on LAP16 and around 5% on both RES16 and LAP15. Due to the widely recognized challenge of sentiment analysis, the achieved improvements can be deemed very considerable.

Furthermore, it can be observed that the performance of GML is highly competitive compared with the supervised DNN techniques. Except GCAE, GML achieves overall better performance than all the other DNN models on both ACSA and ATSA. For instance, for ACSA, GML beats both AT-LSTM and ATAE-LSTM in performance on five out of totally six datasets. For ATSA, GML achieves the best performance on two out of totally four datasets; except GCAE, it outperforms all the other DNN model on at least three out of the four datasets. GML even beats GCAE on the ACSA task of *LAP15* and the ATSA tasks of *LAP15* and *RES15*; their performance on the other datasets are close. It can also be observed that the semi-supervised model of ASVAET can only improve the performance of DNN by small margins (less than 2% in most cases). Therefore, the performance of GML is similarly competitive with the semi-supervised approach. *It is worthy to point out that unlike the supervised and semi-supervised DNN models, GML does not use any labeled training data provided in the benchmark.* These experimental results evidently demonstrate the efficacy of GML.

To further demonstrate the effectiveness of gradual inference, we have also separately compared GML with a trimmed version of SenHint [52, 53], which does not encode the influence of DNN output. Both GML and SenHint begin with the same set of easy instances; they also extract the same sets of word and relational features. Since GML and SenHint have the same performance on easy instances, we only compare their performance on hard instances. Their comparative evaluation results have been presented in Table 6, in which percentage values represent the accuracies of polarity detection achieved on hard instances. It can be observed that GML achieves better performance than SenHint on all test cases of ACSA and ATSA. These evaluation results clearly demonstrate the efficacy of gradual inference used by GML.

6.3. Evaluation of Easy Instance Labeling

In this subsection, we first evaluate the performance of the proposed technique for easy instance labeling, and then its effect on the performance of GML by a comparative study. We have compared our proposed technique with VADER, which has been empirically shown to perform slightly better than LEX-SYN. In our setting, VADER considers an instance as easy if the absolute value of its sentiment score is more than a threshold (e.g., 0.4 and 0.5). Note that the overall sentiment strength for VADER is a float within the range [-1.0, 1.0].

The detailed evaluation results on the ACSA and ATSA tasks are presented in Table 7. It can be observed that

1. A considerable portion of aspect polarities in the test datasets (varying from 48% to 60%) can be identified by our proposed technique as easy, and its accuracy is always high at more than 90%;
2. Compared with our proposed technique, VADER with the threshold of 0.4 (i.e., VADER($thres$=0.4)) can identify more easy instances but with considerably lower accuracy;
3. Compared with our proposed technique, VADER with the threshold of 0.5 (i.e., VADER($thres$=0.5)) can only identify less easy instances, and its accuracy is also lower in most cases.

We have also evaluated the effect of different techniques on the performance of GML. The detailed evaluation results are presented in Table 8, in which GML-vader0.4 (resp. GML-vader0.5) denotes GML that identifies easy instances using VADER with the threshold of 0.4 (resp. 0.5). It can be observed that compared with GML-vader0.4 and GML-vader0.5, GML achieves better performance on seven out of totally eight datasets. Our experimental results have clearly validated the efficacy of our proposed technique for easy instance labeling.

Table 7: Evaluation of easy instance labeling: *Prop* and *Acc* denote the proportion and achieved accuracy of identified easy instances respectively.

		ACSA				ATSA			
		LAP16	**RES16**	**LAP15**	**RES15**	**LAP16**	**RES16**	**LAP15**	**RES15**
VADER	Prop	48.87%	62.34%	51.50%	58.21%	50.52%	69.54%	54.78%	62.57%
(*thres*=0.4)	Acc	85.29%	90.51%	87.02%	85.07%	86.36%	93.69%	88.01%	90.80%
VADER	Prop	33.82%	48.63%	36.98%	46.48%	35.91%	56.59%	41.65%	50.48%
(*thres*=0.5)	Acc	84.65%	93.47%	89.72%	87.24%	84.88%	95.82%	89.64%	92.02%
Our	Prop	48.07%	56.85%	55.41%	48.83%	49.06%	60.30%	56.66%	51.44%
approach	Acc	92.80%	92.64%	94.59%	92.37%	95.32%	93.00%	95.36%	94.03%

Table 8: Performance comparision between GML-vader0.4, GML-vade0.5 and GML.

	ACSA				ATSA			
	LAP16	**RES16**	**LAP15**	**RES15**	**LAP16**	**RES16**	**LAP15**	**RES15**
GML-vader0.4	76.35%	83.29%	79.33%	75.83%	75.37%	**87.13%**	76.59%	79.04%
GML-vader0.5	74.35%	82.86%	75.46%	76.55%	71.69%	86.05%	74.71%	76.47%
GML	**79.84%**	**85.31%**	**83.94%**	**78.57%**	**80.13%**	85.54%	**82.48%**	**80.58%**

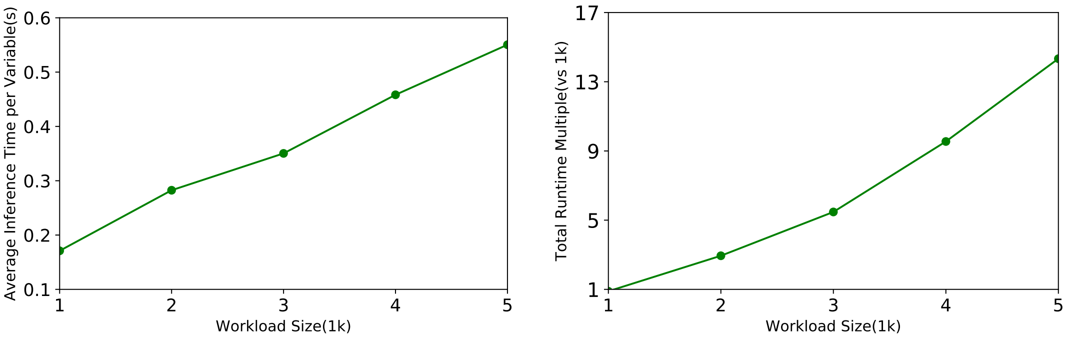
6.4. Sensitivity Evaluation

Table 9: Sensitivity evaluation over ACSA tasks.

		PHO16	**CAM16**	**LAP16**	**RES16**	**LAP15**	**RES15**
w.r.t ***m*** (k = 3)	m = 10	75.91%	81.54%	79.95%	83.93%	84.24%	79.39%
	m = 20	76.14%	81.41%	79.84%	85.31%	83.94%	78.57%
	m = 30	76.26%	81.25%	80.08%	84.79%	83.94%	79.47%
	m = 40	75.61%	80.96%	80.08%	84.76%	84.03%	79.11%
w.r.t ***k*** (m = 20)	k = 1	77.28%	81.58%	79.95%	84.99%	83.06%	79.81%
	k = 3	76.14%	81.41%	79.84%	85.31%	83.94%	78.57%
	k = 5	75.69%	80.91%	77.90%	85.11%	82.97%	79.08%
	k = 7	75.92%	80.54%	80.05%	85.08%	83.16%	78.69%
	$d_{f'}$ d_f	**PHO16**	**CAM16**	**LAP16**	**RES16**	**LAP15**	**RES15**
w.r.t $\boldsymbol{d_{f'}}$ and $\boldsymbol{d_f}$ (m = 20 k = 3)	0.1 0.2	76.60%	81.25%	79.81%	82.91%	84.22%	79.64%
	0.1 0.3	76.67%	80.79%	79.76%	84.53%	83.96%	78.64%
	0.1 0.4	76.14%	81.41%	79.84%	85.31%	83.94%	78.57%
	0.2 0.3	76.45%	80.91%	79.73%	84.82%	84.08%	78.97%
	0.2 0.4	76.41%	81.08%	79.89%	84.59%	84.01%	78.72%
	0.2 0.5	76.26%	81.41%	79.81%	85.05%	84.12%	79.69%
	0.3 0.4	75.95%	80.96%	79.79%	84.62%	83.99%	78.92%
	0.3 0.5	76.22%	80.91%	80.16%	85.25%	83.64%	78.58%

In the sensitivity evaluation, we first vary the values of the parameters m and k as shown in Algorithm 1, which respectively denote the number of candidate variables selected for approximate entropy ranking and the number of candidate variables selected for factor graph inference. We set m=10, 20, 30, 40, and k=1, 3, 5, 7. We then vary the values of the parameters, d_f, $d_{f'}$, d^*_f and $d^*_{f'}$, which denote the degree of uncertainty of word and relational features as shown in Eq. 7, 8, 12 and 13. We set d_f=d^*_f, $d_{f'}$ = $d^*_{f'}$, and $d_{f'}$ < d_f <= 0.5. While evaluating GML's sensitivity to a particular parameter, we fix any other parameter to the same value.

The detailed evaluation results on ACSA are presented in Table 9. Since the standard deviations of accuracy are very similar under different parameter setting, we only report the averages of accuracy in the table. The evaluation results on ATSA are similar, thus omitted here. It can be observed that the performance of GML only fluctuate slightly (≤ 1% in most cases) with different parameter settings. It is noteworthy that the performance of GML does not fluctuate much with various values of m and k. Since most of GML's runtime is spent on factor graph inference, reducing the value of k can effectively improve efficiency. Our experiments show that even with k taking the minimal value of 1, the performance of GML only changes marginally. We also have the similar observation



(a)

(b)

Figure 3: Scalability evaluation.

on the parameter setting of d_f and $d_{f'}$. Various value combinations of d_f and $d_{f'}$ can only result in very marginal performance fluctuations. Our experimental results have clearly shown that the performance of GML is, to a large extent, insensitive to the parameter settings. They bode well for its applicability in real scenarios.

6.5. Scalability Evaluation

In this section, we evaluate the scalability of the proposed scalable approach for GML. We have generated the restaurant workloads with different sizes by retrieving the reviews from Yelp. The workload size varies from 1000 to 5000. All the algorithmic parameters are set to the same values for different workloads. In the experiments, we set m=20, k=3, d_f=d^*_f=0.4, and $d_{f'}$=$d^*_{f'}$=0.1. The detailed evaluation results in terms of runtime are presented in Figure 3. We have observed that most of the runtime is spent on factor graph inference. Even though the total number of extracted features in a task may be large, the number of features a single instance has is usually quite limited. As a result, the size of the subgraph constructed for scalable factor inference on an unlabeled variable generally increases linearly with workload size. Accordingly, the average computational cost of the scalable GML spent on each unlabeled variable increases nearly linearly with workload size. Therefore, as shown in Figure 3, the proposed scalable approach scales well with workload size.

7. Conclusion

In this paper, we have proposed a technical solution for the task of ALSA based on the recently proposed paradigm of gradual machine learning. It begins with some easy instances in an ALSA task, and then gradually labels the more challenging instances based on iterative factor graph inference without any human intervention. Our empirical study on the benchmark datasets has validated the efficacy of the proposed solution.

Our research on gradual machine learning is an ongoing effort. Future work can be pursued on several fronts. Even though GML has been proposed as unsupervised learning approach, human work can be potentially integrated into its process for improved performance. An interesting open challenge is then how to effectively improve the performance of gradual machine learning for ALSA with the minimal effort of human intervention, which include but are not limited to manually labeling some instances. It is also interesting to develop the solution of gradual machine learning for the challenging classification tasks other than entity resolution and sentiment analysis.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2018YFB1003400), National Natural Science Foundation of China (61732014 and 61672432), Fundamental Research Funds for the Central Universities (3102019DX1004) and Natural Science Basic Research Plan in Shaanxi Province of China (2018JM6086).

References

[1] K. Schouten, F. Frasincar, Survey on aspect-level sentiment analysis, IEEE Transactions on Knowledge and Data Engineering 28 (3) (2016) 813–830.

[2] W. Xue, T. Li, Aspect based sentiment analysis with gated convolutional networks, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL, Melbourne, Australia, 2018, pp. 2514–2523.

[3] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based LSTM for aspect-level sentiment classification, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP, Austin, Texas, USA, 2016, pp. 606–615.

[4] S. Ruder, P. Ghaffari, J. G. Breslin, A hierarchical model of reviews for aspect-based sentiment analysis, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP, Austin, Texas, USA, 2016, pp. 999–1005.

[5] S. Kiritchenko, X. Zhu, C. Cherry, S. Mohammad, NRC-Canada-2014: detecting aspects and sentiment in customer reviews, in: Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING, Dublin, Ireland, 2014, pp. 437–442.

[6] J. Saias, Sentiue: target and aspect based sentiment analysis in SemEval-2015 Task 12, in: Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT, Denver, Colorado, USA, 2015, pp. 767–771.

[7] B. Hou, Q. Chen, J. Shen, X. Liu, P. Zhong, Y. Wang, Z. Chen, Z. Li, Gradual machine learning for entity resolution, in: Proceedings of the Web Conference, WWW 2019, San Francisco, CA, USA, 2019, pp. 3526–3530.

[8] B. Hou, Q. Chen, Y. Wang, P. Zhong, M. H. M. Ahmed, Z. Chen, Z. Li, Gradual machine learning for entity resolution, IEEE Transactions on Knowledge and Data Engineering (TKDE) (2020) , accepted, to appear.
URL `https://arxiv.org/abs/1810.12125`

[9] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering 22 (10) (2010) 1345–1359.

[10] Z. Chen, B. Liu, Lifelong machine learning, second edition, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2018.

[11] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: Proceedings of the 26th Annual International Conference on Machine Learning, ICML, Montreal, Quebec, Canada, 2009, pp. 41–48.

[12] R. Mihalcea, Co-training and self-training for word sense disambiguation, in: Proceedings of the 8th Conference on Computational Natural Language Learning, CoNLL, Boston, Massachusetts, USA, 2004, pp. 33–40.

[13] P. Christen, Automatic record linkage using seeded nearest neighbour and support vector machine classification, in: Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining, KDD, Las Vegas, Nevada, USA, 2008, pp. 151–159.

[14] T. Mullen, N. Collier, Sentiment analysis using support vector machines with diverse information sources, in: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP, Barcelona, Spain, 2004, pp. 412–418.

[15] A. Blum, T. M. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the 17th Annual Conference on Computational Learning Theory, COLT, Madison, Wisconsin, USA, 1998, pp. 92–100.

[16] K. Bellare, S. Iyengar, A. G. Parameswaran, V. Rastogi, Active sampling for entity matching, in: Proceedings of 18th ACM International Conference on Knowledge Discovery and Data Mining, KDD, Beijing, China, 2012, pp. 1131–1139.

[17] K. Nigam, A. McCallum, S. Thrun, T. M. Mitchell, Text classification from labeled and unlabeled documents using EM, Machine Learning 39 (2) (2000) 103–134.

[18] H. He, X. Sun, A unified model for cross-domain and semi-supervised named entity recognition in chinese social media, in: Proceedings of the 31st Conference on Artificial Intelligence, AAAI, San Francisco, California, USA, 2017, pp. 3216–3222.

[19] R. Caruana, Multitask learning, Machine Learning 28 (1) (1997) 41–75.

[20] M. P. Kumar, B. Packer, D. Koller, Self-paced learning for latent variable models, in: Advances in Neural Information Processing Systems 23, NIPS, Vancouver, British Columbia, Canada., 2010, pp. 1189–1197.

[21] J. Kivinen, A. J. Smola, R. C. Williamson, Online learning with kernels, in: Advances in Neural Information Processing Systems 14, NIPS, Vancouver, British Columbia, Canada, 2001, pp. 785–792.

[22] J. C. Schlimmer, R. H. Granger, Incremental learning from noisy data, Machine Learning 1 (3) (1986) 317–354.

[23] E. Cambria, Affective computing and sentiment analysis, IEEE Intelligent Systems 31 (2) (2016) 102–107.

[24] K. Ravi, V. Ravi, A survey on opinion mining and sentiment analysis: tasks, approaches and applications, Knowledge Based System 89 (2015) 14–46.

[25] X. Zhang, J. J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: Advances in Neural Information Processing Systems., 2015, pp. 649–657.

[26] R. Johnson, T. Zhang, Deep pyramid convolutional neural networks for text categorization, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 562–570.

[27] Q. Qian, M. Huang, J. Lei, X. Zhu, Linguistically regularized LSTM for sentiment classification, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL, Vancouver, Canada, 2017, pp. 1679–1689.

[28] N. F. F. da Silva, L. F. S. Coletta, E. R. Hruschka, A survey and comparative study of tweet sentiment analysis via semi-supervised learning, ACM Computing Surveys 49 (1) (2016) 1–26.

[29] N. F. F. da Silva, L. F. Coletta, E. R. Hruschka, E. R. Hruschka Jr, Using unsupervised information to improve semi-supervised tweet sentiment classification, Information Sciences 355 (2016) 348–365.

[30] A. Hussain, E. Cambria, Semi-supervised learning for big social data analysis, Neurocomputing 275 (2018) 1662–1673.

[31] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, Adaptive recursive neural network for target-dependent twitter sentiment classification, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL, Baltimore, MD, USA, 2014, pp. 49–54.

[32] D. Tang, B. Qin, X. Feng, T. Liu, Effective LSTMs for target-dependent sentiment classification, in: Proceedings of the 26th International Conference on Computational Linguistics, COLING, Osaka, Japan, 2016, pp. 3298–3307.

[33] S. Wang, S. Mazumder, B. Liu, M. Zhou, Y. Chang, Target-sensitive memory networks for aspect sentiment classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL, Melbourne, Australia, 2018, pp. 957–967.

[34] X. Li, L. Bing, W. Lam, B. Shi, Transformation networks for target-oriented sentiment classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL, Melbourne, Australia, 2018, pp. 946–956.

[35] V. Mnih, N. Heess, A. Graves, K. Kavukcuoglu, Recurrent models of visual attention, in: Advances in Neural Information Processing Systems 27, NIPS, Montreal, Quebec, Canada, 2014, pp. 2204–2212.

[36] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, Attention-based models for speech recognition, in: Advances in Neural Information Processing Systems 28, NIPS, Montreal, Quebec, Canada, 2015, pp. 577–585.

[37] T. Luong, H. Pham, C. D. Manning, Effective approaches to attention-based neural machine translation, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP, Lisbon, Portugal, 2015, pp. 1412–1421.

[38] O. Firat, K. Cho, Y. Bengio, Multi-way, multilingual neural machine translation with a shared attention mechanism, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT, San Diego California, USA, 2016, pp. 866–875.

[39] X. He, D. Golub, Character-level question answering with ttention, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, 2016, pp. 1598–1607.

[40] J. Cheng, S. Zhao, J. Zhang, I. King, X. Zhang, H. Wang, Aspect-level sentiment classification with HEAT (HiErarchical ATtention) network, in: Proceedings of the 2017 ACM Conference on Information and Knowledge Management, CIKM, Singapore, 2017, pp. 97–106.

[41] B. Wang, W. Lu, Learning latent opinions for aspect-level sentiment classification, in: Proceedings of the 32nd Conference on Artificial Intelligence, AAAI, New Orleans, Louisiana, USA, 2018.

[42] D. Ma, S. Li, X. Zhang, H. Wang, Interactive attention networks for aspect-level aentiment classification, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI, Melbourne, Australia, 2017, pp. 4068–4074.

[43] P. Chen, Z. Sun, L. Bing, W. Yang, Recurrent attention network on memory for aspect sentiment analysis, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP, Copenhagen, Denmark, 2017, pp. 452–461.

[44] B. Huang, Y. Ou, K. M. Carley, Aspect level sentiment classification with attention-over-attention neural networks, in: Social, Cultural, and Behavioral Modeling - 11th International Conference, SBP-BRiMS, Washington, DC, USA, 2018, pp. 197–206.

[45] R. He, W. S. Lee, H. T. Ng, D. Dahlmeier, Effective attention modeling for aspect-level sentiment classification, in: Proceedings of the 27th International Conference on Computational Linguistics, COLING, Santa Fe, New Mexico, USA, 2018, pp. 1121–1131.

[46] Q. Liu, H. Zhang, Y. Zeng, Z. Huang, Z. Wu, Content attention model for aspect based sentiment analysis, in: Proceedings of the 2018 Web Conference, WWW, Lyon, France, 2018, pp. 1023–1032.

[47] F. Fan, Y. Feng, D. Zhao, Multi-grained attention network for aspect-level sentiment classification, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP, Brussels, Belgium, 2018, pp. 3433–3442.

[48] Y. Ma, H. Peng, T. Khan, E. Cambria, A. Hussain, Sentic LSTM: a hybrid network for targeted aspect-based sentiment analysis, Cogn. Comput. 10 (4) (2018) 639–650.

[49] Y. Song, J. Wang, T. Jiang, Z. Liu, Y. Rao, Attentional encoder network for targeted sentiment classification, arXiv preprint arXiv:1902.09314.

[50] B. Zeng, H. Yang, R. Xu, W. Zhou, X. Han, Lcf: A local context focus mechanism for aspect-based sentiment classification, Applied Sciences 9 (16) (2019) 3389.

[51] X. Cheng, W. Xu, T. Wang, W. Chu, W. Huang, K. Chen, J. Hu, Variational semi-supervised aspect-term sentiment analysis via transformer, in: Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, 2019, pp. 961–969.

[52] Y. Wang, Q. Chen, X. Liu, M. H. M. Ahmed, Z. Li, W. Pan, H. Liu, Senhint: A joint framework for aspect-level sentiment analysis by deep neural networks and linguistic hints, in: Demonstration of The Web Conference 2018, WWW 2018, Lyon , France, 2018, pp. 207–210.

[53] Y. Wang, Q. Chen, M. Ahmed, Z. Li, W. Pan, H. Liu, Joint inference for aspect-level sentiment analysis by deep neural networks and linguistic hints, IEEE Transactions on Knowledge and Data Engineering (2019) ,accepted, online available.

[54] Y. Xia, E. Cambria, A. Hussain, H. Zhao, Word polarity disambiguation using bayesian model and opinion-level features, Cognitive Computation 7 (3) (2015) 369–380.

[55] T. Singh, M. Kumari, Bbncpd-bayesian belief network based contextual polarity disambiguation in sentiment analysis.

[56] X. Ding, B. Liu, P. S. Yu, A holistic lexicon-based approach to opinion mining, in: Proceedings of the International Conference on Web Search and Web Data Mining, WSDM, Palo Alto, California, USA, 2008, pp. 231–240.

[57] C. J. Hutto, E. Gilbert, VADER: a parsimonious rule-based model for sentiment analysis of social media text, in: Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM, Ann Arbor, Michigan, USA, 2014.

[58] S. Abbasi Moghaddam, Aspect-based opinion mining in online reviews, Ph.D. Thesis, Simon Fraser University, 2013.

[59] M. Marcus, B. Santorini, M. A. Marcinkiewicz, Building a large annotated corpus of english: The penn treebank.

[60] J. Yang, D. Xu, Evidential reasoning rule for evidence combination, Artificial Intelligence 205 (2013) 1–29.

[61] K. Li, X. Zhou, D. Z. Wang, C. E. Grant, A. Dobra, C. Dudley, In-database batch and query-time inference over probabilistic graphical models using UDA-GIST, The VLDB Journal 26 (2) (2017) 177–201.

[62] T. Álvarez-López, J. Juncal-Martínez, et al., GTI at SemEval-2016 Task 5: SVM and CRF for aspect detection and unsupervised aspect-based sentiment analysis, in: Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT, San Diego, CA, USA, 2016, pp. 306–311.

[63] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, SemEval-2015 Task 12: aspect based sentiment analysis, in: Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT, Denver, Colorado, USA, 2015, pp. 486–495.

[64] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. Al-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. D. Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. V. Loukachevitch, E. Kotelnikov, N. Bel, S. M. J. Zafra, G. Eryigit, SemEval-2016 Task 5: aspect based sentiment analysis, in: Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT, San Diego, CA, USA, 2016, pp. 19–30.