Weakly Supervised Gradual Machine Learning for Sentence-Level Sentiment Analysis

Jing Su, Qun Chen, Yanyan Wang, Wei Pan, Zhanhuai Li

[a]*School of Computer Science, Northwestern Polytechnical University, Xi'an, Shaanxi, China*

Abstract

The task of Sentence-Level Sentiment Analysis (SLSA) aims to detect the general sentiment polarity of an entire sentence. The state-of-the-art performance of SLSA has been achieved by DNN models. However, their efficacy depends on large quantities of labeled training data, which may not be readily available in real scenarios. In this paper, we propose a weakly supervised approach for SLSA based on the paradigm of Gradual Machine Learning (GML). The proposed solution begins with only a few labeled samples, and then gradually labels target instances in the order of increasing hardness by iterative inference in a factor graph. Specifically, it performs the task by gradual phases, each of which selects only a proportion of target instances for labeling by the measure of evidential certainty. In each phase, it labels the selected instances in a self-training way with GML as the base model. It randomly selects a portion of pseudo-labels to fine-tune deep models for feature extraction, and then constructs both unary and binary monotonous factors in a factor graph based on the extracted features to fulfill gradual learning. We have empirically evaluated the performance of the proposed solution on real benchmark data. Our extensive experiments show that it performs considerably better than the existing alternatives in the weakly supervised settings. Furthermore, using only a small proportion of labeled training data ($\leq$ 5%), it achieves highly competitive performance compared with the state-of-the-art deep models trained with full training data.

Keywords: Gradual Machine Learning, Sentence-Level Sentiment Analysis, Weak Supervision, Factor Graph Inference

1. Introduction

The task of Sentence-Level Sentiment Analysis (SLSA) aims to analyze a reviewer's general emotion expressed in a sentence [1]. Playing an important role in informed decision making, SLSA has been extensively studied in the literature [2, 3, 4]. However, it remains very challenging due to the complexity of natural languages. To make things worse, a sentence may express conflicting polarities towards different aspects of an entity. Consider the running example shown in Figure1. The sentence of s_2 expresses a positive polarity towards key pad, but a negative polarity towards power key. As a result, its overall emotion is negative. The state-of-the-art performance of SLSA has been achieved by various deep models, especially pre-trained language models (e.g., RoBERTa[5], ALBERT[6], BART[7], DeBERTa[8] and ERNIE3.0[9]). However, these models' efficacy depends on sufficient labeled training data. Unfortunately, in real applications, labeled training data usually require extensive manual effort, thus may not be readily available.

Therefore, there is a need to investigate SLSA in the circumstance where only a limited number of labeled training data are available. We note that previous work has successfully leveraged the non-i.i.d (Independent and Identically Distributed) paradigm of Gradual Machine Learning (GML) for the task of Aspect-Level Sentiment Analysis (ALSA) [10]. Unlike SLSA, ALSA needs to analyze the sentimental polarities expressed towards the finer-grained aspects of an entity in a sentence. Generally, GML begins with some easy instances, which can usually be accurately

Email addresses: sujing@mail.nwpu.edu.cn (Jing Su), chenbenben@nwpu.edu.cn (Qun Chen), wangyanyan@mail.nwpu.edu.cn (Yanyan Wang), panwei1002@nwpu.edu.cn (Wei Pan), lizhh@nwpu.edu.cn (Zhanhuai Li)

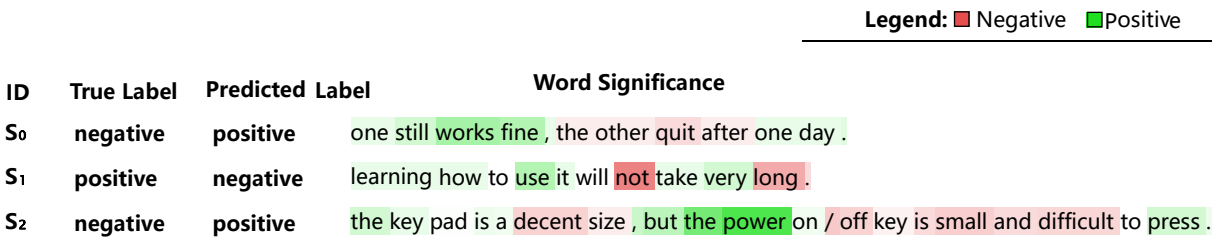

Figure 1: The running examples of SLSA

labeled by the machine, and then gradually labels more challenging instances in the order of increasing hardness by iterative inference in a factor graph. Without exploiting labeled training data, the current unsupervised GML solution for ALSA relies on sentiment lexicons and explicit polarity relations indicated by discourse structures to enable knowledge conveyance. It has been empirically shown that unsupervised GML can achieve competitive performance compared with many supervised deep models. Unfortunately, unsupervised sentiment features are usually incomplete and noisy. Therefore, the performance of gradual learning is still limited by inaccurate and insufficient knowledge conveyance.

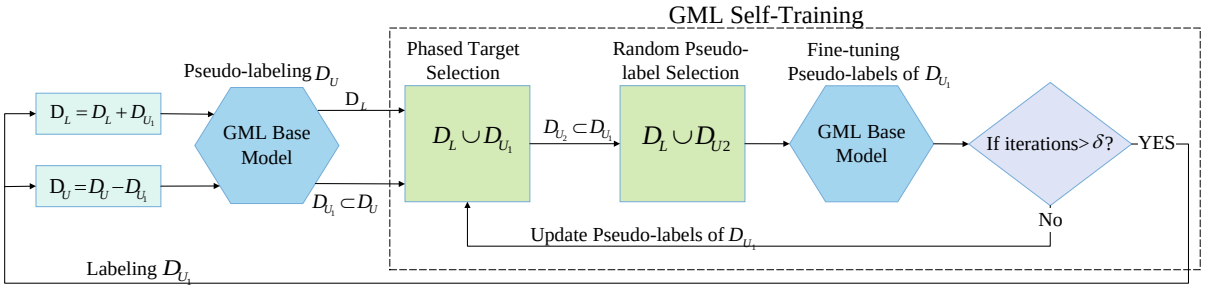

Figure 2: The framework of weakly supervised GML for SLSA.

In this paper, we propose a weakly supervised solution based on the paradigm of GML for the task of SLSA. Since only a few labeled samples are supposed to be available, the challenge is how to generate effective feature representations to facilitate gradual knowledge conveyance. We sketch the framework of the proposed solution in Figure 2. It performs labeling by gradual phases, each of which is supposed to label only a proportion of target instances. In each phase, it selects target instances in the decreasing order of evidential certainty such that the distribution difference between the selected instances and the labeled instances is minimal. Then, it labels the selected instances in a self-training way with GML as the base model, which has been shown in Figure 3. Specifically, it randomly selects a proportion of pseudo-labels to fine-tune the deep models for feature extraction, and then constructs both unary and binary monotonous factors in a factor graph based on the extracted features to fulfill gradual learning. GML self-training is supposed to be conducted in multiple iterations. This process of partition selection and self-training is repeatedly invoked until all the instances in a target workload are labeled.

It can be observed that to compensate for the scarcity of labeled training data, the proposed framework gradually leverages newly generated labels to fine-tune feature representations. However, as a gradual learning approach, it is different from the traditional i.i.d semi-supervised approach in two important ways: 1) to reduce error propagation, it gradually generates labels in the order of increasing hardness as measured by GML evidential certainty; 2) instead of directly using newly generated labels to fine-tune classifiers, it uses them to fine-tune deep models for feature extraction, and then leverages the extracted features to facilitate gradual learning. Since the pre-trained language models (e.g., RoBERTa [5] and SimCSE [11]) can extract implicit features more effectively than the traditional hand-crafted solutions, we leverage them to extract polarity-sensitive features as shown in Figure 3. Specifically, it iteratively fine-tunes the RoBERTa model, the SOTA model for text classification [5], with newly labeled data as well as labeled training data to give polarity predictions, which are then modeled as unary factors in a factor graph. Additionally, it

fine-tunes the SimCSE model to generate a separate polarity-sensitive embedding space, and then constructs binary polarity similarity relations based on the k-nearest neighborhood in the embedding space. The relational features are modeled as binary factors in a factor graph.

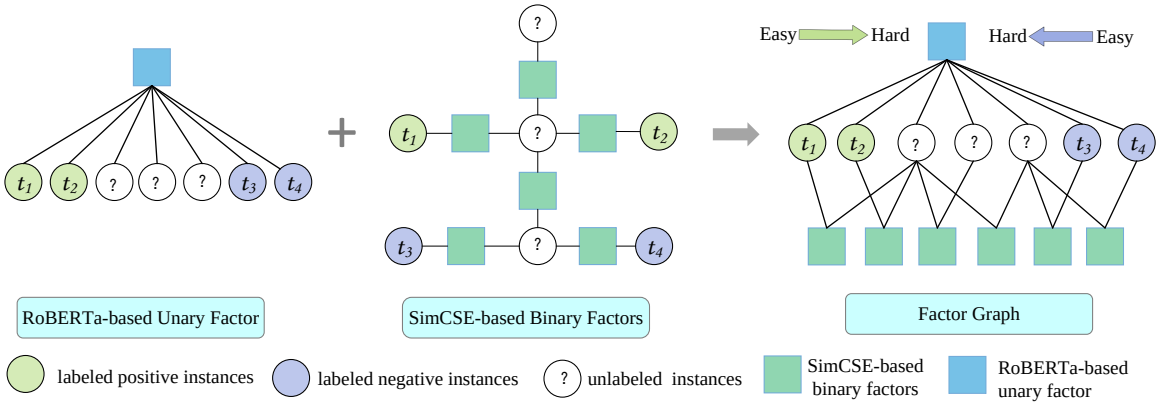

Figure 3: The base GML model.

The proposed framework can process unlabeled training data in the same way as unlabeled target data. It thus provides a unified way to leverage both of them for improved learning. The major contributions of this paper can be summarized as follows:

- We propose a weakly supervised framework based on GML for SLSA, which can effectively leverage a limited number of labeled training data to improve gradual learning;
- We propose a phased approach to perform gradual labeling, and present a self-training technique based on the GML model to fulfill phased labeling.
- We empirically validate the efficacy of the proposed solution by an empirical study on real benchmark data. Our extensive experiments show that it outperforms the existing alternatives by considerable margins, and the margins tend to be more considerable when fewer labeled samples are provided. Furthermore, using only a small proportion of labeled training data ($\leq$ 5%), it achieves highly competitive performance compared with the SOTA deep models trained with full training data.

The rest of this paper is organized as follows: Section 2 discusses more related work. Section 3 introduces the GML framework. Section 4 presents the proposed solution. Section 5 empirically evaluates the proposed solution. Finally, we conclude this paper with some thoughts on future work in Section 6.

2. Related work

Sentiment analysis at different granularity levels, including document, sentence, and aspect levels, has been extensively studied in the literature [12, 13, 14]. At the sentence (resp. document) level, its goal is to detect the general polarity of the entire sentence (resp. document) [1, 15, 9, 16]. In contrast, aspect-level sentiment analysis needs to identify the polarities expressed towards the finer-grained aspects mentioned in text [17, 10, 18, 19].

Various deep neural networks have been developed for the task of SLSA, including contextual sentiment neural network [14], GLRNN [20], CNN-BiLSTM [21] and SR-LSTM [22]. More recent work focused on how to leverage pre-trained language models (e.g., BERT) for improved prediction accuracy, including RoBERTa [5], ALBERT [6], BART [7],DeBERTa [8], DistilBERT [23] and ERNIE 3.0 [9]. It is noteworthy that the efficacy of these deep models usually depends on large quantities of accurately labeled training data. Their performance usually deteriorates considerably if provided with only a limited number of labeled training data. Furthermore, their performance may be very sensitive to parameter settings (e.g., random seed, batch size and iterations) in the weak-supervised circumstance.

To deal with performance degradation due to insufficient training data, semi-supervision has been extensively studied for the broader problem of text classification [24]. Its main purpose is to leverage unlabeled training data for improved learning. The existing work on semi-supervised text classification can be broadly categorized into two categories: 1) self-training, which constructs multiple independent models (e.g., teacher and student) to capture distribution difference between labeled and unlabeled data [25, 26, 27, 28, 29, 30]; 2) co-training, which designs specific loss functions for unlabeled data to optimize training [31, 32]. The existing self-training approaches typically focused on designing appropriate uncertainty estimation metrics for reliable unlabeled sample selection, or effective teacher-student interactive process [33] [27] [28] [29] [30]. For instance, the work of [33] selected data for pseudo-labeling by BALD, whose objective is to maximize the information gain between predictions and model posterior. The work of [27] introduced two teachers and one student to integrate more feature sources. The authors of [28] proposed two-stage semi-supervised learning process, which leveraged pseudo-labels for classifier initialization and labeled data for classifier fine-tuning. The authors of [29] developed a multi-teacher distillation (MTD) framework to learn class distribution in the intersection set and difference set of predicted models. The authors of [30] presented a new task-specific teacher-student model, which used student to predict pseudo-labels for instances that may not be covered by weak rules, and teacher to learn different weights between student pseudo-labels and weak rule labels. The authors of [26] presented DRIFT, which treated teacher-student as a stackelberg game. In contrast, the existing co-training approaches focused on designing complementary noise-robust loss function. For instance, the authors of [32] introduced the MixTex framework, which could co-train labeled, unlabeled and augmented data by supervised loss and consistency loss. The authors of [31] proposed the COSINE framework, which used contrastive regularization and confidence-based reweighting to suppress error propagation. Finally, there also exist some work on the hybrid approach, which combined self-training and co-training. For instance, the authors of [25] proposed CEST, which integrated certainty-driven sample selection into a teacher-student model.

In this paper, we have built the weak-supervised solution for SLSA based on the non-i.i.d paradigm of gradual machine learning. The GML paradigm was first proposed for the task of entity resolution [34]. Since then, it has also been applied to the task of aspect-level sentiment analysis [10, 19]. However, the existing GML solutions are unsupervised. Without exploiting labeled training data, their performance is usually limited by inaccurate and insufficient knowledge conveyance. In this paper, we focus on how to leverage a limited number of labeled examples for improved gradual learning.

3. The GML Framework

In this section, we illustrate the GML framework by the existing unsupervised GML solution for aspect-level sentiment analysis [10]. As shown in Figure 4, the framework consists of the following three essential steps:

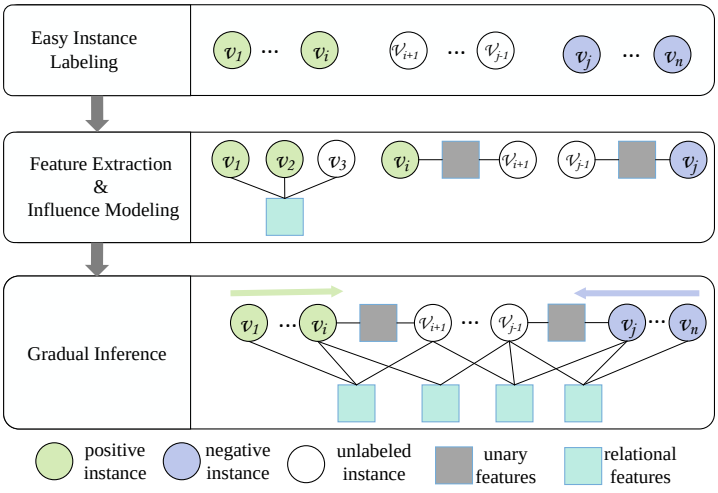

Figure 4: Unsupervised GML Solution for Aspect-level Sentiment Analysis.

Algorithm 1: Scalable Gradual Inference

```
while there exists any unlabeled variable in G do
    V' ← all the unlabeled variables in G;
    for v ∈ 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 ∈ 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 ∈ 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;
```

3.1. Easy instance labeling

Gradual machine learning begins with some easy instances. Therefore, high label accuracy of easy instances is critical for GML's ultimate performance. The existing unsupervised solution for ALSA employs simple user-specified rules to identify non-ambiguous instances as easy ones [10]. Specifically, if a sentence contains some strong positive (resp. negative) sentiment words, but no negation, contrast and hypothetical connectives, it can be reliably reasoned to be positive (resp. negative). *It is noteworthy that since this paper considers SLSA in the weak supervision setting, in which a limited number of training data are supposed to be available, these training data with ground-truth labels can naturally serve as initial easy instances.*

3.2. Feature Extraction and Influence Modeling

Features serve 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 diverse information. For each extracted feature, this step also needs to model its influence over the labels of relevant instances.

The existing unsupervised solution for ALSA presented in [10] relies on sentiment lexicons and explicit polarity relations indicated by discourse structures to enable knowledge conveyance. Specifically, given a sentiment word, *positive* or *negative*, any sentence containing the word is supposed to have the same polarity as the word. Similarly, a similar (resp. opposite) polarity relation between two instances indicates that they are expected to have the same (resp. opposite) polarities. GML models word and relation features as unary and binary factors in a factor graph respectively.

3.3. Gradual Inference

GML fulfills gradual learning by iterative inference on a factor graph, G, which consists of evidence variables representing labeled instances, inference variables representing unlabeled instances and factors representing their features. The values of evidence variables once labeled remain unchanged while the values of inference variables need to be gradually inferred.

Formally, suppose that G consists of a set of evidence variables, $\mathbf{\Lambda}$, a set of inference variables, $\mathbf{V_I}$, and a group of factor functions of variables indicating their correlations, denoted by $\phi_{w_i}(V_i)$. In the case of binary sentiment classification, each variable in the factor graph is a boolean variable indicating polarity, the value of 1 for *positive* and 0 for *negative*. Then, the joint probability distribution over $V = \{\Lambda, V_I\}$ of G can be formulated as

$$P_{\mathbf{w}}(\Lambda, V_I) = \frac{1}{Z_{\mathbf{w}}} \prod_{i=1}^{m} \phi_{w_i}(V_i), \tag{1}$$

where V_i denotes a set of variables, w_i denotes a factor weight, m denotes the total number of factors and $Z_{\mathbf{w}}$ denotes a normalization constant. Factor inference on G learns factor weights by minimizing the negative log marginal likelihood of evidence variables as follows:

$$\hat{\mathbf{w}} = arg\min_{\mathbf{w}} -log \sum_{V_I} P_{\mathbf{w}}(\Lambda, V_I). \tag{2}$$

In each iteration, GML generally chooses to label the inference variable with the highest degree of evidential certainty. Given an inference variable v, GML measures its evidential certainty by the inverse of entropy as follows

$$E(v) = \frac{1}{H(v)} = \frac{1}{-\sum\limits_{i=0,1} P_i(v) \cdot \log_2 P_i(v)}, \tag{3}$$

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 0 or 1. The iteration is repeatedly invoked until all the instances are labeled.

To improve efficiency, GML usually implements gradual inference by a scalable approach as sketched in Algorithm 1. It consists of three steps: measurement of evidential support, approximate ranking of entropy and factor subgraph inference. In the first step, it selects the top-m unlabeled variables with the most evidential support in G as the inference candidates. For each unlabeled instance, GML measures its evidential support from each feature by the degree of labeling confidence indicated by labeled observations, and then aggregates them based on the Dempster-Shafer theory [1]. Secondly, it approximates entropy estimation by an efficient algorithm on the m candidates and selects only the top-k most promising variables among them for factor inference. Finally, it estimates the probabilities of the finally chosen k variables by factor inference.

4. Weakly Supervised GML Solution

The proposed solution, as shown in Figure 2, consists of multiple phases of gradual learning. In each phase, it selects a subset of target instances in the decreasing order of evidential certainty, and then labels them in a self-training way by the base GML model. In this section, we first present the base GML model in Subsection 4.1, then describe the process of phased target selection in Subsection 4.2, and finally describe how to extract features by deep models in Subsection 4.3.

4.1. Base GML Model

The base GML model, as shown in Figure 3, is composed of variables, which correspond to labeled and unlabeled instances, and factors, which encode the correlations between the labels of variables. It has two types of factors, unary factor and binary factor. The unary factor encodes the polarity probabilities as predicted by the RoBERTa model, while the binary factor encodes the neighborhood-based polarity similarities as predicted by the SimCSE model.

As usual, denoting the unary feature by f_u, we model the influence of f_u over a variable, v, as follows:

$$\varphi_{f_u}(v) = \begin{cases} e^{w_{f_u}(v)} & if\ v = 1; \\ 1 & if\ v = 0; \end{cases} \tag{4}$$

where $w_{f_u}(v)$ denotes the factor weight, and

$$w_{f_u}(v) = \theta_{f_u}(v) \cdot \tau_{f_u} \cdot (x_{f_u}(v) - \alpha_{f_u}), \tag{5}$$

in which $\theta_{f_u}(v)$ denotes the confidence on influence modeling of f_u, $x_{f_u}(v)$ denotes the feature value of v, or the probability as predicted by the RoBERTa model, and τ_{f_u} and α_{f_u} denote the steepness and mid-point of a sigmoid function respectively. In our implementation, as in GML for entity resolution [34], we estimate $\theta_{f_u}(v)$ by the theory of regression error bound [35]. The parameter values of τ_{f_u} and α_{f_u} are however supposed to be continuously optimized based evidential observations in the process of gradual learning.

[1] https://en.wikipedia.org/wiki/Dempster-Shafer_theory

Similarly, the base model defines the binary similarity factor by

$$\varphi_{f_b}(v_i, v_j) = \begin{cases} e^{w_{f_b}(v_i, v_j)} & if\ v_i = v_j; \\ 1 & otherwise; \end{cases} \tag{6}$$

in which f_b denotes a binary feature, v_i and v_j denote the two variables sharing the feature of f_b, $w_{f_b}(v_i, v_j)$ denotes the factor weight, and

$$w_{f_b}(v_i, v_j) = \theta_{f_u}(v_i, v_j) \cdot \tau_{f_b} \cdot (x_{f_b}(v_i, v_j) - \alpha_{f_b}), \tag{7}$$

in which $\theta_{f_u}(v_i, v_j)$ denotes the confidence on binary feature influence modeling, $x_{f_b}(v_i, v_j)$ denotes the vector similarity of v_i and v_j, and τ_{f_b} and α_{f_b} denote the steepness and mid-point of a sigmoid function. Similar to the case of the unary feature, $\theta_{f_u}(v_i, v_j)$ is estimated by the theory of regression error bound, while τ_{f_b} and α_{f_b} are supposed to be continuously optimized in the process of gradual learning.

GML Self-Training. As shown in Figure 2, in each phase, the base GML gives initial predictions on the labels of the selected instances in D_{U_1}. It then randomly selects a proportion of these annotated pseudo-labeled data, along with labeled training data, to fine-tune its predictions. This iterative process is repeatedly invoked to predict the final labels of these selected instances.

In each iteration of self-training, it fine-tunes the RoBERTa and SimCSE models using both labeled training data and randomly selected pseudo-labels, and then updates the base GML model based on the fine-tuned features to renew its predictions. In practical implementation, the proportion of psuedo-labels is suggested to be set within the reasonable range between 30% and 60%, while the required number of iterations is suggested to be set to a small value (2 in our implementation). Our experimental evaluation has shown that only a few iterations (≤ 3) can effectively improve labeling accuracy, and the performance of GML self-training is very robust w.r.t both the proportion of pseudo-labels and the number of iterations.

4.2. Phased Target Selection

Since GML labels instances in the order of increasing hardness as measured by evidential certainty, we naturally select the target instances in the order of gradual learning. To reduce error propagation, we incrementally select target instances such that the resulting target set has the minimal distribution shift from the existing set of labeled data. Since recent work has shown that the simple measure of difference of confidence (DoC) can estimate a classifier's performance change over a variety of shifts more accurately than previous proposals, we use the metric of Doc to quantify distribution shift [36].

Specifically, the metric of DoC is measured by

$$\mu_r = \frac{1}{|D_r|} \sum_{x \in D_r} max(\mathcal{M}(x)), \tag{8}$$

$$\mu_t = \frac{1}{|D_t|} \sum_{x \in D_t} max(\mathcal{M}(x)), \tag{9}$$

$$DoC = \mu_r - \mu_t, \tag{10}$$

where $\mathcal{M}$ denotes a prediction model, $\mathcal{M}(x)$ denotes the predicted probabilities of $\mathcal{M}$ on an instance x, $max(\mathcal{M}(x))$ denotes the maximal probability among them, D_r denotes the set of training data, D_t denotes the set of target data, μ_r denotes the mean of all probabilistic predictions on training data, and similarly, μ_t denotes the mean of all probabilistic predictions on target data.

In terms of SLSA, we estimate the DoC values based on the GML model. In the implementation, we estimate the DoC values at multiple potential split points, and then select the split point with the smallest DoC value to construct the phased set of target instances. We illustrate the process of phased target selection by the CR benchmark dataset, which has totally 755 test samples in total. As shown in Table 1, with the split size of 150, the algorithm estimates the DoC values at five split points, 150, 300, 450, 600 and 755 respectively. Since the split point of 300 has the smallest DoC value of 0.000153, GML would choose the first 300 instances as its set of target instances.

Table 1: An illustrative example of phased target selection on CR.

# Partition	#DoC value
[0, 150]	0.001832246
[0, 300]	0.000153
[0, 450]	0.002038
[0, 600]	0.000523
[0, 755]	0.000371

4.3. Feature Extraction by Deep Models

The proposed solution relies on two types of features, a unary feature and a binary feature, which are extracted by the RoBERTa and SimCSE models respectively, to enable gradual learning.

Unary Feature. Since recent studies have shown that the RoBERTa model achieves the state-of-the-art performance on text classification, we use it for extracting unary polarity predictions. To reduce error propagation in the process of gradual learning, it weights the pseudo-labeled instances in the cross-entropy loss by neighborhood labels. Intuitively speaking, the weighting mechanism assumes that the more same-label neighbors a pseudo-label has, the more reliable it is.

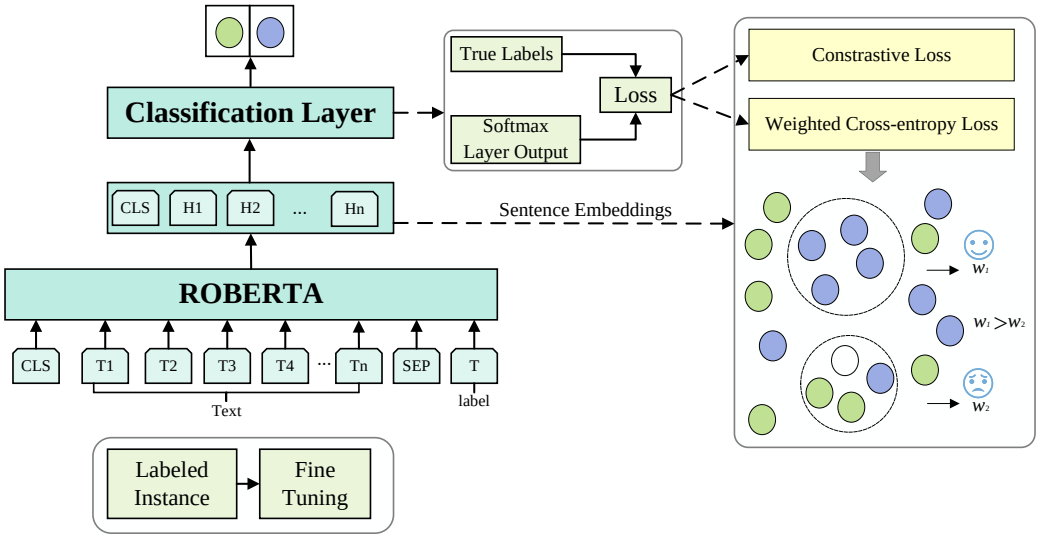

Figure 5: The structure of RoBERTa-based model for unary feature extraction.

The structure of RoBERTa-based model for unary feature extraction is showed in Figure 5. Formally, we define the weighted cross-entropy loss by

$$L_{CE} = \frac{1}{N}\sum_{i=0}^{N} w_i * [Y_i log\hat{Y}_i + (1 - Y_i)log(1 - \hat{Y}_i)], \tag{11}$$

in which N denotes the total number of labeled and pseudo-labeled instances, $\hat{Y}_i$ denotes the probability predicted by the RoBERTa model, Y_i denotes its reference label, and w_i denotes instance weight. If the reference label of an instance is ground-truth, $w_i = 1$; otherwise, we define its weight by

$$w_i = 1 + \frac{a_i \cdot ln(a_i)}{log(B)}, \tag{12}$$

in which B denotes the batch size for regularization (B = 32 in this paper) and a_i denotes the percentage of its k-nearest neighbors having the same label as the instance.

Finally, we define the training objective loss of the ReBERTa-based model by the linear combination of cross-entropy loss and constrastive loss as follows:

$$L = \beta_1 \cdot L_{CE} + \beta_2 \cdot L_{SCL}, \tag{13}$$

in which L_{CE} and L_{SCL} denote the cross-entropy loss and the supervised contrastive loss respectively, and β_1 and β_2 denote their loss weights satisfying $\beta_1 + \beta_2 = 1$.

In practical implementation, we suggest that β_1 is set to be between 0.6 and 0.8. Our experiments have shown that the performance of GML is robust w.r.r the values of β_1 and β_2 provided that they are set within a reasonable range.

Binary Feature. Observing the recent successes of unsupervised sentence embedding learning based on natural language corpus, we leverage the SimCSE model for binary relation extraction. As shown in Figure 6, we add a two-layer linear network, which is represented by MLP (MultiLayer Perceptron) in the figure, to the original SimCSE model for polarity fine-tuning.

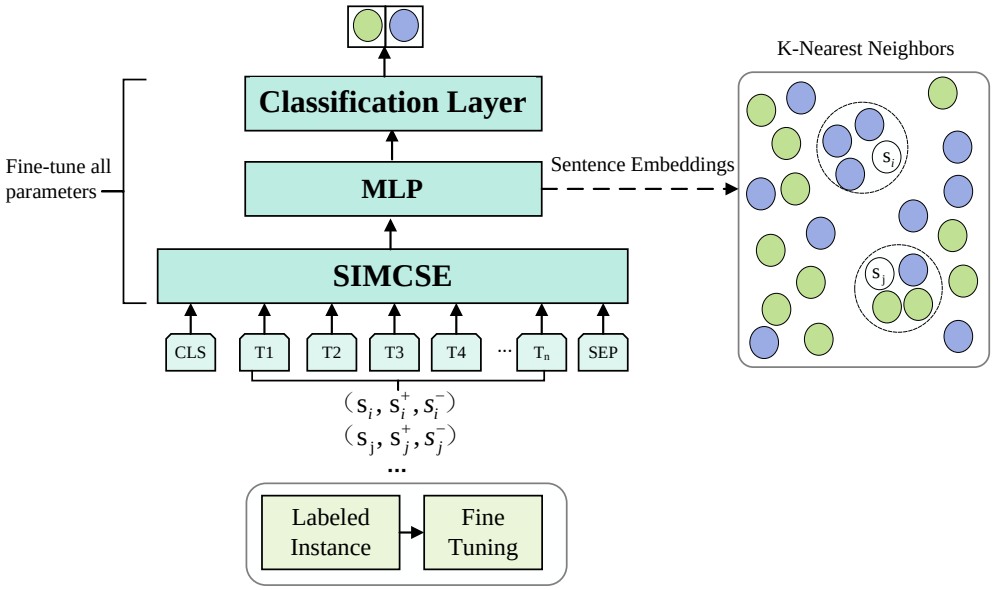

Figure 6: The structure of SimCSE-based model for binary feature extraction.

We fine-tune the parameters of the MLP layer as well as the original SimCSE model by minimizing contrastive objective loss among sentences. Formally, given an input sentence triple, (s_i, s_i^+, s_i^-), where s_i^+ denotes another sentence having the same label as s_i and s_i^- denotes another sentence having the opposite label as s_i, the training structure can be formally defined as follows:

$$x_i = (s_i, s_i^+, s_i^-) \tag{14}$$

$$y_i = SimCSE(x_i; \theta) \tag{15}$$

$$z_i = MLP(y_i; \theta) \tag{16}$$

$$l = \frac{1}{N} \sum_{i \in N} CL(z_i; \theta) \tag{17}$$

where x_i denotes an input triple, θ denotes the model parameters, N denotes the total number of training data, MLP() denotes the multiLayer perceptron, and CL() denotes the contrastive loss. The training process continuously optimizes model parameters to learn representations that encourage data within the same class to have similar representations while keeping data in different classes separated.

In the experiments, for each labeled sentence in training data, we randomly select three positive pairs and three negative pairs. Contrastive loss is measured by cosine similarity [11]. Finally, we extract k-nearest neighbors based on the output of the final linear layer. We suggest to set the value of k for binary feature extraction between 5 and 9 ($k = 9$ in our implementation). Our experiments have shown that the performance of GML is robust w.r.r the value of k provided that it is set within a reasonable range.

5. Experiments

In this section, we empirically evaluate the performance of the proposed solution on real benchmark data by a comparative study. Subsection 5.1 describes the experimental setup. Subsection 5.2 presents the comparative evaluation results. Subsection 5.3 presents the results of ablation study. Finally, Subsection 5.4 presents the results of parameter sensitivity evaluation.

5.1. Experimental Setup

Table 2: The statistics of the test datasets.

Dataset	0.25%Train	0.5%Train	1%Train	3%Train	5%Train	Validation	Test
CR	5	11	22	67	114	754	755
MR	22	42	85	255	426	1066	1067
twitter2013	12	25	51	153	255	915	2034
twitter2016	10	20	39	119	198	1234	10290

For empirical evaluation, we use four popular benchmark datasets for SLSA, which include MR [2], CR [3], Twitter2013 [4] and Twitter2016 [4]. MR is a collection of movie reviews. CR contains the customer reviews of electronic products. Both Twitter2013 and Twitter2016 contain microblog comments, which are usually shorter than movie and product reviews. The detailed statistics of these benchmark datasets are presented in Table 2. To simulate the scenarios of weak supervision, we randomly sample 0.25%, 0.5%, 1% , 3% and 5% respectively from the original split sets of training data.

We compare the proposed solution, denoted by W-GML, with the SOTA weakly supervised solutions as well as the SOTA deep models for SLSA. The compared deep models include

- RoBERTa [5]. It is the classical supervised model for text classification. It has been empirically shown to achieve SOTA performance on various text classification tasks, including SLSA.
- XLNet [37]. As an improvement over BERT, it uses a generalized autoregressive pre-trained model to learn bidirectional contexts by maximizing the expected likelihood over all the permutations of factorization order.
- EFL [38]. Converting class labels into auxiliary sentences, it is a unified model that can model multiple NLP tasks as a textual entailment task.
- DualCL [16]. Recently proposed for sentiment analysis, it can simultaneously learn the features of input samples and the parameters of classifiers in the same embedding space.
- SimCSE [11]. It is a simple contrastive learning framework for sentence embedding. For fair comparison, we use labeled training data to fine-tune the unsupervised SimCSE, which takes an input sentence and predicts itself in a contrastive objective, with only standard dropout used as noise.

Additionally, we compare W-GML with the SOTA semi-supervised solutions proposed for text classification, which can be easily applied to SLSA. They include

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

[3] https://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html#datasets

[4] https://www.dropbox.com/s/byzr8yoda6bua1b/2017_English_final.zip?file_subpath=%2F2017_English_final%2FGOLD%2FSubtask_A

- UST [33]. A teacher-student semi-supervised approach for text classification, it uses an improved self-training method to combine uncertainty estimates of underlying neural networks.
- COSINE [39]. A semi-supervised approach for text classification, it uses contrastive-regularized self training and elaborate confidence-based reweighting to minimize error propagation.
- S2T2 [27]. A recent teacher-student semi-supervised approach for text classification, it employs two teachers, one trained on labeled training data and the other on perturbed labeled data, to collaboratively refine the labeling results on unlabeled data in a bootstrapping manner.

In the implementations of the supervised deep models, we use whatever labeled training data provided to fine-tune pre-trained models. In contrast, the semi-supervised solutions train models using both labeled and unlabeled training data. In the implementations of W-GML, we first label unlabeled training data, and then leverage the resulting pseudo-labels to further fine-tune the GML model for test data. Our experiments show that even though pseudo-labels contain some noise, our implementation can effectively improve performance compared with the alternatives without or only partially exploiting unlabeled training data. In all the experiments, we set the max sentence length to be 128. The droupout rate is set at 0.1. We use Adam to optimize the parameters of deep models. In the implementation of W-GML, we set the learning rate of 2e-5 for the RoBERTa model. As usual, we compare performance on two metrics, Accuracy (ACC) and Macro-F1 (F1). We report both means and standard variances over 5 runs.

5.2. Comparative Evaluation

The detailed comparative results have been presented in Table 3, in which the values in brackets represent standard variances. At the end of the table, we also report the performance of deep models trained with full training data for reference. For simplicity of comparison, we separately report the improvements of W-GML over the semi-supervised alternatives in Table. 4.

It can be observed that with all the weakly supervised settings, which correspond to various sufficiency levels of labeled training data (i.e., 0.25%, 0.5%, 1%, 3% and 5%), W-GML consistently outperforms the alternatives in terms of both accuracy and macro-F1. Specifically, W-GML beats the supervised deep models by large margins when only a few labeled training data are provided (e.g., $\leq$ 1%). For instance, on CR, with 0.25% of labeled training data, the best results achieved by deep models are around 69% and 78% in terms of accuracy and F1 respectively. In comparison, W-GML achieves the performance of 79% and 84% in terms of accuracy and F1, with the improvement margins at 9.41% and 6.5% respectively. The comparative results on the other datasets are similar. These evaluation results clearly demonstrate that the efficacy of deep models to a large extent depends on sufficient labeled training data, and their performance can be severely compromised if provided with a very limited number of labeled training data. Especially, it can be observed that on twitter2016, with only 0.25% training data, the performance of some deep models (e.g., RoBERTa and EFL) is very low, e.g., with F1 even less than 10%. In these cases, with few labeled examples, the deep models seriously underfit and predict the same label on almost all the target data.

It can also be observed that compared with the supervised deep models, the semi-supervised solutions can considerably improve classification accuracy by leveraging unlabeled training data. For instance, again on CR, with 0.25% of training data, the best performer of the semi-supervised solutions, which is UST, achieves around 76% and 81% in terms of accuracy and macro-F1. They represents considerable improvements over the best performance of supervised deep models, which are only 69% and 78% respectively. The comparative results are similar on the other datasets even though with varying margins. These evaluation results validate the efficacy of these semi-supervised solutions.

Finally, as shown in Table. 4, W-GML consistently outperforms the semi-supervised solutions, with margins being considerable when only a few labeled training data are provided ($\leq$ 1%). For instance, on Twitter2016, with only 0.25% of labeled training data, W-GML achieves the performance of around 66% and 69% in terms of accuracy and macro-F1 respectively. Compared with the best semi-supervised solution (COSINE), the improvements are around 5% and 6% respectively. Actually, with 0.25% of training data, W-GML beats the best semi-supervised alternative by the margins of around 2.52%, 5.75%, 4.97% and 5.37% in terms of accuracy on CR, MR, Twitter2013 and Twitter2016 respectively. With 0.5% of training data, the margins are around 2.05%, 2.29%, 3.52% and 4.26% on the four datasets respectively.

It is no surprise that the improvement margins of W-GML over the supervised and semi-supervised alternatives tend to become smaller as more labeled training data are provided. However, it is worthy to point out that W-GML can

Table 3: Comparative evaluation results: ⋄ denotes the supervised approaches and ⋆ denotes the semi-supervised approaches.

Training Data (%)	Models	CR		MR		Twitter2013		Twitter2016	
		Accuracy(%)	F1(%)	Accuracy(%)	F1(%)	Accuracy(%)	F1(%)	Accuracy(%)	F1(%)
0.25%	⋄ RoBERTa	54.70(0.01)	52.08(0.13)	49.77(0.20)	66.46(0.13)	72.52(0.15)	84.07(0.22)	31.89(0.13)	2.09(0.21)
	⋄ XLNet	57.85(0.07)	67.00(0.02)	49.76(0.18)	66.45(0.44)	68.68(0.08)	80.42(0.12)	32.35(0.01)	3.01(0.20)
	⋄ EFL	64.37(0.15)	78.20(0.10)	64.94(0.49)	70.87(0.20)	65.92(0.17)	77.04(0.30)	33.94(0.04)	7.68(0.07)
	⋄ DualCL	62.91(0.06)	63.73(0.02)	62.32(0.16)	54.32(0.27)	65.63(0.20)	62.11(0.31)	54.16(0.20)	46.63(0.15)
	⋄ SimCSE	69.66(0.15)	78.25(0.21)	50.18(0.45)	66.57(0.31)	75.22(0.02)	84.10(0.05)	43.26(0.26)	33.65(0.49)
	⋆ UST	76.55(0.16)	81.77(0.24)	65.13(0.10)	70.69(0.31)	76.19(0.31)	82.70(0.12)	60.68(0.31)	62.87(0.09)
	⋆ COSINE	71.05(0.12)	77.31(0.17)	65.42(0.07)	71.15(0.05)	75.91(0.21)	70.20(0.80)	61.40(0.21)	63.21 (0.51)
	⋆ S2T2	75.63(0.30)	81.71(0.29)	66.35(0.15)	72.19(0.21)	79.15(0.27)	85.31(0.19)	56.79(0.28)	59.91(0.19)
	W-GML	**79.07(0.13)**	**84.75(0.02)**	**72.10(0.28)**	**76.25(0.35)**	**84.12(0.19)**	**88.63(0.14)**	**66.77(0.07)**	**69.63(0.15)**
0.5%	⋄ RoBERTa	50.00(0.21)	39.06(0.24)	62.14(0.21)	71.51(0.17)	72.81(0.07)	84.00(0.09)	35.24(0.01)	12.89(0.30)
	⋄ XLNet	62.32(0.58)	74.03(0.01)	58.07(0.06)	36.58(0.67)	69.02(0.15)	79.61(0.26)	37.44(0.25)	16.99(0.31)
	⋄ EFL	65.17(0.08)	77.46(0.07)	80.97(0.03)	82.20(0.15)	70.50(0.18)	82.26(0.21)	51.10(0.23)	47.30(0.16)
	⋄ DualCL	64.37(0.17)	65.47(0.19)	76.38(0.23)	75.86(0.27)	73.65(0.31)	74.57(0.37)	69.20(0.21)	73.70(0.17)
	⋄ SimCSE	64.15(0.14)	74.34(0.20)	69.82(0.11)	65.59(0.76)	72.41(0.27)	83.53(0.01)	72.18(0.36)	75.56(0.30)
	⋆ UST	81.67(0.30)	83.59(0.28)	82.15(0.17)	82.10(0.14)	82.12(0.21)	85.51(0.24)	73.32(0.07)	76.22(0.09)
	⋆ COSINE	80.01(0.90)	82.11(0.41)	82.76(0.21)	82.74(0.15)	82.65(0.20)	85.42(0.31)	73.00(0.12)	75.92 (0.17)
	⋆ S2T2	79.87(0.22)	82.38(0.37)	82.99(0.24)	81.82(0.35)	82.22(0.88)	85.63(0.03)	71.20(0.42)	73.53(0.21)
	W-GML	**83.72(0.42)**	**85.14(0.31)**	**85.28(0.37)**	**85.99(0.10)**	**86.17(0.25)**	**89.58(0.18)**	**77.58(0.05)**	**81.25(0.02)**
1%	⋄ RoBERTa	64.10(0.14)	79.85(0.07)	81.82(0.32)	80.57(0.16)	78.66(0.20)	86.38(0.31)	36.12(0.08)	16.72(0.14)
	⋄ XLNet	64.11(0.21)	78.13(0.35)	82.09(0.13)	82.31(0.22)	72.57(0.31)	84.08(0.19)	37.66(0.82)	18.04(0.21)
	⋄ EFL	66.49(0.04)	79.10(0.21)	82.94(0.07)	82.87(0.03)	87.89(0.30)	89.63(0.01)	79.83(0.13)	84.74(0.02)
	⋄ DualCL	71.19(0.77)	73.60(0.23)	81.81(0.15)	81.43(0.06)	88.84(0.03)	88.84(0.41)	83.22(0.07)	82.66(0.18)
	⋄ SimCSE	64.23(0.23)	78.19(0.76)	80.85(0.28)	80.97(0.89)	88.69(0.30)	92.16(0.14)	81.63(0.06)	86.01(0.25)
	⋆ UST	86.15(0.22)	89.26(0.69)	82.57(0.23)	82.31(0.20)	83.43(0.59)	88.69(0.27)	84.12(0.15)	86.19(0.29)
	⋆ COSINE	83.57(0.50)	82.35(0.32)	83.72(0.80)	83.70(0.36)	87.71(0.78)	83.91(0.14)	76.14(0.26)	83.15(0.55)
	⋆ S2T2	81.59(0.11)	86.41(0.28)	83.13(0.05)	83.68(0.72)	88.25(0.47)	88.67(0.51)	76.03(0.52)	84.56(0.13)
	W-GML	**88.07(0.25)**	**91.32(0.30)**	**85.84(0.51)**	**86.92(0.07)**	**91.29(0.15)**	**93.76(0.32)**	**87.43(0.27)**	**90.66(0.48)**
3%	⋄ RoBERTa	87.95(0.11)	90.09(0.25)	84.91(0.20)	84.62(0.38)	87.66(0.17)	91.41(0.28)	83.61(0.20)	88.53(0.15)
	⋄ XLNet	82.65(0.52)	86.83(0.20)	84.37(0.24)	84.98(0.14)	85.35(0.10)	90.11(0.27)	77.96(0.35)	85.38(0.19)
	⋄ EFL	90.93(0.35)	93.18(0.89)	83.97(0.02)	84.16(0.31)	89.45(0.28)	92.81(0.03)	84.84(0.05)	89.81(0.23)
	⋄ DualCL	85.30(0.77)	88.33(0.25)	86.46(0.17)	86.21(0.25)	90.78(0.30)	90.75(0.21)	87.38(0.25)	86.66(0.51)
	⋄ SimCSE	83.31(0.14)	88.06(0.51)	86.42(0.19)	86.48(0.06)	90.19(0.16)	93.01(0.13)	87.65(0.21)	91.17(0.05)
	⋆ UST	88.07(0.20)	90.62(0.77)	84.53(0.22)	83.99(0.49)	89.87(0.09)	92.01(0.26)	87.03(0.15)	89.21(0.52)
	⋆ COSINE	89.30(0.01)	88.33(0.03)	85.66(0.60)	85.68(0.72)	90.17(0.60)	93.12(0.68)	85.57(0.19)	87.65 (0.15)
	⋆ S2T2	82.02(0.34)	87.13(0.12)	86.12(0.47)	86.68(0.88)	90.81(0.93)	93.58(0.56)	84.85(0.23)	86.85(0.43)
	W-GML	**92.71(0.45)**	**94.43(0.66)**	**87.53(0.79)**	**87.65(0.31)**	**92.53(0.01)**	**94.89(0.30)**	**89.29(0.18)**	**92.36(0.24)**
5%	⋄ RoBERTa	89.80(0.71)	91.76(0.23)	85.75(0.06)	85.98(0.18)	89.68(0.21)	92.78(0.13)	84.92(0.24)	89.73(0.16)
	⋄ XLNet	88.87(0.19)	91.27(0.23)	86.22(0.77)	86.45(0.13)	88.34(0.17)	92.01(0.20)	83.17(0.13)	87.87(0.17)
	⋄ EFL	92.78(0.20)	94.40(0.27)	86.74(0.10)	86.85(0.29)	91.00(0.15)	93.81(0.09)	88.31(0.02)	91.60(0.08)
	⋄ DualCL	90.79(0.17)	90.59(0.29)	86.93(0.30)	86.88(0.15)	90.88(0.09)	90.92(0.05)	88.53(0.27)	86.63 (0.05)
	⋄ SimCSE	91.12(0.15)	93.23(0.21)	84.87(0.25)	85.90(0.12)	90.46(0.02)	93.37(0.08)	86.17(0.19)	90.54(0.21)
	⋆ UST	92.85(0.16)	94.58(0.27)	87.69(0.05)	87.91(0.13)	91.03(0.21)	93.91(0.06)	89.80(0.12)	92.10(0.30)
	⋆ COSINE	92.89(0.45)	94.87(0.56)	88.28(0.20)	88.31(0.06)	91.95(0.12)	92.12(0.38)	89.14(0.11)	89.73(0.70)
	⋆ S2T2	92.91(0.67)	94.89(0.45)	88.94(0.64)	88.93(0.58)	91.23(0.07)	93.98(0.32)	89.49(0.45)	90.15(0.18)
	W-GML	**93.37(0.20)**	**95.02(0.31)**	**89.85(0.18)**	**89.91(0.30)**	**93.65(0.05)**	**95.59(0.61)**	**90.44(0.24)**	**93.17(0.38)**
100%	⋄ RoBERTa	93.07(0.11)	94.57(0.33)	90.21(0.19)	90.26(0.35)	94.82(0.75)	95.00(0.24)	91.47(0.27)	94.00(0.57)
	⋄ XLNet	92.72(0.17)	94.37(0.23)	90.07(0.32)	90.14(0.45)	93.15(0.17)	93.47(0.25)	91.04(0.35)	93.62(0.38)
	⋄ EFL	93.94(0.04)	95.36(0.12)	92.27(0.32)	92.23(0.22)	94.51(0.02)	94.60(0.40)	92.75(0.12)	94.86(0.69)
	⋄ DualCL	92.64(0.16)	92.78(0.23)	89.43(0.51)	89.10(0.39)	93.75(0.33)	93.69(0.26)	91.76(0.39)	91.67(0.11)
	⋄ SimCSE	93.87(0.11)	95.19(0.68)	91.75(0.07)	91.77(0.03)	94.34(0.10)	96.11(0.25)	93.11(0.07)	95.10(0.21)

achieve good performance with very few labeled training data. For instance, with only 3% of training data (dozens), the performance of W-GML is around 92% in terms of accuracy on both CR and Twitter2013. They are close to the performance of the SOTA models trained with full training data (100%), which is only around 94% on both datasets. Furthermore, it can be observed that on all the four datasets, provided with only 5% of labeled training data, the performance of W-GML is highly competitive with the SOTA deep models trained with full training data.

Table 4: Comparative evaluation results: the improvement margins of W-GML over the semi-supervised alternatives.

Training Data (%)	Models	CR		MR		Twitter2013		Twitter2016	
		Accuracy(%)	F1(%)	Accuracy(%)	F1(%)	Accuracy(%)	F1(%)	Accuracy(%)	F1(%)
0.25%	W-GML *vs* UST	2.52 ↑	2.98 ↑	6.97 ↑	5.56 ↑	7.93 ↑	5.93 ↑	6.09 ↑	6.76 ↑
	W-GML *vs* COSINE	8.02 ↑	7.44 ↑	6.68 ↑	5.10 ↑	8.21 ↑	18.43 ↑	5.37 ↑	6.42 ↑
	W-GML *vs* S2T2	3.44 ↑	3.04 ↑	5.75 ↑	4.06 ↑	4.97 ↑	3.32 ↑	9.98 ↑	9.72 ↑
0.5%	W-GML *vs* UST	2.05 ↑	1.55 ↑	3.13 ↑	3.89 ↑	4.05 ↑	4.07 ↑	4.26 ↑	5.03 ↑
	W-GML *vs* COSINE	3.71 ↑	3.03 ↑	2.52 ↑	3.25 ↑	3.52 ↑	4.16 ↑	4.58 ↑	5.33 ↑
	W-GML *vs* S2T2	3.85 ↑	2.76 ↑	2.29 ↑	4.17 ↑	3.95 ↑	3.95 ↑	6.38 ↑	7.72 ↑
1%	W-GML *vs* UST	1.92 ↑	2.06 ↑	3.27 ↑	4.61 ↑	7.86 ↑	5.07 ↑	3.31 ↑	4.47 ↑
	W-GML *vs* COSINE	4.50 ↑	8.97 ↑	2.12 ↑	3.22 ↑	3.58 ↑	9.85 ↑	11.29 ↑	7.51 ↑
	W-GML *vs* S2T2	6.48 ↑	4.91 ↑	2.71 ↑	3.24 ↑	3.04 ↑	5.09 ↑	11.40 ↑	6.10 ↑
3%	W-GML *vs* UST	4.64 ↑	3.81 ↑	3.00 ↑	3.66 ↑	2.66 ↑	2.88 ↑	2.26 ↑	3.15 ↑
	W-GML *vs* COSINE	3.41 ↑	6.10 ↑	1.87 ↑	1.97 ↑	2.36 ↑	1.77 ↑	3.72 ↑	4.71 ↑
	W-GML *vs* S2T2	10.69 ↑	7.30 ↑	1.41 ↑	0.97 ↑	1.72 ↑	1.31 ↑	4.44 ↑	5.51 ↑
5%	W-GML *vs* UST	0.52 ↑	0.44 ↑	2.16 ↑	2.00 ↑	2.62 ↑	1.68 ↑	0.64 ↑	1.07 ↑
	W-GML *vs* COSINE	0.48 ↑	0.15 ↑	1.57 ↑	1.60 ↑	1.70 ↑	3.47 ↑	1.30 ↑	3.44 ↑
	W-GML *vs* S2T2	0.46 ↑	0.13 ↑	0.91 ↑	0.98 ↑	2.42 ↑	1.61 ↑	0.95 ↑	3.02 ↑

These evaluation results clearly validate the efficacy of our proposed solution and bode well its applicability in real scenarios.

5.3. Ablation Study

To verify the efficacy of the proposed framework, we conduct an ablation study on W-GML. First, we compare the phased W-GML approach with a simpler solution of GML, denoted by 1-GML, which labels unlabeled target instances in one phase instead of multiple phases. Second, we also compare W-GML with another simpler solution of GML, denoted by N-GML, which does not leverage GML self-training for iterative label fine-tuning in each phase.

Table 5: W-GML vs 1-GML

Data name	Training Ratio(%)	Accuracy(%)		F1(%)	
		W-GML	1-GML	W-GML	1-GML
CR	0.25	79.07(0.13)	78.20(0.05)	84.75(0.02)	83.29(0.09)
	0.5	83.72(0.42)	82.93(0.33)	85.14(0.31)	84.72(0.21)
	1	88.07(0.25)	87.49(0.15)	91.32(0.30)	90.73(0.04)
	3	92.71(0.45)	91.52(0.17)	94.43(0.66)	93.60(0.82)
	5	93.37(0.20)	92.84(0.05)	95.02(0.31)	94.40(0.01)
MR	0.25	72.10(0.28)	70.39(0.06)	76.25(0.35)	74.02(0.12)
	0.5	85.28(0.37)	84.96(0.47)	85.99(0.10)	85.00(0.30)
	1	85.84(0.51)	85.66(0.28)	86.92(0.07)	86.54(0.19)
	3	87.53(0.79)	87.34(0.16)	87.65(0.31)	87.19(0.25)
	5	89.85(0.18)	87.55(0.39)	89.91(0.30)	87.51(0.18)
Twitter2013	0.25	84.12(0.19)	83.17(0.22)	88.63(0.14)	85.89(0.16)
	0.5	86.17(0.25)	85.51(0.30)	89.58(0.18)	87.59(0.19)
	1	91.29(0.15)	89.97(0.24)	93.76(0.32)	93.21(0.17)
	3	92.53(0.01)	90.39(0.30)	94.89(0.30)	93.23(0.11)
	5	93.65(0.05)	90.75(0.01)	95.59(0.61)	93.53(0.18)
Twitter2016	0.25	66.77(0.07)	65.86(0.39)	69.63(0.15)	68.00(0.20)
	0.5	77.58(0.05)	76.13(0.11)	81.25(0.02)	79.50(0.04)
	1	87.43(0.27)	85.97(0.05)	90.66(0.48)	87.21(0.70)
	3	89.29(0.18)	88.19(0.27)	92.36(0.24)	90.95(0.33)
	5	90.44(0.24)	90.17(0.18)	93.17(0.38)	92.95(0.74)

Ablation Study 1: W-GML vs 1-GML. The detailed experimental results have been presented in Table 5. It can be observed that W-GML consistently outperforms 1-GML, and the margins are considerable in many cases. For instance, with 0.25% of training data on CR, W-GML beats 1-GML by the margins of around 0.87% and 1.46% in terms of accuracy and F1 respectively. Similarly, with 5% of training data on MR, W-GML beats 1-GML by the margins exceeding 2.3% and 2.4% in terms of accuracy and F1. These results clearly show that when only a few labeled training data are available, leveraging pseudo-labels to fine-tune deep features can usually improve gradual learning, even though the pseudo-labels may contain some noise.

Table 6: W-GML vs N-GML

Data name	Training Ratio(%)	Accuracy(%)		F1(%)	
		W-GML	N-GML	W-GML	N-GML
CR	0.25	79.07(0.13)	76.90(0.24)	84.75(0.02)	81.08(0.26)
	0.5	83.72(0.42)	80.23(0.10)	85.14 (0.31)	83.91(0.91)
	1	88.07(0.25)	85.86(0.19)	91.32(0.30)	88.40(0.72)
	3	92.71(0.45)	90.34(0.14)	94.43(0.66)	92.26(0.03)
	5	93.37(0.20)	92.58(0.26)	95.02(0.31)	94.15(0.15)
MR	0.25	72.10(0.28)	69.43(0.10)	76.25(0.35)	73.70(0.27)
	0.5	85.28(0.37)	84.81(0.12)	85.99(0.10)	83.83(0.03)
	1	85.84(0.51)	83.13(0.16)	86.92(0.07)	83.95(0.10)
	3	87.53(0.79)	86.22(0.02)	87.65(0.31)	86.27(0.39)
	5	89.85(0.18)	85.47(0.31)	89.91(0.30)	85.60(0.08)
Twitter2013	0.25	84.12(0.19)	82.62(0.03)	88.63(0.14)	86.34(0.02)
	0.5	86.17(0.25)	84.50(0.01)	89.58(0.18)	87.05(0.06)
	1	91.29(0.15)	89.62(0.06)	93.76(0.32)	92.65(0.15)
	3	92.53(0.01)	90.21(0.21)	94.89(0.30)	93.14(0.26)
	5	93.65(0.05)	85.15(0.13)	95.59(0.61)	88.92(0.10)
Twitter2016	0.25	66.77(0.07)	61.01(0.23)	69.63(0.15)	62.16(0.15)
	0.5	77.58(0.05)	73.07(0.05)	81.25(0.02)	79.65(0.02)
	1	87.43(0.27)	84.77(0.01)	90.66(0.48)	88.66(0.17)
	3	89.29(0.18)	87.39(0.20)	92.36(0.24)	90.67(0.08)
	5	90.44(0.24)	85.47(0.28)	93.17(0.38)	92.02(0.09)

Ablation Study 2: W-GML vs N-GML. The detailed comparative results have been presented in Table 6. It can be similarly observed that W-GML consistently outperforms N-GML, and the margins are considerable in many cases. For instance, with 5% of training data, self-training improves performance by the margins of 0.79%, 4.38%, 8.5% and 4.97% on CR, MR, Twitter2013 and Twitter2016 respectively in terms of accuracy. These results clearly show that GML self-training can effectively improve labeling accuracy. On one hand, GML self-training can directly improve the labeling accuracy of the current batch of target instances; on the other hand, since W-GML leverages the resulting labels for deep feature extraction, it can also improve knowledge conveyance, thus enhancing W-GML's performance on future batches of target instances.

5.4. Sensitivity Study

In this subsection, we evaluate the performance sensitivity of W-GML w.r.t various algorithmic parameters, which include the value of k indicating k-nearest neighbors in the construction of binary factors, the values of β_1 and β_2 ($\beta_2 = 1 - \beta_1$) as shown in Eq. 13 in the extraction of unary features, and the parameters of GML self-training, i.e., the proportion of randomly selected pseudo-labels and the number of iterations. In the study, we set k between 5 and 9, β_1 between 0.5 and 0.7, while setting the proportion of pseudo-labels within the reasonable range between 20% and 50%, and the number of iterations between 1 and 3. We present our evaluation results on the CR workload provided with 0.25% of training data. The evaluation results on other workloads are very similar, thus omitted here.

Table 7: Sensitivity evaluation results on CR w.r.t the parameter of k.

K_{nn}	0.25%		0.5%		1%		3%		5%	
	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)
5	78.92(0.16)	83.79(0.09)	82.92(0.35)	84.75(0.16)	88.27 (0.01)	91.62(0.15)	92.35(0.11)	93.67(0.28)	92.70(0.03)	94.52(0.30)
7	79.04(0.52)	84.71(0.04)	83.45(0.27)	85.02(0.03)	88.58 (0.69)	91.48(0.31)	92.51(0.16)	93.86(0.15)	92.93(0.10)	94.60(0.27)
9	79.07(0.13)	84.75(0.02)	83.72(0.42)	85.14(0.31)	88.07(0.25)	91.32(0.30)	92.71(0.45)	94.43(0.66)	93.37(0.20)	95.02(0.31)

Table 8: Sensitivity evaluation results on CR w.r.t the parameters of β_1 and β_2.

β_1	β_2	0.25%		0.5%		1%		3%		5%	
		Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)
0.5	0.5	78.75(0.69)	84.38(0.22)	83.45(0.07)	84.71(0.31)	87.85(0.05)	90.73(0.10)	92.68(0.20)	94.35(0.17)	93.30(0.11)	94.92(0.37)
0.6	0.4	79.32(0.10)	84.92(0.26)	83.62(0.35)	84.89(0.10)	87.92(0.41)	90.82(0.19)	92.85(0.10)	94.48(0.55)	93.27(0.31)	94.75(0.16)
0.7	0.3	79.07(0.13)	84.75(0.02)	83.72(0.42)	85.14(0.31)	88.07(0.25)	91.32(0.30)	92.71(0.45)	94.43(0.66)	93.37(0.20)	95.02(0.31)

Table 9: Sensitivity evaluation results on CR w.r.t the ratio of pseudo-label selection.

Ratio(%)	0.25%		0.5%		1%		3%		5%	
	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)
20	80.23(0.19)	85.03(0.41)	83.28(0.14)	84.81(0.25)	87.73(0.30)	90.72(0.19)	92.58(0.15)	93.97(0.40)	92.82(0.04)	94.59(0.10)
30	79.10(0.32)	84.82(0.29)	83.51(0.51)	85.00(0.09)	87.94(0.45)	91.10(0.21)	92.35(0.04)	93.94(0.19)	93.58(0.19)	95.18(0.23)
50	79.07(0.13)	84.75(0.02)	83.72(0.42)	85.14(0.31)	88.07(0.25)	91.32(0.30)	92.71(0.45)	94.43(0.66)	93.37(0.20)	95.02(0.31)

Table 10: Sensitivity evaluation results on CR w.r.t the number of iterations.

Iterations	0.25%		0.5%		1%		3%		5%	
	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)	Acc(%)	F1(%)
1	79.21(0.08)	84.82(0.50)	83.56(0.16)	85.06(0.17)	88.19(0.51)	91.46(0.12)	92.35(0.06)	94.03 (0.09)	93.01(0.16)	94.40(0.59)
2	79.07(0.13)	84.75(0.02)	83.72(0.42)	85.14(0.31)	88.07(0.25)	91.32(0.30)	92.71(0.45)	94.43(0.66)	93.37(0.20)	95.02(0.31)
3	79.83(0.71)	85.01(0.03)	83.45(0.83)	84.98(0.20)	87.76(0.45)	91.05(0.01)	92.45(0.16)	93.97(0.03)	93.18(0.25)	94.53(0.18)

The detailed evaluation results w.r.t the parameter of k, β_1 & β_2, the ratio of pseudo-labels and the number of iterations have been presented in Table 7, 8, 9, and 10 respectively. It can be observed that as the value of a parameter varies, the performance of W-GML only fluctuates marginally. The robustness of W-GML bodes well its efficacy in real applications.

6. Conclusion

In this paper, we have proposed a weakly supervised solution based on gradual machine learning for the task of SLSA. It iteratively labels a proportion of target instances, which are selected based on GML evidential certainty, by gradual learning. By leveraging deep models for feature extraction and GML for gradual inference, the proposed solution achieves considerably better performance than the existing alternatives.

For future work, it can be observed that in the proposed solution, only the deep models for feature extraction are task-specific. For many NLP tasks, similar deep models either already exist or can be easily constructed. Therefore, the proposed framework is potentially applicable to other binary classification tasks, especially NLP tasks. Furthermore, even though this paper focuses on binary classification, in principle, the factor model of GML can be easily extended to handle multi-label classification tasks. Instead of binary values, a variable in the factor graph can take multiple values, each of which corresponds to a specific class. Relational factors can also be similarly constructed to reflect the similar or different label relations between variables. However, the core challenge of how to extract effective relations for multi-label classification tasks needs be be further investigated in future work.

References

[1] V. K. Bongirwar, A survey on sentence level sentiment analysis, International Journal of Computer Science Trends and Technology. 3 (3) (2015) 110–113.

[2] D. Yin, T. Meng, K.-W. Chang, SentiBERT: A transferable transformer-based architecture for compositional sentiment semantics, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3695–3706.

[3] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019, pp. 3982–3992.

[4] R. Xiang, J. Li, M. Wan, J. Gu, Q. Lu, W. Li, C.-R. Huang, Affective awareness in neural sentiment analysis, Knowledge-Based Systems. 226 (2021) 107137.

[5] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, CoRR (2019). arXiv:1907.11692.

[6] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, CoRR (2020). arXiv:1909.11942.

[7] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Meeting of the Association for Computational Linguistics, 2020, pp. 7871–7880.

[8] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, CoRR (2020). arXiv:2006.03654.

[9] Y. Sun, S. Wang, S. Feng, S. Ding, C. Pang, J. Shang, J. Liu, X. Chen, Y. Zhao, Y. Lu, W. Liu, Z. Wu, W. Gong, J. Liang, Z. Shang, P. Sun, W. Liu, X. Ouyang, D. Yu, H. Tian, H. Wu, H. Wang, Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation, CoRR (2021). arXiv:2107.02137.

[10] Y. Wang, Q. Chen, J. Shen, B. Hou, M. H. M. Ahmed, Z. Li, Aspect-level sentiment analysis based on gradual machine learning, Knowledge-Based Systems. 212 (2021) 106509.

[11] T. Gao, X. Yao, D. Chen, Simcse: Simple contrastive learning of sentence embeddings, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 6894–6910.

[12] A. Adhikari, A. Ram, R. Tang, J. Lin, Rethinking Complex neural Network Architectures for Document 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), 2019, pp. 4046–4051.

[13] G. Nikolentzos, A. Tixier, M. Vazirgiannis, Message passing attention networks for document understanding, in: Proceedings of the AAAI Conference on Artificial Intelligence., 2020, pp. 8544–8551.

[14] T. Ito, K. Tsubouchi, H. Sakaji, T. Yamashita, K. Izumi, Contextual Sentiment Neural Network for Document Sentiment analysis, Data Science and Engineering. 5 (2020) 180–192.

[15] C. Liu, Z. Mengchao, F. Zhibing, P. Hou, Y. Li, FLiText: A faster and lighter semi-supervised text classification with convolution networks, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 2481–2491.

[16] Q. Chen, R. Zhang, Y. Zheng, Y. Mao, Dual Contrastive Learning: Text Classification via Label-Aware Data Augmentation, CoRR (2022). arXiv:2201.08702.

[17] 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 2014), Association for Computational Linguistics, 2014, pp. 437–442.

[18] B. Pang, L. Lee, Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval. 2 (2008) 1–135.

[19] M. H. M. Ahmed, Q. Chen, Y. Wang, Y. Nafa, Z. Li, T. Duan, Dnn-driven Gradual Machine Learning for Aspect-term Sentiment Analysis, in: Findings of the Association for Computational Linguistics, 2021, pp. 488–497.

[20] S. Garg, G. Ramakrishnan, BAE: BERT-based adversarial examples for text classification, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020, pp. 6174–6181.

[21] M. Rhanoui, M. Mikram, S. Yousfi, S. Barzali, A CNN-BiLSTM Model for Document-Level Sentiment Analysis, Machine Learning and Knowledge Extraction. 1 (3) (2019) 832–847.

[22] G. Rao, W. Huang, Z. Feng, Q. Cong, LSTM with sentence representations for document-level sentiment classification, Neurocomputing. 308 (2018) 49–57.

[23] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of bert: smaller, faster, cheaper and lighter, CoRR (2019). arXiv:1910.01108.

[24] S. Yang, Xiangli, I. Zixing, King, A Survey on Deep Semi-supervised Learning, IEEE Transactions on Knowledge and Data Engineering. (2021) 1–20.

[25] L.-C. F. Austin Cheng-Yun Tsai, Sheng-Ya Lin, Contrast-enhanced semi-supervised text classification with few labels, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 11394–11402.

[26] S. Zuo, Y. Yu, C. Liang, H. Jiang, S. Er, C. Zhang, T. Zhao, H. Zha, Self-training with differentiable teacher, in: Findings of the Association for Computational Linguistics, 2022, pp. 933–949.

[27] H. T. N. Qian Lin, A semi-supervised learning approach with two teachers to improve breakdown identification in dialogues, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 11011–11019.

[28] N. K. Payam Karisani, A. I. . Claims, Semi-supervised text classification via self-pretraining, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021, pp. 40–48.

[29] W. Li, T. Qian, From consensus to disagreement: Multi-teacher distillation for semi-supervised relation extraction, CoRR (2021). arXiv:2112.01048.

[30] G. Karamanolakis, S. Mukherjee, G. Zheng, A. H. Awadallah, Self-training with weak supervision, in: Association for Computational Linguistics, 2021, pp. 845–863.

[31] Y. Yu, S. Zuo, H. Jiang, W. Ren, T. Zhao, C. Zhang, Fine-tuning pre-trained language model with weak supervision:a contrastive-regularized self-training, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2020, pp. 1063–1077.

[32] D. Y. Jiaao Chen, Zichao Yang, Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification, in: Association for Computational Linguistics, 2020, pp. 2147—2157.

[33] S. Mukherjee, A. H. Awadallah, Uncertainty-aware self-training for few-shot text classification, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020, pp. 21199–21212.

[34] B. Hou, Q. Chen, Y. Wang, Y. Nafa, Z. Li, Gradual machine learning for entity resolution, IEEE Transactions on Knowledge and Data Engineering. 34 (4) (2022) 1803–1814.

[35] S. X. Chen, Empirical likelihood confidence intervals for linear regression coefficients, Journal of Multivariate Analysis. 49 (1) (1994) 24–40.

[36] D. Guillory, V. Shankar, S. Ebrahimi, T. Darrell, L. Schmidt, Predicting with Confidence on Unseen Distributions, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 1114–1124.

[37] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. V. Le, Xlnet Generalized Autoregressive Pretraining for Language Understanding, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019, pp. 5753–5763.

[38] S. Wang, H. Fang, M. Khabsa, H. Mao, H. Ma, Entailment as Few-Shot Learner, CoRR (2021). arXiv:2104.14690.

[39] Y. Yu, S. Zuo, H. Jiang, W. Ren, C. Zhang, Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2020, pp. 1063–1077.