CORT: A New Baseline for Comparative Opinion Classification by Dual Prompts

Yequan Wang[1], **Hengran Zhang**[2,3], **Aixin Sun**[4], **Xuying Meng**[2]
[1]Beijing Academy of Artificial Intelligence, Beijing, China
[2]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[3]University of Chinese Academy of Sciences, Beijing, China
[4]School of Computer Science and Engineering, Nanyang Technological University, Singapore
tshwangyequan@gmail.com, axsun@ntu.edu.sg, {zhanghengran22z,mengxuying}@ict.ac.cn

Abstract

Comparative opinion is a common linguistic phenomenon. The opinion is expressed by comparing multiple targets on a shared aspect, *e.g.,* "camera A is better than camera B in picture quality". Among the various subtasks in opinion mining, comparative opinion classification is relatively less studied. Current solutions use rules or classifiers to identify opinions, *i.e., better*, *worse*, or *same*, through feature engineering. Because the features are directly derived from the input sentence, these solutions are sensitive to the order of the targets mentioned in the sentence. For example, "camera A is better than camera B" means the same as "camera B is worse than camera A"; but the features of these two sentences are completely different. In this paper, we approach comparative opinion classification through prompt learning, taking the advantage of embedded knowledge in pre-trained language model. We design a twin framework with dual prompts, named CORT. This extremely simple model delivers state-of-the-art and robust performance on all benchmark datasets for comparative opinion classification. We believe CORT well serves as a new baseline for comparative opinion classification.

1 Introduction

Comparative opinion classification (Liu, 2012) aims to find the relative opinion preference on a specific aspect towards two or more compared targets. Sentences containing comparative opinions may not express a direct positive or negative opinion, but a comparison. In this example sentence, "*BMW's handling is better than that of Mercedes-Benz.*", there are two targets: *BMW* and *Mercedes-Benz*. The aspect in comparison is *handling*, and the opinion is target t_1 (*BMW*) is better than target t_2 (*Mercedes-Benz*). Note that, the comparison does not imply that the opinion towards Mercedes-Benz is negative. Hence, performing typical sentiment classification as a whole is less applicable to comparative text.

Comparative opinion plays a vital role in consumers' purchasing decisions. It is common that a consumer identifies a few candidate products and makes a comparison on all aspects of his/her interest. Comparative sentences are also widely observed in product reviews and online forums.

The Research Problem. In comparative opinion mining, there are a predefined set of opinions $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$. Given a comparative sentence, denoted by $S = [w_1, w_2, \ldots, w_n]$, the task is to predict a four-tuple (t_1, t_2, a, o). Here, t_1 and t_2 are the targets to be compared, a denotes the aspect, and o is the opinion.

This task can be decomposed into two subtasks: (i) *comparative elements extraction* to extract targets t_1, t_2 in comparison and aspect a, and (ii) *comparative opinion classification* to predict opinion o with the assumption that targets t_1, t_2 and the aspect a are given. In this paper, we focus on the second subtask. That is, we assume that targets and aspects are pre-extracted. Generally, the opinion set includes *better*, *worse*, *same* and *incomparable*. However, *incomparable* can be filtered in the element extraction stage, so we only consider the remaining three opinions.

Existing studies for comparative opinion are mainly rule-based or machine-learning methods. In general, comparative elements extraction is first performed, to identify comparative sentences and to extract compared targets and aspects. Jindal and Liu (2006a) first identify comparative sentences from review. Hu and Liu (2006); Ding et al. (2009); Xu et al. (2009) extract comparative elements from the identified comparative texts. With comparative text and its comparative elements, Ganapathibhotla and Liu (2008) design six rules based on context and pre-defined pros and cons in review to classify comparative opinion. Panchenko et al. (2019) evaluate a few classifiers with features for comparative opinion classification. Despite that deep learning based solutions have significantly advanced the

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 7064–7075
December 7-11, 2022 ©2022 Association for Computational Linguistics

area of sentiment analysis in recent years, to the best of our knowledge, no dedicated deep learning models have been proposed for comparative opinion classification.

A major challenge in comparative opinion classification is that, opinion o depends on the order of targets t_1, t_2 (*i.e.,* t_1 is better than t_2 means that t_2 is worse than t_1) except when o is *same*. For this reason, sentiment analysis models that directly predict positive or negative, cannot well handle comparative opinion classification. To overcome this problem, we design a novel twin framework to detect comparative opinion. In this framework, a primary channel and a mirror channel are designed to capture both the original (*i.e.,* for the order t_1, t_2) and the reversal (for order t_2, t_1) comparative opinions. Both channels are realized by prompt-based learning in our framework.

Specifically, our proposed CORT (**C**omparative **O**pinion **R**epresentations from **T**win network) model contains two opinion channels (*i.e., primary channel* and *mirror channel*), and a *comparative module*. Each channel includes three cells: *prompter*, *encoder*, and *classifier*. Given an input in the form of (text, target t_1, target t_2, aspect), the prompter generates a template like "[target t_1] is $[\text{MASK}]$ than [target t_2] in [aspect]". Then the encoder encodes the original input text and the template to get a global representation (*i.e.,* encoding at the $[\text{CLS}]$ position) and the opinion representation (*i.e.,* encoding at the $[\text{MASK}]$ position). Lastly, the comparative opinion is predicted by a classifier. Mirror channel shares the same configurations as the primary channel. The only difference is that the two targets are swapped in the generated template.

To the best of our knowledge, this is the first attempt to design a prompt-based learning framework for comparative opinion classification. We demonstrate that CORT achieves state-of-the-art performance against all existing baselines on three public datasets, namely CameraReview, CompSent-19, and CompSent-08. More importantly, our CORT model is robust and is insensitive to the order of targets in comparison.

2 Related Work

Comparative opinion expresses opinions by comparing similar targets, which is different from directly expressing an opinion about targets and their aspects (Liu, 2012). Simply put, comparative opinion mining is the analysis of the contrast between multiple targets/objects (Ganapathibhotla and Liu, 2008). Generally, there are two main subtasks: (i) elements extraction, to extract comparative sentences, targets, aspects, and (ii) comparative opinion classification. This paper focuses on the latter. Very few studies consider both subtasks (Liu et al., 2013, 2021b). As our model is built on top of prompt-based learning, we also briefly review pre-trained language models for sentiment analysis.

2.1 Comparative Opinion Classification

The task of comparative opinion mining was formulated between 2006 to 2008 (Jindal and Liu, 2006b; Ganapathibhotla and Liu, 2008). Early approaches are mostly based on feature engineering and manually defined rules.

Rule-based Methods. Ganapathibhotla and Liu (2008) design six rules to identify which target is more preferred. After that, Tkachenko and Lauw (2014) propose a generative model for comparative sentences from online reviews, to define comparative directions of targets. Again, rules are used to predict the preference between multiple targets. In general, rule-based method is expensive to maintain and is heavily domain-dependent.

Traditional Machine Learning. Feature engineering with classifier (*i.e.,* Logistic Regression, Random Forest, Support Vector Machine *et al.*) was the mainstream approach in the past. Panchenko et al. (2019) build a corpus of comparative sentences, then evaluate multiple supervised models. In their evaluation, comparative sentences are divided into three categories including *none*, *better*, and *worse*. They do not take into account the instances with *same* opinion.

2.2 Pre-trained Language Model

Pre-trained language models (PLMs) including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019) *et al.*, have been widely used in natural language processing tasks, including sentiment analysis. Fine-tuning (Ding et al., 2021; Han et al., 2021) is a popular method for downstream tasks *e.g.,* classification (Xu et al., 2019), generation (Liu et al., 2019) *et al.*. However, fine-tuning methods typically require a large amount of annotated data (Chen et al., 2021). Then, prompt learning is proposed to solve this problem by filling the gap of objective forms between pre-training and fine-tuning (Brown et al., 2020; Han et al., 2021; Liu et al., 2021a; Ben-David et al.,

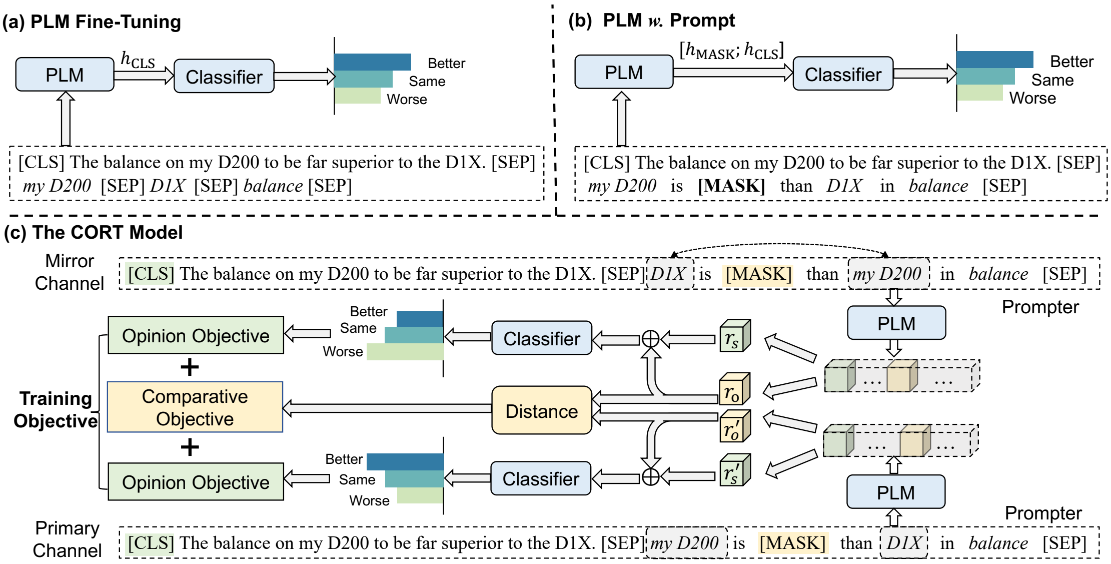
(a) PLM Fine-Tuning
PLM
h_{CLS}
Classifier
Better
Same
Worse
[CLS] The balance on my D200 to be far superior to the D1X. [SEP]
my D200 [SEP] D1X [SEP] balance [SEP]
(b) PLM w. Prompt
PLM
$[h_{\text{MASK}}; h_{\text{CLS}}]$
Classifier
Better
Same
Worse
[CLS] The balance on my D200 to be far superior to the D1X. [SEP]
my D200 is [MASK] than D1X in balance [SEP]
(c) The CORT Model
Mirror Channel
[CLS] The balance on my D200 to be far superior to the D1X. [SEP] D1X is [MASK] than my D200 in balance [SEP]
PLM
Prompter
Opinion Objective
Better
Same
Worse
Classifier
r_s
r_o
Training Objective
Comparative Objective
Distance
r_o'
r_s'
Opinion Objective
Better
Same
Worse
Classifier
PLM
Prompter
Primary Channel
[CLS] The balance on my D200 to be far superior to the D1X. [SEP] my D200 is [MASK] than D1X in balance [SEP]

Figure 1: The architectures of (a) PLM Fine-Tuning, (b) PLM *w.* Prompt, and (c) CORT. The prompter of CORT is the same as PLM *w.* Prompt. For each opinion channel, r_s and r_o denote the global representation ([CLS] position) and opinion representation ([MASK] position), respectively. The classifier of each channel takes the concatenated representation $r = r_s \oplus r_o$ as input, and predicts opinion distribution $\mathcal{P}$.

2022). To the best of our knowledge, neither dedicated PLM nor prompt learning has been applied to comparative opinion classification.

3 CORT Model

The **C**omparative **O**pinion **R**epresentations from **T**win network (**CORT**) has its root in prompt learning. We first brief Pre-trained Language Model with Prompt (PLM *w.* Prompt) for comparative opinion classification. Then we detail the design of CORT model and its optimization method.

3.1 PLM *w.* Prompt

How to effectively use target information is vital, when classifying the opinion in a comparative sentence with two targets. The straightforward approach is to use the fine-tuning method for prediction, by using the global representation obtained at the [CLS] position, shown in Figure 1 (a). Here, the targets and the corresponding aspect are appended to the comparative sentence. As the opinion is sensitive to the order of targets, it is more reasonable to adopt prompt learning, by including more contextual information about the targets through prompts, see Figure 1 (b).

Preliminary: Prompt Learning. Prompt learning relies on a pre-defined set of label words V^* and a template $\mathcal{T}$. Given an input text x, $\mathcal{T}$ modifies the original text into a prompt input, by adding some words including [MASK] to the original input. Conventionally, the representation at the location of [MASK] is used to predict the masked word w. For each label y in $\mathcal{Y}$, a label word set $V_y = \{v_1, v_2, \ldots, v_n\}$ is defined, which is a subset of vocabulary in PLM. With each label maps to a set, all sets together form a set of label words V^*. Thus, in prompt learning, a classification problem is transferred into a mask learning problem, or formally as follows:

$$p(y|x) = p\left([\text{MASK}] = w|\mathcal{T}(x)\right) \qquad (1)$$

The architecture of the proposed PLM *w.* Prompt is shown in Figure 1 (b). Given a comparative sentence as input, we generate the input of PLM with a template: "[CLS] text [SEP] [target t_1] is [MASK] than [target t_2] in [aspect] [SEP]". Because the comparative opinion is nuanced, we use the global representation to enhance the representation of the masked location. Hence, we do not use the vocabulary table to predict the opinion. Instead, we use a softmax classifier to classify the opinion based on the concatenated representation of [CLS] and that of [MASK] position. Cross entropy objective is used to optimize this model.

3.2 The CORT Model

Figure 1 (c) depicts the architecture of CORT. Based on the twin framework, CORT has two opinion channels (*i.e., primary channel* and *mirror* *channel*), and a *comparative module*. Each channel consists of three cells: *prompter*, *encoder*, and *classifier*. As the name suggests, *prompter* assigns template for input text; *Encoder* encodes the input with template, to produce the text representation r_s at [CLS] position and the opinion representation r_o at [MASK] position; *Classifier* then predicts the opinion of the channel. Parameters for encoders and classifiers are shared across two channels.

Last, *comparative module* is used to contrast differences between opposite opinions, which could improve the robustness of the model from the stance of contrastive representations.

3.2.1 Twin Opinion Channels

The difference between *primary channel* and *mirror channel* is the order of the two targets in comparison. In the example shown in Figure 1, the template for primary channel is "*my D200* is [MASK] than *D1X* in *balance*", while the template for mirror channel is "*D1X* is [MASK] than *my D200* in *balance*". Correspondingly, the ground truth labels for the two templates are opposite in training, *e.g.,* better and worse respectively in this example. Due to their similar structure, we only describe the primary channel.

Prompter Cell. The prompter cell in CORT has the same structure as PLM *w.* Prompt. Given a sentence S, targets t_1, t_2, aspect a, the prompter cell generates input with template S_p for primary channel: "[CLS] S [SEP] t_1 is [MASK] than t_2 in a [SEP]". Similarly, the generated text with template S_m for mirror channel is:"[CLS] S [SEP] t_2 is [MASK] than t_1 in a [SEP]".

Encoder Cell. The generated text is encoded by PLM. Taking the input S_p in primary channel, we obtain r_s and r_o, denoting the representations of entire context (*i.e.,* hidden representation at [CLS] position) and the opinion representation (*i.e.,* hidden representation at [MASK] location), respectively.

$$r_s, r_o = \mathrm{PLM}(S_p). \tag{2}$$

The final representation is the concatenation of the two: $r = r_s \oplus r_o$. In our experiments, we evaluate three popular PLMs, namely, RoBERTa, BERT, and XLNet.

Classifier Cell. The opinion distribution $\mathcal{P}$ is computed by a $\mathrm{softmax}$ classifier based on the learned representation r:

$$\mathcal{P} = \mathrm{softmax}(W_p r + b_p), \tag{3}$$

where W_p and b_p are the learnable parameters.

The twin framework is designed to reflect semantic meaning of comparative opinion. For instance, "my D200 is *better* than the D1X in auto white balance" and "the D1X is *worse* than my D200 in auto white balance" mean the same, despite the order change in targets. Hence, mirror channel is computed in a similar manner:

$$r_s', r_o' = \mathrm{PLM}(S_m) \tag{4}$$

$$\mathcal{P}' = \mathrm{softmax}(W_p r' + b_p) \tag{5}$$

Here, r_s' and r_o' are the global representation and the mask representation of the mirror channel. The opinion distribution $\mathcal{P}'$ is computed in the same manner using the final representation $r' = r_s' \oplus r_o'$. Note that the ground truth labels of $\mathcal{P}$ and $\mathcal{P}'$ are opposite when the opinion is not *same*.

During testing, CORT generates two probability distributions $\mathcal{P}$ and $\mathcal{P}'$. We use the maximum value from them to assign the comparative opinion.

3.2.2 Comparative Module

Again, due to the order change in targets, when the opinion in the primary channel is *better*, the corresponding opinion in mirror channel is *worse*, and vice versa. Hence, the comparative module aims to maximize the distance between two opinion representations when the opinion is *better* or *worse*. Simultaneously, the module minimizes the distance of two opinion representations when the opinion is *same*. To be detailed next, we design our training objective by considering the distance computed by comparative module:

$$d = 1 - \cos(W_o r_o + b_o, W_o r_o' + b_o), \tag{6}$$

where d is the distance by $\cos$ similarity; W_o and b_o are learnable parameters.

3.3 Training Objective

CORT model has two learning objectives: opinion objective and comparative objective. Opinion objective is to minimize the cross-entropy of the opinion probability distributions for both channels. Comparative objective is to maximize the distance of [MASK] representations if the opinion is *better* or *worse*, and minimize the distance for *same*.

Opinion Objective. From the two channels, we have two opinion probability distributions $\mathcal{P}$ and $\mathcal{P}'$ for an instance i. Inspired by Wang et al. (2016, 2019); Liu et al. (2022); Lin et al. (2022), the opinion probability objective $J(\theta)$ adopts cross-entropy losses on both channels:

$$\Phi = \sum_i \text{cross-entropy}(y_i, \mathcal{P}_i) \tag{7}$$

$$\Phi' = \sum_i \text{cross-entropy}(y'_i, \mathcal{P}'_i) \tag{8}$$

$$J(\theta) = \lambda\Phi + \mu\Phi' \tag{9}$$

Here, Φ and Φ' are losses of the two channels; y_i and y'_i are the annotated opinions of the two channels for instance i; λ and μ are hyperparameters.

Comparative Objective. We use hinge loss for comparative objective $U(\theta)$:

$$U(\theta) = \sum_i \begin{cases} d_i, & \text{if } o_i = \text{same} \\ \max(0, 1 - d_i), & \text{if } o_i \neq \text{same} \end{cases} \tag{10}$$

Considering both objectives, the final objective L is the sum of J and U:

$$Loss(\theta) = J(\theta) + \xi U(\theta), \tag{11}$$

where ξ is a hyperparameter.

4 Experiment

We now evaluate the proposed base model PLM *w.* Prompt and CORT for comparative opinion classification, against baselines.

4.1 Dataset

CameraReview. Created by Kessler and Kuhn (2014), this dataset[1] contains comparative sentences about camera reviews in English. Each instance is annotated with labels (target t_1, target t_2, aspect, opinion). The set of opinion is {*better*, *worse*, *same*}. We select the sentences with clear direction in two targets from this dataset, and split the instances with the ratio of 7:1:2 for training, validation, and testing. Table 1 reports statistics of this dataset.

CompSent-19. This dataset[2] annotates comparative sentences on computer science with compared targets like programming languages (*e.g.,* C++, Python, Java), database products (*e.g.,* MYSQL, Oracle) and technology standards (*e.g.,* Bluetooth or Ethernet) (Panchenko et al., 2019). The annotation format is (target t_1, target t_2, opinion), without aspect. The set of opinion is {*better*, *worse*}.

Table 1: The statistics of CameraReview, CompSent-19 and CompSent-08 datasets, marked as "C.R.", "C.S.-19" and "C.S.-08" in this table. Note that CompSent-08 is very small, so we only use it as a test set. CameraReview and CompSent-19 are split with the ratio 7:1:2 for training, validation and testing.

Opinion	Dataset	C.R.	C.S.-19	C.S.-08
Better	Train	809	746	–
	Valid	124	113	–
	Test	231	232	119
Worse	Train	220	348	–
	Valid	25	44	–
	Test	71	82	30
Same	Train	216	–	–
	Valid	27	–	–
	Test	55	–	37
Total		1,778	1,565	186

CompSent-08. Despite its small size, this dataset[3] by Jindal and Liu (2006a) contains comparative sentences in various domains, ranging from digital cameras to soccer. The sentences are taken from reviews, blog posts, and forum discussions. The annotation format is similar to CameraReview, including target t_1, target t_2, aspect, and opinion. Due to the small size, we only use this dataset as a test set, to evaluate the generalization performance of our model.

4.2 Compared Methods

For completeness, we compare our models with several baselines including rule-based methods, traditional machine learning methods, and neural models.

Rule. Ganapathibhotla and Liu (2008) develop six rules to find which target is more preferred. It does not consider the *same* opinion. Because the codes are not released, and the *same* opinion is missing, we implement the same six rules, and develop additional rules for *same* in our experiments. Specifically, our rules for *same* opinion are built with opinion words (*i.e.,* same, like, similar, equal) reflecting the *same* opinion.

Traditional Machine Learning. We follow the methods in Panchenko et al. (2019) to experiment with traditional machine learning methods: Logis-

[1] https://wiltrud.hwro.de/research/data/reviewcomparisons.html

[2] https://github.com/uhh-lt/comparative/tree/master/Classification/code/data

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

Table 2: The accuracy, $F1$, and detailed $F1$ of methods on CameraReview and CompSent-19 datasets. The baseline methods marked with $*$ are our own implementation, and the methods marked with † are implemented based on the open codes (Panchenko et al., 2019). Best results are in bold face and second best underlined.

Method	CameraReview					CompSent-19			
	Acc.	$F1$	$F1$-B.	$F1$-W.	$F1$-S.	Acc.	$F1$	$F1$-B.	$F1$-W.
Rule*	0.609	0.467	0.753	0.263	0.384	–	–	–	–
Majority Class†	0.647	0.262	0.786	0.000	0.000	0.739	0.425	0.850	0.000
Extra Trees†	0.647	0.285	0.784	0.000	0.070	0.764	0.574	0.859	0.289
Random Forest†	0.661	0.327	0.792	0.054	0.136	0.777	0.614	0.865	0.364
k-Neighbors†	0.636	0.373	0.782	0.083	0.254	0.736	0.604	0.832	0.376
SGD Classifier†	0.675	0.439	0.808	0.141	0.368	0.739	0.662	0.823	0.500
Decision Tree†	0.616	0.451	0.779	0.351	0.222	0.707	0.649	0.792	0.505
AdaBoost†	0.661	0.488	0.778	0.346	0.341	0.726	0.653	0.812	0.494
SVM†	0.726	0.552	0.838	0.460	0.359	0.758	0.684	0.837	0.531
XGBoost†	0.731	0.559	0.830	0.379	0.468	0.768	0.685	0.846	0.523
Logistic Regression†	0.720	0.583	0.821	0.441	0.488	0.761	0.673	0.843	0.503
CRF*	0.777	0.649	0.850	0.394	0.703	–	–	–	–
RNN-Capsule†	0.675	0.529	0.809	0.328	0.451	0.672	0.563	0.781	0.344
Multi-Stage$_{\text{BERT}}{}^{*}$	0.661	0.498	0.785	0.228	0.482	0.790	0.683	0.867	0.500
RoBERTa Fine-Tuning	0.857	0.802	0.916	0.677	0.814	0.844	0.797	0.895	0.699
+ data augmentation	0.852	0.822	0.897	0.711	0.857	0.908	0.881	0.937	0.824
RoBERTa *w.* Prompt	0.877	0.826	**0.931**	0.719	0.829	0.892	0.857	0.927	0.788
+ data augmentation	0.871	0.847	0.906	0.736	**0.900**	0.924	0.899	0.949	0.850
CORT	**0.885**	**0.861**	0.918	**0.784**	0.880	**0.933**	**0.913**	**0.955**	**0.871**

tic Regression, XGBoost, SVM, AdaBoost, Decision Tree, SGD classifier, k-Neighbors, Random Forest, Extra Trees, and Majority Class.

CRF. Conditional Random Field (CRF) is used for comparative elements extraction (Sutton et al., 2007). We build five features to adopt CRF to comparative opinion classification. The five features are *word*, *position*, *entity*, *POS tag* and *word label*. Here, *word label* indicates whether a word is part of t_1, t_2, a, or opinion words. If yes, then *word label* is a special tag, otherwise, *word label* is "None" (See Appendix A.2 for more details about the others).

RNN-Capsule. RNN-Capsule (Wang et al., 2018) is a powerful model for sentiment classification (*e.g.,* positive, negative, and neutral). To adapt to comparative opinion classification, we enrich the original input by appending the comparative elements (*i.e.,* target t_1, target t_2, and aspect) to the end of the original sentence, as input to RNN-Capsule.

Multi-Stage$_{\text{BERT}}$. Multi-Stage$_{\text{BERT}}$ (Liu et al., 2021b) extracts comparative elements and detects comparative opinion, in a pipeline setting. For a fair comparison of the comparative opinion classification subtask, we use the ground truth targets and the aspect as input, instead of the extracted elements by the model. Note that this model needs a text span with an opinion as input, but the datasets do not provide such annotations. So we only use the remaining elements, including target t_1, target t_2, and aspect.

PLM Fine-Tuning. Based on PLM (*e.g.,* BERT, RoBERTa, XLNet), fine-tuning method makes prediction with an extra linear layer after PLM (Chen et al., 2021). Similar to RNN-Capsule, we add the comparative elements (*i.e.,* target t_1, target t_2, and aspect) to the end of the original text as the input to PLM Fine-Tuning. Then the [CLS] representation is used to predict the opinion.

For a fair comparison, we also experiment *PLM Fine-Tuning* and *PLM w. prompt* with data augmentation (by adding a copy of a training instance with target order changed and opinion reversed, if the opinion is *Better* or *Worse*). Data augmentation benefits the model with additional training data and naturally avoids data imbalance. Conceptually, this setting is similar to training with dual channels.

4.3 Overall Performance Comparison

We use accuracy, macro-$F1$, and detailed $F1$ of each opinion, to compare all methods. Results on CameraReview and CompSent-19 datasets are reported in Table 2.[4] Because rule and CRF models

[4] Because CompSent-08 is very small, it is only used as test data and evaluated in Section 4.6.

Table 3: The accuracy, $F1$, and detailed $F1$ of neural models with swapping targets on CameraReview dataset. The downwards/upwards arrow indicates the performance change compared to the setting without swapping targets in test set.

Model	Accuracy	$F1$	$F1$-Better	$F1$-Worse	$F1$-Same
RNN-Capsule	0.249 ↓ 0.43	0.284 ↓ 0.25	0.235 ↓ 0.57	0.199 ↓ 0.13	0.418 ↓ 0.03
Multi-Stage$_{\text{BERT}}$	0.300 ↓0.36	0.334 ↓0.16	0.287 ↓0.50	0.253 ↑ 0.03	0.463 ↓ 0.02
RoBERTa Fine-Tuning	0.221 ↓ 0.64	0.356 ↓ 0.45	0.143 ↓ 0.77	0.064 ↓ 0.61	0.862 ↑ 0.05
RoBERTa *w.* Prompt	0.252 ↓0.63	0.354 ↓0.47	0.134 ↓0.80	0.162 ↓0.56	0.767 ↓0.06
CORT	**0.885** ↓ 0.00	**0.861** ↓ 0.00	**0.784** ↓ 0.13	**0.918** ↑ 0.13	**0.880** ↓ 0.00

need features that heavily depend on the specific dataset, it is expensive to build domain-specific features for CompSent-19. Their results are unavailable on CompSent-19, marked with "-" in Table 2.

Discussion. On both datasets, as expected, PLM-based models outperform all other baselines, revealing the powerful ability of PLM models. Among PLM-based models, CORT is the winner, followed by RoBERTa *w.* Prompt with data augmentation. The performance gap between them clearly indicates the ability of our proposed twin framework.

Reported in the Table 2, the use of data augmentation leads to 2.0 and 2.1 points increase in $F1$ on CameraReview, compared to the models without data augmentation, respectively. In addition to the increase in $F1$, the models using data augmentation become stable on reversed data, and produce similar $F1$ scores (see Table 4). On the other hand, both models remain much poorer than CORT.

Compared to PLM Fine-Tuning or PLM *w.* Prompt (using data augmentation or not), CORT benefits from the following design to achieve the best results: (1) The input to the classifier of both channels is the concatenation of [CLS] and [MASK] representations, because both of them provide important information for classification. (2) The input to the comparative module considers the [MASK] representations only, but not [CLS]. This is because [CLS] representation denotes the whole representation of the input text. By design, the order of the targets (t_1 and t_2) in the two channels are different, hence [CLS] representations are always different and there is no need to compare. For [MASK] representations, comparative module needs to distinguish the cases when the comparison is *Same*, and the cases when the comparison is *Better* or *Worse*.

Traditional machine learning methods outperform rule-based method. CRF is the best performing traditional machine learning method, with $F1$ of 0.649 on CameraReview. RNN-Capsule does not deliver good performance as it is not designed for comparative sentence classification. Further, RNN-Capsule is based on pre-trained word embeddings, not PLM. Surprisingly, Multi-Stage$_{\text{BERT}}$ is poorer than many traditional machine learning models. One reason is that its $\mathrm{softmax}$ classifier only takes the concatenation of representations of compared targets and the aspect as input. That is, even if PLM is used, an effective design is essential for good performance. Methods like Extra Trees and Majority Class are very sensitive to the data distribution of labels. They tend to predict all instances to the majority label *i.e., better*, resulting in very low $F1$'s for *worse* and *same*.

4.4 Robustness of CORT on Reversal Data

As aforementioned, comparative opinion classification requires semantic understanding in the sense that, if a is better than b, then b is worse than a. In this set of experiments, we evaluate model robustness by swapping the targets in comparison. That is, a well behaved model shall be able to predict the opposite for *better* and *worse* in reversal data, and *same* when the original opinion is *same*.

Accordingly, in this set of experiments, we keep the training and validation set unchanged but swap the targets t_1 and t_2 in testing, and their corresponding ground truth label. Table 3 reports model performance on CameraReview, and the performance changes against their original performance (see Table 2). In this set of experiments, RoBERTa is used

Table 4: The accuracy and $F1$ of PLM based models on original and reversal test data of CameraReview.

PLM	Module	Original Test Data		Reversal Test Data	
		Accuracy	$F1$	Accuracy	$F1$
RoBERTa	*w.* Fine-Tuning	0.857	0.802	0.221	0.356
	w. Prompt	0.877	0.826	0.252	0.354
	w. CORT	**0.885**	**0.861**	**0.885**	**0.861**
BERT	*w.* Fine-Tuning	0.787	0.714	0.224	0.320
	w. Prompt	0.807	0.733	0.238	0.335
	w. CORT	**0.818**	**0.775**	**0.818**	**0.775**
XLNet	*w.* Fine-Tuning	0.793	0.731	0.246	0.362
	w. Prompt	0.796	0.727	0.244	0.348
	w. CORT	**0.846**	**0.804**	**0.846**	**0.804**

Table 5: Models are trained on CameraReview, then evaluated on CompSent-08 and CompSent-19 as two test sets.

Test set	Model	Accuracy	$F1$	$F1$-Better	$F1$-Worse	$F1$-Same
CompSent-08	RoBERTa Fine-Tuning	0.833	0.787	0.888	**0.646**	0.827
	RoBERTa *w.* Prompt	**0.855**	**0.798**	**0.900**	0.561	**0.933**
	CORT	0.828	0.781	0.879	0.635	0.829
CompSent-19	RoBERTa Fine-Tuning	0.778	0.702	0.855	0.550	–
	RoBERTa *w.* Prompt	0.785	0.740	0.862	0.618	–
	CORT	**0.863**	**0.844**	**0.909**	**0.780**	–

as the PLM encoder.

Our proposed CORT does not change in overall accuracy, F_1 and detailed F_1. In Table 3, the shown changes of CORT are caused by the reversed labeled opinion on *Better* and *Worse* of the original labels. However, big drops are observed for all other models including RNN-Capsule, Multi-Stage$_{\text{BERT}}$, RoBERTa Fine-Tuning, and RoBERTa *w.* Prompt. In particular, $F1$ scores for *better* and *worse* opinions decrease sharply. Thanks to the twin channel design in CORT, our model is trained to handle comparison targets in either order, and experiment results well support the robustness of our design.

4.5 Choices of PLMs for CORT

PLMs have contributed to significant improvements in various NLP tasks. We evaluate the mainstream PLMs including RoBERTa, BERT, and XLNet, on CORT. Table 4 reports the performance of the twin framework based on different PLMs on both the original and the reversal test sets of CameraReview.

The model based on RoBERTa performs the best, followed by XLNet and BERT. The other two models *i.e.,* PLM *w.* Prompt and Fine-Tuning, share the same trend, mainly due to the much larger training data used in RoBERTa. On the reversal test set, our CORT is unaffected and delivers the same performance as the original data for both measures. Significant performance drops happen to PLM *w.* Prompt and Fine-Tuning, and BERT gives slightly worse performance for these two models, compared to other PLMs.

4.6 Cross Dataset Evaluation

To the best of our knowledge, CameraReview is the largest public dataset that comes with comparative opinion annotations. CompSent-08 dataset has only 186 instances, and is insufficient to train a model. As the annotation scheme of CompSent-08 is similar to that of CameraReview, it is interesting to find out whether the CORT model trained on CameraReview dataset could be used to identify comparative opinion on CompSent-08. For completeness, we further evaluate the model trained on CameraReview on the full CompSent-19 dataset as a test set.

Reported in Table 5, all PLM-based models perform very well on CompSent-08, even though these models are trained on a different dataset *i.e.,* CameraReview. Interestingly, PLM *w.* Prompt performs better than CORT on CompSent-08. Through manual investigation, we note that a few incorrect predictions on CompSent-08 lead to big changes in performance numbers due to its small size. When using the full CompSent-19 as a test set, the proposed CORT shows clear superiority over alterna-

Table 6: Ablation study of CORT on CameraReview.

Model	Accuracy	$F1$
CORT	**0.885**	**0.861**
$w.o.$ comparative module	0.863	0.830
$w.o.$ mirror channel	0.885	0.841
$w.o.$ [CLS] representations	0.868	0.828
$w.o.$ [MASK] representations	0.874	0.845

tives. Note that, due to the differences in train and test sets, the results in Table 5 cannot be directly compared to the numbers in Table 2. Nevertheless, the high accuracy and $F1$ numbers in Table 5 do suggest that our CORT is generalized to similar comparative opinion classification tasks.

4.7 Ablation Study

We conduct ablation study on CameraReview dataset. Thanks to the simple design of the twin framework, we could evaluate the effectiveness of the comparative module and the mirror channel easily. In addition, to study the effect of [CLS] and [MASK] representations, we also conduct experiment without these representations.

Table 6 reports the results of detailed comparison: (i) Removal of comparative module leads to 3.1 points decrease in $F1$. Further, $F1$ decreases 2.0 points after removal of the mirror channel. These shows that both comparative module and the twin opinion channels are effective. (ii) Removal of [CLS] representations (r_s and r'_s) on both channels leads to 3.3 points decrease in $F1$. This result suggests that the context of the entire sentence in both channels is helpful for the classification task. (iii) Removal of [MASK] representations (r_o and r'_o), leads to a drop of 1.6 points in $F1$. This result shows that not only opinion representations are important, but also text representations are vital for classification.

5 Conclusion

In this paper, we focus on comparative opinion classification, a specific sentiment analysis subtask. Built on the top powerful pre-trained language models, we show that comparative opinion classification can be addressed by prompt learning with promising accuracy. In our proposed CORT, we designed two channels for comparative targets arranged in either order, to facilitate the model to learn the semantics behind the comparative opinion, *e.g.,* a is better than b vs. b is worse than a. Experiments show that the proposed CORT achieves state-of-the-art performance compared to various baselines, on all comparative datasets. We show that CORT based on the twin framework with different pre-trained language models performs beautifully on both the original and reversal data. We also show that the model achieves good performance in cross-dataset setting, demonstrating its effectiveness and robustness. We believe, as a simple and effective model, CORT well serves as a new baseline for comparative opinion classification.

Limitations

There are two main limitations for comparative opinion classification: dataset and model design. Comparative opinion statements comprise over 10% of the total opinionated text (Kessler and Kuhn, 2013). Hence it is important to study this common linguistic phenomenon. The largest dataset has only about $1k$ instances, which is considered small for neural models. The lack of high-quality and large datasets heavily limits the development in this area.

In this paper, we make the very first attempt to perform comparative opinion classification by dual prompts. By design, the proposed CORT only considers two targets on one aspect. However, comparative text may be expressed in a more complex way. For example, there may be multiple compared targets, on multiple compared aspects. Further, the proposed CORT does not consider the situation that one of the compared targets is a pronoun. All of these are important factors for further exploration.

Acknowledgments

This work was supported by the National Key R&D Program of China (2020AAA0105200) and the National Science Foundation of China (NSFC No. 62106249, 61902382, 61972381).

References

Eyal Ben-David, Nadav Oved, and Roi Reichart. 2022. Pada: Example-based prompt learning for on-the-fly adaptation to unseen domains. Transactions of the Association for Computational Linguistics, 10:414–433.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. CoRR, abs/2104.07650.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021. Prompt-learning for fine-grained entity typing. CoRR, abs/2108.10604.

Xiaowen Ding, Bing Liu, and Lei Zhang. 2009. Entity discovery and assignment for opinion mining applications. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009, pages 1125–1134. ACM.

Murthy Ganapathibhotla and Bing Liu. 2008. Mining opinions in comparative sentences. In COLING 2008, 22nd International Conference on Computational Linguistics, Proceedings of the Conference, 18-22 August 2008, Manchester, UK, pages 241–248.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. PTR: prompt tuning with rules for text classification. CoRR, abs/2105.11259.

Minqing Hu and Bing Liu. 2006. Opinion feature extraction using class sequential rules. In Computational Approaches to Analyzing Weblogs, Papers from the 2006 AAAI Spring Symposium, Technical Report SS-06-03, Stanford, California, USA, March 27-29, 2006, pages 61–66. AAAI.

Nitin Jindal and Bing Liu. 2006a. Identifying comparative sentences in text documents. In SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006, pages 244–251. ACM.

Nitin Jindal and Bing Liu. 2006b. Mining comparative sentences and relations. In Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA, pages 1331–1336. AAAI Press.

Wiltrud Kessler and Jonas Kuhn. 2013. Detection of product comparisons - how far does an out-of-the-box semantic role labeling system take you? In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1892–1897. ACL.

Wiltrud Kessler and Jonas Kuhn. 2014. A corpus of comparisons in product reviews. In Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014, pages 2242–2248. European Language Resources Association (ELRA).

Ting Lin, Aixin Sun, and Yequan Wang. 2022. Aspect-based sentiment analysis through edu-level attentions. In Advances in Knowledge Discovery and Data Mining - 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China, May 16-19, 2022, Proceedings, Part I, volume 13280 of Lecture Notes in Computer Science, pages 156–168. Springer.

Bing Liu. 2012. Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Chengxiang Liu, Ruifeng Xu, Jie Liu, Peng Qu, He Wang, and Chengtian Zou. 2013. Comparative opinion sentences identification and elements extraction. In International Conference on Machine Learning and Cybernetics, ICMLC 2013, Tianjin, China, July 14-17, 2013, pages 1886–1891. IEEE.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. CoRR, abs/2107.13586.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692.

Yiyi Liu, Yequan Wang, Aixin Sun, Xuying Meng, Jing Li, and Jiafeng Guo. 2022. A dual-channel framework for sarcasm recognition by detecting sentiment conflict. In Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022, pages 1670–1680. Association for Computational Linguistics.

Ziheng Liu, Rui Xia, and Jianfei Yu. 2021b. Comparative opinion quintuple extraction from product reviews. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 3955–3965. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.

Alexander Panchenko, Alexander Bondarenko, Mirco Franzek, Matthias Hagen, and Chris Biemann. 2019. Categorizing comparative sentences. In Proceedings of the 6th Workshop on Argument Mining, ArgMining@ACL 2019, Florence, Italy, August 1, 2019, pages 136–145. Association for Computational Linguistics.

Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. 2007. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. J. Mach. Learn. Res., 8:693–723.

Maksim Tkachenko and Hady Wirawan Lauw. 2014. Generative modeling of entity comparisons in text. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014, pages 859–868. ACM.

Yequan Wang, Minlie Huang, Li Zhao, and Xiaoyan Zhu. 2016. Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, pages 606–615. The Association for Computational Linguistics.

Yequan Wang, Aixin Sun, Jialong Han, Ying Liu, and Xiaoyan Zhu. 2018. Sentiment analysis by capsules. In Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018, pages 1165–1174. ACM.

Yequan Wang, Aixin Sun, Minlie Huang, and Xiaoyan Zhu. 2019. Aspect-level sentiment analysis using as-capsules. In The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019, pages 2033–2044. ACM.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 2324–2335. Association for Computational Linguistics.

Kaiquan Xu, Stephen Shaoyi Liao, Raymond Y. K. Lau, Heng Tang, and Shanshan Wang. 2009. Building comparative product relation maps by mining consumer opinions on the web. In Proceedings of the 15th Americas Conference on Information Systems, AMCIS 2009, San Francisco, California, USA, August 6-9, 2009, page 179. Association for Information Systems.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 5754–5764.

Appendix

A Implementation Details

We list the implementation details of the proposed models, to support reproduction.

A.1 Model Hyperparameters of CORT

In our implementation of PLMs, we choose the version "bert-base-uncased" (Devlin et al., 2019), "roberta-base" (Liu et al., 2019), "xlnet-base-cased" (Yang et al., 2019) based on Transformers (Wolf et al., 2020) for BERT, RoBERTa and XLNet models, respectively. In all our experiments, default parameters are used for BERT, RoBERTa, XLNet *encoder*, except batch size, dropout, and learning rate. Specifically, we set batch size to be 16 instances. Dropout is 0.1 for the representations of all models except CORT with BERT. CORT with BERT does not use dropout.

All models are implemented on Pytorch (version 1.11.0), and model parameters are randomly initialized. For the models including PLM Fine-Tuning, PLM *w.* Prompt, and CORT based on RoBERTa and XLNet, AdamW (Loshchilov and Hutter, 2019) is used as optimizer, and we use $1e-5$, $1e-5$ and $1e-4$ as learning rate for RoBERTa, XLNet, and BERT, respectively.

During training, the hyperparameters λ, μ of opinion objective in Equation 9 and ξ of comparative objective in Equation 11 are vital. We use a greedy method to optimize them on both datasets. Experiments results show that the optimized λ, μ and ξ are "0.5, 0.5 and 1.0", "0.9, 0.1 and 1.0", "1.0, 1.0 and 1.0" for RoBERTa, BERT, and XLNet, respectively.

A.2 Details of CRF Baseline

For CRF model, we define five features: *word*, *position*, *entity*, *POS tag*, *word label*. Word label has been described in the paper. *Position* denotes the distance between the current word and the first word in a given sentence. *Entity* refers to entity, recognized by StanfordNLP[5]. *POS tag* is part-of-speech tagging. The implementation is based on CRF++-0.54 [6].

[5] https://stanfordnlp.github.io/stanfordnlp/index.html

[6] https://sourceforge.net/projects/crfpp/files/crfpp/0.54/