Journal of Information Systems and Telecommunication
Vol.12, No.3, July-September 2024, 208-219
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Improving Opinion Mining Through Automatic Prompt Construction

Arash Yousefi Jordehi[1], Mahsa Hosseini Khasheh Heyran[1], Saeed Ahmadnia[1], Seyed Abolghasem Mirroshandel[1*], Owen Rambow[2]

[1].Department of Computer Engineering, Faculty of Engineering, University of Guilan, Rasht, Guilan, Iran
[2].Department of Linguistics, Stony Brook University, Stony Brook, NY, USA

Received: 20 Apr 2024/ Revised: 14 Sep 2024/ Accepted: 30 Sep 2024

Abstract

Opinion mining is a fundamental task in natural language processing. This paper focuses on extracting opinion structures: triplets representing an opinion, a part of text involving an opinion role, and a relation between opinion and role. We utilize the T5 generative transformer for this purpose. It also adopts a multi-task learning approach inspired by successful previous studies to enhance performance. Nevertheless, the success of generative models heavily relies on the prompts provided in the input, as prompts customize the task at hand. To eliminate the need for human-based prompt design and improve performance, we propose Automatic Prompt Construction, which involves fine-tuning. Our proposed method is fully compatible with multi-task learning, as we did so in our investigations. We run a comprehensive set of experiments on Multi-Perspective Question Answering (MPQA) 2.0, a commonly utilized benchmark dataset in this domain. We observe a considerable performance boost by combining automatic prompt construction with multi-task learning. Besides, we develop a new method that re-uses a model from one problem setting to improve another model in another setting as a Transfer Learning application. Our results on the MPQA represent a new state-of-the-art and provide clear directions for future work.

Keywords: Opinion Mining/Sentiment Analysis; Statistical and Machine Learning Methods; Large Language Models; MPQA; Automatic Prompt Construction.

1- Introduction

Extracting opinion entities, such as opinion expressions, opinion holders, and opinion targets is one of the most interesting problems in Opinion Mining (OM), which clarifies *Who expressed or experienced what type of cognitive state toward which entity?* [1, 2, 3, 4]. A cognitive state can be defined as the state of a source (holder or experiencer of the cognitive state, also known as the agent) holding an attitude (via an opinionated expression within the text) toward a target [5, 6]. In this paper, we refer to the opinion expression as the expression, the opinion holder as the agent, and the opinion target as the target. We refer to the (expression, role, relation) triplet as an opinion structure. In fine-grained OM, an expression might be coupled with one or more roles (agents and/or targets). An expression may also not have any agent or target [7, 4]. In this paper, we focus on detecting expressions, agents, and targets, and the structures they form. Similar to much previous research [8, 2, 3, 1], we have focused on a subset of the Multi-Perspective Question Answering (MPQA) 2.0 dataset in our paper. This subset is frequently used as a benchmark dataset in research focused on detecting opinion expressions and identifying their roles, such as agent and target. The MPQA Corpus contains news articles and other text documents manually annotated for opinions and other cognitive states. Examining prior work indicates that using the aforementioned dataset for fine-grained OM problems is a suitable choice, as most studies have relied solely on this dataset. In our perspective, MPQA is highly complex and has many aspects that remain unexplored. Mastering MPQA, of course, requires significant time. Previous work typically concentrated on using a tagging mechanism in order to label tokens and extract opinion expression and roles. However, the main drawback of using this approach is that it cannot capture cases where there is an overlap between opinion arguments (i.e., roles) of two different

✉ **Seyed Abolghasem Mirroshandel**
mirroshandel@guilan.ac.ir

opinion expressions, as one word can be assigned to only one tag. This approach does not adequately capture the essence of the problem. Xia et al. [3] proposed "SpanOM", in which a two-step algorithm is adopted: 1) Binary prediction on word span in order to detect that the word span is an expression or role or neither. 2) Allocation of opinion relations to the pairs of (expression, role). Their approach solves the issue of overlapping opinion roles mentioned earlier, but on the other hand it has a high computational complexity and lack of explicit interaction between expressions and roles. The most recent research [9], solved the problem by proposing a neural transition model which is highly affected by language knowledge provided to the system. We propose a generative system called Generative Opinion Mining, the "GenOM" system. GenOM avoids the weaknesses of some previous studies and mainly uses modern architectures. Recent studies have demonstrated the success of using transformers [10, 11, 3, 12, 13, 14, 15, 16], such as Text-to-Text Transfer Transformer (T5) [17], which improve the performance compared to traditional machine learning algorithms, manual feature engineering and also deep CNNs and RNNs. In our research, due to the successful results of T5, we aim to utilize it. Furthermore, our model is not dependent on any external natural language prerequisites. In the current study, we address two settings. First, we predict the (expression, role, relation) triplet directly from a sentence (i.e. end-to-end setting). Second, we include the expression in the input and predict its roles (i.e. agents and targets) (called given expression setting). These two settings are also used in a number of previous studies and our proposed GenOM system is capable of performing in both. The problem can be viewed as several related sub-tasks, namely detecting the expression, the agent, and the target from the sentence, and detecting the agent and the target from the sentence and the expression. Because we have distinct but related sub-tasks, we can apply Multi-task Learning (MTL) which strongly improves performance. Following the standard methodology for fine-tuning, we prepend to the sentence (or to the combination of sentence and expression in the given-expression setting) a prompt indicating the sub-task, which then prompts the model to generate that sub-task's output. It turns out that finding an efficient prompt for a given problem by hand and trial-and-error is very time-consuming and problem-dependent. Hence, we propose an approach for finding an efficient prompt automatically, Automatic Prompt Construction (APC). GenOM synthesizes each sub-task's output and uses these results to assemble the predicted triplet (end-to-end setting) or pair (given-expression setting). Then, we measure our system's performance by metrics used in the previous studies. Finally, by comparing our work to other research, we observe successful results of our proposed methods. We also show that using either MTL or APC strongly improves performance, compared to the simple use of transformers, and that these improvements are additive.

The main contributions of our paper lie in the following key points:

- Proposing a generative approach based on MTL to solve the OM problem consists of three main tasks: 1) Expression prediction, 2) Roles prediction through an end-to-end manner, and 3) Roles prediction employing the given-expression method, referred to as the Opinion Role Labeling (ORL) problem in prior literature.
- Suggesting and implementing the APC approach to improve the efficiency of generative prompt-based text-to-text models and obviate the requirement for manually crafted prompts. It offers a novel and efficient approach for optimizing prompts in today's widely used text-to-text language models. This approach could be used for any pre-trained large language model or transformer which accepts (or affected by) prefixes or prompts. It is worth mentioning that the T5 transformer has not been the only choice in the past years. Other text-to-text transformers such as BART [18] and FLAN-T5 [19] have been available since their presentation. However, we presume adopting models like T5 or BART was not successful in previous endeavors of other researchers as they are significantly dependent on input/output structure and prompts provided to them.
- Achieving state-of-the-art (SOTA) and near SOTA results in all benchmark tasks without using external sources of knowledge (e.g., parse tree information), and using only the base version (i.e., medium size in terms of parameters) of T5.

The remainder of this paper is arranged as follows. In section 2, we review previous work. Then, in Section 3, we explain our Generative Opinion Mining, the "GenOM", algorithm. Section 4 introduces the benchmark dataset, the experimental setup, and hyper-parameters with all essential details. In Section 5, after presenting our experimental results, we discuss our results. We also conducted a comprehensive comparison of our approaches with existing

successful algorithms. Finally, in Section 6, we conclude and outline the future work.

2- Related Work

Research in the OM field can be categorized into four groups. The first group consists of opinion expression extraction and labeling [20, 21, 22]. The second group is opinion structure recognition in an end-to-end fashion [23, 24, 25, 9]. The third group of research is Opinion Role Labeling (ORL), which includes the expression in the input (i.e. given-expression setting) as a means to detect its corresponding opinion roles [1, 8, 2]. Afterwards, Xia et al. [3] proposed a system to address OM in the most comprehensive way (i.e., including the end-to-end and given-expression tasks) and to overcome issues observed in the earlier works. As an illustration, inability to capture the semantic dependencies between words which are far apart is mentioned as one drawback of previous research.

Prior research frequently used BMESO-based tags to unravel the problem. BMESO-based tagging tags every token with one of the BMESO tags. B, M, and E tags encode the beginning, middle, and ending word of a role, and the S and O tags represent single-word roles and other words [8]. Therefore, techniques based on Conditional Random Fields (CRF) seemed to be a good choice [21]. Another study [23] recommended using a specific type of recurrent neural networks, called Bi-directional Long Short Term Memory (BiLSTM), in combination with CRF to construct the BiLSTM-CRF model. Their intention is to assign a label to each word in the sentence. Subsequently, they designate the relation to the expression with the set of two features: assigned label and distance to the expression. In other research [25], they designed an end-to-end transition-based system which actually determines the expressions and roles. In other words, they encode the input sentence by a multi-layer BiLSTM. Then, they detect opinion expressions and roles by using manually designed transition actions. Quan et al. [24] derived Bidirectional Encoder Representations from Transformers (BERT) [26] contextualized representations of sentences in order to synthesize BERT and BiLSTM-CRF. In consonance with what was mentioned, models based on sequence tagging are not able to detect opinion roles (agent/target) corresponding to distinct expressions in a sentence.

Table 1. Summary of Key Opinion Mining Studies, Methods, Models, and Identified Research Gaps.

Study	**Methodology**	**Utilized Model**	**Main Findings**	**Limitations**	**Research Gap Addressed by The Study**
Xia et al. [3]	Unified span-based approach with syntactic constituents	SpanOM	Improved detection of opinion expressions and roles using a span-based method.	High computational complexity, lack of explicit interaction between expressions and roles.	Overcomes complexity and enhances interaction through generative modeling.
Wu et al. [8]	Neural transition model joined with PointNet	Neural Transition Model	Successfully detects opinion structures in an end-to-end fashion.	Highly reliant on external syntactic knowledge, limited to end-to-end detection only.	Proposes a generative approach that does not depend on external syntactic knowledge.
Zhang et al. [7]	MTL with Semantic Role Labeling (SRL)	Semantic-aware BiLSTM-CRF	Enhances opinion role labeling by incorporating SRL outputs as inputs.	Depends heavily on SRL outputs, which may not always be available or reliable.	Uses automatic prompt construction without reliance on external knowledge.
Quan et al. [23]	End-to-end joint opinion role labeling with BERT	BiLSTM-CRF with BERT representations	Combines BERT with BiLSTM-CRF for improved contextual understanding.	Struggles with capturing complex opinion relationships and dependencies between	Employs a generative model capable of handling complex opinion structures directly.
Proposed Approach (This Study)	Generative framework using MTL and APC	T5 Transformer with MTL and APC	Achieves state-of-the-art performance, optimizes prompt construction automatically, and integrates end-to-end and given-expression settings for improved accuracy.	Does not rely on external syntactic or semantic knowledge, simplifies model training, and reduces manual prompt design efforts.	Introduces a novel generative approach combining MTL and APC, setting new benchmarks for opinion mining.

On the benchmark dataset, there is some research to adopt a variety of external knowledge to boost the performance. Marasović and Frank [1] used MTL with Semantic Role Labeling (SRL) to address the scarcity of data by leveraging the semantic knowledge. Another team of researchers [8] utilized the SRL outputs as inputs to the OM system which results in a significant boost in performance. In another study [11], they used the rich representations of BERT to be fed in a deep BiLSTM-CRF model. Xia et al. [3] suggested a new method instead of BMESO, that consists of three sub-tasks: 1) Opinion expression detection. 2) Opinion role detection. 3) Opinion relation detection. They perform these sub-tasks in the MTL fashion. In addition, they used syntactic constituents to enhance their performance. However, as noted by Wu et al. [9], it suffers from some issues. For instance, the computational complexity of their approach is very high (i.e., $\mathcal{O}(n^4)$), due to the necessity of processing all possible spans. Also, when their model tries to capture interplays between opinion expressions and roles explicitly, it ends in failure. Recently, Wu et al. [9] designed a complex system for detecting opinion structures only in the end-to-end way. Their system comprises a neural transition model joined with a PointNet [27] in order to accurately find the boundaries of opinion expressions and roles. Similar to some other past research efforts, they utilize external syntax knowledge to improve their system. More precisely, there is a requirement of dependency structure and part-of-speech tags for each input. In Table 1, we provide a concise overview of the studies discussed in this section. This table highlights the key methods, models, and findings of each work, along with the research gaps our proposed approach aims to address.

3- Proposed Method

3-1- Formal Task Definition

We adopt the task definition presented by Xia et al. [3]. Given an arbitrary sentence, say s as input, where $s = w_1, w_2, \dots, w_n$, the system tries to predict the gold-standard opinion triplets $\mathcal{Y} \subseteq \mathrm{E} \times \mathrm{O} \times \mathrm{R}$, where E is the set of opinion expressions defined mathematically as $\mathrm{E} = \{w_i, \dots, w_j \mid 1 \leq i \leq j \leq n\}$, O is the set of opinion roles defined as $\mathrm{O} = \{w_i, \dots, w_j \mid 1 \leq i \leq j \leq n\}$, and R is the set of opinion relations ($\{\text{agent}, \text{target}\}$). While Xia et al. [3] use indices to represent text spans, we take a generative approach and actually generate words. Our proposed method is two-step: we recognize expressions (we can generate expressions standalone because they are not dependent on opinion roles explicitly), and then we predict the opinion role-expression pairs separately. More specifically, we will define three sub-tasks: i) Predicting expressions, ii) Predicting agent-expression pairs, and iii) Predicting target-expression pairs. These tasks could be done separately but we do them jointly, and after the prediction, we form triplets by linking these three sub-tasks' outputs. See Section 3.6 for more details.

3-2- T5 for Conditional Generation

T5 [17] is an encoder-decoder transformer [28] which has been proposed to tackle problems in a generative manner supported by text-to-text learning. More precisely, we use T5 based on conditional generation [29]. The text generation task can be defined as learning a mapping $f: X \longrightarrow Y$ from input X to output Y. Usually, X and Y are sequences of tokens (words), which are denoted by $X = X_1 X_2 \dots X_n$ and $Y = Y_1 Y_2 \dots Y_m$, where $X_i (1 \leq i \leq n)$ and $Y_j (1 \leq j \leq m)$ show the i^{th} and j^{th} token of input and output, respectively. In this kind of problem, the model intends to find Y to maximize the probability (we denote probability of event A by $Pr(A)$ throughout this paper) $Pr_\theta(Y|X)$ based on parameters of the model, θ.

It is possible to insert some additional information in the input of the model. Suppose $P = \{p_1, p_2, p_3, \dots, p_k\}$ is a series of tokens called "prompt tokens" which we prepend to the input X, which gives us the probability $Pr_\theta(Y|[P;X])$. To see the effect of an individual prompt, θ remains fixed. Instead of bounding ourselves to a fixed P, we make P parameterized by θ, and hence it will have its own specific updatable parameters θ_P. This is the basis of the idea of our technique called APC we describe in Section 3.4.

3-3- Multi-Task Leaning (MTL)

As explained in Section 3.1, the expression, agent, and target prediction tasks are related. Previous research [1, 3] stressed the issue of data scarcity and they address it by taking advantage of MTL. We follow them in working with MTL. T5 accepts "prefix" terms, prepended to inputs. Prefixes can be thought as a specific type of prompt (described in Section 3.2). By adding several distinct prompts to the input, we can learn multiple tasks simultaneously, in which we are telling the model what task should be processed, and the model generates output appropriate for that task. When we apply MTL, we are increasing the number of data items (since we can bring in data items for different but related tasks). It can be also considered as a way of data augmentation method. We are comparing this approach to a scenario where a sentence is input into a generative model, and it is expected that the comprehensive output will encompass all tasks.

✉ **Seyed Abolghasem Mirroshandel**
mirroshandel@guilan.ac.ir

3-4- Automatic Prompt Construction (APC)

Inspired by the idea of "prompt tuning" [29], we propose APC approach (as our novel contribution) consisting of two phases:

1) Finding the optimal prompt tokens for a specific task automatically (which is usually called "soft prompt" tuning). We prepend prompt tokens (i.e., tokens of P) to the input tokens, and we try to maximize the likelihood of Y by $Pr_{\theta;\theta_P}(Y|[P;X])$ as the new conditional generation task. By doing backward propagation, gradient updates to the parameter θ_P will take place. After passing input (i.e., X) tokens to the T5 tokenizer, each token is converted to an ID. Then, T5 builds a n by d matrix ($X_e \in \mathbb{R}^{n\times d}$), where n is the length of input tokens and d is the size of embedding vectors contrived for T5 (because T5 comes in different sizes such as small, base, and large). The learnable prompt tokens embedding defined by us are represented as a matrix $P_e \in \mathbb{R}^{k\times d}$. The next thing to do in phase 1 of our method is to append the T5 standard embedding of original input sequence to our updatable prompt embeddings. So, the concatenation of these two forms $[P_e; X_e] \in \mathbb{R}^{(k+n)\times d}$. As a conclusion, in phase 1 only the parameters in P_e are updated.

2) Transferring the optimal prompt tokens learned in phase 1 to be used in the model's fine-tuning. In other words, P_e learned from phase 1 acts as a series of normal tokens, but with the difference that these embeddings representing these tokens might not corresponds to a real word in language. They are new tokens known as "virtual tokens" in the Natural Language Processing (NLP) community.

It should be noted that APC approach is **efficient** regarding the size of trainable parameters. Mathematically, the number of trainable parameters added to the simple fine-tuning is $\mathcal{O}(k \cdot d)$. Even though we consider maximum values, the number of parameters is negligible in comparison to the model parameters when doing fine-tuning.

3-5- MTL + APC

APC can be applied on MTL-based tasks as well. This methodology is our novel contribution. The proposed approach is a combination of fixed prompt (also known as "hard prompt") and soft prompt. More precisely, we extend P a bit more and it is now equal to $P = \{p_1, p_2, p_3, \dots, p_k, p_{k+1}, \dots, p_{k+l}\}$ where the first k tokens are the same trainable tokens as in Section 3.4 which actually is shared among all tasks, and the remaining l tokens are hard prompts that customizes each task.

It is possible to consider a dedicated soft prompt for each task, but our initial experiments indicate no improvement to the results, and furthermore, it is not as efficient as our method in the number of training parameters and runtime. To the best of our knowledge, there are no other similar works for tackling MTL problems in the context of prompt tuning in such a way. In this scenario, the number of tasks does not affect the number of trainable parameters.

As an illustration, Fig. 1 shows the difference between soft prompts and hard prompts. Soft prompts consist of a set of learnable parameters or word embeddings that can be optimized through standard training procedures. In contrast, hard prompts are fixed character strings (i.e., text) that are manually determined and remain constant throughout the process.

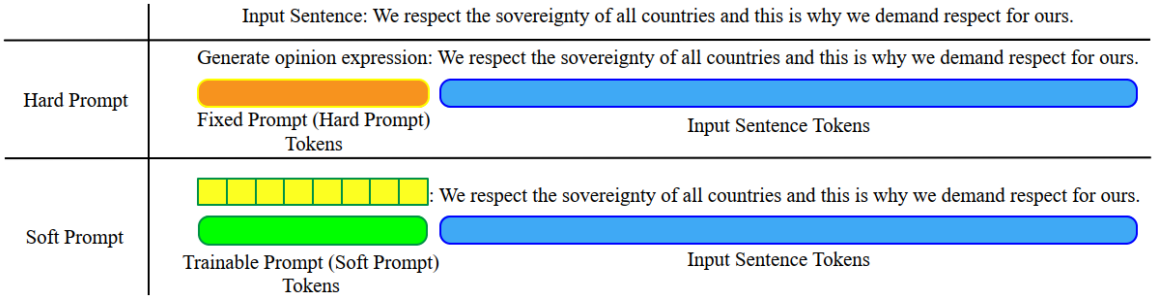

Fig. 1: The comparison between soft and hard prompt.

3-6- Triplet Forming Algorithm

After generating outputs by model, we need to link them in order to find triplets as (expression, role, relation).

3-6-1-End-to-end Setting

In the end-to-end manner, we consider outputs of the expression prediction task as a set of expressions, say $\hat{E}$. Also, the set of predicted agent-expression pairs is designated as $\widehat{AE}$, and set of predicted target-expression pairs as $\widehat{TE}$. We form the set of final predictions as the union of:

$$\{(\xi, \alpha, agent) | \xi \in \hat{E}, (\alpha, \xi) \in \widehat{AE}\} \cup \{(\xi, \tau, target) | \xi \in \hat{E}, (\tau, \xi) \in \widehat{TE}\}$$

which yields us the proper triplets of the task definition presented in Section 3.1. It is notable that when the model predicts an incorrect expression, its agent and target will be ignored in model's evaluation. In other words, correct expression prediction is the precondition for agents and targets evaluation.

3-6-2-Given-Expression Setting

To address the problem in the given-expression setting, we have fed expression with the sentence in the input. Hence, the set of expressions, say E, is revealed and given. With this condition, we only have two outputs: the set of predicted agent items $\hat{A}$, and the set of predicted target items $\hat{T}$. We form the set of final predictions as the union of:

$$\{(\xi, \alpha, agent) | \xi \in E, (\alpha, \xi) \in \widehat{AE}\} \cup \{(\xi, \tau, target) | \xi \in E, (\tau, \xi) \in \widehat{TE}\}$$

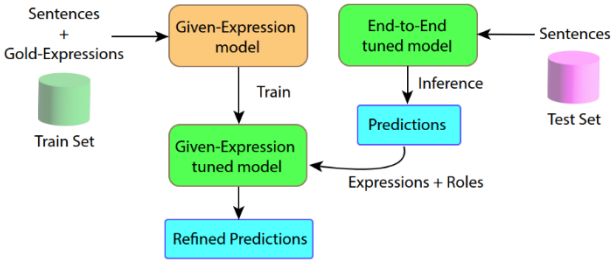

Fig.2: Integrating given-expression model to improve end-to-end predictions by feeding predicted expressions and roles.

3-7- Integrating Given-Expression Model (Int)

As it is shown in Fig. 2, we use predicted outputs of the end-to-end model to be fed into our saved given-expression model in order to see the effect by querying several tasks. It turns out by applying integrating idea, we get a boost in different prediction tasks. In our point of view, based on the error analysis (provided in Section 5.2) and observations of miss-matches, a percentage of false predicted items differs in not important words such as stop-words. Therefore, we believe that the fine-tuned end-to-end model outputs are of sufficiently high quality to accurately identify expression spans, even though some words at the beginning or end may occasionally be omitted. Performance of our system in the given-expression setting demonstrates its excellence as well. By supplying these expressions to the given-expression model, we can identify those that closely resemble gold standard expressions and determine the corresponding roles for each one. Then, by comparing the new predictions with the old ones, we can revise the predictions. This revision process mainly relies on the correlation rate between the two sets of predictions and is based on the improvements observed in the development set performance. We suggest this application of models to be considered as a novel idea of Transfer Learning in NLP.

In this part of our research, we explain the dataset we exploited, the evaluation metrics we report, the setup of our experiments and other details involving training procedure.

3-8- Dataset and Settings

As mentioned earlier, we employed the most frequently used dataset, MPQA 2.0, in order to carry out our experiments. We mimic the data split of previous work [9, 3, 8, 2] and conduct 5-fold cross-validation run. We set the random seed to a constant number to make the results reproducible. It's worth mentioning that, in line with prior research as well as considering the intricate complexity and granularity of the MPQA, our exclusive focus has been on this dataset.

Our models were developed using the PyTorch [1] deep learning framework and we performed the models on a single NVIDIA A100-SXM4-40GB GPU. We also utilized packages such as spaCy[2] and NLTK[3], along with the scikit-learn library [4], NumPy [5], and Matplotlib [6]. The T5-base model and its tokenizer, which were obtained from the

[1] https://pytorch.org/
[2] https://spacy.io/
[3] https://www.nltk.org/
[4] https://scikit-learn.org/stable/
[5] https://numpy.org/
[6] https://matplotlib.org/

✉ **Seyed Abolghasem Mirroshandel**
mirroshandel@guilan.ac.ir

Hugging Face Transformers library[1], were also employed in our implementation.

3-9- Details of Input/Output Design

Since we are using T5 to solve this OM problem in the generative way, the design of input and output structure is essential. Hence, we will go into detail by examples in this section. Please note that samples of input and output are presented here are real ones used in implementations. To develop the input, we use prompts to make tasks distinguishable and more learnable by T5. In the end-to-end approach, we use one prompt set, but it also possible to have more. For the given-expression setting, we give the sentence and the expression using two different prompt sets. Upper parts of Fig. 3, 4, and 5 indicate input structure used in our system. At the output, in the end-to-end setting we use "=" (equal sign) to show allocation of an opinion role (agent/target) to an expression, i.e. "agent1 = expression1". If there are more than one item, we split them by "|" (pipe) symbol. It operates precisely in accordance with the triplet forming algorithm described in the end-to-end setup (Section 3.6.1). Finally, for the given expression manner, we only divide opinion role spans by "|" sign. As depicted in the lower part of Fig. 3, outputs for opinion roles are forming pairs of (role, expression), which are grouped with the "=" character. It functions accurately in alignment with the triplet forming algorithm outlined in the given-expression setup (Section 3.6.2). Despite this configuration and design of this type of output reflecting the concept pretty well, it also results the best among other choices we examined in our initial experiments.

Nevertheless, we do not require the expression prediction sub-task to produce its outputs in this way. The reason we are doing this is to make a harmony between all of tasks' outputs. It seems if input and output of several tasks at hand executing by T5, be quite identical in format, the model performs better. For illustration, see Fig. 3, 4, and 5. Prefixes are highlighted in red. Sentences are highlighted in aqua. Opinion roles and expressions are highlighted in yellow and green respectively. Prefix (prompt) phrases are depicted in all pictures are examples and they could get changed and replace by the APC mechanism. In our assertion that the fixed set of prompts could be substituted by the APC mechanism, we are referring to a system that automatically adds a collection of learnable vectors to the prompts throughout the training process. These vectors are dynamically updated, thereby improving the model's performance across various tasks. It is essential to highlight that these vectors may not directly represent actual words (unlike our standard prompt words); rather, they exist exclusively within the embedding space and are inserted at appropriate positions in the input sequence when converting words into their embedding vectors.

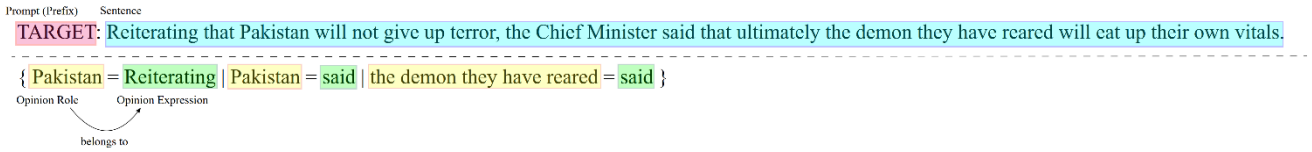

Fig. 3: An example of input (up) and output (down) with multiple opinion role-expression pairs used in the experiments in the end-to-end setting.

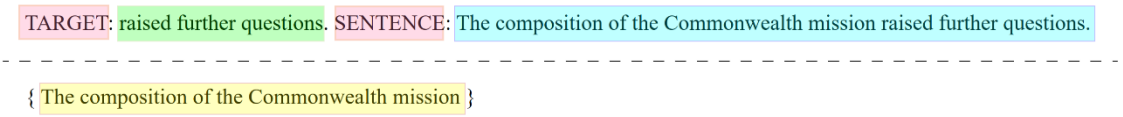

Fig. 4: An example of input (up) and output (down) with one opinion role used in the experiments in the given-expression setting when querying its targets

[1] https://github.com/huggingface/transformers

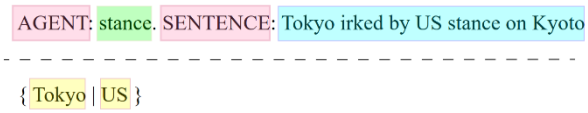

Fig. 5: An example of input (up) and output (down) with two opinion roles used in the experiments in the given-expression setting when querying its agents.

3-10- Hyper-parameters and Training Details

The number of prompt tokens is an important hyper-parameter in our experiments. We tried different number of tokens in our experiments (i.e., 20, 50, 100, and 150 tokens). In the end-to-end setting, we get the best results with 100 tokens. On the other hand, in the given-expression setting, the results are reported using 20 tokens. Dropout rate of T5 transformer equals to its default value. The batch size is 16 and we use the Adam optimizer. The learning rate (LR) and number of epochs (Epochs) for each model are depicted in Table 2. The loss function considered for training is the default one for training T5 models. We select the model (or those weights we require) that performs best on the development set data.

By following previous research on the issue of prompt tokens initialization [30, 29], we categorize all methods in three groups: 1) Uniformly sample from the range $[-0.5, 0.5]$. 2) Select from vocabulary (or a specific subset of the whole vocabulary). 3) Select from class labels. As we do not encounter a classification problem, we only tested methods 1 and 2. In the experiments, we did not observe a notable variation in the results. Therefore, due to the quicker convergence of method 2, we chose to sample random tokens from the vocabulary.

Table 2: Learning rate and epoch number count of different models.

Model Type	LR	Epochs
End-to-end fine-tuning	1e-4	70
Given-expression fine-tuning	1e-4	100
Prompt tokens learning	0.3	300

3-11- Evaluation Metrics

To keep up with previous works (e.g., Xia et al. [3]) and make our results comparable, we employ Precision, Recall, and F1 score (in some cases, we only show F1) to evaluate our experimental results using the Exact match setting (i.e., *Exact P, R, and F1*), in which we have a true positive (TP) for calculating recall and precision if and only if the entire sequence of tokens is predicted exactly. Additionally, we utilize two auxiliary metrics known as Binary (i.e., *Binary F1*) and Proportional match (i.e., *Proportional F1*). The proportional metric measures the maximum portion that a predicted item matches its gold-standard item, and counts this fraction as a TP in calculating recall and precision. The binary metric yields a TP if a predicted sequence overlaps with its gold-standard sequence in at least one token. We present the formulas for Precision, Recall and F1 score as follows.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

4- Results Analysis and Discussion

In this section, the experimental results in both end-to-end and given-expression settings are presented. Furthermore, we compared our method with other established successful methods. In the proceeding tables, “GenOM” shows our proposed method and “Standalone” shows that the model solely predicts expressions. “MTL”, “Prompt”, and “Int” show our multi-task learning, automatic prompt construction, and integration ideas, respectively. “+” sign indicates the combination of certain methods within our approach. In the end-to-end setting, we compare our results to: BiLSTM-CRF1 [23], Trans [25], SpanOM [3], PtrTrans [9], and BiLSTM-CRF2 [24] methods.

In the given-expression setting (i.e., ORL problem), we also compare our results to: EnhanceORL [8], SynAwareORL [2], and ORL [1] methods. “BERT” means integrating BERT representations into the model. Also, some methods utilized external knowledge as: i) “SynCons” means syntactic constituent features. ii) “Syn” means syntax-enhanced version. iii) “SynDep” means dependency syntax knowledge. iv) “SRL” means Semantic Role Labeling that involves semantic knowledge.

The best results are in bold. The best results <u>without involving any external knowledge</u> are underlined.

✉ **Seyed Abolghasem Mirroshandel**
mirroshandel@guilan.ac.ir

4-1- Results in the end-to-end Setting

The results from Table 3 show a substantial improvement when we apply MTL on the expression prediction task. To follow, F1 score of "Standalone" from 61.4% boosts up to 62.8% when we use MTL method. Also, we observe a 3.6% increase when we adopt APC and MTL jointly. Furthermore, leveraging integration method improves the result by 0.5 percent and set the new SOTA for expression prediction task. The APC method also demonstrates a notable improvement itself. This is evident when comparing the F1 scores of the "Standalone + Prompt" to the "Standalone", which shows a notable increase of 2.8%. This observation emphasizes the effectiveness of our proposed APC method.

Another point of achievement in our methods, is the convergence of Precision and Recall. These two measures are both going up and shows a successful balance between them, which helps to gain the SOTA performance on F1 score. Although the highest Precision was achieved by "SpanOM + Prompt" model, but the low value of Recall lowers their F1 score.

Table 3: Results and comparison of the expression prediction on the exact match metric in the end-to-end setting. "-" means results are not reported in their paper.

Models	Exact Match		
	P	R	F1
Trans	60.2	48.5	53.0
SpanOM	64.9	52.6	58.1
SpanOM + BERT	**67.2**	60.6	63.7
PtrTrans	-	-	58.1
PtrTrans + Syn	-	-	59.9
PtrTrans + BERT	-	-	63.9
PtrTrans + BERT + Syn	-	-	65.3
GenOM			
Standalone	62.0	61.0	61.4
Standalone + Prompt	64.6	63.9	64.2
MTL	63.2	62.5	62.8
Prompt + MTL	64.5	65.5	65.0
Prompt + MTL + Int	65.1	**65.9**	**65.5**

On the other hand, in opinion roles prediction (Table 4), in overall value of all metrics (i.e., exact and auxiliary), we outperform other systems and set a new SOTA performance. In all auxiliary metrics (i.e. proportional and binary) our results are superior compare to other related research. In Exact matching, agent performance is remarkably better than the history of results, however we could achieve second best and best without using external knowledge for target role.

Generally, we observe a considerable improvement in all conditions when we adopt each of our novel ideas. This magnifies our enunciation about using raw text-to-text transformer in the correct way.

Table 4: Experimental results of our GenOM system and comparison with previous research works on the MPQA 2.0 benchmark dataset in the end-to-end setting. "-" means results are not available and/or not presented in their paper.

Model	Exact F1			Binary F1			Proportional F1		
	Overall	Agent	Target	Overall	Agent	Target	Overall	Agent	Target
BiLSTM-CRF1	-	-	-	-	58.2	55.0	-	-	-
Trans	-	47.0	31.5	-	60.9	56.4	-	-	-
SpanOM	43.1	52.9	32.4	51.0	56.5	45.1	48.9	55.6	41.7
PtrTrans	43.7	53.2	33.2	-	57.9	47.0	-	56.9	42.8
PtrTrans + Syn	44.4	54.7	35.0	-	58.3	47.7	-	57.1	43.6
BiLSTM-CRF2 + BERT	-	-	-	-	55.5	50.4	-	46.6	34.3
SpanOM + BERT	49.9	58.2	41.1	57.8	62.0	53.3	55.7	61.2	49.9
SpanOM + BERT + SynCons	50.5	58.5	41.8	-	-	-	-	-	-
PtrTrans + BERT	50.1	58.3	42.0	-	62.3	53.7	-	61.7	50.4
PtrTrans + BERT + Syn	51.6	59.5	44.0	-	63.2	55.2	-	62.3	52.0
GenOM									
MTL	46.4	56.2	36.8	56.6	60.4	52.8	53.4	59.5	47.4
Prompt + MTL	48.9	58.3	39.8	60.0	63.1	57.0	56.8	61.8	51.9
Prompt + MTL + Int	51.8	61.1	42.6	62.5	65.3	59.7	59.4	64.3	54.6

4-2- Results in the given-expression setting

As shown in Table 5, our system achieves SOTA results for all opinion roles (overall, agent and target) using all metrics. By adopting MTL, we obtain new SOTA results for agent and overall using the exact match metric, but we are not better in target. After applying APC, we observe a substantial boost in F1 score, so that on exact match, we establish a new SOTA for overall, agent and target.

4-3- Discussion and Error Analysis

By running an error analysis on the predicted items in development set, which is depicted in Table 6, we understand that a considerable portion of wrong matches are due to the mismatch of opinion expressions. Hence, we aim to focus more deeply on expression prediction task in future. Although Unmatch (means no overlap) items seems to be legitimate errors, but we observe some samples which the system prediction and gold-standard are actually pointing to one specific entity. As an illustration, consider the sentence No.1 from Table 7. Our system predicts "he" as an agent for expression "said", but the gold-standard agent for this expression is "Syed Hamid". Note that in this sample, system successfully determined the gold-expression. Obviously, "he" corresponds to "Syed Hamid" and they are actually one unique entity. Therefore, it seems leveraging "Anaphora resolution" techniques could be helpful and correct some miss-matches. As it is reported in Table 5, partial matches are also considerable. Our analyses indicate a variety of conflicts between the predicted and gold-standard occurs at the boundaries but the interesting point is that most of these discrepancies are about stop-words. We did an automatic analysis by using *Levenshtein distance algorithm* in order to align predicted and gold-standard spans and find disparate segments between them. The most frequent words causing discrepancies in target are *to, the, a, and, of, in, on, is, be*, and in agent are *the, of, and, an, at, a*. In the expression prediction task, the rate of partial errors is higher, and the discrepancies are also the same. The most common words that cause conflict in expression prediction task are *to, of, the, is, are, by, a, in*.

Table 5: Experimental results of our GenOM system and comparison with previous research works on the MPQA2.0 benchmark dataset in the given-expression setting. "-" means results are not available and/or not presented in their paper

Model	Exact F1			Binary F1			Proportional F1		
	Overall	Agent	Target	Overall	Agent	Target	Overall	Agent	Target
EnhanceORL	58.3	73.1	42.7	75.2	81.6	68.3	70.6	79.4	61.2
SynAwareORL	58.8	73.1	44.2	75.4	81.2	69.5	71.0	79.3	62.5
SpanOM	59.6	72.4	45.8	71.6	78.1	64.5	68.1	76.7	58.7
ORL + SRL	61.5	75.6	46.4	-	-	-	-	-	-
EnhanceORL + SRL	63.7	77.0	51.0	-	-	-	-	-	-
SynAwareORL + BERT	64.7	76.7	52.6	80.6	85.5	75.7	76.5	83.6	69.3
SynAwareORL + BERT + SynDep	68.1	79.5	56.6	-	-	-	-	-	-
SpanOM + BERT	66.0	76.5	55.0	77.9	82.7	72.9	74.6	81.5	67.4
SpanOM + BERT + SynCons	68.0	78.3	57.0	-	-	-	-	-	-
GenOM									
MTL	68.2	79.3	56.7	84.1	87.5	80.6	79.4	85.5	73.0
Prompt + MTL	68.7	79.9	57.1	84.3	87.8	80.7	79.5	85.7	73.2

Table 6: Percentage of different types of errors among all predicted items of development set in each task. EM stands for Expression Miss-match, PM stands for Partial Match and U means Unmatch or legitimate errors

Task	EM	PM	U
Agent	55.7	15.5	28.8
Target	39.3	30.4	30.3
Expression	-	57.7	42.3

As reported in other studies like Xia et al. [3], we also observed some peculiarities and errors in annotations of MPQA, which might provide false information to our system. For instance, sentence No.2 from Table 7, there is an expression marked in corpus as "The", but we think "The" is not a reasonable expression. On the other hand, in some sentences, there are predictions by our system which seems to be correct but they are absent in annotated data. For instance, consider sentence No.3 from Table 7. The system predicted the agent of "oppose" expression as "many poor" which is correct. But there is not any agent marked for this expression in the corpus. To mitigate these **gold errors** in the future, it is crucial to enhance the quality

✉ **Seyed Abolghasem Mirroshandel**
mirroshandel@guilan.ac.ir

of MPQA annotation. This improvement will pave the way for more accurate and reliable results.

We also did some preliminary experiments with a variant of T5, called FLAN-T5 [19], but the results did not indicate superiority.

Our proposed generative approach utilizing the T5 transformer in conjunction with MTL and APC shows notable performance enhancements on the MPQA 2.0 dataset. However, several limitations must be addressed for application in real-world scenarios. First, the model's dependence on specific characteristics of the dataset may restrict its adaptability to other domains where opinion structures vary significantly or where annotated data is limited. Additionally, the complexity and computational requirements of the generative model could present challenges for deployment in resource-limited environments or in applications that necessitate real-time processing. Moreover, while APC effectively optimizes prompts for this dataset, its performance in entirely different contexts or languages may differ, requiring further tuning or adaptation.

Table 7: Some sentences of MPQA 2.0 corpus.

No.	Sentence
1	Syed Hamid said the international community must deal with terrorism rationally and form a new "security architecture" to combat what he described as a "new dimension of crime against humanity" in the long term.
2	The CIA was given the task to topple governments and install rulers of its own choice.
3	However, 78 percent of those polled believe there are many poor who oppose him.

5- Conclusion and Future Work

This research introduces a novel generative framework for opinion mining, leveraging the T5 transformer model through Multi-Task Learning (MTL) and Automatic Prompt Construction (APC). Our approach achieves remarkable performance improvements on the MPQA 2.0 dataset, setting new state-of-the-art records without relying on external knowledge. The MTL strategy enables the model to learn interconnected sub-tasks concurrently, enhancing the detection of opinion expressions and their associated roles. Meanwhile, APC facilitates the automatic optimization of prompts, effectively addressing the challenges posed by manual prompt engineering and ensuring more efficient task customization. The results indicate that the synergy between MTL and APC significantly elevates precision, recall, and F1 scores across various evaluation metrics. By integrating predictions from both the end-to-end and given-expression settings, our method achieves a more accurate recognition of opinion structures. These findings highlight the effectiveness of generative models in capturing complex opinion relationships within text.

Looking ahead, future research can build on these findings by integrating additional syntactic and semantic knowledge into generative models and further refining the APC technique. Extending the application of our methods to other datasets and domains is also critical. Investigating the use of more advanced generative transformers or combining our approach with alternative machine learning strategies could yield additional improvements. Furthermore, enhancing the quality of existing datasets and developing new benchmarks will be essential for validating the generalizability and effectiveness of these methods across a broader range of contexts.

Reference

[1] A. Frank and A. Marasović, "SRL4ORL: Improving Opinion Role Labeling Using Multi-Task Learning with Semantic Role Labeling," in Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, Louisiana, 2018.

[2] B. Zhang, Y. Zhang, R. Wang, Z. Li and M. Zhang, "Syntax-Aware Opinion Role Labeling with Dependency Graph Convolutional Networks," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 2020.

[3] Q. Xia, B. Zhang, R. Wang, Z. Li, Y. Zhang, F. Huang, L. Si and M. Zhang, "A Unified Span-Based Approach for Opinion Mining with Syntactic Constituents," in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 2021.

[4] S. Ahmadnia, A. Yousefi Jordehi, M. Hosseini Khasheh Heyran, S. Mirroshandel and O. Rambow, "Opinion Mining Using Pre-Trained Large Language Models: Identifying the Type, Polarity, Intensity, Expression, and Source of Private States," in Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Torino, Italia, 2024.

[5] J. Wiebe, "Identifying subjective characters in narrative," in COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics, 1990.

[6] J. Wiebe, T. Wilson and M. Bell, "Identifying collocations for recognizing opinions," in Proceedings of the ACL-01 Workshop on Collocation: Computational Extraction, Analysis, and Exploitation, 2001.

[7] T. A. Wilson, Fine-grained subjectivity and sentiment analysis: recognizing the intensity, polarity, and attitudes of private states, University of Pittsburgh, 2008.

[8] M. Zhang, P. Liang and G. Fu, "Enhancing Opinion Role Labeling with Semantic-Aware Word Representations from Semantic Role Labeling," in Proceedings of the 2019

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,, Minneapolis, Minnesota, 2019.

[9] S. Wu, H. Fei, F. Li, D. Ji, M. Zhang, Y. Liu and C. Teng, "Mastering the explicit opinion-role interaction: Syntax-aided neural transition system for unified opinion role labeling," in Proceedings of the AAAI conference on artificial intelligence, Online, 2022.

[10] Z. Gao, A. Feng, X. Song and X. Wu, "Target-Dependent Sentiment Classification With BERT," IEEE Access, vol. 7, pp. 154290-154299, 2019.

[11] Y. Z. R. W. Z. L. M. Z. Bo Zhang, "Syntax-Aware Opinion Role Labeling with Dependency Graph Convolutional Networks," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 2020.

[12] W. Zhang, X. Li, Y. Deng, L. Bing and W. Lam, "Towards Generative Aspect-Based Sentiment Analysis," in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 2021.

[13] X. Bao, W. Zhongqing, X. Jiang, R. Xiao and S. Li, "Aspect-based Sentiment Analysis with Opinion Tree Generation," in Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, Vienna, 2022.

[14] P. Kavehzadeh, M. M. Abdollah Pour and S. Momtazi, "Deep Transformer-based Representation for Text Chunking," Journal of Information Systems and Telecommunication (JIST), vol. 3, pp. 176-184, 2023.

[15] S. Chakraborty, M. Borhan Uddin Talukdar, P. Sikdar and J. Uddin, "An Efficient Sentiment Analysis Model for Crime Articles' Comments using a Fine-tuned BERT Deep Architecture and Pre-Processing Techniques," Journal of Information Systems and Telecommunication (JIST), vol. 12, pp. 1-11, 2024.

[16] N. Jadhav, "Hierarchical Weighted Framework for Emotional Distress Detection using Personalized Affective Cues," Journal of Information Systems and Telecommunication (JIST), vol. 10, pp. 89-101, 2022.

[17] Liu, C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li and P. J., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," Journal of Machine Learning Research (JMLR), vol. 21, no. 140, pp. 1-67, 2020.

[18] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 2020.

[19] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros and Marie, "Scaling instruction-finetuned language models," arXiv preprint arXiv:2210.11416, 2022.

[20] E. Breck, Y. Choi and C. Cardie, "Identifying expressions of opinion in context.," in International Joint Conference on Artificial Intelligence, Hyderabad India, 2007.

[21] B. Yang and C. Cardie, "Joint Inference for Fine-grained Opinion Extraction," in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 2013.

[22]B. Yang and C. Cardie, "Joint Modeling of Opinion Expression Extraction and Attribute Classification," Transactions of the Association for Computational Linguistics, p. 505–516, 2014.

[23] A. Katiyar and C. Cardie, "Investigating LSTMs for Joint Extraction of Opinion Entities and Relations," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 2016.

[24] W. Quan, J. Zhang and X. T. Hu, "End-to-end joint opinion role labeling with bert," in IEEE International Conference on Big Data (Big Data), 2019.

[25] M. Zhang, Q. Wang and G. Fu, "End-to-end neural opinion extraction with a transition-based model," Information Systems, vol. 80, pp. 56-63, 2019.

[26] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, 2019.

[27]O. Vinyals, M. Fortunato and N. Jaitly, "Pointer networks," Advances in neural information processing systems, vol. 28, 2015.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, {. Kaiser and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems}, vol. 30, 2017.

[29] B. Lester, R. Al-Rfou and N. Constant, "The Power of Scale for Parameter-Efficient Prompt Tuning," in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, 2021.

[30] D. C. Senadeera and J. Ive, "Controlled text generation using T5 based encoder-decoder soft prompt tuning and analysis of the utility of generated text in AI," in arXiv preprint arXiv:2212.02924, 2022.