Uncertainty-Aware Unlikelihood Learning Improves Generative Aspect Sentiment Quad Prediction

Mengting Hu[1] **Yinhao Bai**[1] **Yike Wu**[1*] **Zhen Zhang**[1]
Liqi Zhang[2] **Hang Gao**[3] **Shiwan Zhao**[†] **Minlie Huang**[3]
[1] Nankai University, [2] Tiangong University, [3] Tsinghua University
{mthu, wuyike}@nankai.edu.cn, {yinhao, zhangzhen}@mail.nankai.edu.cn
gaohang@mail.tsinghua.edu.cn, aihuang@tsinghua.edu.cn

Abstract

Recently, aspect sentiment quad prediction has received widespread attention in the field of aspect-based sentiment analysis. Existing studies extract quadruplets via pre-trained generative language models to paraphrase the original sentence into a templated target sequence. However, previous works only focus on *what to generate* but ignore *what not to generate*. We argue that considering the negative samples also leads to potential benefits. In this work, we propose a template-agnostic method to control the token-level generation, which boosts original learning and reduces mistakes simultaneously. Specifically, we introduce Monte Carlo dropout to understand the built-in uncertainty of pre-trained language models, acquiring the noises and errors. We further propose marginalized unlikelihood learning to suppress the uncertainty-aware mistake tokens. Finally, we introduce minimization entropy to balance the effects of marginalized unlikelihood learning. Extensive experiments on four public datasets demonstrate the effectiveness of our approach on various generation templates[1].

1 Introduction

Recently, aspect sentiment quad prediction (ASQP) has received extensive attention in the field of aspect-level sentiment analysis. ASQP targets a comprehensive sentiment understanding and extracts four elements of aspect sentiment, including 1) *aspect term (at)* which is the concrete aspect description; 2) *opinion term (ot)* suggesting the specific opinion expression towards the aspect; 3) *aspect category (ac)* denoting the aspect class; 4) *sentiment polarity (sp)* indicating the sentiment class of the aspect. For example, given a comment sentence *"Service was good and food was wonderful"*, ASQP aims to recognize two quadruples (*Service*, *good*, *service general*, *positive*) and (*food*, *wonderful*, *food quality*, *positive*).

Inputs-1	*The food is good.*
Label-1	*(food, good, food quality, positive)*
Pred-1	*(foods, good, food quality, positive)* ✖
Inputs-2	*Yamato is an excellent place to go.*
Label-2	*(Yamato, excellent, restaurant miscellaneous, positive)*
Pred-2	*(Yamato, great, restaurant miscellaneous, positive)* ✖

Figure 1: Two predicted error cases are shown. Pred denotes the prediction. The results of Label and Pred are shown in the order of (at, ot, ac, sp), and the highlighted parts are the predicted error items.

Existing works have pointed out two promising research directions. Cai et al. (2021) propose a pipeline-based method, using the properties of four elements and designing first-extract-then-classify two stages. Another direction leverages generation-based pre-trained language models. ASQP is addressed in an end-to-end manner by *"re-writing"* a sentence into a structured target sequence (Zhang et al., 2021b,a; Hu et al., 2022). With pre-defined templates, quadruples can be easily decoded from the target sequence. Due to its simplicity and effects, the second paradigm gradually becomes the main streaming in ASQP (Hu et al., 2022).

However, no matter designing good templates (Zhang et al., 2021a; Bao et al., 2022) or using data augmentation (Hu et al., 2022), previous generation-based works only focus on *what to generate* but ignore *what not to generate*. Learning signals of negative effects are also crucial for accurate extraction. The reason is that ASQP is not a typical text-generation task, such as dialog (Liu et al., 2021) or storytelling (Xu et al., 2020b). Semantic-similar or ambiguous words are harmful for extraction. Two failed cases of pre-trained language models are presented in Figure 1. In the first case, the aspect term *"food"* is easily confused with *"foods"*. And the second case also implies

[*] Corresponding author.
[†] Independent researcher.
[1] Experimental codes and data are available at: https://github.com/byinhao/UAUL.

Findings of the Association for Computational Linguistics: ACL 2023, pages 13481–13494
July 9-14, 2023 ©2023 Association for Computational Linguistics

that the opinion term *"excellent"* can be wrongly decoded as *"great"*. Though these words do not obviously change the semantics, they lead to complete mistakes for ASQP. Therefore, how to make language models to avoid errors motivates us.

In this paper, we propose uncertainty-aware unlikelihood learning (UAUL) to guide the likelihood learning (*what can be generated*) and marginalized unlikelihood learning (*what not to generate*) simultaneously. Concretely, *what to generate* is in light of the sequence-to-sequence learning objective. Target sequences are constructed with predefined templates, providing semantic and syntactic structured information. As for *what not to generate*, we argue that the noise and errors present in the pre-trained model are due to the uncertainty of the model itself. Therefore, we introduce the Monte Carlo dropout (MC dropout) (Gal and Ghahramani, 2016) to obtain built-in negative samples of pre-trained models. By dropping out random parameters of the last layer followed by the decoder, i.e. language model head, multiple predictions can be attained, which further tell the inherent errors of language models.

Moreover, with uncertainty-aware negative samples, we further propose marginalized unlikelihood learning (MUL) to suppress the probability of them. The marginalization could promote the gap between correct and error tokens, making models to better distinguish semantically similar or ambiguous words. Finally, MUL reduces the probability of noises. This might enlarge the probability of other errors, since the vocabulary set of language models is with the scale. Hence, to balance the influences of MUL, we propose to minimize the entropy of uncertainty-aware probability distributions.

In summary, the contributions of this paper are as follows:

- We study generative ASQP task from the view of *what not to generate*. To the best of our knowledge, this is the first work to study negative samples in this task. We propose uncertainty-aware unlikelihood learning to avoid the intrinsic mistakes of pre-trained language models.

- Specifically, the model uncertainty is comprehended with MC dropout. And the built-in errors are reduced with the proposed marginalized unlikelihood learning and minimization entropy. Our method is template-agnostic and can be easily applied to various target templates.

- Experimental results on four public datasets Rest15, Rest16, Restaurant, and Laptop demonstrate that UAUL has universal effectiveness on various templates.

2 Methodology

2.1 Formulation and Overview

Given a sentence $\boldsymbol{x}$, aspect sentiment quad prediction (ASQP) aims to predict all aspect-level quadruplets $\{(at, ot, ac, sp)\}$. Following the previous generation-based works (Zhang et al., 2021a; Hu et al., 2022), we define projection functions to map the quadruplets (at, ot, ac, sp) into semantic values $(x_{at}, x_{ot}, x_{ac}, x_{sp})$. Concretely, 1) if aspect term at is explicit, $x_{at} = at$, otherwise $x_{at} =$*"it"*; 2) if opinion term ot are explicitly mentioned, $x_{ot} = ot$, otherwise it is mapped as *"NULL"* if being implicitly expressed; 3) aspect category ac is transformed into words, such as $x_{ac} =$*"food quality"* for $ac =$*"food#quality"*; 4) the sentiment polarity $sp \in \{$*positive*, *neutral*, *negative*$\}$, is mapped into words with sentiment semantics $\{$*great*, *ok*, *bad*$\}$, respectively.

Based on the above rules, the values are fed into a template $\mathcal{T}$ to form the target sequence. For instance, a template follows the cause and effect semantic relationship "x_{ac} is x_{sp} because x_{at} is x_{ot}" (Zhang et al., 2021a) or uses special markers "[AT] x_{at} [OT] x_{ot} [AC] x_{ac} [SP] x_{sp}" (Hu et al., 2022). If a sentence contains multiple quadruplets, the templated sequences are concatenated with a special marker [SSEP] to obtain the final target sequence $\boldsymbol{y}$.

As shown in Figure 2, an input sentence is first fed into the encoder-decoder framework. We exploit the pre-trained language model T5 (Raffel et al., 2020). To deal with negative samples[2], we first acquire multiple uncertain-aware samples via MC dropout for each decoding time step. Then these samples are fed to calculate marginalized unlikelihood learning loss. Finally, to enhance the learning of target sequence and balance the effects of MUL, we design enhanced likelihood learning for uncertainty-aware samples. Next, we will introduce the components in detail.

2.2 Uncertainty-Aware Samples Acquisition

As depicted in Figure 1, semantic-similar or ambiguous words lead to complete error predictions

[2] It is worth noting that negative samples indicate the noisy tokens in the vocabulary set rather than a sentence.

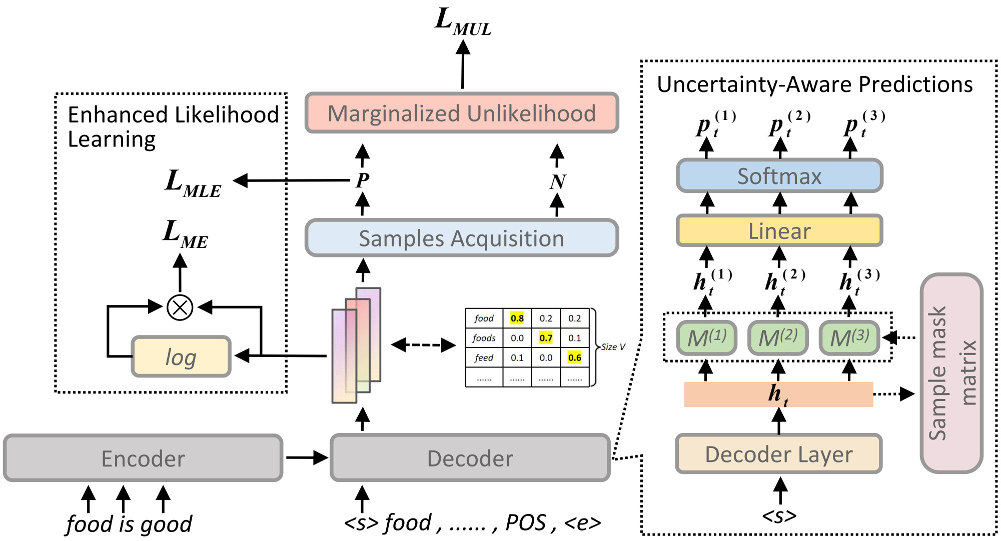
L_{MUL}
Enhanced Likelihood Learning
L_{MLE}
L_{ME}
log
Marginalized Unlikelihood
P
N
Samples Acquisition
food 0.8 0.2 0.2
foods 0.0 0.7 0.1
feed 0.1 0.0 0.6
....
Size V
Encoder
Decoder
food is good
<s> food , ... , POS , <e>
Uncertainty-Aware Predictions
$p_t^{(1)}$ $p_t^{(2)}$ $p_t^{(3)}$
Softmax
Linear
$h_t^{(1)}$ $h_t^{(2)}$ $h_t^{(3)}$
$M^{(1)}$ $M^{(2)}$ $M^{(3)}$
h_t
Decoder Layer
<s>
Sample mask matrix

Figure 2: An overview of the proposed uncertainty-aware unlikelihood learning (UAUL). We present the details via an example of the first decoding time step. The beginning token *"<s>"* yields its next token, i.e. *"food"*, where the output is enhanced as three uncertainty-aware probability distributions $\{\boldsymbol{p}_t^{(i)}\}$. The largest probability is highlighted and chosen as a negative sample.

for ASQP task. The in-depth reason relies on that language models are pre-trained based on distributional semantics theory (Boleda, 2020), producing alike representations for words that frequently appear in similar contexts, such as *"excellent"* and *"great"*. Then when extracting aspect quadruplets, language models are not sure which one is more accurate. Understanding the inherent uncertainty of language models may have potential benefits. To achieve this goal, we re-design the decoder of T5 and adopt MC dropout (Gal and Ghahramani, 2016) to obtain valuable samples.

Uncertainty-Aware Predictions The target sequence $\boldsymbol{y}$ is fed into the decoder as teacher forcing (Williams and Zipser, 1989) during training. The decoder's inner layers are depicted in the right plot of Figure 2. Here, we use the beginning token "<*s*>" as an example to illustrate the details of each time step. We obtain a representation for each token with multiple transformer-based self-attention mechanisms of the decoder layer.

$$\boldsymbol{h_t} = \mathrm{Enc} - \mathrm{DecLayer}(\boldsymbol{x}, \boldsymbol{y_{<t}}) \tag{1}$$

where $\boldsymbol{h_t}$ is calculated based on the input sequence $\boldsymbol{x}$ and previous outputs $\boldsymbol{y_{<t}}$. $\mathrm{Enc} - \mathrm{DecLayer}$ indicates the encoder module and the decoder layer.

Then, following Vazhentsev et al. (2022), we only exploit the last dropout layer, which is much less computationally expensive than the standard MC dropout. Specifically, an uncertain-aware representation is obtained by sampling a random mask matrix $M^{(i)}$.

$$\boldsymbol{h}_t^{(i)} = M^{(i)} \times \boldsymbol{h_t} \tag{2}$$

where sampling $M^{(i)}$ follows the Bernoulli distribution $Bernoulli(1 - p)$ and $p \in [0, 1]$ is the dropout rate. Then the output is calculated based on the uncertainty-aware representations.

$$\boldsymbol{p}_t^{(i)} = \mathrm{softmax}(W^{\mathrm{T}} \boldsymbol{h}_t^{(i)}) \tag{3}$$

where W maps $\boldsymbol{h}_t^{(i)}$ into a vector and $\boldsymbol{p}_t^{(i)}$ indicates the probability distribution over the vocabulary set. We dropout multiple times and attain multiple output distributions at t-th time step $\{\boldsymbol{p}_t^{(i)}\}$ and $i \in [1, K]$. K is the number of MC forward computations.

Samples Acquisition Based on multiple uncertain-aware output probability distributions, we then acquire key samples as described in Algorithm 1. Note that this algorithm displays the acquisition of samples for time step t. For the positive samples, we mainly concentrate on the probability of the ground-truth token y_t out of each distribution. For the negative samples, we choose the largest wrong prediction. In this way, both the positive and negative samples are integrated with the uncertainty of language models (i.e. various probabilities). Meanwhile, negative samples are also difficultly distinguishable ones.

Algorithm 1 Samples Acquisition

```
Input: Output probability distributions $\{\boldsymbol{p}_t^{(i)}\}$, ground-truth $y_t$
Define: $P_t = \varnothing, N_t = \varnothing$
for $i = 1, 2, ..., K$ do
    $c \leftarrow \text{argmax}(\boldsymbol{p}_t^{(i)})$
    if $c \mathrel{!=} y_t$ then
        $N_t \leftarrow N_t \cup \boldsymbol{p}_t^{(i)}[c]$
    end if
    $P_t \leftarrow P_t \cup \boldsymbol{p}_t^{(i)}[y_t]$
end for
return $P_t, N_t$
```

2.3 Marginalized Unlikelihood Learning

Then with these chosen key samples, we propose marginalized unlikelihood learning to explicitly control their optimization. As an example shown in Figure 2, we have three output distribution $\{\boldsymbol{p}_t^{(1)}, \boldsymbol{p}_t^{(2)}, \boldsymbol{p}_t^{(3)}\}$. The highlighted probabilities are the largest in that distribution. With Algorithm 1, we obtain $P_t = \{0.8, 0.2, 0.2\}$ and $N_t = \{0.7, 0.6\}$, where P_t and N_t are sampled positive and negative samples, respectively. These probabilities are further utilized to calculate in Eq. (4).

$$\mathcal{L}_{MUL} = \sum_{t=1}^{n} \log[1 + \sum_{k=1}^{|P_t|} \sum_{l=1}^{|N_t|} \exp(\alpha(N_t^l - P_t^k + m))] \quad (4)$$

where α is the scale hyperparameter. m is the margin between positive and negative samples. n is the length of the target sequence. $|P_t|$ means the number of samples in P_t.

It is worth noting that the proposed MUL is based on the largest probability in every uncertainty-aware distribution. Putting it to N_t according to whether it is correct. The reason is that softmax probabilities tend to be overconfident (Guo et al., 2017), making all other probabilities very small except for the largest one. Then our method can better select easily-mistaken samples from multiple distributions.

2.4 Enhanced Likelihood Learning

Except for dealing with the noise issue, *what to generated* is still important to obtain task-specific semantic and structured knowledge. We exploit the original likelihood training to optimize the positive sample probabilities on multiple uncertainty-aware probability distributions.

$$\mathcal{L}_{MLE} = -\frac{1}{K} \sum_{i=1}^{K} \sum_{t=1}^{n} \boldsymbol{y_t} \log \boldsymbol{p}_t^{(i)} \quad (5)$$

where $\boldsymbol{y_t}$ denotes a ground true one-hot vector at the time step t.

Datasets	Train		Dev		Test	
	#S	#Q	#S	#Q	#S	#Q
`Rest15`	834	1354	209	347	537	795
`Rest16`	1264	1989	316	507	544	799
`Restaurant`	2934	4172	326	440	816	1161
`Laptop`	1530	2484	171	261	583	916

Table 1: Data statistics. #S and #Q denote the number of sentences and quadruplets respectively.

Though MUL reduces the probability of noise, it might enlarge the probability of other errors. Take Figure 2 as an example, reducing the probability of *"foods"* may disperse the numerical to other noisy words, such as *"feed"* or *"apple"*. Thus, the optimization of MUL and likelihood is not fully consistent, which in turn affects the likelihood of learning. To balance likelihood and MUL, we introduce the minimization entropy (ME) loss term.

$$\mathcal{L}_{ME} = -\sum_{i=1}^{K} \sum_{t=1}^{n} \boldsymbol{p}_t^{(i)} \log \boldsymbol{p}_t^{(i)} \quad (6)$$

By minimizing Eq. (6), $\boldsymbol{p}_t^{(i)}$ will become more peak, that is to say, suppressing the noises simultaneously. In this way, our approach seeks a balance between MUL and likelihood, ensuring both accurate extraction and negative sample reduction.

2.5 Joint Training Objective

The final training objective is jointly to combine the above three losses.

$$\mathcal{L} = \mathcal{L}_{MLE} + \mathcal{L}_{MUL} + \mathcal{L}_{ME} \quad (7)$$

3 Experiments

3.1 Datasets

To evaluate the proposed approach, we choose four publicly available datasets. `Rest15` and `Rest16` datasets are proposed by Zhang et al. (2021a). They are based on previous SemEval tasks (Pontiki et al., 2015, 2016), and expanded with quadruplet annotations. Cai et al. (2021) propose `Restaurant` and `Laptop` datasets. The `Restaurant` dataset is constructed based on the SemEval 2016 Restaurant dataset (Pontiki et al., 2016) and its expansion datasets (Fan et al., 2019; Xu et al., 2020a). The Laptop dataset is annotated based on the data collected on Amazon between 2017 and 2018. The statistics of datasets are displayed in Table 1.

3.2 Compared Methods

We choose the following strong baseline methods and divided them into two types: i.e. *non-generation* and *generation*.

Non-Generation Baselines: Traditional paradigm designs various stages to extract individual information separately.

- **Double-Propagation** (Qiu et al., 2011) It is a classical method for triple extraction. Cai et al. (2021) adapt it to ASQP. All $\{at, ot, sp\}$ triplets are first extracted using double propagation, and then each triplet is assigned ac to attain quad.
- **JET** (Xu et al., 2020a) It is an end-to-end framework for detecting triplet. Cai et al. (2021) first obtain $\{at, ot, sp\}$ triples with JET and then leveraged BERT to obtain ac.
- **HGCN-BERT+BERT** (Zhang et al., 2021a) It is designed for learning syntactic dependencies for ASQP. Its variants include HGCN-BERT+BERT-Linear and HGCN-BERT+BERT-TFM according to the last layer.
- **TAS-BERT** (Wan et al., 2020) It recognize triplets $\{at, ac, sp\}$ via learning dependencies. Cai et al. (2021) reformulate TAS-BERT to filter out invalid $\{ac, at\}$ pairs to get the final quadruplet. Zhang et al. (2021a) extend TAS-BERT to detect ot for ASQP task. The variants are TASO-BERT-Linear and TASO-BERT-CRF.
- **Extract-Classify-ACOS** (Cai et al., 2021) It first extracts aspect-opinion and then classifies category and sentiment, yielding the final quad.

Generation Baselines: Aspect sentiment quadruplets are fed into semantic templates to obtain a target sequence for generation learning.

- **GAS** (Zhang et al., 2021b) It is the first work to reformulate all ABSA tasks as generation problems, and process all sub-tasks in a unified generation framework.
- **Paraphrase** (Zhang et al., 2021a) It transforms the quadruplet extraction into a paraphrase generation through a predefined template.
- **Special_Symbols** (Hu et al., 2022) It distinguishes the type of element in each position by special symbols.
- **DLO** (Hu et al., 2022) It designs dataset-level data augmentation via template-order permutation. The templates use special symbols.
- **ILO** (Hu et al., 2022) ILO designs data augmentation for each instance to find the good template order. The templates adopt special symbols.

3.3 Experimental Results

3.3.1 Overall Results

Experimental results are reported in Table 2 and Table 3. Firstly, it can be observed that our method is effective in almost all experimental settings. Especially, compared with a strong baseline Paraphrase, Paraphrase+UAUL gains absolute F1 score improvements by +2.45% (+5.22% relatively), +1.47% (+2.54% relatively), +0.34% (+0.57% relatively), and +1.45% (+3.37% relatively) in Rest15, Rest16, Restaurant and Laptop datasets, respectively. Similarly, Special_Symbols+UAUL achieves consistent improvements on all datasets, performing the best on Rest15 dataset. Compared with DLO, DLO+UAUL also performs consistently better, achieving the best F1 scores of 60.50% and 60.78% on the Rest16 and Restaurant datasets, respectively. These results demonstrate that the proposed UAUL can be easily applied to various templates with universal effectiveness.

Moreover, we also see a few exceptions. For example, on Laptop dataset, UAUL causes the performances of GAS and ILO slightly decline. A possible reason is that Laptop dataset has a larger proportion of implicit information. Template treats implicit aspect term as *"it"* and implicit opinion term as *"NULL"*. Such implicit information makes it hard to understand quadruplets accurately.

3.3.2 Low-Resource Scenario

To further explore the performance of our proposed method in a low-resource environment, we train the model only with subsets of Rest15. The results are reported in Table 4. We can see that for both baseline methods, i.e. Special_Symbols and Paraphrase, UAUL can bring consistent improvements with various data scales. In particular, with only 15% training data, UAUL improves Special_Symbols and Paraphrase significantly by +3.47% (+11.22% relatively) and +4.89% (+17.01% relatively) on F1 score, respectively. This verifies that UAUL shows more significant effectiveness in low-resource scenarios. A rational explanation is that low-resource might boost the overfitting of language models to the small-scale data. Then mistakes will occur

Methods	Rest15			Rest16		
	Pre	Rec	F1	Pre	Rec	F1
HGCN-BERT+BERT-Linear* (Zhang et al., 2021a)	24.43	20.25	22.15	25.36	24.03	24.68
HGCN-BERT+BERT-TFM* (Zhang et al., 2021a)	25.55	22.01	23.65	27.40	26.41	26.90
TASO-BERT-Linear* (Zhang et al., 2021a)	41.86	26.50	32.46	49.73	40.70	44.77
TASO-BERT-CRF* (Zhang et al., 2021a)	44.24	28.66	34.78	48.65	39.68	43.71
Extract-Classify-ACOS* (Cai et al., 2021)	35.64	37.25	36.42	38.40	50.93	43.77
GAS* (Zhang et al., 2021b)	45.31	46.70	45.98	54.54	57.62	56.04
+UAUL	**46.39**	**47.82**	**47.10**	**55.95**	**58.30**	**57.10**
Paraphrase* (Zhang et al., 2021a)	46.16	47.72	46.93	56.63	59.30	57.93
+UAUL	**48.96**	**49.81**	**49.38**	**58.28**	**60.58**	**59.40**
Special_Symbols* (Hu et al., 2022)	48.24	48.93	48.58	58.74	60.35	59.53
+UAUL	**49.12**	**50.39**	**49.75**	**59.24**	**61.75**	**60.47**
DLO* (Hu et al., 2022)	47.08	49.33	48.18	57.92	61.80	59.79
+UAUL	**48.03**	**50.54**	**49.26**	**59.02**	**62.05**	**60.50**
ILO* (Hu et al., 2022)	47.78	50.38	49.05	57.58	61.17	59.32
+UAUL	46.84	49.53	48.15	**58.23**	**61.35**	**59.75**

Table 2: Evaluation results compared with baseline methods in terms of precision (Pre, %), recall (Rec, %) and F1 score (F1, %). The results of baseline methods, marked with *, are obtained from this work (Hu et al., 2022).

Methods	Restaurant			Laptop		
	Pre	Rec	F1	Pre	Rec	F1
DP* (Cai et al., 2021)	34.67	15.08	21.04	13.04	00.57	08.00
JET* (Cai et al., 2021)	59.81	28.94	39.01	44.52	16.25	23.81
TAS-BERT* (Cai et al., 2021)	26.29	46.29	33.53	47.15	19.22	27.31
Extract-Classify-ACOS* (Cai et al., 2021)	38.54	52.96	44.61	45.56	29.48	35.80
GAS (Zhang et al., 2021b)	57.09	57.51	57.30	43.45	43.29	43.37
+UAUL	**58.69**	**59.26**	**58.97**	**43.58**	42.98	43.28
Paraphrase (Zhang et al., 2021a)	59.85	59.88	59.87	43.44	42.56	43.00
+UAUL	**60.39**	**60.04**	**60.21**	**44.91**	**44.01**	**44.45**
Special_Symbols (Hu et al., 2022)	59.98	58.40	59.18	43.58	42.72	43.15
+UAUL	**61.22**	**59.87**	**60.53**	**44.38**	**43.65**	**44.01**
DLO (Hu et al., 2022)	60.02	59.84	59.93	43.40	43.80	43.60
+UAUL	**61.03**	**60.55**	**60.78**	**43.78**	43.53	**43.65**
ILO (Hu et al., 2022)	58.43	58.95	58.69	44.14	44.56	44.35
+UAUL	**59.46**	**59.12**	**59.29**	43.92	43.46	43.69

Table 3: Evaluation results compared with baseline methods in terms of precision (Pre, %), recall (Rec, %) and F1 score (F1, %). The results of baseline methods, marked with *, are obtained from this work (Cai et al., 2021).

Ratio	SS	+UAUL	Δ	Para	+UAUL	Δ
10%	29.05	**31.48**	2.43	26.68	**30.55**	3.87
15%	30.92	**34.39**	3.47	28.74	**33.63**	4.89
20%	36.83	**37.07**	0.24	33.60	**36.70**	3.10
25%	37.76	**39.20**	1.44	35.01	**37.71**	2.70
30%	39.41	**41.03**	1.62	37.12	**40.25**	3.13
35%	41.83	**42.52**	0.69	38.10	**42.30**	4.20
40%	42.46	**44.17**	1.71	39.48	**44.35**	4.87
45%	42.84	**44.86**	2.02	40.49	**43.28**	2.79
50%	44.64	**45.69**	1.05	42.48	**45.30**	2.82

Table 4: Evaluation results of low-resource scenario in terms of F1 (%). Radio indicates the proportion of Rest15 dataset's training data. SS and Para are Special_Symbols and Paraphrase methods for short. Δ denotes the absolute improvements.

more frequently, which are potentially distributed within the model and are caused by the uncertainty of the model itself. Our method helps the models to understand these potential errors well and addresses them to some extent. Therefore, UAUL is not only template-agnostic but also resource-friendly.

3.3.3 Ablation Study

To validate the effectiveness of individual components, we perform a systematic ablation study based on Special_Symbols+UAUL. The experimental results are presented in Table 5. It is worth noting that -MC dropout represents removing the MC dropout and directly sampling the output of a network layer for negative samples. -MUL+UL means replacing the marginalized unlikelihood

Datasets	Model	Pre	Rec	F1
Rest15	**Our**	**49.12**	**50.39**	**49.75**
	-ME	49.11	50.18	49.64
	-MUL	48.36	49.45	48.90
	-MUL + UL	48.40	49.13	48.76
	-MUL -ME + UL	47.99	49.05	48.52
	-MC dropout	48.54	49.96	49.24
Rest16	**Our**	**59.24**	**61.75**	**60.47**
	-ME	58.81	60.78	59.77
	-MUL	58.53	60.88	59.68
	-MUL + UL	58.74	61.48	60.08
	-MUL -ME + UL	58.52	60.80	59.64
	-MC dropout	58.07	60.83	59.41
Restaurant	**Our**	**61.22**	**59.87**	**60.53**
	-ME	60.19	58.76	59.46
	-MUL	60.86	59.45	60.15
	-MUL + UL	61.00	59.74	60.36
	-MUL -ME + UL	60.58	59.10	59.84
	-MC dropout	59.70	58.36	59.02
Laptop	**Our**	**44.38**	**43.65**	**44.01**
	-ME	43.31	42.58	42.94
	-MUL	43.87	43.10	43.48
	-MUL + UL	42.89	42.03	42.45
	-MUL -ME + UL	43.08	42.12	42.59
	-MC dropout	44.19	43.12	43.65

Table 5: Evluation results of ablation study. The minus “-” denotes removing components and the addition “+” denotes adding components.

learning with the original unlikelihood.

Firstly, it can be observed that by removing various components, the performances on four datasets are consistently decreasing. This validates the effectiveness of the constituent part of UAUL. Concretely, we see that removing MUL causes significant performance declines on all datasets. This presents that MUL is effective and telling language models *what not to generate* successfully makes quadruplets extraction more accurate. Moreover, replacing MUL with naive UL also leads to performance drops. This further demonstrates that only using UL is not enough. The proposed MUL can widen the gap between correct and easily-mistaken words and is beneficial for quadruplets prediction.

Secondly, it is found that the full model slightly outperforms the variant of removing ME, suggesting that ME is able to enhance likelihood learning and balance its effects with MUL. We also observe that -MUL-ME+UL, brings consistent degradation. In most experimental settings, it is less performed than -MUL+UL. This further demonstrates the effectiveness of ME.

Finally, we also see that the full model consistently outperforms the variant of removing MC dropout on four datasets. The observation shows that understanding the uncertainty of language models is important to choose crucial mistakes. Making language models distinguish these easily-mistaken words contributes to ASQP.

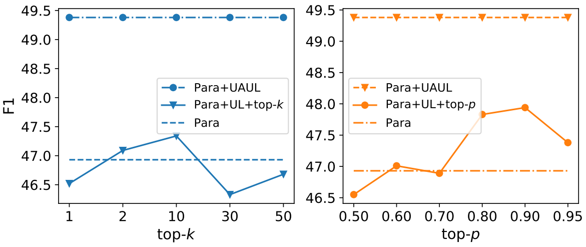

Figure 3: Evaluation results of other strategies on Rest15. Para denotes the Paraphrase method.

3.3.4 Comparison with Other Strategies

We further compare our method with choosing negative samples via top-k (Fan et al., 2018) and top-p (Holtzman et al., 2020) strategies. Previously, these two strategies are exploited in the inference phase of text generation. Here we borrow their idea to select negative samples in the training phase. Specifically, except for the ground-truth token, all other samples from top-k and top-p are regarded as negative. The evaluation results are depicted in Figure 3.

We first see that introducing unlikelihood learning with top-k or top-p strategies can both bring some gains by setting specific k or p values. This demonstrates that learning negative information is effective for ASQP. Yet the gains are very limited. Then it can be observed that Paraphrase+UAUL achieves significant improvements, showing the effectiveness of our approach. This suggests that considering the uncertainty of language models can successfully choose more valuable samples.

3.3.5 Hyperparameter Study

The effects of two hyperparameters are also studied, i.e. m and p, where m is the margin in Eq. (4) and p is the MC dropout rate. The curves are depicted in Figure 4.

Hyperparameter m It determines the gap extent to learn from negative samples. Fixing dropout to 0.4 and keeping all other parameters the same, we vary m from -1.0 to 0.2. In the left plot of Figure 4, it is found that with most of the values, Special_Symbols+UAUL outperforms the original model. It shows that this hyperparameter has robustness to some extent. Then setting m too small or too large leads to a decrease in performance. If the gap extent is too large, it probably causes

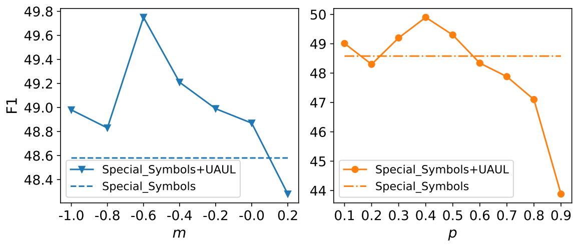

Figure 4: Effect of hyperparameters on Rest15.

Inputs-1	*was considering an apple product but i ' m much happier with this !*
Label-1	*(it, happier, laptop general, positive)*
Pred-1	*(apple product, happier, laptop general, positive)* ✖
Inputs-2	*i then noticed a dead pixel on the screen .*
Label-2	*(screen, NULL, display quality, negative)*
Pred-2	*(screen, dead, display quality, negative)* ✖

Figure 5: Two error cases predicted by Paraphrase+UAUL from the testing set of Laptop dataset.

overfitting, while if too small, the influences of uncertainty-aware negative samples are limited.

Dropout Rate p It determines the proportions of neural connections to drop. Setting various values of p will lead to different extents of uncertainty for language models. As shown in the right plot of Figure 4, we find that keeping p within 0.1 to 0.5 yields good results. However, if p is set to a large value, the scale of weights for optimizing is limited, which affects likelihood training and in turn, causes performance degradation.

3.3.6 Error Analysis

To better understand the limitations of UAUL, we choose a typical method Paraphrase+UAUL, and conduct an error analysis. Two failed cases are shown in Figure 5.

The first sentence implicitly describes an aspect term that the user is *"much happier with"* rather than *"apple product"*. Thus the ground-truth aspect term is *"it"*, yet our approach predicts *"apple product"*. Similarly, the second case expresses the negative opinion towards *"screen"* since it has a *"a dead pixel"*. The opinion term is also implicit, but our approach predicts wrongly to an adjective *"dead"*. In summary, an aspect/opinion term may be described implicitly, which requires deep semantic understanding. Though UAUL achieves consistent performance improvements for various generation-based methods, it struggles to deal with implicit information.

4 Related Work

Aspect-Base Sentiment Analysis (ABSA) Early studies of ABSA stay at the level of individual elements, such as extracting aspect terms (Xu et al., 2018), detecting aspect categories (Bu et al., 2021; Brauwers and Frasincar, 2022), predicting the sentiment polarity given an aspect term (Huang and Carley, 2018) or an aspect category (Hu et al., 2019). Subsequently, researchers (Schouten and Frasincar, 2015; Zhang et al., 2022) pay attention to the dependencies of multiple elements and recognize them simultaneously. Peng et al. (2020) focus on the triplet of aspect opinion sentiment. Recently, ASQP has drawn much attention, dealing with the whole elements, i.e. aspect sentiment quadruplets. To address ASQP, pipeline method (Cai et al., 2021) and generation-based method (Zhang et al., 2021b,a) are proposed. Due to the simplicity and end-to-end manner, the generation paradigm has become the main research direction. Promising works design novel approaches based on tree structure (Mao et al., 2022; Bao et al., 2022), contrastive learning (Peper and Wang, 2022) and data augmentation (Hu et al., 2022). Different from the above works, we study ASQP from the perspective of what not to generate and design novel uncertainty-aware unlikelihood learning for the ASQP task.

Unlikelihood Learning It is originally proposed in the field of neural text generation (Welleck et al., 2020). It aims to deal with the generation repetition problem, which records the words that have been decoded and suppress their probabilities in the following decoding time steps. Li et al. (2020) introduce unlikelihood loss into dialog generation to address the utterance repetition, frequent words, and logical flaw issues. Song et al. (2021) leverage unlikelihood training to improve the understanding of character consistency in the persona-based dialogue. In this work, semantic-similar or ambiguous tokens are negative information for ASQP. We acquire them via the inherent uncertainty of language models and propose novel marginalized unlikelihood learning to deal with negative samples.

5 Conclusion

Generation-based paradigm has become a new trend for ASQP. Yet previous works mainly consider what to generate but ignore what not to generate. In this work, we propose a template-agnostic uncertainty-aware unlikelihood learning (UAUL) method to address negative samples. We acquire easily-mistaken samples by modeling the built-in uncertainty of language models. Then based on the mistakes, we propose marginalized unlikelihood learning to promote the distinguishable of the noises and errors. To balance the impact of marginalized unlikelihood learning, we design minimization entropy. Extensive experiments on various generate-based methods demonstrate UAUL has universal effectiveness towards different templates.

Limitations

Our work is the first study of generative ASQP task from the view of what not to generate. Despite the state-of-the-art performance and template-agnostic effectiveness, our work still has limitations that may guide the direction of future work.

Firstly, implicit information is still challenging for UAUL. Failed cases in error analysis §3.3.6 demonstrate that tough cases require in-depth semantic understanding. Though UAUL achieves wide improvements in the generation paradigm, it struggles to deal with implicit cases.

Secondly, in this work, we only design token-level marginalized unlikelihood learning. Since aspect sentiment quadruplets contain four types of information, considering span-level and whole sequence-level negative sample learning may attain further gains.

Thirdly, UAUL increases the training time, as shown in Table 8. We optimize the implementation by parallel computation. Meanwhile, MC dropout is only adopted in the last dropout layer. The training time is still significantly enlarged. Nevertheless, our method does not require additional human labor, which has obvious advantages in real applications.

Acknowledgements

We sincerely thank all the anonymous reviewers for providing valuable feedback. This work is supported by the youth program of National Science Fund of Tianjin, China (Grant No. 22JC-QNJC01340), the Fundamental Research Funds for the Central University, Nankai University (Grant No. 63221028 and No.63232114), the key program of National Science Fund of Tianjin, China (Grant No. 21JCZDJC00130).

References

Xiaoyi Bao, Z Wang, Xiaotong Jiang, Rong Xiao, and Shoushan Li. 2022. Aspect-based sentiment analysis with opinion tree generation. *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 4044–4050.

Gemma Boleda. 2020. Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6:213–234.

Gianni Brauwers and Flavius Frasincar. 2022. A survey on aspect-based sentiment classification. *ACM Computing Surveys*, pages 1–37.

Jiahao Bu, Lei Ren, Shuang Zheng, Yang Yang, Jingang Wang, Fuzheng Zhang, and Wei Wu. 2021. ASAP: A Chinese review dataset towards aspect category sentiment analysis and rating prediction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 2069–2079.

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 340–350.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.

Zhifang Fan, Zhen Wu, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2019. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (NAACL)*, pages 2509–2518.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pages 1321–1330.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, pages 1–16.

Mengting Hu, Yike Wu, Hang Gao, Yinhao Bai, and Shiwan Zhao. 2022. Improving aspect sentiment quad prediction via template-order data augmentation. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 7889–7900.

Mengting Hu, Shiwan Zhao, Li Zhang, Keke Cai, Zhong Su, Renhong Cheng, and Xiaowei Shen. 2019. CAN: Constrained attention networks for multi-aspect sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4601–4610.

Binxuan Huang and Kathleen Carley. 2018. Parameterized convolutional neural networks for aspect level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1091–1096.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2020. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4715–4728.

Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. 2021. Towards emotional support dialog systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 3469–3483.

Yue Mao, Yi Shen, Jingchao Yang, Xiaoying Zhu, and Longjun Cai. 2022. Seq2path: Generating sentiment tuples as paths of a tree. In *Findings of the Association for Computational Linguistics (ACL)*, pages 2215–2225.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 8600–8607.

Joseph J Peper and Lu Wang. 2022. Generative aspect-based sentiment analysis with contrastive learning and expressive structure. *arXiv preprint arXiv:2211.07743*.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *International workshop on semantic evaluation (SemEval 2016)*, pages 19–30.

Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 486–495.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion Word Expansion and Target Extraction through Double Propagation. *Computational Linguistics*, 37(1):9–27.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)*, 21(140):1–67.

Kim Schouten and Flavius Frasincar. 2015. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830.

Haoyu Song, Yan Wang, Kaiyan Zhang, Wei-Nan Zhang, and Ting Liu. 2021. BoB: BERT over BERT for training persona-based dialogue models from limited personalized data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (ACL-IJCNLP)*, pages 167–177.

Artem Vazhentsev, Gleb Kuzmin, Artem Shelmanov, Akim Tsvigun, Evgenii Tsymbalov, Kirill Fedyanin, Maxim Panov, Alexander Panchenko, Gleb Gusev, Mikhail Burtsev, et al. 2022. Uncertainty estimation of transformer predictions for misclassification detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8237–8252.

Hai Wan, Yufei Yang, Jianfeng Du, Yanan Liu, Kunxun Qi, and Jeff Z Pan. 2020. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 9122–9129.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *International Conference on Learning Representations (ICLR)*, pages 1–17.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

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

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020a. Position-aware tagging for aspect sentiment triplet extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2339–2349.

Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. 2020b. MEGATRON-CNTRL: Controllable story generation with external knowledge using large-scale language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2831–2845.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021a. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9209–9219.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021b. 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 (ACL-IJCNLP)*, pages 504–510.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A survey on aspect-based sentiment analysis: tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*.

A Experimental Details

A.1 Software and Hardware

The details of the software and hardware environments are as follows.

- **System**: Ubuntu 9.4.0; Python3.8; PyTorch 1.7.0
- **CPU**: 12 vCPU Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz
- **GPU**: NVIDIA GeForce RTX 3090

A.2 Datasets Details

All four of our datasets are publicly available and are linked as follows:

- `Rest15`, `Rest16`: https://github.com/IsakZhang/ABSA-QUAD
- `Restaurant`, `Laptop`: https://github.com/NUSTM/ACOS

Models	**Learning Rate**	**Beam Size**
GAS	3e-4	1
Paraphrase	3e-4	1
Special_Symbols	3e-4	1
DLO	1e-4	5
ILO	1e-4	5

Table 6: Hyperparameters of baseline methods. Beam size is the number of paths searched by the beam search at the inference stage. And beam size is 1, indicating that the inference stage uses the greedy search for decoding.

Datasets	m	p	α	MC forward num
`Rest15`	-0.6	0.4	10	5
`Rest16`	-0.3	0.4	10	5
`Restaurant`	-0.6	0.4	10	5
`Laptop`	-0.1	0.4	10	5

Table 7: Hyperparameters of the proposed UAUL.

A.3 Implementations of Baseline Methods

We choose the following open-source generation-based methods to evaluate the proposed UAUL.

- **GAS**: https://github.com/IsakZhang/Generative-ABSA
- **Paraphrase**: https://github.com/IsakZhang/ABSA-QUAD
- **Special_Symbols**, **DLO**, **ILO**: https://github.com/hmt2014/AspectQuad

For all baseline methods, the number of epochs is set to 20. The batch size is 16. Other parameters are shown in Table 6. In addition, we also depict the template details of each baseline method in Figure 6. It is worth noting that ILO and DLO also follow the special symbols templates but combine multiple template orders as data augmentation.

A.4 Implementations of Our Method

In the experiments, all the reported results are the average of 5 runs. For all baselines and our methods, we use T5-base (Raffel et al., 2020) as our pre-trained language model. And when applying our method, we keep all the parameters the same as in the baseline method 6. The hyperparameter details are presented in Table 7.

Inputs Sentence	*The food is good.*
Quadruplet (*at, ot, ac, sp*)	(*food*, *good*, *food quality*, *positive*)
Semantic Mapping (x_{at}, x_{ot}, x_{ac}, x_{sp})	(*food*, *good*, *food quality*, *great*)
GAS	(*at, ot, ac, sp*)
Target sequence	(*food*, *good*, *food quality*, *positive*)
Paraphrase	x_{ac} *is* x_{sp} *because* x_{at} *is* x_{ot}
Target sequence	*food quality is great because food is good*
Special Symbols	[AT] x_{at} [OT] x_{ot} [AC] x_{ac} [SP] x_{sp}
Target sequence	[AT] *food* [OT] *good* [AC] *food quality* [SP] *great*

Figure 6: Template details of the various methods.

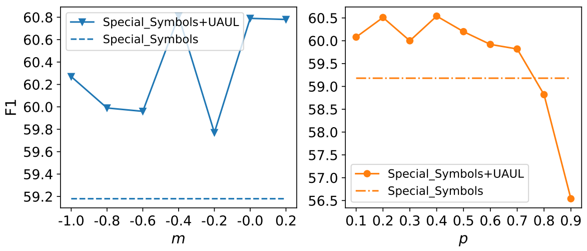

Figure 7: Effect of hyperparameters on Restaurant

Models	Rest15	Restaurant
GAS +UAUL	356s 616s	556s 1043s
Paraphrase +UAUL	347s 609s	555s 1034s
Special_Symbols +UAUL	316s 605s	623s 1204s
DLO +UAUL	1400s 1895s	2346s 3516s
ILO +UAUL	1352s 1902s	2325s 3640s

Table 8: Average running time of each model.

B Additional Experimental Results

B.1 Additional Hyperparameter Study

Hyperparameter studies on Restaurant and Rest16 are depicted in Figure 7 and Figure 8. It can be found that on Restaurant, our method outperforms Special_Symbols on various values of m. On Rest16, our method also outperforms Special_Symbols in most cases. This demonstrates that hyperparameter m has high robustness. For the dropout rate p, it is found that keeping p within 0.1 to 0.5, the results are the best, but setting p larger than 0.7 causes performance degradation. A possible explanation is that dropping out too much scale of neural connections reduces the proportion of learnable parameters.

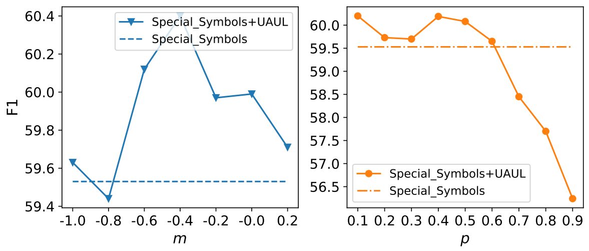

Figure 8: Effect of hyperparameters on Rest16

B.2 Training Time Analysis

The average running time of each model is shown in Table 8. We can observe that on five generation-based methods, UAUL consistently causes more training time. Even UAUL has already used parallel computation and the last layer MC dropout in the training phase, the training time is still extremely enlarged. Admittedly the time overhead is a limitation of our approach, but our method does not require additional human labor, which is also very beneficial in practical applications.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section limitation

☒ A2. Did you discuss any potential risks of your work?
We are performing sentiment analysis on sentences commented by others, without any potential risk involved.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

section 3.1

☑ B1. Did you cite the creators of artifacts you used?
section 3.1, section A.2

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
This dataset is a public dataset, and the paper proposing the dataset is already cited in the text, so no additional description is needed.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
This is the labeled data and the goal is to train the neural network, so this part does not need to be discussed separately.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We use publicly available datasets without invading the privacy of others

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
All use a specific publicly available dataset for sentiment analysis, and no additional discussion is needed for this part.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
section 3.1

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

section 3.3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
section A.1, section B.2

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
section A.3, section A.4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
section 3.3, section A.4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
section A.4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☒ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☒ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.