UniCOQE: Unified Comparative Opinion Quintuple Extraction as a Set

Zinong Yang[1], **Feng Xu**[2], **Jianfei Yu**[1], **and Rui Xia**[1*]
[1]School of Computer Science and Engineering,
Nanjing University of Science and Technology, Nanjing, China
[2]School of Accountancy,
Nanjing University of Finance and Economics, China
[1]{znyang, jfyu, rxia}@njust.edu.cn, [2]breezewing@nufe.edu.cn

Abstract

Comparative Opinion Quintuple Extraction (COQE) aims to identify comparative opinion sentences in product reviews, extract comparative opinion elements in the sentences, and then incorporate them into quintuples. Existing methods decompose the COQE task into multiple primary subtasks and then solve them in a pipeline manner. However, these approaches ignore the intrinsic connection between subtasks and the error propagation among stages. This paper proposes a unified generative model, UniCOQE, to solve the COQE task in one shot. We design a generative template where all the comparative tuples are concatenated as the target output sequence. However, the multiple tuples are inherently not an ordered sequence but an unordered set. The pre-defined order will force the generative model to learn a false order bias and hinge the model's training. To alleviate this bias, we introduce a new *"predict-and-assign"* training paradigm that models the golden tuples as a set. Specifically, we utilize a set-matching strategy to find the optimal order of tuples. The experimental results on multiple benchmarks show that our unified generative model significantly outperforms the SOTA method, and ablation experiments prove the effectiveness of the set-matching strategy.

1 Introduction

As an essential branch of opinion mining, comparative opinion mining aims to explore the information of comparisons in product reviews. Jindal and Liu (2006b) first proposed the concept of comparative opinion mining and introduces two primary subtasks: Comparative Sentence Identification (CSI) and Comparative Element Extraction (CEE). The former aims to identify whether a given sentence is a comparative sentence, while the latter aims to extract all the comparative elements in a given comparative sentence. Panchenko et al. (2019) further proposed the Comparative Preference Classification (CPC) task, which seeks to predict the comparative preference (BETTER, WORSE, or NONE) of a provided comparative sentence.

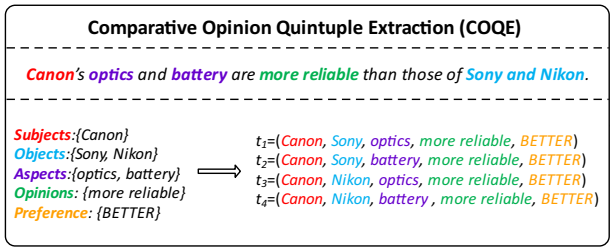

Figure 1: An example of the Comparative Opinion Quintuple Extraction (COQE) task. Given a product review sentence, COQE aims to identify whether it is a comparative sentence, extract all the comparative elements (if existing), and incorporate them into quintuples.

In order to integrate various subtasks of comparative opinion mining, Liu et al. (2021) first proposed the Comparative Opinion Quintuple Extraction (COQE) task (shown in Fig. 1). COQE aims to identify comparative opinion sentences in product reviews and extract five comparative opinion elements in the sentences, i.e., comparative subject (*sub*), comparative object (*obj*), comparative aspect (*ca*), comparative opinion (*co*) and comparative preferences (*cp*), and then incorporate them into a quintuple (*sub, obj, ca, co, cp*). Liu et al. (2021) adopted a multi-stage model, decomposing the COQE task into primary sub-tasks (CSI, CEE, and CPC formerly mentioned), and then solving them one by one in a pipeline manner. However, the pipeline model ignores the internal connection between multiple subtasks of comparative opinion mining, and the error propagation between each stage heavily strains the model's performance.

To this end, we employ a generative extraction model called UniCOQE for the first time on the COQE task. We utilize T5 (Raffel et al., 2020) as the backbone and propose a generative template to adapt to the COQE task, identifying comparative

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 12229–12240

sentences and extracting all quintuples once and for all.

In the generation paradigm, we concatenate all the golden comparative tuples together as the target output sequence of the model. However, multiple tuples are essentially not an ordered sequence but an unordered set. If a pre-defined order is imposed, it will introduce an order bias, forcing the generative model to learn the bias, which hinders the model's training. Taking Fig. 1 as an example, there are four target tuples: t_1, t_2, t_3, and t_4. Theoretically, $A_4^4 = 24$ types of permutations of target tuples are all correct. During training, the model would get "confused": Why $t_1; t_2; t_3; t_4$ is correct but $t_4; t_3; t_2; t_1$ is unacceptable?

In order to alleviate this order bias problem, we introduce a "predict-and-assign" training paradigm to the generative model. During the training phase, we first let the model autoregressively predict comparative tuples in the given sentence. Subsequently, we model the golden tuples as a set and use the Hungarian algorithm (Kuhn, 1955) to match the set of golden tuples with the predicted sequence to find the optimal order of golden tuples.

Finally, we validate the performance of our approach on three COQE benchmarks. Experimental results show that our model significantly outperforms SOTA methods, and the effectiveness of the set-matching strategy is demonstrated through ablation experiments.

The contributions of this paper can be summarized as follows:

- We propose a generative comparative opinion quintuple extraction model to solve the error propagation problem of previous multi-stage models.
- We introduce the "predict-and-assign" training paradigm based on a set-matching strategy to alleviate the order bias of the generative model during training.
- Our model significantly outperforms previous SOTA models, and ablation experiments verify the effectiveness of the set-matching strategy.

2 Related Works

As an important subtask of opinion mining, the task of comparative opinion mining was first proposed by Jindal and Liu (2006a,b), which aims to identify comparative sentences in product reviews and extract all the comparative opinion elements (entities, features, and comparative keywords). Specifically, it used class sequential rules(Hu and Liu, 2006) to identify comparative sentences and label sequential rules to extract comparative elements.

Some subsequent studies concentrated on the comparative sentence identification (CSI) task. Huang et al. (2008) used diverse features (e.g., keywords and sequential patterns) to recognize comparative sentences. Park and Blake (2012) exploited semantic and grammatical features to explore the task of identifying comparative sentences in scientific texts. Liu et al. (2013) recognized comparative sentences on Chinese documents based on keywords, sentence templates, and dependency analysis.

On the comparative element extraction (CEE) task, Hou and Li (2008) used semantic role labeling (SRL) to analyze the structure of comparative sentences and trains a conditional random field (CRF) to extract comparative features. Some studies (Song et al., 2009; Huang et al., 2010; Wang et al., 2015a) also used CRF as the extraction model. Kessler and Kuhn (2013) further explored the application of existing SRL methods to comparative element extraction. Arora et al. (2017) proposed applying deep learning methods to comparative opinion mining, mainly using an LSTM-CRF framework to extract comparative elements.

Considering the early comparative opinion mining tasks did not include the author's comparative preference, Ganapathibhotla and Liu (2008) proposed the Comparative Preference Classification (CPC) task for the first time, aiming to predict which entity is preferred given a comparative sentence and its comparative elements. It utilized a keyword-based approach to identify comparative preferences. Panchenko et al. (2019) used a pretrained encoder to encode sentences and classified sentences' comparative preference based on XGBoost (Chen and Guestrin, 2016). Ma et al. (2020) employed a graph attention network to model the syntactic parsing information of comparative sentences to better predict comparative preferences. Nevertheless, the premise of the CPC task is that the two entities to be compared are annotated in advance, which is challenging to apply in real-world scenarios.

Liu et al. (2021) first introduced the task of comparative opinion quintuple extraction (COQE),

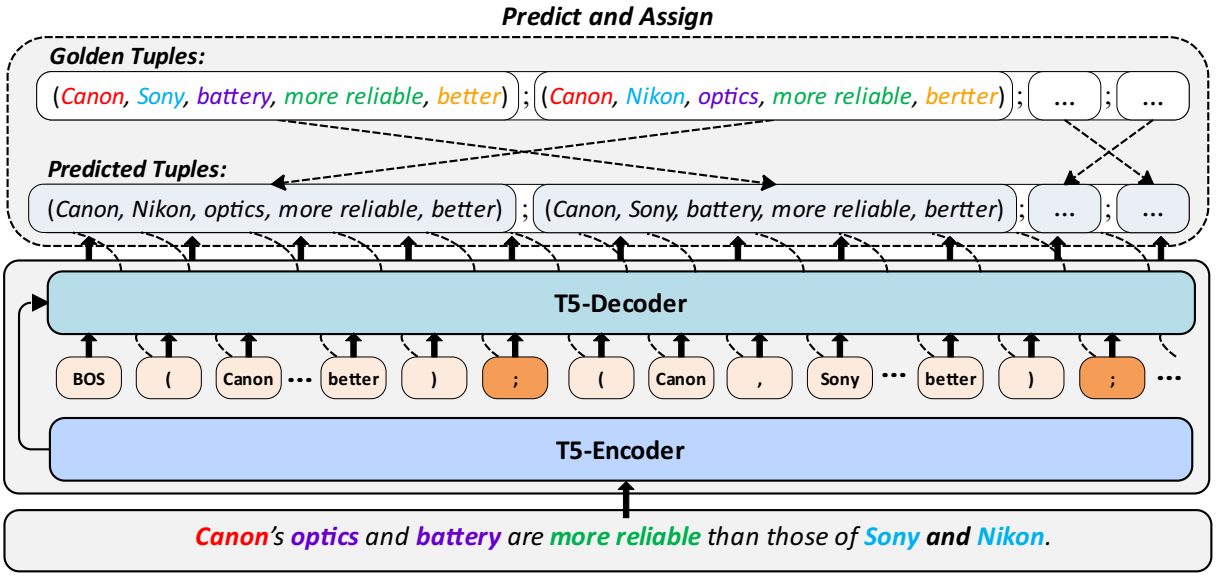

Figure 2: An overview of the UniCOQE framework. We utilize T5 as the backbone of our generative model and employ a ***"predict-and-assign"*** training paradigm to alleviate the order bias of the vallina generative model. During training, we temporarily turn off the model's gradient and let the model autoregressively ***predict*** the tuples. Then we model the golden tuples as a set and use the Hungarian algorithm to match the set of golden tuples with the predicted sequence to ***assign*** the optimal order of golden tuples.

which aims to extract quintuples (comparative subject, comparative object, comparative aspect, comparative opinion, comparative preference). Specifically, it utilized a multi-stage model based on BERT (Devlin et al., 2019) performing CSI, CEE, and CPC tasks at each stage. Although this method serialized multiple subtasks of comparative opinion mining in a pipeline manner, the error propagation across multiple stages undermined the model's performance.

In addition to subtasks such as CSI, CEE, CPC, and COQE, some research directions are also closely related to comparative opinion mining. Comparative question answering system (Alhamzeh et al., 2021; Chekalina et al., 2021) allows the machines to automatically answer the comparative question "Is X better than Y with respect to Z?'. Opinion tuple extraction (Jian et al., 2016; Peng et al., 2020) and quadruple (Cai et al., 2021) extraction in traditional aspect-based sentiment analysis aim to extract fine-grained opinion information in the text.

Several studies have also explored the use of set-matching strategies for generative models. In the keyphrase extraction task, Ye et al. (2021) concatenate all the keyphrases as target outputs of Transformer (Vaswani et al., 2017) without predefining an order. In the event argument extraction task, Ma et al. (2022) introduces a scheme for optimal span assignments of BART (Lewis et al., 2020). These studies demonstrate the effectiveness of set matching strategies in generative models, highlighting their potential for improving the performance of generative LMs.

3 Methodology

This section introduces the UniCOQE framework in detail (as shown in 2). In this framework, we model the COQE task as a natural language generation task. We use the generative pre-trained language model T5 (Raffel et al., 2020) as the backbone model and adopt a generation template to directly identify comparative sentences and output the comparative quintuples therein in an end-to-end manner. To further alleviate the order bias problem of the generative models, we introduce the ***"predict-and-assign"*** training paradigm.

3.1 Task Formulation

We first formulate the COQE task as follows: Given a product review sentence $X = \{x_1, ..., x_n\}$ containing n tokens, COQE aims to identify whether it is a comparative sentence and (if so) extract all comparative quintuples in it:

$$\begin{aligned} S_X &= \{tup_1, ..., tup_k\} \\ &= \{(sub_1, obj_1, ca_1, co_1, cp_1), ..., \\ &\quad (sub_k, obj_k, ca_k, co_k, cp_k)\} \end{aligned} \quad (1)$$

where k is the number of comparative quintuples extracted from comparative sentence X. $tup = (sub, obj, ca, co, cp)$ is an extracted quintuple, where sub is the subject entity, obj is the object entity, ca is the aspect being compared, co is the opinion of the author reflecting a comparative preference. $cp \in$ {WORSE, EQUAL, BETTER, DIFFERENT} is the comparative preference of the author.

3.2 COQE with Generative Paradigm

In this section, we introduce the generative paradigms for the COQE task. We design a T5 generation template for end-to-end extraction of quintuples. Examples are as follows:

> ***Input***: *Canon's optics and battery are more reliable than those of Sony and Nikon.*
> ***Target***:
> *(Canon, Sony, optics, more reliable, BETTER);*
> *(Canon, Sony, battery, more reliable, BETTER);*
> *(Canon, Nikon, optics, more reliable, BETTER);*
> *(Canon, Nikon, battery, more reliable, BETTER)*

> ***Input***: *Canon's optics and battery are so great.*
> ***Target***: *(unknown, unknown, unknown, unknown, unknown)*

In the generative paradigm, k golden quintuples are concatenated with " ; " as the target sequence of the model. If a comparison element does not exist, it is padded with the word "*unknown*". If the target sequence is"*(unknown, unknown, unknown, unknown, unknown)*", the corresponding input sentence X is then considered a non-comparative sentence. We call this approach the *Vallina* generative paradigm.

Still, a problem exists with the Vallina generative paradigm: The k target tuples are essentially an unordered set, rather than an ordered sequence. The training of the generative model is fundamentally based on the cross-entropy loss, depending heavily on the order of the target text sequence. In multi-tuple scenarios, artificially predefining an order can introduce a false order bias during training, undermining the model's performance.

3.3 Improving Generative COQE with Predict-and-Assign Paradigm

To address the order bias problem, we introduce a "predict-and-assign" training paradigm. The paradigm incorporates two steps: predicting step and assigning step.

3.3.1 Predicting Stage

For the input sentence $X = \{x_1, ..., x_n\}$, during the training phase, we temporarily turn off the gradient backpropagation of the model and send X into the T5-encoder to get the latent representation of the sentence :

$$h^{enc} = \mathbf{Encoder}(X) \tag{2}$$

We then used T5-decoder to predict all the comparative quintuples autoregressively. At the c_{th} moment of the decoder, h^{enc} and the previous output tokens: $t_{1:c-1}$ are utilized as the input into the decoder:

$$h_c^{dec} = \mathbf{Decoder}(h^{enc}, t_{1:c-1}) \tag{3}$$

The conditional probability of token t_c is defined as follows:

$$P(t_c|t_{1:c-1}, X) = \mathrm{Softmax}(h_c^{dec}W + b) \tag{4}$$

where $W \in \mathbb{R}^{d_h \times |\mathcal{V}|}$, $b \in \mathbb{R}^{|\mathcal{V}|}$. $\mathcal{V}$ here refers to the vocabulary size of T5. Then the final predicted sequence of tuples is:

$$T_{pred} = t_{1:m} = \{t_1, ..., t_m\} \tag{5}$$

where m is the length of the predicted sequence. We split T_{pred} with the semicolon symbol " ; " to get a set of comparative quintuple predicted by the model: $Q_{pred} = \{tup_1^{pred}, ..., tup_l^{pred}\}$.

3.3.2 Assigning Stage

Given two tuples: p and g, we define the similarity score between p and g as follows:

$$sim(p, g) = \frac{1}{n}\sum_{k=1}^{n} \mathrm{IoU}\Big(p^{(k)}, g^{(k)}\Big) \tag{6}$$

where n is the number of elements in tuples. In our case, $n = 5$ constantly for we have five elements(i.e., *sub*, *obj*, *ca*, *co*, and *cp*) in the comparative quintuples. IoU here refers to the "intersection over union" of the two token sequences, and k refers to the index of the element (e.g., $k = 3$ for *ca*). Therefore, $\mathrm{IoU}(p^{(k)}, g^{(k)})$ calculates the IoU score of the k-th element of both tuples. We eventually take the average IoU score of all five elements as the similarity score of two tuples. For example, in Fig. 3, we have tuple p_1 = *(Canon, Nikon, sensors, less stable, WORSE)*, and g_2 = *(Canon, Sony, sensors, less stable, WORSE)*, the element-wise IoU scores are 1, 0, 1, 1, and 1, respectively. So the similarity score between p_1 and g_2 is 0.8.

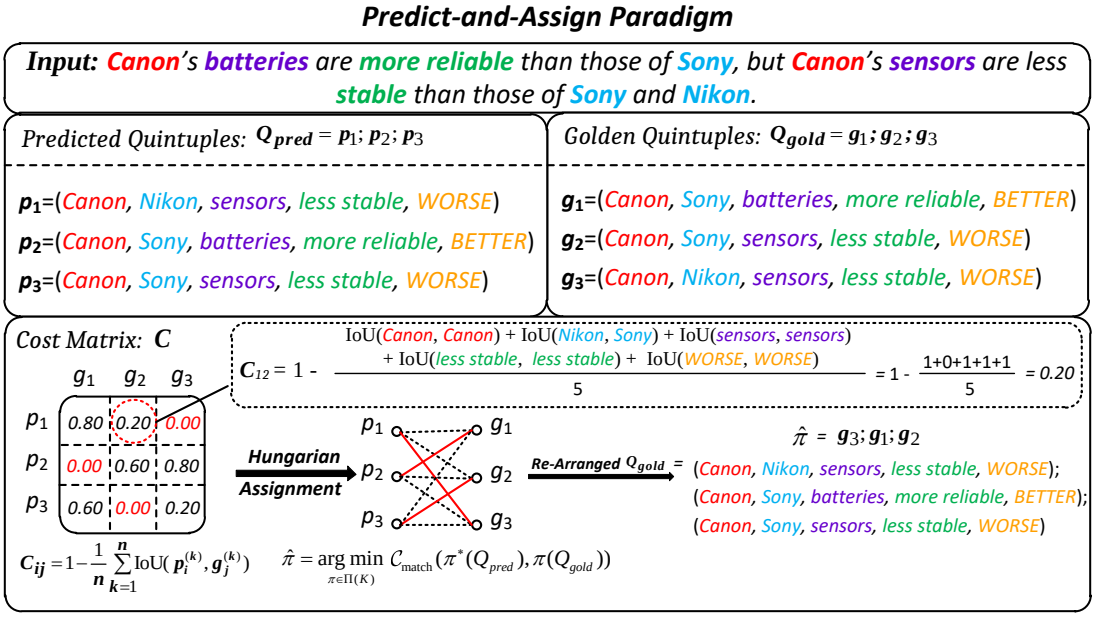

Figure 3: An example of the predict-and-assign paradigm.

We then define the assignment cost between p and q:

$$cost(p, g) = 1 - sim(p, g) \quad (7)$$

For the ground-truth tuple set $Q_{gold} = \{tup_1^{gold}, ..., tup_K^{gold}\}$, we aim to find a permutation $\hat{\pi}$ of Q_{gold}, so that $\hat{\pi}(Q_{gold})$ is the most similar sequence to the tuples predicted by the model in predicting stage (Section.3.3.1). This is essentially an assignment (a.k.a binary matching) problem.

Formally, to find an optimal order of ground-truth tuples Q_{gold}, we search for a permutation $\hat{\pi}$ that minimizes the total assignment cost:

$$\hat{\pi} = \underset{\pi \in \Pi(K)}{\arg\min}\, \mathcal{C}_{\text{match}}\Big(\pi^*(Q_{pred}), \pi(Q_{gold})\Big) \quad (8)$$

where K is the number of tuples in Q_{gold}. $\Pi(K)$ is the space of permutations of K tuples in Q_{gold}. $\pi^*(Q_{pred})$ is the predicted sequence of tuples in Formula (5). This process of finding the optimal assignment can be solved efficiently by Hungarian algorithm (Kuhn, 1955). $\mathcal{C}_{\text{match}}(\pi^*, \hat{\pi})$ is the total pair-wise matching cost between permutation π^* and permutation $\hat{\pi}$. The assignment cost can be defined as follows:

$$\begin{aligned} &\mathcal{C}_{\text{match}}\Big(\pi^*(Q_{pred}), \pi(Q_{gold})\Big) \\ &= \sum_{i=1}^{s} cost\Big(\pi^*(Q_{pred})_i, \pi(Q_{gold})_i\Big) \end{aligned} \quad (9)$$

where $s = \min(|Q_{pred}|, |Q_{gold}|)$ is the minimum number of tuples between Q_{pred} and Q_{gold}.

	Car-COQE	Ele-COQE	Camera-COQE
#Subject	1520	950	1649
#Object	2121	1980	1316
#Aspect	1917	1602	1368
#Opinion	2171	2089	2163
#Preference	2695	2289	2442
#Comparative	1747	1800	1705
#Non-Comparative	1800	1800	1599
#Multi-Comparisons	550	361	500
#Comparisons Per Sent	1.5	1.3	1.4

Table 1: Statistics of three COQE datasets.

$\pi^*(Q_{pred})_i$ and $\pi(Q_{gold})_i$ refer to the i_{th} tuple in $\pi^*(Q_{pred})$ and $\pi(Q_{gold})$ respectively.

After assigning the new order of the golden tuples, we take the new order as the training target of the model and re-open the gradient backpropagation to restart training.

4 Experiments

4.1 Datasets

We conduct experiments on three COQE datasets released by Liu et al. (2021): Camera-COQE, Car-COQE, and Ele-COQE:

- **Camera-COQE** contains English product reviews in the camera domain. This dataset is based on Kessler and Kuhn (2014), completing the annotations of comparative opinions (co) and comparative preferences (cp).
- **Car-COQE** contains Chinese product reviews in the automobile domain. This dataset is based on the Car dataset in the

Models	Camera-COQE		Car-COQE		Ele-COQE	
	CSI	COQE	CSI	COQE	CSI	COQE
Multi-Stage$_{\text{CSR-CRF}}$	65.38	3.46	86.90	5.19	88.30	4.07
Joint$_{\text{CRF}}$	82.14	4.88	89.85	8.65	85.97	4.71
Multi-Stage$_{\text{LSTM}}$	87.14	9.05	92.68	10.28	96.25	14.90
Multi-Stage$_{\text{BERT}}$	93.04	13.36	97.39	29.75	98.31	30.73
UniCOQE	**95.21**	**31.95**	**98.28**	**36.55**	**98.41**	**35.46**

Table 2: Results of different approaches for CSI and COQE under the Exact Match metric.

COAE2012/2013 (Tan et al., 2013), supplemented with annotations of comparative opinions (*co*) and comparison preferences (*cp*).

- **Ele-COQE** similarly derives from the electronic product review dataset in COAE2012/2013 (Tan et al., 2013), which contains Chinese comparative product reviews of electronic products.

The statistics of the three datasets are demonstrated in Table 1. Each dataset contains both non-comparative and comparative sentences. #Comparative indicates the number of comparative sentences, and #Non-Comparative refers to the number of non-comparative sentences. #Multi-Comparisons is the number of comparative sentences containing multiple comparisons.

4.2 Experimental Setup

We employ T5 as the backbone model. We utilize T5 for the English dataset and Multilingual T5 (mT5) (Xue et al., 2021) for the Chinese datasets. We did not choose the Chinese T5 model because there are multiple non-Chinese characters (i.e., product names and versions) in the Car-COQE and Ele-COQE. We employ T5-base and mT5-base provided by Huggingface[1] library for experiments. For T5 and mT5, we set the batch size to 24 and 10, respectively. The learning rates of both models are set to 3e-4. We train T5 for 60 epochs and mt5 for 30 epoches.

4.3 Evaluation Metrics

Following the setting of Liu et al. (2021), for the comparative sentence identification (CSI) task, We report the Accuracy metric. For the COQE task, we consider three matching strategies: Exact Match, Proportional Match, and Binary Match. These three metrics measure the F1 scores to varying degrees on the predicted tuples by the models.

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

Specifically, for the three metrics, we define $\#correct_e$, $\#correct_p$, $\#correct_b$ as follows:

$$\#correct_e = \begin{cases} 0, & \exists(g_k \neq p_k) \\ 1, & \text{otherwise} \end{cases} \quad (10)$$

$$\#correct_p = \begin{cases} 0, & \exists(g_k \neq p_k = \varnothing) \\ \frac{\sum_k len(g_k \cap p_k)}{\sum_k len(p_k)}, & \text{otherwise} \end{cases} \quad (11)$$

$$\#correct_b = \begin{cases} 0, & \exists(g_k \cap p_k = \varnothing) \\ 1, & \text{otherwise} \end{cases} \quad (12)$$

where g_k is the kth element of a golden comparison quintuple, and p_k is the kth element of a predicted comparison quintuple. $len(\cdot)$ represents the length of the comparison element.

4.4 Baseline Models

We take the following baseline models for comparison :

Multi-Stage$_{\text{CSR-CRF}}$ (Jindal and Liu, 2006a) uses an SVM based on CSR features to identify comparative sentences and uses a CRF to extract comparative elements.

Joint$_{\text{CRF}}$ (Wang et al., 2015b) uses CRF to jointly extract comparative sentences and comparative elements.

Multi-Stage$_{\text{LSTM}}$ (Liu et al., 2021) utilizes an LSTM as a text encoder. The method decomposes the COQE task into three subtasks: comparative sentence identification, comparative element extraction, and comparative preference classification, and solves these subtasks successively in a pipeline manner.

Multi-Stage$_{\text{BERT}}$ (Liu et al., 2021) is a variant of **Multi-Stage$_{\text{LSTM}}$**, specifically, replacing the text encoder with BERT.

4.5 Main Results

In Table.2, we report the performance of all five methods on the two tasks of CSI and COQE on

Dataset	Model	Exact	Proportional	Binary
Camera-COQE	Vallina Gen	28.88	39.95	41.88
	UniCOQE	**31.95**	**42.39**	**44.44**
Car-COQE	Vallina Gen	34.85	48.27	50.42
	UniCOQE	**36.55**	**51.60**	**53.80**
Ele-COQE	Vallina Gen	35.08	50.86	53.40
	UniCOQE	**35.46**	**51.47**	**54.05**

Table 3: Ablation study of the set-matching strategy.

Dataset	Model	Exact	Proportional	Binary
Camera-COQE (mt)	Vallina Gen	31.38	38.11	39.03
	UniCOQE	**35.25**	**41.70**	**42.65**
Car-COQE (mt)	Vallina Gen	29.58	40.05	42.10
	UniCOQE	**31.32**	**43.80**	**45.85**
Ele-COQE (mt)	Vallina Gen	25.37	39.91	42.54
	UniCOQE	**27.07**	**41.94**	**44.23**

Table 4: Results under multi-tuple scenarios. "mt" indicates we use the **multi-tuple** data in the test set for evaluation.

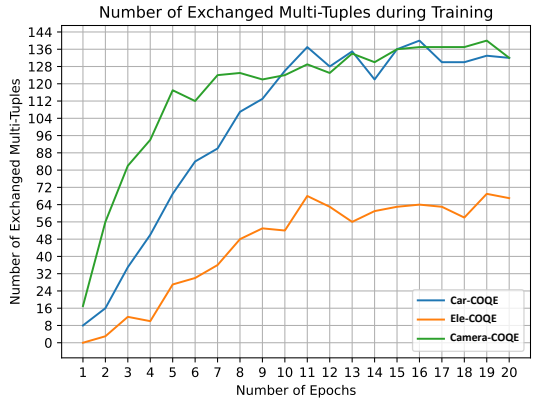

Figure 4: Number of exchanged multi-tuples during training.

the three datasets: Camera-COQE, Car-COQE, and Ele-COQE. For CSI, we report the Accuracy metric. All indicators are in the case of Exact Match.

Experimental results show that the UniCOQE model achieves the best performance on all three datasets on both the CSI task and the COQE task. The two CRF-based methods generally yield the lowest performance on both tasks. **Multi-Stage$_{\text{LSTM}}$** achieves relatively better performance. On the CSI task, **Multi-Stage$_{\text{BERT}}$** has already achieved rather satisfactory results of Accuracy: 93.04, 97.39, and 98.31 on three datasets. However, it is notable that our UniCOQE model still outperforms **Multi-Stage$_{\text{BERT}}$** by 2.17, 0.89, 0.10 percent.

On the COQE task, the UniCOQE model achieves 18.59, 6.80, and 4.73 percent of improvement on the Camera-COQE Car-COQE and Ele-COQE datasets, respectively. It is worth noting that the advantage of our UniCOQE model over other models is more evident on the English dataset than on the Chinese datasets. One possible explanation is that mT5, a multilingual version of the T5, involves the pre-training of multiple languages and has a more expansive vocabulary list, which would weaken the model's performance on monolingual datasets.

4.6 Influence of the Set-Matching Strategy

In Table.3, we show the impact of the set-matching strategy over the generative model. The experimental results show that compared with the Vallina generative model, the set-matching strategy has improved the model's performance on Camera-COQE, Car-COQE, and Ele-COQE datasets under all three metrics. It reveals that the set-matching strategy indeed finds a better order of tuples, helping the model better learn the data distribution.

4.7 Multi-Tuple Scenarios Results

To measure the model's effectiveness on the multi-tuple data, we only use the multi-tuple data in the test set for evaluation. We demonstrate the multi-tuple scenario results in Table.4. The experimental results show that the set-matching strategy has considerably improved the model's performance on multi-tuple data. Taking the Exact match metric as an example, compared to the Valiina generative

Example.1 @ *1st epoch*
Input: *The main reason I chose this model over both the SD 550 and the SD 450 , even though the 550 had a higher megapixel CCD , was that it has more of these features .*
Default Target: *(550, this model, megapixel CCD, higher, better) ; (this model, SD 550, features, over, better) ; (this model, SD 450, features, over, better)*
Cross Entropy Loss: 1.435
Re-ordered Target: *(this model, SD 550, features, over, better) ; (this model, SD 450, features, over, better) ; (550, this model, megapixel CCD, higher, better)*
Cross Entropy Loss: 0.598
Example.2 @ *15th epoch*
Input: *Frankly , it 's just as capable as the D200 EXCEPT for the lower frame rate .*
Default Target: *(it, D200, NONE, as capable, equal) ; (it, D200, frame rate, lower, worse)*
Cross Entropy Loss: 2.244
Re-ordered Target: *(it, D200, frame rate, lower, worse) ; (it, D200, NONE, as capable, equal)*
Cross Entropy Loss: 0.032

Figure 5: Case study of the set-matching strategy.

paradigm, UniCOQE obtains 3.87, 1.74, and 1.70 percent of improvements on the Camera-COQE, Car-COQE, and Ele-COQE, respectively.

4.8 Exchanges of Multi-Tuples

Fig. 4 exhibits the number of exchanges of multi-tuples during the training process of UniCOQE. During the first ten epochs, the number of tuple exchanges keeps on increasing. Around the 11th epoch, all three datasets reach their peak, and the number becomes stabilized. The number of tuple exchanges on Camera and Car is both stabilized at around 140. In contrast, the Electronic dataset is stabilized at around 60, for the Electronic domain has fewer multi-tuple data.

4.9 Case Study

In Fig. 5, we illustrate the effect of the tuple-matching strategy on T5's training procedure. Taking Example.1 as an instance, we can observe that at the very beginning of the model's training (epoch 1), if we follow the default "golden" sequence order, the calculated cross-entropy loss will be 1.453. However, if we assign a new tuple order according to our set-matching strategy, the new loss will become 0.598. The phenomenon is more evident as the training epoch increases. As demonstrated in Example 2, at epoch 15, the default tuple order would end up with a loss of 2.244, whereas the loss of the newly assigned order is much smaller: 0.032.

5 Conclusion

In this paper, we investigate the task of comparative opinion quintuple extraction. To overcome the error propagation problem of previous pipeline models, we propose an extraction model based on the generative paradigm. We further introduce a set-matching strategy based on the Hungarian algorithm to alleviate the order bias of the generative model during training. The experimental results show that our model significantly outperforms the SOTA models, and we verify the effectiveness of the set-matching strategy through in-depth experiments.

6 Limitations and Future Works

We summarize the limitations of our work as follows:

- We only validate the effectiveness of the set-matching strategy for generative models on the COQE task.
- We observe that the scale of the COQE datasets is quite small and has caused the model's overfitting problem.

In the future, we will conduct further research from the following perspectives:

- Explore further application of the set-matching strategy in multiple research directions, such as information extraction, sentiment analysis, etc.
- Utilize unsupervised data to better help the models mine comparative opinion information.
- Design data augmentation methods to relieve the data sparsity problem.

Ethics Statement

We perform experiments on three datasets formerly established by (Liu et al., 2021), namely Camera-COQE, Car-COQE, and Ele-COQE. These datasets do not include personal information or contain any objectionable content that could potentially harm individuals or communities. It's important to note that certain product reviews may include subjective comparisons between products given by anonymous customers, which do not necessarily reflect the preferences of this study.

Acknowledgements

This work was supported by the Natural Science Foundation of China (No. 62076133, 62006117, and 72001102), and the Natural Science Foundation of Jiangsu Province for Young Scholars (No. BK20200463) and Distinguished Young Scholars (No. BK20200018).

References

Alaa Alhamzeh, Mohamed Bouhaouel, Elöd Egyed-Zsigmond, and Jelena Mitrovic. 2021. Distilbert-based argumentation retrieval for answering comparative questions. In *Proceedings of CLEF*, pages 2319–2330.

Jatin Arora, Sumit Agrawal, Pawan Goyal, and Sayan Pathak. 2017. Extracting entities of interest from comparative product reviews. In *Proceedings of CIKM*, pages 1975–1978.

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of ACL*, pages 340–350.

Viktoriia Chekalina, Alexander Bondarenko, Chris Biemann, Meriem Beloucif, Varvara Logacheva, and Alexander Panchenko. 2021. Which is better for deep learning: Python or MATLAB? answering comparative questions in natural language. In *Proceedings of EACL*, pages 302–311.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of KDD*, pages 785–794.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pages 4171–4186.

Murthy Ganapathibhotla and Bing Liu. 2008. Mining opinions in comparative sentences. In *Proceedings of COLING*, pages 241–248.

Feng Hou and Guo-hui Li. 2008. Mining chinese comparative sentences by semantic role labeling. In *Proceedings of ICML*, pages 2563–2568.

Minqing Hu and Bing Liu. 2006. Opinion feature extraction using class sequential rules. In *Proceedings of AAAI*, pages 61–66.

Gao-Hui Huang, Tian-Fang Yao, and Quan-Sheng Liu. 2010. Mining chinese comparative sentences and relations based on crf algorithm. *Chinese Computer Application Research*, pages 2061–2064.

Xiaojiang Huang, Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2008. Learning to identify comparative sentences in chinese text. In *Proceedings of PRICAI*, pages 187–198.

Liao Jian, Li Yang, and Wang Suge. 2016. The constitution of a fine-grained opinion annotated corpus on weibo. In *Proceedings of CCL*, pages 227–240.

Nitin Jindal and Bing Liu. 2006a. Identifying comparative sentences in text documents. In *Proceedings of SIGIR*, pages 244–251.

Nitin Jindal and Bing Liu. 2006b. Mining comparative sentences and relations. In *Proceedings of AAAI*, pages 1331–1336.

Wiltrud Kessler and Jonas Kuhn. 2013. Detection of product comparisons - how far does an out-of-the-box semantic role labeling system take you? In *Proceedings of EMNLP*, pages 1892–1897.

Wiltrud Kessler and Jonas Kuhn. 2014. A corpus of comparisons in product reviews. In *Proceedings of LREC*, pages 2242–2248.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, pages 83–97.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*, pages 7871–7880.

Quanchao Liu, Heyan Huang, Chen Zhang, Zhenzhao Chen, and Jiajun Chen. 2013. Chinese comparative sentence identification based on the combination of rules and statistics. In *AMDA*, pages 300–310.

Ziheng Liu, Rui Xia, and Jianfei Yu. 2021. Comparative opinion quintuple extraction from product reviews. In *Proceedings of EMNLP*, pages 3955–3965.

Nianzu Ma, Sahisnu Mazumder, Hao Wang, and Bing Liu. 2020. Entity-aware dependency-based deep graph attention network for comparative preference classification. In *Proceedings of ACL*, pages 5782–5788.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of ACL*, pages 6759–6774.

Alexander Panchenko, Alexander Bondarenko, Mirco Franzek, Matthias Hagen, and Chris Biemann. 2019. Categorizing comparative sentences. In *Proceedings of the 6th Workshop on Argument Mining*, pages 136–145.

Dae Hoon Park and Catherine Blake. 2012. Identifying comparative claim sentences in full-text scientific articles. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse*, pages 1–9.

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 AAAI*, pages 8600–8607.

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*, pages 140:1–140:67.

Rui Song, Hongfei Lin, and Fuyang Chang. 2009. Chinese comparative sentences identification and comparative relations extraction. *Journal of Chinese Information Processing*, pages 102–107.

Songbo Tan, Liu Kang, Wang Suge, and Liao Xiangwen. 2013. Overview of chinese opinion analysis evaluation 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*, pages 5998–6008.

Wei Wang, TieJun Zhao, GuoDong Xin, and YongDong Xu. 2015a. Exploiting machine learning for comparative sentences extraction. *International Journal of Hybrid Information Technology*, pages 347–354.

Wei Wang, TieJun Zhao, GuoDong Xin, and YongDong Xu. 2015b. Extraction of comparative elements using conditional random fields. *Acta Automatica Sinica*, pages 1385–1393.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of NAACL*, pages 483–498.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. One2Set: Generating diverse keyphrases as a set. In *Proceedings of ACL*, pages 4598–4608.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Yes. We discuss the limitations of our work in Section 6: Limitations and Future Works.

☐ A2. Did you discuss any potential risks of your work?
Not applicable. The main theme of our work is to mine comparative opinions in publically available product reviews, and all the datasets utilized in this paper are also open source.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Yes. Abstract and Section 1: Introduction summarize the paper's main claims. We demonstrate the topic, challenge, and contributions of our paper in both sections.

☒ A4. Have you used AI writing assistants when working on this paper?
No.

B ☐ Did you use or create scientific artifacts?

Not applicable. Not applicable. We do not use or create scientific artifacts in this paper.

☐ B1. Did you cite the creators of artifacts you used?
Not applicable. Not applicable. We do not use or create scientific artifacts in this paper.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Not applicable. We do not use or create scientific artifacts in this paper.

☐ 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)?
Not applicable. Not applicable. We do not use or create scientific artifacts in this paper.

☐ 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?
Not applicable. Not applicable. We do not use or create scientific artifacts in this paper.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Not applicable. We do not use or create scientific artifacts in this paper.

☐ 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.
Not applicable. Not applicable. We do not use or create scientific artifacts in this paper.

C ☑ Did you run computational experiments?

Yes. In section 4: Experiments.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Yes. In section 4.2: Experimental Setup.

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Yes. In section 4.2: Experimental Setup.

☑ 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?
Yes. In section 4.5 Main Results.

☑ 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.)?
Yes. In section 4.2: Experimental Setup.

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

Not applicable. We do not involve human annotations in this paper.

☐ 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.?
Not applicable. Not applicable. We do not involve human annotations in this paper.

☐ 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)?
Not applicable. Not applicable. We do not involve human annotations in this paper.

☐ 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?
Not applicable. Not applicable. We do not involve human annotations in this paper.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Not applicable. We do not involve human annotations in this paper.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Not applicable. We do not involve human annotations in this paper.