Generating Complement Data for Aspect Term Extraction with GPT-2

Amir Pouran Ben Veyseh[1], **Franck Dernoncourt**[2],
Bonan Min[3] **and Thien Huu Nguyen**[1]
[1]Department of Computer Science,
University of Oregon, Eugene, Oregon, USA
[2]Adobe Research, San Jose, CA, USA
[3]Raytheon BBN Technologies, USA
{apouranb, thien}@cs.uoregon.edu
franck.dernoncourt@adobe.com, bonan.min@raytheon.com

Abstract

Aspect Term Extraction (ATE) is the task of identifying the word(s) in a review text toward which the author express an opinion. A major challenges for ATE involve data scarcity that hinder the training of deep sequence taggers to identify rare targets. To overcome these issues, we propose a novel method to better exploit the available labeled data for ATE by computing effective complement sentences to augment the input data and facilitate the aspect term prediction. In particular, we introduce a multi-step training procedure that first obtains optimal complement representations and sentences for training data with respect to a deep ATE model. Afterward, we fine-tune the generative language model GPT-2 to allow complement sentence generation at test data. The REINFORCE algorithm is employed to incorporate different expected properties into the reward function to perform the fine-tuning. We perform extensive experiments on the benchmark datasets to demonstrate the benefits of the proposed method that achieve the state-of-the-art performance on different datasets.

1 Introduction

Aspect Term Extraction (ATE) is one of the fundamental tasks in Aspect-based Sentiment Analysis (ABSA). Its goal is to recognize the terms upon which a sentiment opinion is expressed in text. For instance, in the sentence "*The staff of the restaurant were good but the quality of the food was terrible*", an ATE system should recognize the two aspect terms (targets) "*staff*" and "*quality of food*". ATE finds its applications in ABSA systems to identify targets toward which sentiment analysis is done.

A major challenge for ATE is the lack of enough training data. For instance, the widely-used SemEval datasets, e.g., Res15 (Pontiki et al., 2015) or Res16 (Pontiki et al., 2016), contain less than 2,000 training samples with only 20% of the words appearing more than five times (Chen and Qian, 2020a). This small size of training data hinders the deep sequence taggers to achieve optimal performance, especially for the tail targets (i.e., targets with few examples in the dataset) (He et al., 2018; Chen and Qian, 2019). In order to alleviate this issue, prior work has resorted to data augmentation techniques to exploit additional training signals from different sources, including data from related tasks, e.g., ABSA (performing multi-tasking learning (Luo et al., 2020; Chen and Qian, 2020b)), new labeled data for ATE produced by pre-trained sequence-to-sequence models (Li et al., 2020), and soft prompts that are generated by pre-trained language models (Chen and Qian, 2020a). As such, the critical requirements for such prior methods involve annotation for related tasks of ATE (e.g., ABSA), or large in-domain corpora to train the sequence-to-sequence/language models for data generation (called external data). Unfortunately, these requirements might be unavailable or very expensive to obtain in different domains, making it less applicable for various scenarios in practice.

To this end, this work aims to solve the issue of data scarcity for ATE without relying on annotated data for related tasks and large in-domain corpora. In particular, our main proposal is to fine-tune existing large-scale language models so they can generate complement sentences for input sentences in existing labeled datasets for ATE (i.e., not using external data as in prior works). Here, the motivation is that data scarcity might present a challenge for ATE models, especially on tail examples with rare aspect terms and context patterns (Chen and Qian, 2020a). The complement sentence thus aims to provide supporting evidence and facilitate the recognition of aspect terms for the input sentences.

As such, our method first seeks to obtain complement sentences for all the sentences in a given ATE dataset via a multi-step training procedure. In the first step, we train a base ATE model on a labeled training dataset to encode the available

Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing, pages 203 - 213

knowledge about aspect terms in the dataset. However, due to data scarcity, the base model might not be exposed sufficiently to aspect term patterns, thus limiting the ability of the produced representations for the input sentences to fully capture relevant information/features for ATE. To achieve complement sentences for ATE datasets, in the second step, we thus propose to learn optimal representation vectors/word embeddings that can be combined (e.g., via adding) to improve the representation vectors from the base model for ATE (called complement representations). Our motivation is that the insufficient coverage of aspect term patterns in the representations would cause the base model to exhibit poor performance (i.e., high loss) on the validation dataset. To this end, we propose to infer the complement representations for each validation sentence by incorporating them into the base model as additional parameters and minimizing the loss of the augmented model on the validation data. In the implementation, we divide the training data for an ATE dataset into k folds. By choosing one fold as validation data and treating the $k-1$ remaining folds as training data, the aforementioned procedure can return the optimal complement representations for each sentence in the validation fold. As such, we repeat this process for k possible choices of the validation fold that in all produce complement representations for each sentence in the training data.

To employ the complement representations for training data, we can introduce them into the base model for retraining. However, this will cause a mismatch in the test time where labels for sentences are not available and the complement representations cannot be obtained. To this end, instead of directly using the learned complement representations, we propose to first transform them into complement sentences based on the GloVe word embeddings (Pennington et al., 2014). This is done by introducing constraints to encourage the learned representation vectors to belong to the same space with GloVe word embeddings. The complement representations can then be mapped into complement sentences by finding the words whose GloVe embeddings are closest to the complement representations. In this way, each original training sentence for ATE can be associated with a complement sentence. Using pairs of original and complement sentences as training data, in the next step, we propose to train a generative model that can transform the original sentences into their complement versions. As such, in the test time, we can apply the generative model to generate complement sentences for test data and use GloVe embeddings to produce complement representation vectors for data augmentation.

In this work, we propose to fine-tune the language model GPT-2 (Radford et al., 2019) on the original and complement sentence pairs to obtain the generative model. Our motivation stems from the small number of the pairs for the original and complement sentences (due to the small size of ATE datasets) that might not be sufficient to train a generative model well. By leveraging the pre-trained GPT-2 model, we expect that its language priors can compensate for the data scarcity issue and bootstrap the learning process from complement data. Finally, we use REINFORCE (Williams, 1992) to fine-tune GPT-2 to facilitate the enforcement of expected properties for the generated sentences (i.e, the similarity or the length comparability with respect to the complement sentences produced in prior step). We perform extensive evaluations for the proposed method on different benchmark datasets for ATE. Our experiments reveal the superior performance and demonstrate the effectiveness of the proposed method.

2 Model

Problem Definition: ATE is formulated as a sequence labeling problem. Formally, given the input sentence $S = [w_1, w_2, \ldots, w_n]$, the goal is to predict the gold label sequence $Y = [y_1, y_2, \ldots, y_n]$ where $y_i \in \{B, I, O\}$, B stands for the "Beginning of a target", I stands for "Inside a target", and O stands for "Other". Our proposed model consists of a four-step procedure: (I) Training a base model for ATE using the available labeled data, (II) Finding the word representations of the optimal complement sentences for training data, (III) Fine-tuning a the language model GPT-2 to produce complement sentences for input sentences, and (IV) Training a final ATE model on the training data augmented with complement sentences.

2.1 Training a Base ATE Model (Step I)

For the first step, we train a base model on an available labeled ATE dataset. The trained model will serve as a base to find the optimal complement representations for input sentences of the ATE dataset in the next step. To this end, we employ a Bi-LSTM

base model. In particular, the input sentence S is first fed into the pre-trained BERT model (Devlin et al., 2019) to obtain the contextualized word embeddings $X = [x_1, x_2, \ldots, x_n]$ (x_i is the average of the representation vectors for the wordpieces of w_i in the last layer of BERT). As such, to further abstract the word embeddings X for ATE, we feed X into a Bidirectional LSTM (Bi-LSTM) network to obtain the hidden states $H = [h_1, h_2, \ldots, h_n]$. Afterward, the vectors in H are sent into a two-layer feed-forward layer FF to generate the label probability distribution $P(\cdot|S, w_i)$ for i-th word: $P(\cdot|S, w_i) = FF(h_i)$. Finally, to train the base model, we use negative log-likelihood loss: $\mathcal{L}_b = -\frac{1}{n}\sum_{i=1}^{n} \log P(y_i|S, w_i)$.

2.2 Finding Complement Representations (Step II)

As mentioned in the introduction, the limited size of the ATE datasets might prevent the base model from being imposed sufficiently to training samples to learn necessary aspect term patterns in the representations, potentially leading to inferior performance (i.e., high loss on validation data), especially on tail targets. As such, it is necessary to enhance the representation learning capability of the base model by imposing it to further information. To achieve this goal, prior work has resorted to data augmentation (Li et al., 2020) or soft-prompts (Chen and Qian, 2020a) in which the training data is augmented with new sentences (e.g., generated by a pre-trained language model) to provide more evidences for aspect terms. However, the limitation of the prior work is that the generated sentences to augment ATE data is either ignorant of the ATE task (Chen and Qian, 2020a) or constrained on some heuristics (i.e., replacing non-aspect terms with other words generated by a language model) (Li et al., 2020). As such, we argue that these data augmentation methods might not achieve the optimal augmentation for the available ATE data. We thus posit that the optimal augmentation for an input sentence is the one whose combination with the sentence could directly reduce the objective loss on validation data. Concretely, to find the optimal augmentation for a sentence S in the validation data, we search for the sentence S' whose combination with S (i.e., by adding their word representations) could further reduce the objective loss $\mathcal{L}_b$ computed on validation data. Since this augmentation is optimized over validation data and not bound to any heuristics-based constraints, we expect it to be the optimal augmentation for the input sentence. Note that the optimality of the sentence S' is with respect to the objective loss $\mathcal{L}_b$ and changing the criteria could lead to a different sentence S'.

To find the optimal complement sentence S' for S in the validation data, since this is a discrete variable, we first attempt to find the representation vectors X' for its words w_i. That is, S' is parameterized by a set of learnable vectors X' which are combined with the word embeddings X and are updated with the objective loss $\mathcal{L}_b$ over validation data. In this work, the combination of X and X' is defended as the sum of their corresponding vectors x_i and x'_i. As such, the number of tokens of X' should be equal to the number of tokens of X, i.e., $X' = [x'_1, x'_2, \ldots, x'_n]$. In addition, the dimension of the vectors should also match, i.e., $|x_i| = |x'_i| = D$, where D is the dimensionality of the word embedding vectors. Hence, the total number of parameters defined for all representation vectors is $N \times n \times D$, where N is the total number of sentences in the validation set.

In the next step, we seek to optimize the representation parameters complement sentences by reducing the objective loss $\mathcal{L}_b$ over validation data. In particular, for the sentence S with embeddings X and the complement sentence S' with parameters (i.e., embeddings) X', we compute the sum of the corresponding vectors for the i-th token: $\hat{x}_i = x_i + \lambda x'_i$ where λ is a trade-off parameter (i.e., the data augmentation in this work). The vectors $\hat{X} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n]$ will be sent to the base model architecture (i.e., BiLSTM followed by a feed-forward layer) to obtain the label distribution $P(\cdot|S, S', w_i)$. As such, the objective loss for this training step, i.e., $\mathcal{L}_f$, is defined similar to $\mathcal{L}_b$: $\mathcal{L}_f = -\frac{1}{n}\sum_{i=1}^{n} \log P(y_i|S, S', w_i)$ (computed over validation data). Note that in this training step, the original parameters of the trained base model is fixed so the only parameters to be updated are the parameters for the complement sentences S', i.e., the vectors X'.

Embedding Regularization: To further improve the complement embeddings and facilitate the mapping to complement sentences S' later, we introduce two additional regularization terms for the learning objective of complement embeddings. The first regularization seeks to encourage the complement embeddings X' to capture different (i.e., complementary) information from those

for the embeddings X of the input sentence S, thus enhancing the contribution of complement embeddings. In particular, we compute the representation vectors R_S and $R_{S'}$ for the original and complement sentences using the max-pooling operation: $R_S = MAX\_POOL(x_1, x_2, \ldots, x_n)$ and $R_{S'} = MAX\_POOL(x'_1, x'_2, \ldots, x'_n)$. Afterward, the complementary nature of embeddings is enforced by introducing the dot product $\mathcal{L}_{reg}$ between R_S and $R_{S'}$ into the loss function for minimization (i.e., minimizing the similarity between R_S and $R_{S'}$): $\mathcal{L}_{reg} = R_S \odot R_{S'}$.

For the second regularization, we aim to align the complement embeddings X' to the space of the GloVe embeddings (Pennington et al., 2014) to facilitate the transformation to complement sentences in the next step. Here, we use the GloVe embeddings for convenience and leave other possible pre-trained embeddings for future work. In particular, for each vector $x'_i \in X'$, we use a feed-forward network F to transform x'_i into the vector $F(x'_i)$ of the same dimension with Glove embeddings. Afterward, we find the vector e_i in the GloVe embedding table that is closest to $F(x'_i)$ based on the Euclidean distance. The Euclidean distance between $F(x'_i)$ and e_i is then incorporated into the loss function to promote the alignment of complement and GloVe embeddings: $\mathcal{L}_{GloVe} = \frac{1}{n}\sum_{i=1}^{n} \|F(x'_i) - e_i\|_2^2$. Finally, the overall loss function to learn the complement representations X' is: $\mathcal{L}_{emb} = \mathcal{L}_f + \alpha_{reg}\mathcal{L}_{reg} + \alpha_{GloVe}\mathcal{L}_{GloVe}$ where α_{reg} and α_{GloVe} are the trade-off parameters. Note that the parameters for F are also optimized in this process.

As such, this training step produces the complement embedding X' for each sentence in the validation data. To maximize the use of data, we implement this training step in a 10-fold validation fashion described in the introduction. In particular, we train the base model on 9 folds of the training data (i.e., Section 2.1) and use the remaining fold for the validation data in the complement representation optimization. By alternating the choice of validation fold, we can obtain a complement representation sequence X' for each sentence in the original training data.

2.3 Generating Complement Sentences (Step III)

As mentioned in the introduction, the complement embeddings X' can be used directly to augment training data and train a model for ATE. However, as the optimization for complement embeddings cannot be done in the test time (due to the unavailability of labels for data), the direct augmentation will cause a mismatch between the training and test phases. To enable the generation of complement embeddings in the test time, we thus propose to first transform the complement embeddings X' into a complement sentence $S' = [w'_1, w'_2, \ldots, w'_n]$ where w'_i is the word whose Glove embedding is closest to the transformed complement vector $F(x'_i)$ for w_i. The set of every pair (S, S') for sentences S in training data is then employed to train a generative language model that seeks to consume S and produce its complement sentence S'. In this way, we can apply the generative model in the test time to generate complement sentences for test data, that, in turn, can be transformed into complement embeddings by mapping words into Glove embedding vectors for data augmentation.

One potential issue is that the number of original and complement sentence pairs (S, S') might be small due to the limited size of ATE datasets, thus hindering the training of effective generative models for our complement sentence goal. As such, we propose to leverage the language priors in the pre-trained generative model GPT-2 (Radford et al., 2019) as the bootstrap knowledge for the complement generation. In particular, we propose to fine-tune the GPT-2 model on the sentence pairs (S, S') in this step. The policy-gradient method REINFORCE (Williams, 1992) is utilized for the fine-tuning process to facilitate the incorporation of different expected properties for complement sentences. Concretely, the input to GPT-2 consists of the S sentence "$w_1 w_2 \ldots w_n SEP$" from which GPT-2 will generate the sentence S''. To compute the reward for the generated sentence S'', we propose three objectives: (1) **Similarity with Complement Sentence**: The generated sentence S'' should be similar to the actual complement sentence S'. To compute the similarity of the two sentences, we employ the CIDEr score (Vedantam et al., 2015) for S'': $R_{sim} = CIDEr(S'')$; (2) **Length Penalty**: As discussed earlier, since we use sum of the corresponding word embeddings of the original and complement sentences for data augmentation, it is intuitive to encourage the generated sentences S'' to have the same length as the original sentence S. Thus, we introduce the length penalty as a part of the reward: $R_{len} = ||S| - |S''||$;

(3) **Difference with Original Sentence**: Similar to embedding regularization $\mathcal{L}_{reg}$ presented earlier for complement embeddings, here we also aim to promote the semantic difference between the generated sentence S'' and the original sentence S (for complementary information). To this end, we represent each sentence using the max-pooled representation of their word embeddings obtained from the GloVe embedding table, i.e., $\hat{R}_S$ and $\hat{R}_{S''}$. Next, their dot-product is employed for the difference reward $R_{diff} = \hat{R}_S \odot \hat{R}_{S''}$. Consequently, the overall reward to train the generative model is computes as $R(S'') = R_{sim} - \beta R_{len} - \gamma R_{diff}$. With REINFORCE, we seek to minimize the negative expected reward $R(S'')$ over the possible choices of S'': $\mathcal{L}_{tune} = -\mathbb{E}_{\hat{S}'' \sim P(\hat{S}''|S)}[R(\hat{S}'')]$. The policy gradient is then estimated by: $\nabla \mathcal{L}_{tune} = -\mathbb{E}_{\hat{S}'' \sim P(\hat{S}''|S)}[(R(\hat{S}'') - b) \nabla \log P(\hat{S}''|S)]$. Using one roll-out sample, we further estimate $\nabla \mathcal{L}_{tune}$ via the generated sentence S'': $\nabla \mathcal{L}_{tune} = -(R(S'') - b) \nabla \log P(\hat{S}''|S)$ where b is the baseline to reduce variance. In this work, we obtain the baseline b via: $b = \frac{1}{|B|} \sum_{i=1}^{|B|} R(S''_i)$, where $|B|$ is the mini-batch size and S''_i is the generated sentence for the i-th sample in the mini-batch.

2.4 Training a Final ATE Model (Step IV)

To achieve a consistency in the training and testing phase, we use the generated sentences from the fine-tuned GPT-2 model as the complement sentences for data augmentation in both phases. In particular, for the training data, similar to the complement embedding optimization Section 2.2, the fine-tuning of GPT-2 is performed with 10-fold cross validation. In particular, the GPT-2 model is fine-tuned on the (S, S') pairs of 9 folds and then employed to generate S'' for each sentence in the remaining fold. To this end, each sentence S in the training data is associated with a generated sentence S''. For test data, we simply apply the fine-tuned GPT-2 model directly to generate a complement sentence for each sentence in that data.

As such, for each sentence S (in the training or test data), its complement sentence from GPT-2 $S'' = [w''_1, w''_2, \ldots, w''_n]$ is first transformed into a representation vector sequence $X'' = [x''_1, x''_2, \ldots, x''_n]$ based on the mappings for their words w''_i from GloVe embeddings[1]. Next, the augmented representation $\bar{X} = [\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n]$ from the two sentences is computed by the sum of their corresponding word representations: $\bar{x}_i = x_i + \lambda G(x''_i)$, where G is a feed-forward network to match the dimensions of the GloVe embedding $G(x''_i)$ and x_i. Finally, following the base ATE architecture, the resulting vectors $\bar{X}$ are sent to a BiLSTM network followed by a feed-forward layer to obtain the label distribution $P(\cdot|S, S'', w_i)$ for the i-th word. This distribution is used for prediction in the test phase while the training phase employs the negative log-likelihood (over training data) to train the final ATE model: $\mathcal{L}_{final} = -\frac{1}{n} \sum_{i=1}^{n} \log P(y_i|S, S'', w_i)$.

[1]Note that the generated sentence S'' might have a different length from the original sentence S. For these cases, we pad (with zero vectors) or truncate the vector sequence X'' to have the same length as S.

Datasets	Lap14		Res14		Res15		Res16	
Type	Train	Test	Train	Test	Train	Test	Train	Test
Sentences	3045	800	3041	800	1315	685	2000	676
Aspects	2342	650	3686	1134	1209	547	1757	622

Table 1: Statistics of the SemEval datasets

3 Experiments

Datasets & Parameters: To evaluate the effectiveness of the proposed method, we employ the commonly used SemEval datasets for ATE. Specifically, we use the datasets of SemEval 2014 Task 4 (Pontiki et al., 2014), i.e., Lap14 and Res14 reviews for laptops and restaurants, and SemEval 2015 Task 12 (Pontiki et al., 2015) and SemEval 2016 Task 5 (Pontiki et al., 2016), i.e., Res15 and Res16 with reviews for restaurants. Following prior work (Chen and Qian, 2020a), we employ the official train/test splits and randomly select 150 samples from training data as validation data for those datasets. Table 1 shows the statistics for the datasets.

In our experiments, we use the development set of the Lap14 dataset to tune the hyper-parameters. Based on our experiments, the following values are selected: 200 dimensions for the BiLSTM layer and the feed-forward layers; 1 layer for BiLSTM and 2 layers for the feed-forward networks; the BERT$_{base}$ version for the encoding layer with fixed parameters; GPT-2 small for sentence generation; 0.3 for the λ in the word vector augmentation; 0.1 for the trade-off parameters α_{reg} and α_{GloVe} in the complement embedding optimization; 0.1 and 0.05 for the trade-off parameters β and γ in the reward for GPT-2 fine-tuning; 0.3 for the learning rate for the Adam optimizer; and 50 for batch-size.

Baselines: Following prior work, we compare our model with: (1) the winners of the SemEval tasks:

IHS-RD (Chernyshevich, 2014), **DLIREC** (Toh and Wang, 2014), **EliXa** (San Vicente et al., 2015), **NLANGP** (Toh and Su, 2016); (2) Deep joint models, i.e., jointly train ATE with Opinion Term Extraction (OTE) or Aspect-Based Sentiment Analysis (ABSA): **RNCRF** (Wang et al., 2016), **MIN** (Li and Lam, 2017), **CMLA** (Mao et al., 2021), **HAST** (Li et al., 2018), **RACL** (Chen and Qian, 2020b), **Dual-MRC** (Mao et al., 2021); (3) Deep models trained on ATE datasets augmented with external in-domain corpora and resources: **BiLSTM-CRF** (Li et al., 2020), **Seq2Seq** (Ma et al., 2019), **BERT** (Li et al., 2020), **DE-CNN** (Xu et al., 2018), **BERT-PT** (Li et al., 2020), **SoftProto** (Chen and Qian, 2020a). We also compare with **CL-BERT** (Yang et al., 2020) which employs constituency trees for ATE. For the evaluation metric, following prior work, we report the F1 score for aspect term prediction. A prediction is counted as correct if its boundaries match the gold aspect term. We name our model ATEOA which stands for Aspect Term Extraction with Optimal Augmentation.

Results: The main results are shown in Table 2. This table shows that the proposed model can effectively improve the performance compared to existing joint inference and data augmentation methods. This achievement is significant as the proposed model does not utilize any external in-domain data nor extra supervision from other related tasks. This is important for domains with limited data where collecting large-scale in-domain data or supervision from related tasks could be prohibitively expensive. Moreover, as the proposed model employs pre-trained language models (i.e., GPT-2) to generate effective augmentation sentences for training/test data, it can directly benefit from growing advances in pre-trained language models.

Ablation Study: The proposed ATEOA model is trained in four major training steps. In this section, we study the role of those proposed steps for the performance of the ATE model. For each training step, we aim to answer two questions: (i) Whether the proposed step is beneficial for ATEOA? and (ii) Is the current configuration for the step optimal? To this end, we consider the following ablated models: (1) **- Base Model Training (Step I)**: This model ignores step I to train a base ATE model in Section 2.1. In particular, step II for finding complement embeddings will only employ an ATE base model with randomly initialized parameters. Here, the parameters for the base model are not fixed; they are jointly updated with the complement embeddings in step II of the training; (2) **- Complement Representation Finding (Step II)**: This model excludes step II of the training procedure that makes the optimized complement representations unavailable for the fine-tuning of GPT-2 in step III. As such, to achieve a fair access to the trained base model in step I in this model, we change step III by fine-tuning the GPT-2 model with the reward of F1 score of the trained base model (from step I) on the validation data. Here, the base model is applied on the representation combinations of the original and GPT-generated sentences (also using GloVe embeddings for the words in the generated sentences); (3) **- Embedding Regularization** $\mathcal{L}_{reg}$: This model removes the regularization loss $\mathcal{L}_{reg}$ in step II of the training process; (4) **- GloVe Alignment** $\mathcal{L}_{GloVe}$: This model excludes the regularization $\mathcal{L}_{GloVe}$ for representation alignment with GloVe in step II; (5) **- Language Model (Step III)**: This ablated models does not utilize step III, thus eliminating the GPT-2 model trained over the original and complement sentence pairs (S, S'). As such, in step IV, we directly retrain the base model on the augmented data with the complement representations X' (i.e., $x_i + \lambda x'_i$) and do not apply data augmentation for test data (i.e., applying the train model on x_i directly); (6) **- Language Model + FForward**: This model is similar to (5) (i.e., excluding GPT-2 in step III). However, to allow the

IHS-RD	74.55	79.62	-	-
DLIREC	73.78	84.01	-	-
EliXa	-	-	70.04	-
NLANGP	-	-	67.12	72.34
RNCRF	78.42	84.93	67.74	69.72
MIN	77.58	-	-	73.44
CMLA	77.80	85.29	70.73	72.77
HAST	79.52	85.61	71.46	73.61
RACL-BERT	81.99	85.37	72.82	-
Dual-MRC	82.51	86.60	75.08	-
BiLSTM-CRF	74.28	-	-	71.44
Seq2Seq	78.68	-	-	74.01
BERT	81.14	-	-	75.89
DE-CNN	81.58	-	-	75.19
BERT-PT	85.33	-	-	80.29
SoftProto	83.19	87.39	73.27	76.98
CL-BERT	85.61	-	-	81.14
ATEOA (ours)	**86.71**	**88.99**	**75.41**	**82.58**

Table 2: F1 scores on the test sets of the SemEval datasets. The proposed model ATEOE is significantly better than prior work ($p < 0.05$).

augmentation on test data, a feed-forward network is trained on pairs (x_i, x'_i) to directly transform the representation vectors x_i of the original sentence S into the complement representations for data augmentation in both training and test phases of step IV; (7) **- Similarity Reward**: For this model, we do not use the similarity reward R_{sim} in the reward function to fine-tune GPT-2 in step III; (8) **- Length Penalty**: This model does not employ the length penalty R_{len} in the reward for tuning GPT-2; (9) **- Difference Reward**: For this baseline, the reward based on difference with original sentence, i.e., R_{diff}, is ablated from the reward for GPT-2 fine-tuning; (10) **- Final Training (Step IV)**: This baseline skips the last step of the proposed training procedure. As such, the combined representations of the original sentence S and the complement sentence S'' generated by GPT-2 (i.e., $x_i + \lambda G(x''_i)$) are directly sent into the base ATE model (trained over the entire training data) from step I for prediction; and (11) **- Generated Data in Final Training**: Finally, to demonstrate the benefit of augmenting training data with generated sentences from the fine-tuned GPT-2 model in step IV, we report the performance of the base model that is instead trained on the combination of the word representations X and the optimized complement representations X', i.e., $x_i + \lambda x'_i$ in step II (as in (5)). The fine-tuned GPT-2 model is still used to generate complement sentences for data augmentation in the test phase for this model.

The performance of the models on the test sets of the SemEval datasets is reported in Table 3. This table clearly shows that all training steps in the proposed procedure are necessary as skipping any of these steps will hurt the performance. In particular, among the four steps, removing step III has the most negative impact as the ablated model "- *Language Model*" has the lowest performance across datasets. We attribute the importance of step III to its ability to enable augmentation consistency for training and test data (i.e., the fine-tuned GPT-2 can generate complement sentences for both training and test data). This is further highlighted by the worse performance of the "**- Generated Data in Final Training**" model where the training data is augmented with X', but GPT-generated data is used to augment test data. Table 3 also shows that among the three awards for GPT-2 fine-tuning, the similarity reward is most important. This is expected as the primary goal of fine-tuning is to generate sentences that are similar to the complement sentences S'.

Model	Lap14	Res14	Res15	Res16
ATEOA (Full)	**86.71**	**88.99**	**75.41**	**82.58**
- Base Model Training (Step I)	84.39	86.91	74.18	78.91
- Comp. Rep. Finding (Step II)	84.96	86.18	74.22	79.65
- Embedding Regularization $\mathcal{L}_{reg}$	85.04	87.93	74.31	80.32
- Glove Alignment $\mathcal{L}_{GloVe}$	86.02	88.12	75.31	81.95
- Language Model (Step III)	84.22	85.91	73.17	78.91
- Language Model + FForward	84.13	86.94	73.22	80.51
- Similarity Reward	83.33	85.98	73.54	79.05
- Length Penalty	85.10	87.99	73.91	81.18
- Difference Reward	85.11	88.02	73.88	80.04
- Final Training (Step IV)	84.40	87.12	74.09	80.01
- Generated Data in Final Train.	84.01	86.92	73.81	79.88

Table 3: Performance of the ablated models on test sets.

4 Analysis

Generative Language Models: As it is evident in the ablation study, exploiting a pre-trained generative model (i.e., GPT-2) for ATEOA is preferable since it can provide language priors to support the sentence generation learning from limited ATE datasets. In this section, we study how the performance of the model changes if we alter the generative language model used in step III of ATEOA. Concretely, we compare the performance of three different models: (1) **GPT-2** (Radford et al., 2019): This transformer-based model is pre-trained on WebText corpus. We examine its small version with 117 million parameters; (2) **T5** (Raffel et al., 2019): This language model employs the encoder-decoder architecture in Transformer for sequence-to-sequence tasks. We explore its base version with 220 million parameters. We use the input sentence S as the source sequence and the complement sentence S' as the target sequence to fine-tune the T5 model; and (3) **BART** (Lewis et al., 2019): This model is a transformer-based auto-encoder language model. We also utilize its base version with 139 million parameters. Similar to T5, this is a sequence-to-sequence generative model that is fine-tuned by treating S and S' as the source and target sequences respectively.

To compare the performance of the three language models, we use them in the training step III of the final ATE model and report the corresponding performance. Furthermore, we compare the language models on their capability to generate sentences that are similar to the complement sentences S'. In particular, using the Lap14 dataset, we seek to find a complement sentence S' for each sentence in the test data portion with the proposed method.

Language Model	Lap14	Res14	Res15	Res16
GPT-2	**86.71**	**88.99**	**75.41**	**82.58**
BART	84.32	88.05	74.91	79.49
T5	84.18	86.95	73.16	79.15

Table 4: Performance of the final ATE model on test sets with different language models in step III.

Language Model	BLUE-4	METEOR	ROUGE-1	ROUGE-2
GPT-2	12.05	12.25	31.89	10.33
BART	9.39	10.14	29.05	9.06
T5	13.10	11.92	30.33	8.95

Table 5: Similarity between the generated sentences from the language models and the "ground-truth" complement sentences on test data of Lap14.

To this end, a base model is first trained on the training data portion using step I; the complement representations X' are then computed for each sentence in the test data portion using step II; each X' is then mapped into the complement sentence S' with the GloVe embeddings. Here, S' serves as the "ground-truth" complement sentence for the test sentences in our approach. Next, we use the fine-tuned language model to generate the complement sentence S'' for each test sentence (i.e., prompting the language model with test data). Finally, we evaluate the similarity of the generated sentence S'' and the "ground-truth" complement sentence S' (for the test data) using ROUGE-1, ROUGE-2, METEOR (Banerjee and Lavie, 2005) and BLUE4 as the similarity metrics. The results for this experiment are shown in Tables 4 and 5. Both tables clearly demonstrate the capacity of GPT-2 to generate better complement sentences to augment ATE data (i.e., yielding better performance for ATE in Table 4 and generating more similar sentences to the obtained complement sentences S' in Table 5).

Tail Aspect Term Analysis: Following prior work (Chen and Qian, 2020a), we evaluate the performance for our model on the tail aspect terms in test data (i.e., aspect terms occurring less than 5 times in the training sets). As such, we compare our model with prior work that reports their performance in this analysis, i.e., **DE-CNN** (Xu et al., 2018) and **SoftProto** (Chen and Qian, 2020a). Note that we replace the contextualized BERT representations (i.e., X) in our model with the GloVe embeddings to achieve a fair comparison with prior work in this section. The results are provided in Table 6 that clearly shows the superiority of ATEOA to recognize tail aspect terms and further highlights the benefits of the proposed method.

Model	Lap14	Res14	Res15	Res16
DE-CNN	74.37	77.61	70.00	70.68
SoftProto	79.85	82.22	76.80	70.93
ATEOA (ours)	**81.92**	**83.69**	**77.39**	**73.49**

Table 6: Performance for tail aspect terms on test data.

Case Study: To provide more insight into the quality of the complement sentences generated by the pre-trained GPT-2 model, Table 7 showcases some examples from the laptop and restaurant domains whose aspect terms can only be correctly predicted by our proposed method (i.e., prior work fails to recognize aspect terms in these cases). The table suggests that although the generated sentences might not look natural, they clearly provide more evidence and emphasis on the aspect terms which makes the task easier for the ATE model. Specifically, in the first example, the generated complement sentence emphasizes the target word "*touchpad*" in the original sentence by replicating it and including the related word "*mouse*". The same pattern of emphasis can be seen in the second example where the model excludes the word "*money*" and includes the related word "*Food*" (that are more related to the target word "*meal*") in the generated sentence.

5 Related Work

ATE has been first approached with rule-based (Hu and Liu, 2004; Wu et al., 2009) or feature-based (Li et al., 2010; Chen et al., 2014; Toh and Su, 2016) methods. Recently, ATE methods have focused on neural networks such as LSTM (Liu et al., 2015), CNN (Xu et al., 2018) or Transformer (Li et al., 2020). An ATE system can be used in downstream applications such as sentiment analysis (Wang et al., 2019; Pouran Ben Veyseh et al., 2020b; Orbach et al., 2021) or opinion term extraction (Pouran Ben Veyseh et al., 2020a). One of the challenges for this task is the scarcity of training data which hinders the training of large neural networks. To alleviate this issue, two major directions have been explored in the literature: (I) Joint Training, i.e., jointly solving ATE task with another related task such as ABSA (Wang et al., 2016; Mao et al., 2021; Chen and Qian, 2020b) or Opinion Term Extraction (OTE) (Li and Lam, 2017; Dai and Song, 2019); and (II) Data Augmentation, i.e., augmenting ATE models with in-domain unlabeled data (Xu et al., 2018; Li et al., 2020; Chen and Qian,

Input Sentences	Generated Complement Sentences	Targets
however , there are major issues with the touchpad which render the device nearly useless .	Although , exist some problems with the touchpad and mouse makes touchpad useless and touchpad useless	touchpad
way too much money for such a terrible meal .	Food costs so much for such a bad meal .	meal

Table 7: Generated complement sentences by GPT-2.

2020a). In this work, we also propose a method to augment the training data for ATE. However, unlike prior work that requires large in-domain corpus, our approach employs an existing large-scale language model (i.e., GPT-2) to facilitate the generation of complement sentences for the available ATE datasets. Using GPT-2 to address data scarcity has been shown to be effective in other domains and tasks (Papanikolaou and Pierleoni, 2020; Pouran Ben Veyseh et al., 2021; Peng et al., 2020). In this work, we demonstrate the viability of this technique for aspect term extraction.

6 Conclusion

We introduce a new training procedure for Aspect Term Extraction. In the proposed procedure, the available ATE dataset is employed to train a deep model which is further used to find complement representations for input sentences in training data. Later, to obtain the complement sentences at the inference time, we tine-tune the pre-trained language model GPT-2 to generate sentences similar to the complement sentences found in the previous steps (with GloVe mapping). Our extensive experiments on benchmark datasets reveal the superiority of the proposed model, leading to the state-of-the-art performance for the datasets. Moreover, our analysis show that all steps of the proposed procedure are necessary and effective for the ATE task. In future, we will explore the application of this procedure in other related task such as OTE and ABSA.

Acknowledgement

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112 and the NSF grant CNS-1747798 to the IU-CRC Center for Big Learning. This research is also based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect extraction with automated prior knowledge learning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 347–358, Baltimore, Maryland. Association for Computational Linguistics.

Zhuang Chen and Tieyun Qian. 2019. Transfer capsule network for aspect level sentiment classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 547–556, Florence, Italy. Association for Computational Linguistics.

Zhuang Chen and Tieyun Qian. 2020a. Enhancing aspect term extraction with soft prototypes. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2107–2117, Online. Association for Computational Linguistics.

Zhuang Chen and Tieyun Qian. 2020b. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3685–3694, Online. Association for Computational Linguistics.

Maryna Chernyshevich. 2014. IHS R&D Belarus: Cross-domain extraction of product features using CRF. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages

309–313, Dublin, Ireland. Association for Computational Linguistics.

Hongliang Dai and Yangqiu Song. 2019. Neural aspect and opinion term extraction with mined rules as weak supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5268–5277, Florence, Italy. Association for Computational Linguistics.

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

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Exploiting document knowledge for aspect-level sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 579–585, Melbourne, Australia. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. 2010. Structure-aware review mining and summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 653–661, Beijing, China. Coling 2010 Organizing Committee.

Kun Li, Chengbo Chen, Xiaojun Quan, Qing Ling, and Yan Song. 2020. Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7056–7066, Online. Association for Computational Linguistics.

Xin Li, Lidong Bing, Piji Li, Wai Lam, and Zhimou Yang. 2018. Aspect term extraction with history attention and selective transformation. In *IJCAI*.

Xin Li and Wai Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark. Association for Computational Linguistics.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1433–1443.

Huaishao Luo, Lei Ji, Tianrui Li, Daxin Jiang, and Nan Duan. 2020. GRACE: Gradient harmonized and cascaded labeling for aspect-based sentiment analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 54–64, Online. Association for Computational Linguistics.

Dehong Ma, Sujian Li, Fangzhao Wu, Xing Xie, and Houfeng Wang. 2019. Exploring sequence-to-sequence learning in aspect term extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3538–3547, Florence, Italy. Association for Computational Linguistics.

Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis. In *AAAI*.

Matan Orbach, Orith Toledo-Ronen, Artem Spector, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2021. YASO: A targeted sentiment analysis evaluation dataset for open-domain reviews. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9154–9173, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yannis Papanikolaou and Andrea Pierleoni. 2020. Dare: Data augmented relation extraction with gpt-2. In *SciNLP workshop at the Conference on Automated Knowledge Base Construction (AKBC)*.

Baolin Peng, Chenguang Zhu, Michael Zeng, and Jianfeng Gao. 2020. Data augmentation for spoken language understanding via pretrained language models. *arXiv preprint arXiv:2004.13952*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris 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, Denver, Colorado. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021. Unleash GPT-2 power for event detection. 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)*, pages 6271–6282, Online. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Nasim Nouri, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020a. Introducing syntactic structures into target opinion word extraction with deep learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8947–8956, Online. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Nasim Nouri, Franck Dernoncourt, Quan Hung Tran, Dejing Dou, and Thien Huu Nguyen. 2020b. Improving aspect-based sentiment analysis with gated graph convolutional networks and syntax-based regulation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4543–4548, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Iñaki San Vicente, Xabier Saralegi, and Rodrigo Agerri. 2015. EliXa: A modular and flexible ABSA platform. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 748–752, Denver, Colorado. Association for Computational Linguistics.

Zhiqiang Toh and Jian Su. 2016. NLANGP at SemEval-2016 task 5: Improving aspect based sentiment analysis using neural network features. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 282–288, San Diego, California. Association for Computational Linguistics.

Zhiqiang Toh and Wenting Wang. 2014. DLIREC: Aspect term extraction and term polarity classification system. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 235–240, Dublin, Ireland. Association for Computational Linguistics.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Jingjing Wang, Changlong Sun, Shoushan Li, Jiancheng Wang, Luo Si, Min Zhang, Xiaozhong Liu, and Guodong Zhou. 2019. Human-like decision making: Document-level aspect sentiment classification via hierarchical reinforcement learning. 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 5581–5590, Hong Kong, China. Association for Computational Linguistics.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 616–626, Austin, Texas. Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Kluwer Academic*.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1541, Singapore. Association for Computational Linguistics.

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)*, pages 592–598, Melbourne, Australia. Association for Computational Linguistics.

Yunyi Yang, Kun Li, Xiaojun Quan, Weizhou Shen, and Qinliang Su. 2020. Constituency lattice encoding for aspect term extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 844–855, Barcelona, Spain (Online). International Committee on Computational Linguistics.