TEASER: Towards Efficient Aspect-based SEntiment analysis and Recognition

Vaibhav Bajaj, Kartikey Pant, Ishan S Upadhyay, Srinath Nair and **Radhika Mamidi**
IIIT Hyderabad
Hyderabad, India
{vaibhav.bajaj,kartikey.pant}@research.iiit.ac.in
{ishan.sanjeev,srinath.nair}@research.iiit.ac.in
radhika.mamidi@iiit.ac.in

Abstract

Sentiment analysis aims to detect the overall sentiment, i.e., the polarity of a sentence, paragraph, or text span, without considering the entities mentioned and their aspects. Aspect-based sentiment analysis aims to extract the aspects of the given target entities and their respective sentiments. Prior works formulate this as a sequence tagging problem or solve this task using a span-based extract-then-classify framework where first all the opinion targets are extracted from the sentence, and then with the help of span representations, the targets are classified as positive, negative, or neutral. The sequence tagging problem suffers from issues like sentiment inconsistency and colossal search space. Whereas, Span-based extract-then-classify framework suffers from issues such as half-word coverage and overlapping spans. To overcome this, we propose a similar span-based extract-then-classify framework with a novel and improved heuristic. Experiments on the three benchmark datasets (Restaurant14, Laptop14, Restaurant15) show our model consistently outperforms the current state-of-the-art. Moreover, we also present a novel supervised movie reviews dataset (Movie20) and a pseudo-labeled movie reviews dataset (moviesLarge) made explicitly for this task in English language and report the results on the novel Movie20 dataset as well.

1 Introduction

Online reviews and tweets play an essential role in consumer decision making. Hence, it becomes crucial to efficiently and effectively extract user opinions from an unstructured text (user review, tweet). Aspect-Based Sentiment Analysis is a fundamental task in mining opinions and sentiment analysis (Pang and Lee, 2008; Liu, 2012). This task requires detecting the aspects of the target entities mentioned (Aspect Extraction) and detecting the sentiment attached, i.e., the polarity of the target entity (Sentiment Classification). Hence, it is more challenging than the traditional sentence-level sentiment analysis (Lin and He, 2009; Kim, 2014), where we predict the text's polarity as a whole. As shown in Table 1, given a sentence, the task is to extract the aspects "acting" and "editing" and predict the corresponding polarity, which is positive, negative, respectively.

Sentence	Great acting dreadful editing
Targets	[acting], [editing]
Polarities	Positive, Negative

Table 1: Example Sentence

Most of the previous works treat this Aspect-Based Sentiment Analysis (ABSA) task as a combination of two different subtasks, namely, Aspect Extraction (AE) and Sentiment Classification (SC). Researchers often treat these two subtasks as independent and work on both of them individually, or some even tried to combine the two and propose a joint model that can extract the targets and predict the polarity. Much work has been done in the field of AE. Jakob and Gurevych (2010); Liu et al. (2015); Wang et al. (2016a); Poria et al. (2016); Shu et al. (2017); He et al. (2017); Xu et al. (2018) formulate AE as a sequence tagging problem. In sequence tagging, the task is to mark each word with a set of tags (e.g. B, I, O). The second subtask, SC, i.e., marking each extracted target term as Positive, Negative, or Neutral, has also been widely studied (Tang et al., 2016; Wang et al., 2016b; Chen et al., 2017; Xue and Li, 2018; Fan et al., 2018). The main issue with most of these sentiment classifiers is that, they assume the target is already given.

Zhang et al. (2015) and Li et al. (2019) tried to combine the two subtasks and solve the task in a more integrated way by jointly extracting targets and predicting their sentiments. The main idea

Proceedings of Recent Advances in Natural Language Processing, pages 102–110
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_013

here is either jointly marking the words with a set of tags for the task of AE (e.g., B, I, O) and also marking them as Positive, Negative, Neutral for SC or use a more collapsed version of marking (e.g., B-Sentiment, I-Sentiment, O).

There are many disadvantages to the above BIO annotation scheme. As shown by Lee et al. (2016), using BIO tags for extractive question answering (in our case extracting opinion terms) have issues like colossal search space since the model must consider the power set of all words in a sentence. This results in it being less effective, and in the case of polarity classification, sequence tagging is not optimal because tagging the polarity over each word fails to capture the semantics of the entire opinion-target. Moreover, there could be sentiment inconsistencies in a multi-word target-term as predicted polarities over different words in a target could be different.

In this paper, we make the following contributions:

1. We propose TEASER, a span-based labeling scheme methodology exploiting the extract-then-classify framework for aspect-based sentiment analysis that reduces the search space while dealing with half-word coverage issues and overlapping spans.

2. We conduct extensive experiments that demonstrate our model to consistently outperform the current state-of-the-art in Aspect Extraction and overall ABSA task on the three benchmark datasets (*Restaurant14*, *Restaurant15*, *Laptop14*).

3. We also present two novel datasets[1] in the domain of movie-reviews. The Movie20 dataset is a Supervised dataset of 1162 sentences made explicitly for this task, and moviesLarge is a Pseudo-labeled dataset of 14373 sentences.

2 Related Work

Hu and Liu (2004) proposed Aspect-based Sentiment Analysis for the first time. Since then, it has been widely studied, especially in recent years (Zhang et al., 2018).

Most existing works treat this as a combination of Aspect Extraction and Sentiment Classification. The task of AE has been widely studied with various methods tried and tested for this task. Jakob and Gurevych (2010); Wang et al. (2016a); Shu et al. (2017) made use of Conditional Random Fields (CRFs) for prediction. Poria et al. (2016) tried a deep learning approach for this task of AE for the first time. They proposed a CNN based model to tag words in sentences as an aspect or non-aspect. Xu et al. (2018) Used a double embedding Mechanism with Convolutional Neural Networks (CNNs) to solve this task. Liu et al. (2015) tried to tackle this with Recurrent Neural Networks (RNNs). Similar to AE, the subtask of SC has been widely studied (Jiang et al., 2011; Vo and Zhang, 2015; Wang et al., 2018; Zhu and Qian, 2018; Chen and Qian, 2019; Zhu et al., 2019). Dong et al. (2014) proposed an RNN based approach, Chen and Qian (2019) attempted to solve this using Transfer Capsule Network, and Li and Lam (2017) used Memory networks.

Few works tried to propose a joint (unified) model for both the tasks of AE and SC. There are mainly two ways: Joint training and Collapsed tagging. In the former, a multi-task learning framework is built where both the subtasks, AE and SC have individual tags and are trained independently, and they may have some shared features. Then the two models are combined during inference. Meanwhile, in the latter, a collapsed set of tags e.g. B-Sentiment, I-Sentiment, O are used, and then a single model is trained combinedly for both the tasks.

Mitchell et al. (2013) formulated this task as a sequence tagging model and proposed a model using CRFs for the same. Li et al. (2019) made use of the collapsed tagging scheme, involving two stacked RNNs and a gate mechanism to maintain sentiment consistency. Zhou et al. (2019) proposed a span-based joint model using BiLSTMs and an attention mechanism to compute the sentiment information towards each span. Hu et al. (2019) proposed a pipelined span-based extract-then-classify framework using BERT (Devlin et al., 2018) as a backbone network jointly trained on AE and SC. Chen and Qian (2020) tried to exploit the interactive relation between the two subtasks by constructing a multi-layer multi-task framework with a relation propagation mechanism and thereby boosting the performance of both the subtasks.

[1]The datasets can be found here https://github.com/vaibhavb26/Movie-Reviews-Datasets

Sentence	Great acting dreadful editing	
Pipeline	Target Start: 2, 4	Target End: 2, 4
	Polarity: +, -	
Collapsed	Target Start: 2+, 4-	Target End: 2+, 4-

Table 2: Span-based labeling scheme

Sentence	The service was exceptional - sometime there was a feeling that we were served by the army of friendly waiters
Predicted Aspects	[service], [served], [waiter], [waiters]
Gold	[service], [waiters]

Table 3: Half Word Coverage and Overlapping spans

3 Methodology

3.1 Preliminaries

Hu et al. (2019) proposed a span-based labeling scheme, as shown in Table 2, i.e., annotating each opinion target with its span boundary followed by its sentiment polarity. While this model reduces the search space marginally, the approach has issues like overlapping spans, half-word coverage.

As shown in the example in Table 3, the sub-word [waiter] is being predicted twice; i.e., it is part of two different predicted spans ([waiter] and [waiters]). This is a case of overlapping spans as each sub-word should be a part of no more than one span, and since this output will then be sent to the sentiment classifier, there could be a problem of sentiment inconsistency for the word "waiters". Also, half-word coverage is evident here as "waiter" should not be considered as an aspect, instead, "waiters" is more appropriate, because if "waiter" is considered as an aspect it will lead to having two different tags for the word "waiters" (one tag for "waiter" and one tag for "s") which is incorrect. Though the work clearly states that they remove redundant spans with the word-level F1 function but since BertTokenizer tokenizes the words into sub-words (e.g. Waiters being tokenized to "waiter" and "s"), the redundancy issue persists.

3.2 Supervised Method

We formulate ABSA in a different way as compared to most of the previous works which treat ABSA as a sequence-tagging problem. As shown by (Lee et al., 2016), it is more beneficial to predict the two endpoints of a span as compared to sequence-tagging(BIO prediction). Hence, we use a similar approach of predicting the two endpoints. Similar to (Hu et al., 2019), we make use of span-based labeling scheme which is as follows: given an input sentence x of length n i.e. $x = \{x_1, x_2, ..., x_n\}$, we make three different lists, *start_positions*, *end_positions* and *polarities*, each of length *m* where *m* is the number of targets in the sentence. *start_positions* is a list containing the start position of each target in a sentence. Similarly, *end_positions* is a list containing the end position of each target and *polarities* is the list containing the polarities(Positive, Negative, Neutral) of each target.

We build two different models for the two subtasks of Aspect Extraction and Sentiment Classification. These two models are separately trained and combined as a pipeline during inference. The pre-trained BERT model can be finetuned with just one additional layer to create state-of-the-art results in many tasks (Devlin et al., 2018). Therefore we use the BERT encoder as the primary network in both the subtasks. Using the pre-trained transformer blocks (Vaswani et al., 2017), the word embeddings are mapped to contextualized token representations. An Aspect extractor is used to extract the multiple possible targets from the sentence. Then, a polarity classifier (Hu et al., 2019) is used to predict the polarity of each extracted target using the summarized span representation.

3.2.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) can achieve state-of-the-art results in a lot of NLP tasks (Devlin et al., 2018) and hence we use it as our main network. Given a sentence x, we first tokenize the sentence using BertTokenizer(based on wordpiece) with a vocabulary of 30522 tokens. Then we put a [CLS] token at the start and [SEP] token at the end of the tokenized sentence to form a new sentence y of length $n + 2$

(considering n as the length of y). For each token $y_i \in y$, its input representation is constructed by summing the corresponding token, segment and position embeddings (Devlin et al., 2018). Now the input representation is passed to the series of L stacked transformers blocks($L = 12$ for BERT-base and $L = 24$ for BERT-Large) to get the contextual representations. It has been used in various downstream tasks including GLUE (Devlin et al., 2018), subjective bias detection (Pant et al., 2020), and sarcasm detection (Pant and Dadu, 2020). We suggest readers go through (Devlin et al., 2018) and (Vaswani et al., 2017) to get an in-depth understanding of BERT and the transformer block architecture, respectively.

3.2.2 Aspect Extractor

The aim of Aspect extractor is to extract all possible opinion targets from a given sentence. Instead of tagging the sentence sequentially, we detect the target by predicting the *start* and *end* positions of the targets (Hu et al., 2019). We add another layer on top of the BERT model. Using this, we get the confidence score for the start and end position as shown in Equation 1, where h is the contextual representation of the input (Output of BERT) and w_s, w_e are trainable weight vectors.

$$c_s = w_s h,\ p_s = softmax(c_s)$$
$$c_e = w_e h,\ p_e = softmax(c_e) \quad (1)$$

For training, we then generate two lists, a list of *starts* and a list of *ends*, each of length $n + 2$. Each position in the *starts* signifies if any span in the training sentence starts at the given position. Similarly, the list of *ends* signifies if any span in the training sentence ends at the given position. And the probabilities p_s and p_e are calculated as shown in Equation 1. The training objective is the sum of the negative log probabilities of the true *start* and *end* positions on the two predicted probabilities (Hu et al., 2019). The training objective is shown in Equation 2 where y_s and $y_e \in \mathbb{R}^{n+2}$, and each element y_s^i indicates whether the i-th token starts a target and y_e^j indicates whether the j-th token ends a target(Hu et al., 2019).

$$L = -\sum_{i=1}^{n+2} y_s^i log(p_s^i) - \sum_{j=1}^{n+2} y_e^j log(p_e^j) \quad (2)$$

Algorithm 1: TEASER's Heuristic for Aspect Extraction

Input: c_s, c_e, α, K
/* c_s: score of start position */
/* c_e: score of end position */
/* α: threshold value */
/* K : maximum proposed targets */
P, Out, H = {}, {}, {}
/* P: preliminary predictions */
/* Out: output list */
/* H: heuristic score */
$selected$ = {}
$starts$, $ends$ = Top-M indices of c_s, c_e
for s_i *in* $starts$ **do**
  for e_j *in* $ends$ **do**
    if $s_i \leq e_j$ *and* $c_s[i] + c_e[j] \geq \alpha$ **then**
      $target = [s_i, e_j]$
      $score = c_s[i] + c_e[j] - \sqrt{j - i + 1}$
      $P = P \cup target$
      $H = H \cup score$
P.sort() /* sort based on Heuristic score in reverse order */
for $pred$ *in* P **do**
  if *size*$(Out) < K$ **then**
    $s_i, e_i = pred.start\_position,$ $pred.end\_position$
    if $\forall\ i \in [s_i, e_i] \notin selected$ **then**
      $Out = Out \cup pred$
      $selected = selected \cup [s_i, e_i]$
  else
    break
return Out

Once we get the confidence scores c_s and c_e, the objective is to choose the non-overlapping spans ensuring no half-word coverage that has maximum value of $c_s^i + c_e^j$ such that $i <= j$. As shown by (Hu et al., 2019), choosing the top k spans could lead to overlapping spans. Hence, we present a TEASER's heuristic for Aspect Extraction as shown in Algorithm 1. The algorithm helps remove the overlaps as well as half-word coverage. Firstly, we choose top M indices from both the confidence scores($starts$, $ends$). For each pair s_i, e_j such that $s_i \in starts$ and $e_j \in ends$, $s_i <= e_j$ and s_i is a start of a word and e_j is an end of a word, The *heuristic score* is defined as $c_s^i + c_e^j - \sqrt{(length\ of\ the\ target)}$. It is inter-

esting to note that, the *heuristic score* is a function of the length of the target and this is very important for the performance of the model as the targets are usually short entities. If the *heuristic score* of these two indices is greater than a certain threshold (manually tuned), we add it to the list of *preliminary predictions*. *preliminary predictions* is a list of predictions which follow the heuristic condition but it also has overlapping targets. To remove the overlaps we maintain another list of selected tokens, *selected*, which helps in identifying if a token was a part of a better prediction.

We sort the *preliminary predictions* in reverse order with the most confident prediction being in the first place and so on. We then iterate through the list, let the *start position* and *end position* of the current prediction we are looking at be t_s, t_e respectively. If any $token \in [t_s, t_e]$ was already present in a previous prediction (which is calculated using the $selected$ list), we discard the current prediction. If no token is present in the *selected* list, we add the prediction to the list of targets and mark all $tokens \in [t_s, t_e]$ as *selected*.

We repeat this until we reach the end of preliminary predictions or the maximum number of targets are extracted. The pseudocode of the algorithm is as shown in Algorithm 1.

3.2.3 Polarity Classifier

Instead of using sequence tagging methods, we calculate a summarized vector from the contextualized sentence vectors according to the span boundary (Hu et al., 2019). The summarized vector is calculated using attention mechanism (Bahdanau et al., 2015) and the sentiment polarity is predicted with the help of feed-forward neural networks.

We obtain the polarity score by applying a linear transformation followed by a Tanh activation and another linear transformation which is then normalized using the softmax function as shown by (Hu et al., 2019).

$$\begin{aligned} \mathbf{g}^p &= \mathbf{W}_p tanh(\mathbf{W}_v \mathbf{v}) \\ \mathbf{p}^p &= softmax(\mathbf{g}^p) \end{aligned} \tag{3}$$

where $\mathbf{W}_v \in \mathbb{R}^{h*h}$ and $\mathbf{W}_p \in \mathbb{R}^{k*h}$ are two trainable parameter matrices.

We minimize the negative log probabilities of the true polarity on the predicted probability. We calculate the polarity probability for each candidate target span present in the set $\mathbf{O}$ during inference and choose the sentiment class with the highest $\mathbf{p}^p$.

4 Semi-Supervised Learning

4.1 Dataset Creation

We scrape the 15535 sentences from 3200 movie reviews from a leading movie review website. Users rate a movie on a scale of 1 (very bad) to 10 (very good). To avoid any potential bias, we chose the most popular movies with the most reviews, and the reviews were chosen uniformly on a rating scale of 1 to 10. We then divide the dataset into two parts: 14373 sentences (moviesLarge) for training the semi-supervised model and 1162 sentences (Movie20) to validate the semi-supervised model.

Two human annotators with proficiency in English and linguistic background performed the annotation of the dataset's validation split (Movie20). The annotation was performed according to the original guidelines as set in (Pontiki et al., 2014b) on the following aspects:

1. **Opinion Targets**: Given a sentence, identify all the aspect terms present in the sentence. e.g., "Stunning visuals, amazing storyline." The aspect terms in the sentence are "visuals", "storyline".

2. **Target Sentiment**: Assuming that we know the aspect terms beforehand, determine the sentiment attached, i.e., the polarity of each term (Positive, Negative, Neutral). e.g., visuals - Positive, storyline - Positive (Considering the example mentioned above.)

Table 4 shows a few example instances from the novel Movie20 dataset.

For validating the quality of the annotation process, we use the Inter-Annotator Agreement of both the tasks through Cohen's Kappa Coefficient (Fleiss and Cohen, 1973). We obtain a Kappa score of 0.8326 for the annotation process. The Kappa score implies that the annotation process is of high quality, with the annotators showing a high degree of agreement.

4.2 Pseudo-Labeling

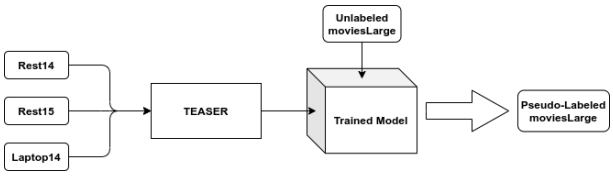
The moviesLarge dataset has 14373 sentences. Since human annotation to such a large dataset is very time-consuming and complex, we use the Pseudo-Labeling technique. In Pseudo-Labeling, instead of manually annotating the dataset, we approximate labels to the dataset based on available labeled data.

Sentences	Aspects	Polarity
It has very good cinematography shots and it is very entertaining.	[cinematography shots]	POS
I like it but the beginning was very long and slow while the end was all over the place trying to explain everything.	[beginning], [end]	NEG, NEG
It is action packed, fantasy filled and thoroughly exciting.	[action], [fantasy]	NEU, NEU
Great Acting dreadful editing.	[Acting], [editing]	POS, NEG
the whole cinematic experience is not there.	[cinematic experience]	NEG

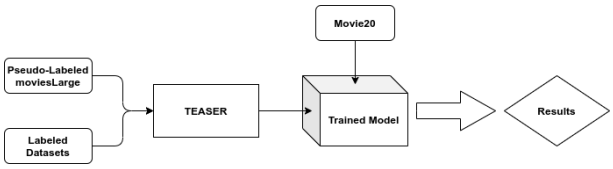
Table 4: Example instances from Movie20 dataset.

Datasets	Restaurant14		Restaurant15		Laptop14	
	Train	Test	Train	Test	Train	Test
#Sentences	3040	800	1313	685	3045	800
#Aspects	3603	1122	1209	547	2302	634

Table 5: The statistics of the three datasets



(a) Pseudo-Labeling



(b) Model Training

Figure 1: Semi-Supervised Learning: Model illustration

Figure 1 shows the Pipeline of Semi-Supervised Learning. As shown, it can be divided into two parts: Pseudo-Labeling (1a) and final Model Training (1b). In case of pseudo-labeling, we combine all the three existing datasets, *Laptop14*, *Restaurant14*, and *Restaurant15* and using this as our training data, we train our model TEASER. We then make use of the trained model to predict the labels of the unlabeled moviesLarge dataset. Since these labels aren't manually annotated, these are the approximate labels and hence the process is called pseudo-labeling.

Once we have the pseudo-labeled moviesLarge dataset, we combine moviesLarge and the labeled datasets *Laptop14*, *Restaurant14*, and *Restaurant15* and train the model using this as our training data. Finally, we test this model on our novel Movie20 dataset and derive the results. The details are discussed in subsection 6.2.

5 Experiments

5.1 Datasets

For all the experiments, we use the three benchmark datasets from various domains. The datasets were taken from SemEval 2014 (Pontiki et al., 2014b) and SemEval 2015 (Pontiki et al., 2014a) tasks which include 2 datasets from restaurant domain, *Restaurant14* and *Restaurant15* [2]. Moreover, we use another dataset *Laptop14* made by using customer reviews from the laptop domain. The statistics of the datasets are as shown in the Table 5.

5.2 Metrics

We use the F1 score as the evaluation metric for the ABSA task. To analyze our TEASER's heuristic for Aspect Extraction's performance, we also compare using an F1 score between various models for the subtask of Aspect Extraction.

5.3 Experimental Settings

We use the pre-trained BERT-Large model for all the experiments. It has 24 layers (transformer blocks), 16 attention heads. For more details about BERT-Large parameters, readers can refer to (Devlin et al., 2018). The batch size is 32, M (number of candidate spans) is set to 20, while K, the maximum number of proposed targets is 10, and the threshold is manually tuned. We use Adam optimizer with a learning rate of 2e-5.

[2]We do not use the erroneous *rest_total* dataset as prescribed by the authors of Li et al. (2019).

Model	Restaurant14 F1 score		Restaurant15 F1 score		Laptop14 F1 score	
	AE	ABSA	AE	ABSA	AE	ABSA
MNN	83.05	63.87	70.24	56.57	76.94	53.80
E2E-TBSA	83.92	66.60	69.40	57.38	77.34	55.88
SpanABSA	86.71	73.68	74.63	62.29	82.34	61.25
DE-CNN	82.79	-	68.52	-	79.38	-
RACL-BERT	86.38	75.42	73.99	66.05	81.79	63.40
TEASER	**88.76**	**75.53**	**79.78**	**67.34**	**87.16**	**68.93**

Table 6: Results of the Aspect Extraction Task and the Aspect-based Sentiment Analysis task.

5.4 Baseline Methods

We compare our proposed model with the following approaches [3]:

1. MNN (Wang et al., 2018) - This work proposed a unified (collapsed) tagging scheme for both the tasks of Aspect Extraction and Sentiment Classification.

2. SpanABSA (Hu et al., 2019) - It is a pipelined model with a multi-target extractor and a polarity classifier. It uses BERT-Large as the backbone network for both the subtasks.

3. E2E-TBSA (Li et al., 2019) - It has two stacked RNNs(Recurrent Neural Networks) with multi-task learning over a collapsed tagging scheme.

4. DE-CNN (Xu et al., 2018) - It is a model exclusively for Aspect Extraction, which uses a double embedding mechanism with CNNs(Convolutional Neural Networks).

5. RACL-BERT (Chen and Qian, 2020) - This is the current state-of-the-art method that proposes a Relation Aware Collaborative Learning (RACL) framework which allows the subtasks to work coordinately via the multi-task learning and relation propagation mechanisms in a stacked multi-layer network.

6 Results

6.1 Supervised Model

The Table 6 shows the comparison for all the methods. For the task of Aspect Extraction, our model achieves 2.05%, 5.15%, 4.82% absolute gains over the three benchmark datasets, which proves the efficacy of our model. Also, for the overall ABSA task, our model achieves 0.09%, 1.29%, 5.43% absolute gains, which is significant. The AE results prove that span-based extraction performs better than any of the other methods proposed. The overall ABSA results suggest that it is better to use two different models for the two subtasks and then combine via pipeline over jointly learning to predict them simultaneously. This further concretizes the fact mentioned by (Hu et al., 2019) that Target Extraction and Sentiment Classification are loosely coupled, i.e., there is a weak connection between them.

6.2 Semi-Supervised Model

As shown in Table 7, we report the Precision, Recall, and F1-score of the model on our novel Movie20 dataset for the AE and ABSA task. The F1 score for AE is 63.74% and 58.23% for ABSA. Through this, we set a strong benchmark for semi-supervised aspect-based sentiment analysis on a movie-based dataset. We further analyze the model's predictions and discover the following patterns in the errors made by the model:

1. The model usually failed to mark aspects preceded by rare adjectives (i.e., adjectives that occurred in the dataset with less frequency). For example, in the following sentence, "Great Acting dreadful editing", the words "acting" and "editing" are the targets with polarities Positive and Negative respectively. However, the model recognized "acting" but failed to recognize "editing" because the word "dreadful" occurs very rarely in the dataset. Fewer examples in the dataset could have caused the model to fail in such cases.

2. The model also failed to mark sentences where specific experiential knowledge about a movie's good and bad aspects was required (usually in the absence of any clear adjec-

[3] We use the results as in Chen and Qian (2020).

Model	Movie20 Precision		Movie20 Recall		Movie20 F1 score	
	AE	ABSA	AE	ABSA	AE	ABSA
TEASER	81.91	79.91	52.17	45.80	63.74	58.23

Table 7: Results of the Semi-supervised Aspect Extraction Task and the Aspect-based Sentiment Analysis task.

tives). For example, in the following sentence, "We know the bliss can't last. Thus, tears stream down your face during the third act"; the phrase "third act" should be marked as an aspect with a positive sentiment since movies that can connect with the audience's emotions are considered good. However, the model does not have this experiential knowledge and hence failed to recognize the aspect. A similar example is the following sentence, "I could not relate to any character and did not care about the outcome.", where the model failed to mark "character" as an aspect with negative sentiment since the model does not have the experiential knowledge that relatable characters make for a good movie.

7 Conclusion

In this work, we proposed TEASER, an extract-then-classify network for Aspect-based Sentiment Analysis with pre-trained BERT-Large as the main network. We also presented an Aspect extractor with a novel heuristic, which helps extract all the targets of a given sentence. Experiments show that our method consistently outperforms the current state-of-the-art in the task of AE and also in ABSA. We also presented two datasets, Movie20, a supervised dataset of 1162 sentences with a Cohen Kappa Score of 0.8326, and moviesLarge, a pseudo-labeled dataset of around 14373 sentences. Lastly, using Semi-supervised learning, we benchmarked TEASER on the Movie20 dataset. We analyzed the model to reason where the model failed to perform, and according to the findings, aspects preceded with rare adjectives and aspects with an absence of a clear adjective were the primary reasons for the failure.

References

Dzmitry Bahdanau, K. Cho, and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *EMNLP*.

Z. Chen and Tieyun Qian. 2020. Relation-aware collaborative learning for unified aspect-based sentiment analysis. In *ACL*.

Zhuang Chen and T. Qian. 2019. Transfer capsule network for aspect level sentiment classification. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. Cite arxiv:1810.04805Comment: 13 pages.

Li Dong, F. Wei, Chuanqi Tan, Duyu Tang, M. Zhou, and K. Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *ACL*.

Feifan Fan, Yansong Feng, and Dongyan Zhao. 2018. Multi-grained attention network for aspect-level sentiment classification. In *EMNLP*.

Joseph L. Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement*, 33(3):613–619.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *ACL (1)*, pages 388–397. Association for Computational Linguistics.

Minghao Hu, Y. Peng, Z. Huang, D. Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. In *ACL*.

Minqing Hu and B. Liu. 2004. Mining and summarizing customer reviews. In *KDD '04*.

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single and cross-domain setting with conditional random fields. In *EMNLP*, pages 1035–1045. ACL.

L. Jiang, Mo Yu, M. Zhou, X. Liu, and T. Zhao. 2011. Target-dependent twitter sentiment classification. In *ACL*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. Cite arxiv:1408.5882Comment: To appear in EMNLP 2014.

Kenton Lee, T. Kwiatkowski, Ankur P. Parikh, and D. Das. 2016. Learning recurrent span representations for extractive question answering. *ArXiv*, abs/1611.01436.

Xin Li, Lidong Bing, Piji Li, and W. Lam. 2019. A unified model for opinion target extraction and target sentiment prediction. In *AAAI*.

Xin Li and W. Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In *EMNLP*.

C. Lin and Y. He. 2009. Joint sentiment/topic model for sentiment analysis. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Pengfei Liu, Shafiq R. Joty, and Helen M. Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *EMNLP*, pages 1433–1443. The Association for Computational Linguistics.

M. Mitchell, J. Aguilar, T. Wilson, and B. V. Durme. 2013. Open domain targeted sentiment. In *EMNLP*.

B. Pang and L. Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1-2):1–135.

Kartikey Pant and Tanvi Dadu. 2020. Sarcasm detection using context separators in online discourse. In *FIGLANG*.

Kartikey Pant, Tanvi Dadu, and Radhika Mamidi. 2020. Towards detection of subjective bias using contextualized word embeddings. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 75–76, New York, NY, USA. Association for Computing Machinery.

Maria Pontiki, D. Galanis, Haris Papageorgiou, S. Manandhar, and Ion Androutsopoulos. 2014a. Semeval-2015 task 12: Aspect based sentiment analysis. In *SemEval@NAACL-HLT*.

Maria Pontiki, D. Galanis, John Pavlopoulos, Haris Papageorgiou, Ion Androutsopoulos, and S. Manandhar. 2014b. Semeval-2014 task 4: Aspect based sentiment analysis. In *COLING 2014*.

Soujanya Poria, E. Cambria, and Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowl. Based Syst.*, 108:42–49.

Lei Shu, Hu Xu, and Bing Liu. 2017. Lifelong learning crf for supervised aspect extraction. In *ACL*.

Duyu Tang, B. Qin, X. Feng, and T. Liu. 2016. Effective lstms for target-dependent sentiment classification. In *COLING*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, A. Gomez, L. Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *IJCAI*.

Shuai Wang, S. Mazumder, B. Liu, Mianwei Zhou, and Yi Chang. 2018. Target-sensitive memory networks for aspect sentiment classification. In *ACL*.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016a. Recursive neural conditional random fields for aspect-based sentiment analysis. In *EMNLP*, pages 616–626. The Association for Computational Linguistics.

Yequan Wang, Minlie Huang, X. Zhu, and L. Zhao. 2016b. Attention-based lstm for aspect-level sentiment classification. In *EMNLP*.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *ACL*.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. *ArXiv*, abs/1805.07043.

L. Zhang, Shuai Wang, and B. Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2015. Neural networks for open domain targeted sentiment. In *EMNLP*.

Y. Zhou, Longtao Huang, T. Guo, Jizhong Han, and Songlin Hu. 2019. A span-based joint model for opinion target extraction and target sentiment classification. In *IJCAI*.

Peisong Zhu, Zhuang Chen, Haojie Zheng, and Tieyun Qian. 2019. Aspect aware learning for aspect category sentiment analysis. *ACM Transactions on Knowledge Discovery from Data*, 13:1–21.

Peisong Zhu and T. Qian. 2018. Enhanced aspect level sentiment classification with auxiliary memory. In *COLING*.