ZHIXIAOBAO at SemEval-2022 Task 10: Apporoaching Structured Sentiment with Graph Parsing

Yangkun Lin[*], **Chen Liang**[*], **Jing Xu, Chong Yang**[†], **Yongliang Wang**
Ant Group
Hangzhou, China
{linyangkun.lyk, liangchen.liangche, jill.xj,
yangchong.yang, yongliang.wyl}@antgroup.com

Abstract

This paper presents our submission to task 10, Structured Sentiment Analysis of the SemEval 2022 competition. The task aims to extract all elements of the fine-grained sentiment in a text. We cast structured sentiment analysis to the prediction of the sentiment graphs following (Barnes et al., 2021), where nodes are spans of sentiment holders, targets and expressions, and directed edges denote the relation types between them. Our approach closely follows that of semantic dependency parsing (Dozat and Manning, 2018). The difference is that we use pre-trained language models (e.g., BERT and RoBERTa) as text encoder to solve the problem of limited annotated data. Additionally, we make improvements on the computation of cross attention and present the suffix masking technique to make further performance improvement. Substantially, our model achieved the **Top-1** average Sentiment Graph F1 score on seven datasets in five different languages in the monolingual subtask.

1 Introduction

SemEval 2022 task 10 is a structured sentiment analysis task, aiming to predict all of the opinion tuples in a text. Each opinion O is a tuple (t, h, e, p), where h is a holder who expresses a polarity p towards a target t through a sentiment expression e. In practical, the task of structured sentiment analysis can help machines understand how people perceive ideas, policy etc.

This paper describes the system developed by the team ZHIXIAOBAO for SemEval-2022 Task 10. We follow the work of (Barnes et al., 2021) to cast the task as dependency graph parsing problem. The predicted opinion tuples are denoted by a directed graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, for each sentence. As shown in Figure 1, all tokens in a sentence are presented as nodes and there are directed edges between the nodes to represent their relations. Each node in $\mathcal{V}$ can point to multiple nodes, and can have multiple incoming edges too. Sentiment expressions are regarded as roots in the structured sentiment graph (e.g., the token "5" and "don't" in Figure 1). Notice that not all nodes connect to the other nodes (e.g., the token "give" in Figure 1). The isolated tokens are the none-sentiment elements in the sentence, thus we should be able to predict the edge type of "null" in our model. Original edge types are defined as *holder*, *target*, and *expression*. Following the work of (Barnes et al., 2021), we also tried the **"+inlabel"** style of definition, where none-"null" edge types consist of $holder_{in}$, $holder_{out}$, $target_{in}$, $target_{out}$, $expression_{in}$, $expression_{out}$. Foot markers *in*, *out* denotes the in-span and out-span edges respectively. For example, the edge from "5" to "some" belongs to $holder_{out}$, while the edge from "Some" to "others" belongs to $holder_{in}$.

As demonstrated in previous work (Barnes et al., 2021), formulating the task as a graph structure prediction problem is superior to that of solving it by the span extraction and relation prediction approaches. The former can better extract overlapping spans than the latter. Thus, our model mainly follows the solution of dependency parsing to directly predict between-word relations. The model consists of a text encoder to extract contextual features of tokens, and a classifier to predict edges between each pair of tokens. A bilinear or biaffine cross attention is applied in the classifier layer to make multiplicative interactions between the features of a pair of tokens. In our model, pre-trained language models (e.g., BERT and RoBERTa) are used as the text encoder. We discover that fine-tuning the pre-trained language model brings huge enhancements in our experiments. In addition, as the meaning of in-span and out-span edge types are totally different, we leverage two cross attention for in-span edge types and out-span edge types prediction respectively. We also present a suffix mask-

[*] The first two authors contributed equally to this work.
[†] Corresponding author.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1343 - 1348

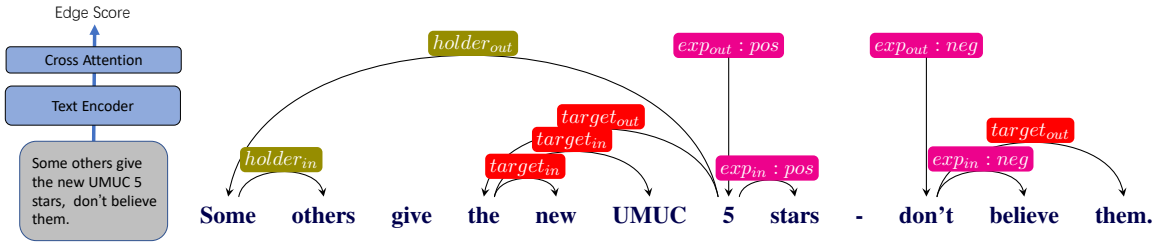

Figure 1: The framework of our model.

Dataset	Language	train	dev	test
NoReC$_{\text{Fine}}$	Norwegian	8634	1531	1272
MultiB$_{\text{CA}}$	Catalan	1174	168	336
MultiB$_{\text{EU}}$	Basque	1064	152	305
OpeNER$_{\text{ES}}$	Spanish	1438	206	410
OpeNER$_{\text{EN}}$	English	1746	249	499
MPQA	English	4500	1622	1681
DS$_{\text{Unis}}$	English	2253	232	318

Table 1: Summary of the datasets.

ing technique to reduce noise in the data. These techniques greatly improve the performance of our model compared with the original dependency parsing model.

Our model ranked **1st** out of 32 participating teams on monolingual subtask and got the highest average F1 score on 4 datasets.

2 Task Description

SemEval task 10 focuses on predicting all elements of the structured sentiment in a text, represented by opinion tuples (t, h, e, p), where h is a holder who expresses a polarity p towards a target t through a sentiment expression e. The evaluation is on seven datasets in five languages, the statistics of which are shown in Table 1.

NoReC$_{\text{Fine}}$ (Øvrelid et al., 2019) is the largest structured sentiment multi-domain dataset with professional reviews in Norwegian. **MultiB**$_{\text{EU}}$ and **MultiB**$_{\text{CA}}$ (Barnes et al., 2018) are hotel reviews datasets in Basque and Catalan, respectively. **OpeNER**$_{\text{EN}}$ and **OpeNER**$_{\text{ES}}$ (Agerri et al., 2013) are polarity-enhanced datasets with customer reviews in Spanish and English respectively. **MPQA** (Wiebe et al., 2005) annotates news articles in English from the world press. Finally, **DS**$_{\text{Unis}}$ (Toprak et al., 2010) is an annotated English reviews dataset of online universities and e-commerce.

Previous shared tasks on Aspect-Based Sentiment Analysis (ABSA) focus on extracting sentiment targets and classifying the polarity directly. Most previous methods follow the information extraction pipeline, which firstly extract the span of holders, targets, expressions and subsequently predict the relations. However, splitting structured sentiment analysis into subtasks may cause the error propagation problem. We follow the work that solving the problem by dependency graph parsing (Barnes et al., 2021) to achieve better performance in our model.

3 System Overview

The overview of our system is shown in Figure 1. We use a pre-trained language model to extract text information as the node features in the graph. Then, a cross attention layer is used to compute the predicted score of each edge type. After we get the edge score for each token pair, a graph parsing algorithm is presented to transform the predicted score to opinion tuples.

3.1 Text Encoder

We use the pre-trained language models, **BERT** and **RoBERTa**, to generate the contextual features of the text in multiple languages. Both of them are Transformer-based language models using a huge amount of text with a masked language model objective. These pre-trained language models have shown great superiority in a low-resource scenario like this task. Compared with BERT, RoBERTa removes next sentence prediction(NSP) loss and applies larger batch size and sequence length during the pre-training step, leading to a better performance in most cases. We tried the monolingual version of **RoBERTa**$_{\text{LARGE}}$ (Liu et al., 2019a) and **BERT**$_{\text{LARGE}}$ (Devlin et al., 2019) for each dataset, as our feature extractor. Our experimental results demonstrate that **RoBERTa**$_{\text{LARGE}}$ performs better than **BERT**$_{\text{LARGE}}$ in all the datasets.

3.2 Discrete Cross Attention

After extracting the text features, we can then use bilinear or biaffine attention to produce a score for

each pair of tokens. The score includes multiplicative interactions among pairs, and can be used to predict edge types. Different from the previous work (Barnes et al., 2021), we model heads and dependents of in-span and out-span separately, because we think it is better to cast the in-span and out-span label prediction as two "different" tasks. According to our observation, they have different properties in the corpus. Thus, inspired by the multi-task learning framework, we propose to use **Discrete Cross Attention (DCA)** to make them share same bottom features in the text encoder, but have non-shared parameters in the computation of cross attention.

The contextual features C extracted from text encoder are processed with four layers of the feed-forward neural networks(FNN), FNN^{in}_{head}, FNN^{in}_{dep}, FNN^{out}_{head} and FNN^{out}_{dep} creating representations of potential heads and dependents for in-span and out-span respectively. And then a bilinear score is computed for each kind of edge types using a trainable parameter matrix A. The discrete cross attention can be formulated as bellow,

$$h_i^{in} = FNN_{head}^{in}(c_i) \quad (1)$$

$$d_i^{in} = FNN_{dep}^{in}(c_i) \quad (2)$$

$$score_{ij}^{in} = h_i^{inT} A_{in} d_j^{in} \quad (3)$$

$$h_i^{out} = FNN_{head}^{out}(c_i) \quad (4)$$

$$d_i^{out} = FNN_{dep}^{out}(c_i) \quad (5)$$

$$score_{ij}^{out} = h_i^{outT} A_{out} d_j^{out} \quad (6)$$

$$score_{ij} = softmax(score_{ij}^{in} \oplus score_{ij}^{out}) \quad (7)$$

$score_{ij}$ represents the final score list for each edge type, which is the softmax score of the concatenation of in-span edge scores $score_{ij}^{in}$ and out-span scores $score_{ij}^{out}$.

3.3 Graph Parsing

We set a threshold θ to determine whether the edge exists, i.e., if $max(score_{ij}) > \theta$, we set the predicted edge type to be $argmax(score_{ij})$, or we make the predicted edge to be "null". We set $\theta = 0.5$ in our experiments. We use two kinds of graph parsing representations, head-first and head-final, following (Barnes et al., 2021). For head-first, we use the first token in the target/holder/expression spans as the head of the span and the other tokens within the span as the dependent. For head-final, we take the opposite way, i.e., set the final token of the target/holder/expression spans as the heads.

Algorithm 1: Graph parsing

Input: Sentiment graph $\mathcal{G}$
Output: Opinion Tuples (H, T, E)
Data: Opinion set Op_{set}

```
for e_{r,i} in G do
    if e_{r,i} ∈ ExpTypes then
        E ← FindSpan(i, Type(e_{r,i}));
          new H_set, T_set;
        for e_{i,j} in G do
            if Type(e_{i,j}) ∈ HolTypes then
                H_set ← FindSpan(j, hol)
            if Type(e_{i,j}) ∈ TrgTypes then
                T_set ← FindSpan(j, trg)
        for H in H_set do
            for T in T_set do
                Op_set ← (H, T, E)
```

The algorithm of converting structured sentiment graph to opinion tuples (H, T, E) is in shown in **Algorithm 1**. H, T, E denote $holder$, $target$ and $expression$ respectively. $ExpTypes$, $HolTypes$, $TrgTypes$ are the edge type sets for $expression$, $holder$, $target$ respectively. $e_{i,j}$ denotes the predicted edge type between token i and token j, and r is the root nodes. $FindSpan(\cdot)$ is a function to find the complete span for a certain edge type, which can be simply implemented by merging linked tokens with the same edge type. As shown in **Algorithm 1**, we should first find the expressions, and then add the linked holders and targets for each expression to the opinion tuples. Notice that if we cannot find holder or target spans for an expression, we shall append an empty token into H_{set} or T_{set}.

3.4 Suffix Masking Trick

In both training and predicting procedure, a sentence is first tokenized by byte-pair encoding (BPE) before it is inputted into the text encoder, i.e., BERT

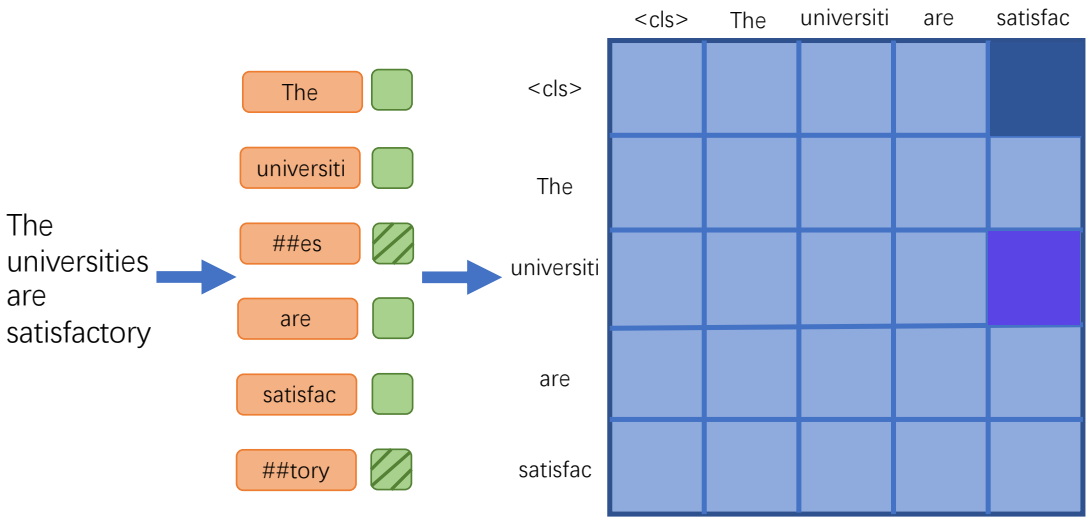

Figure 2: Suffix masking.

and RoBERTa. Some words are splitted into prefixes and suffixes in the procedure. For example, "universities" is tokenized to "universiti", "##es" as shown in Figure 2. We think these suffixes are often noises and provide few supervision signals for the edge prediction, since they are shared by many distinguished words. Inspired by this intuition, we mask these suffixes in the computation of edge scores. As shown in Figure 2, we mask the suffix, "##es" and "##tory", before the prediction of edges. In this way, the edges only exist between the pair of $(< cls >, satisfac)$ and $(universiti, satisfac)$.

4 Experiments

The experiments details and main results are shown in this section.

4.1 Experiment Details

The implementation of our model depends on pytorch and huggingface. In our experiments, **BERT**$_{\text{LARGE}}$ represents the monolingual version for each dataset, which all can be found in the huggingface website. As for **RoBERTa**$_{\text{LARGE}}$, the version of *xlm-roberta-large* (Conneau et al., 2019) is used on models for **MultiB**$_{\text{CA}}$, **MultiB**$_{\text{EU}}$ and **OpeNER**$_{\text{ES}}$, and *roberta-large-en-cased* (Liu et al., 2019b) is used on the English datasets, i.e., **MPQA**, **OpeNER**$_{\text{EN}}$ and **DS**$_{\text{Unis}}$.

We use Adam as our optimizer with the learning rate to be 1e-5 for the fine-tuning of pre-trained language models and 1e-4 for the other parameters in the model. The batch size is set to 12 with the gradient accumulation steps to be 48. The dropout rate is 0.3 and the hidden state size of FNN layers is set to 256. Our models are run on a maximum of 1000 epochs. We train all the models with 5 different seeds on the training set released by the orginizer and choose the best results based on the performance on development datasets. The training run on two Tesla V100 GPUs with 32G memory.

It has to be noted that we add a "<cls>" token when encoding the sentences and set the "<cls>" token as the root of the sentiment graph. In Figure 1, there actually exists an edge between <cls> token and the head of expression spans.

4.2 Metrics

To measure how well a system is able to capture the full sentiment graph, submitted systems are evaluated on sentiment graph F_1 (SF_1) following (Barnes et al., 2021). A true positive is defined as an exact match at graph-level, weighting the overlap in predicted and gold spans for each sentiment element, averaged across all three kinds of spans, i.e., expression, holder, target. For precision we weight the number of correctly predicted tokens divided by the total number of predicted tokens (for recall, we divide instead by the number of gold tokens).

4.3 Main Results

The main experimental results are shown in Table 2. It can be seen that using monolingual **BERT**$_{\text{LARGE}}$ pre-trained on larger language-specific corpus as text encoder is better than the multilingual **BERT**$_{\text{base}}$ used in (Barnes et al., 2021), and fine-tuning the pre-trained language models brings more improvements than freezing the parameters. An in-

Methods	NoReC$_{\text{Fine}}$	MultiB$_{\text{CA}}$	MultiB$_{\text{EU}}$	OpeNER$_{\text{ES}}$	OpeNER$_{\text{EN}}$	MPQA	DS$_{\text{Unis}}$
mBERT$_{\text{base}}$ w/o fine-tune + lstm (Barnes et al., 2021)	39.4	55.8	57.4	-	-	18.8	27.3
BERT$_{\text{LARGE}}$ w/o fine-tune + lstm	42.9	63.7	62.1	65.4	65.0	33.5	36.7
BERT$_{\text{LARGE}}$ fine-tune + lstm	50.4	69.1	65.2	67.1	69.2	42.1	40.8
BERT$_{\text{LARGE}}$ fine-tune	50.8	70.7	65.7	68.6	70.8	43.5	42.4
BERT$_{\text{LARGE}}$ fine-tune + inlabel	50.9	70.5	65.9	68.4	71.5	43.3	42.8
BERT$_{\text{LARGE}}$ fine-tune + inlabel + DCA	51.7	71.1	67.3	69.1	73.1	43.6	44.5
BERT$_{\text{LARGE}}$ fine-tune + inlabel + DCA + mask	**52.9**	71.7	70.5	71.6	75.4	43.9	47.2
RoBERTa$_{\text{LARGE}}$ fine-tune + inlabel + DCA + mask	-	**72.8**	**73.9**	**72.2**	**76.0**	**44.7**	**49.4**

Table 2: Main results. **mBERT**$_{\text{base}}$ denotes the multilingual BERT (Xu et al., 2019). "+lstm" denotes adding an LSTM layer after the text encoder. "+inlabel", "DCA" and "mask" denote the "+inlabel" style of edge types, the discrete cross attention and the suffix masking technique presented in last section.

teresting discovery is that adding an LSTM layer between text encoder and cross attention leads the decreasing of SF_1 score. Thus, we remove the LSTM layer in our final submitted models. In addition, we can see that the **"+inlabel"** style definition of edge types is indeed helpful in this task. Furthermore, the presented discrete cross attention and suffix masking technique significantly improve the performance of our model.

The results prove the effectiveness of fine-tuning **RoBERTa** in this task. As shown in the Table 2, methods with **RoBERTa**$_{\text{LARGE}}$ as the text encoder on six datasets achieve the best performance. The best SF_1 scores on **MPQA**, **OpeNER**$_{\text{EN}}$ and **DS**$_{\text{Unis}}$ are 44.7, 76.0, and 49.4, where *roberta-large-en-cased* is used in the model. For **MultiB**$_{\text{CA}}$, **MultiB**$_{\text{EU}}$ and **OpeNER**$_{\text{ES}}$, *xlm-roberta-large* (Conneau et al., 2019) outperforms **BERT**$_{\text{LARGE}}$ and we have 72.8 on **MultiB**$_{\text{CA}}$, 73.9 on **MultiB**$_{\text{EU}}$ and 72.2 on **OpeNER**$_{\text{ES}}$. The results demonstrate that a pre-trained language model with more parameters and trained on the larger corpus performs very well in downstream tasks as feature extractor. For **NoReC**$_{\text{Fine}}$, we use *nb-bert-large* (Kummervold et al., 2021) and the SF_1 score is 52.9. We do not get results on **NoReC**$_{\text{Fine}}$ in the last line of the table because monolingual **RoBERTa**$_{\text{LARGE}}$ for Norwegian is not available at the time of our experiments. We guess that the model on **NoReC**$_{\text{Fine}}$ using **RoBERTa**$_{\text{LARGE}}$ will achieve a better result.

There are some failed attempt during the period of competition, too. We tried to enrich the contextual features of a sentence with word embedding, POS tag embedding, lemma embedding by using tools like SpaCy (Honnibal et al., 2020), Stanza (Qi et al., 2020) and UDPipe (Straka and Straková, 2017). But, we find that it is not superior to directly fine-tuning the pre-trained language model. We also tried to pre-train the RoBERTa-large on a Norwegian corpus, then the SF_1 score continuously grows with the increasing of training steps. However, due to the limitation of computation resources and time, we only trained the model with 2M steps. The final version does not outperform that using **BERT**$_{\text{LARGE}}$ because of the inadequate training.

5 Conclusion

In this paper, we have presented the implementation of the ZHIXIAOBAO system submitted to the SemEval-2022 Task 10. We propose an enhanced dependency parsing model for sentiment graph analysis. We leverage the fine-tuning technique of pre-trained language models, **BERT**$_{\text{LARGE}}$ and **RoBERTa**$_{\text{LARGE}}$ to increase the ability of model generalization. Furthermore, we present the discrete cross attention and suffix masking technique to achieve a significant performance improvement. Our model ranked **1st** out of 32 participating teams on the monolingual subtask with the highest SF_1 score on 4 datasets.

6 Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments.

References

Rodrigo Agerri, Montse Cuadros, Seán Gaines, and German Rigau. 2013. Opener: Open polarity enhanced named entity recognition. *Procesamiento Del Lenguaje Natural*, 51:215–218.

Jeremy Barnes, Robin Kurtz, Stephan Oepen, Lilja Øvrelid, and Erik Velldal. 2021. Structured sentiment analysis as dependency graph parsing. *arXiv preprint arXiv:2105.14504*.

Jeremy Barnes, Patrik Lambert, and Toni Badia. 2018. Multibooked: A corpus of basque and catalan hotel

reviews annotated for aspect-level sentiment classification. *arXiv: Computation and Language*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Per E Kummervold, Javier de la Rosa, Freddy Wetjen, and Svein Arne Brygfjeld. 2021. Operationalizing a national digital library: The case for a norwegian transformer model.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized bert pretraining approach.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2019. A fine-grained sentiment dataset for norwegian. *arXiv preprint arXiv:1911.12722*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Cigdem Toprak, Niklas Jakob, and Iryna Gurevych. 2010. Sentence and expression level annotation of opinions in user-generated discourse. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 575–584.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. Bert post-training for review reading comprehension and aspect-based sentiment analysis.