ETMS@IITKGP at SemEval-2022 Task 10: Structured Sentiment Analysis Using A Generative Approach

R Raghav[*] and **Adarsh Vemali**[*] and **Rajdeep Mukherjee**

Indian Institute of Technology Kharagpur, India

{rraghav5600, adarsh.vemali, rajdeep1989}@iitkgp.ac.in

Abstract

Structured Sentiment Analysis (SSA) deals with extracting opinion tuples in a text, where each tuple (h, e, t, p) consists of h, the holder, who expresses a sentiment polarity p towards a target t through a sentiment expression e. While prior works explore graph-based or sequence labeling-based approaches for the task, we in this paper present a novel unified generative method to solve SSA, a SemEval-2022 shared task. We leverage a BART-based encoder-decoder architecture and suitably modify it to generate, given a sentence, a sequence of opinion tuples. Each generated tuple consists of seven integers respectively representing the indices corresponding to the start and end positions of the holder, target, and expression spans, followed by the sentiment polarity class associated between the target and the sentiment expression. We perform rigorous experiments for both Monolingual and Cross-lingual subtasks, and achieve competitive Sentiment F1 scores on the leaderboard in both settings.

1 Introduction

Structured Sentiment Analysis (SSA) is the task of extracting structured information around sentiment expressions present in text in the form of opinion tuples $O = \{O_1, O_2, ..., O_n\}$, where each opinion tuple $O_i = (h, t, e, p)$ consists of h, the *holder* (or *source*, used interchangeably) who expresses a sentiment polarity p towards a *target* (or *aspect*) t using an opinion or sentiment expression e (Barnes et al., 2021a). Prior works (Liu, 2012; Peng et al., 2020) have highlighted the importance of addressing sentiment analysis as a structured prediction problem in order to capture the complete information around various opinions expressed in the text. The task of SSA thus expects to exploit the pairwise interactions between the members of the same opinion tuple during the extraction process.

With the exponential growth of online marketplaces and user-generated content therein, SSA or near similar tasks of aspect-sentiment-opinion triplet extraction (Peng et al., 2020; Mukherjee et al., 2021a; Yan et al., 2021), and aspect-category-sentiment-opinion quad extraction (Cai et al., 2021), the newest additions under the broader umbrella of aspect-based sentiment analysis (ABSA) (Pontiki et al., 2014a,b) have become more important than ever (Mukherjee et al., 2021b). In the face of ever-expanding choices, it becomes a challenging necessity to take educated explainable decisions from past user reviews. SSA guides the learning in the proper direction by facilitating an automated way to focus on major sentiment or opinion indicators. As a result, the task has wide applications in various market segments, such as e-commerce, food delivery, healthcare, ride sharing, travel and hospitality, to name a few.

Previous efforts on SSA have primarily focused on two approaches: sequence labeling-based (He et al., 2019), and graph-based (Barnes et al., 2021b). The former tries to first predict the presence/absence of targets and expressions in the text by sequentially labeling each text token using BIOES[1] tags, before modeling their interaction to predict the sentiment polarity. The latter models the task as a dependency graph parsing problem, where the sentiment expression is considered as the root node, and the other elements are connected via arcs that represent their relationships. Different from these, we present a novel generative approach to solve SSA. More specifically, we take motivation from a unified generative framework recently proposed by Yan et al. (2021) to solve several ABSA tasks. We suitable modify their BART-based encoder-decoder architecture to adapt it for SSA. Given a sentence, the model is

---

[*] Equal contribution

[1] BIOES is a tagging scheme commonly used for sequence labeling tasks. B, I, E, O, and S respectively denote the *begin, inside, outside, end*, and *single* tags corresponding to an entity.

Dataset Name	Language	% Null (Train, Dev)	Size (Train, Dev, Test)
NoReC_fine (Øvrelid et al., 2020)	Norwegian	(47.24%, 46.37%)	(8634, 1531, 1272)
MultiBooked_eu (Barnes et al., 2018)	Basque	(15.43%, 21.05%)	(1064, 152, 305)
MultiBooked_ca (Barnes et al., 2018)	Catalan	(14.65%, 16.17%)	(1174, 168, 336)
OpeNER_es (Agerri et al., 2013)	Spanish	(12.93%, 15.53%)	(1438, 206, 410)
OpeNER_en (Agerri et al., 2013)	English	(19.72%, 20.48%)	(1744, 249, 499)
MPQA (Wiebe et al., 2005)	English	(77.92%, 79.18%)	(5873, 2063, 2112)
Darmstadt_unis (Toprak et al., 2010)	English	(69.77%, 64.66%)	(2253, 232, 318)

Table 1: Dataset Statistics

```
{
    "sent_id": "../example/dataset/train/book/demo_sample",

    "text": "I would not suggest this book .",

    "opinions": [
            {
                "Source": [["I"], ["0:1"]],
                "Target": [["this book"], ["20:29"]],
                "Polar_expression": [["would not suggest"], ["2:19"]],
                "Polarity": "Negative",
                "Intensity": "Average"
            }
        ]
}
```

Figure 1: Annotation Format. The value of "opinions" in the JSON is a list of opinion tuples present in the "text". Each item in the list is a dictionary, with keys being the tuple elements, and the values corresponding to their representation in the text. *Source, Target* and *Polar_expression* are annotated with the the actual word spans appearing in the text along and their respective character indices. *Polarity* represents the sentiment expressed in the tuple, and *Intensity* represents its strength.

trained to generate a sequence of tuples, each consisting of seven integer outputs corresponding to the start and end indices of the holder, target and sentiment expression spans appearing in the text, and finally the polarity class. An example is shown and described in Figure 2.

We participate in the SemEval 2022 Task 10: Structured Sentiment Analysis (Barnes et al., 2022) hosted on CodaLab. In order to demonstrate the efficacy of our proposed solution, we attempt both the *monolingual* and *cross-lingual* subtasks and achieve competitive performance on the leaderboard in both settings. As part of the (sub)tasks, we testify our approach on multiple datasets spanning across five different languages - *English* (Darmstadt_unis, OpeNER_en, MPQA), Basque (MultiBooked_eu), *Catalan* (MultiBooked_ca), *Norwegian* (NoReC_fine) and *Spanish* (OpeNER_es). A summary of dataset statistics is reported in Table 1. While the evaluation scripts were made available

to us by the task organizers to analyze our performance, the final leaderboard scores were obtained on a hidden test set.

2 Task Overview

2.1 Task Definition

SSA aims to predict all the structured sentiment graphs present in a given text. A graph is formally represented by opinion tuples $O = O_1, O_2, ..., O_n$, where each opinion tuple O_i consists of a quadruple of the holder h, the target t, the sentiment expression e, and the sentiment polarity p.

2.2 Datasets

As summarized in Table 1, we are provided with a total of 7 datasets, as part of the shared task, spanning across 5 different languages. Each dataset is a collection of sentences, along with their corresponding annotated opinion tuples, each consisting of (*Source, Target, Polar Expression*, and *Polarity*).
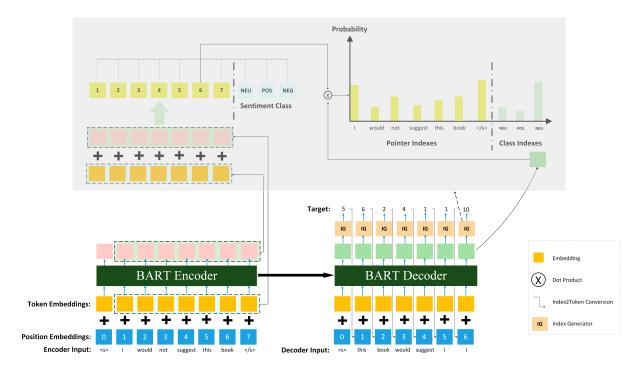
Figure 2: Model Architecture. The above figure shows an example where the input is "<s> I would not suggest this book </s>", and the corresponding output is "5, 6, 2, 4, 1, 1, 10" (Only partial decoder sequence is shown. Here, 7 (</s>) should be the next generation index). The "Index2Token Conversion" module converts the pointer indices back to the corresponding tokens in the source text, and the class index to the corresponding sentiment polarity.

While the *Intensity* of the expressed sentiment is also provided as part of the annotations, intensity classification/regression is not included as part of the task. An example is shown in Figure 1. All the data is provided through CodaLab, and GitHub.

3 Methodology

3.1 Task Formulation

We take a generative approach to formulate SSA as a structured prediction problem. We note here that predicting the holder (source), target (aspect), and polar expression spans correspond to extraction tasks, whereas sentiment polarity prediction is a classification task. Following (Yan et al., 2021), we model both these tasks in a unified generative framework by representing span entities with their start and end pointer indices corresponding to the text, and sentiment polarity with a class index.

We denote the holder, target, polar expression and sentiment polarity as h, t, pe, and sp respectively. The start and end index of each term is represented using superscripts s and e. For an input $X = [x_1, ..., x_n]$, where x_i is the ith word in the text, the target $Y = [t_i^s, t_i^e, pe_i^s, pe_i^e, h_i^s, h_i^e, sp_i, ...]$ is defined as a sequence of tuples, each consisting of seven indices corresponding to an opinion tuple

(h, t, pe, sp). An example sentence along with its target sequence are shown in Figure 2.

3.2 System Overview

Our model, as shown in Figure 2, consists of an encoder-decoder architecture with BART (Lewis et al., 2019) as its backbone. Given an input $X = [x_1, ..., x_n]$, the model is trained to produce an output $Y = [y_1, ..., y_m]$ (with y_0 representing the start-of-sequence token, $< s >$). The probability distribution is modeled as:

$$P(Y|X) = \prod_{t=1}^{m} P_t \qquad (1)$$

Here $P_t = P(y_t|X, Y_{<t})$ represents the index probability distribution for the tth time step.

3.2.1 Encoder

BART comprises of a bi-directional encoder. We denote the encoded vector of the input sentence X as H^e. For the sake of simplicity, we ignore the start-of-sequence token ($< s >$) in the equations.

$$H^e = BARTEncoder([x_1, ..., x_n]) \qquad (2)$$

Here $H^e \in R^{n \times d}$, with d as the hidden dimension.

Dataset	% Null Instances
NoReC_fine	47.24% (As in the dataset)
MultiBooked_eu	10%
MultiBooked_ca	14.65 % (As in the dataset)
OpeNER_es	4%
OpeNER_en	14%
MPQA	50%
Darmstadt_unis	69.77 % (As in the dataset)

Table 2: % null instances in processed train sets.

3.2.2 Index2Token Conversion

Since the entity spans are decoded as corresponding start and end indices, and the sentiment polarity is decoded as corresponding class index, the indices need to be converted back to tokens before the BART Decoder can use them along with the encoder hidden state H^e for generating the next token (index) in the tth time step. For each $y_t \in Y_{<t}$, we therefore use the following conversion strategy:

$$\hat{y}_t = \begin{cases} X_{y_t} & \text{if } y_t \text{ is a pointer index,} \\ Pol_{y_t - n} & \text{if } y_t \text{ is a class index} \end{cases} \quad (3)$$

where $Pol = [p_1, p_2, p_3]$ is the list of polarity classes. In our implementation, $y_t \in [1, n+3]$. The first sentence token x_1 has the pointer index 1.

3.2.3 Decoder

Our BART decoder now uses H^e and the converted decoder outputs $\hat{Y}_{<t}$ to obtain the tth decoder hidden state:

$$H_t^d = BARTDecoder(H^e, \hat{Y}_{<t}) \quad (4)$$

where $H_t^d \in R^d$. Finally, H_t^d is used to predict the token probability distribution P_t. We request our readers to refer to (Yan et al., 2021) for additional details.

3.2.4 Training and Workflow

Teacher forcing with negative log likelihood as the loss function is used to train the model. During inference, beam search is used to generate the target sequence Y in an auto-regressive manner. Finally, the generated sequence is translated back into the phrase spans and sentiment polarity. As shown in Figure 2, we now illustrate the working of our proposed method using an example sentence:
I would not suggest this book .

1. The input <s> I would not suggest this book </s> is sent as input to the BART Encoder.

As specified in Section 3.2.2, the word "I" is mapped to position index 1. Accordingly, </s> is mapped to index 7. Thereafter, each sentiment polarity is assigned a class index in sequence. In this case, the polarity values neutral, positive, and negative, are assigned class indices 8, 9 and 10 respectively.

2. The BART Decoder is trained to generate a sequence of indices till the end-of-sequence index (here 7) is generated. Corresponding to each opinion tuple, the decoder respectively predicts the start and end word indices for the target, polar expression, and source and finally, the polarity class index.

3. In our case, the expected target sequence is 5, 6, 2, 4, 1, 1, 10, 7. Here, (5, 6) represents the *target* phrase "this book", (2, 4) represents the *polarity expression* phrase "would not suggest", (1, 1) represents the *holder* phrase "I", and 10 represents the *negative* polarity class.

4. During inference, a decoding algorithm is employed making use of the *Index2Token Conversion* module to respectively convert the indices back to the text tokens and polarities before presenting to the end user.

4 Experiments

4.1 Data Preprocessing

In addition to fully annotated sentences, we observe the following two kinds of instances in all the datasets: (a) no opinion tuples at all - hereby referred to as the null examples, and (b) few empty entities in a single opinion tuple. We note here that our tuple representation scheme expects position indices of words appearing in the sentence. In order to accommodate for the above mentioned cases, we add a string *None* in the beginning of every sentence. This helps us to map the missing entities (e.g. holder or aspect in an opinion tuple) to a phrase present in the sentence.

After rigorous experiments, we set an optimal threshold for the proportion of null examples to be used for training in each of the datasets as reported in Table 2. During training, we found that limiting the proportion of null examples to the reported values significantly helped us in achieving the best performance on the respective datasets across monolingual and cross-lingual settings.

Dataset	% null	Batch Size	LR	F1 Score	Best Epoch
NoReC_fine	10	16	1E-04	0.320	35
NoReC_fine	10	16	2E-05	0.310	8
NoReC_fine	10	16	5E-05	0.323	11
OpeNER_es	10	8	1E-04	0.351	33
OpeNER_es	10	8	2E-05	0.482	36
OpeNER_es	10	8	5E-05	0.566	33
OpeNER_en	10	8	1E-04	0.674	43
OpeNER_en	10	8	2E-05	0.661	32
OpeNER_en	10	8	5E-05	0.675	7
Darmstadt_unis	10	16	1E-04	0.259	30
Darmstadt_unis	10	16	2E-05	0.276	34
Darmstadt_unis	10	16	5E-05	0.289	19

Table 3: Learning Rate Tuning

Dataset	% null	Batch Size	LR	F1 Score	Best Epoch
NoReC_fine	15	8	5E-05	0.317	8
NoReC_fine	30	16	5E-05	0.295	17
NoReC_fine	As in Dataset	16	5E-05	0.357	28
MultiBooked_eu	5	8	5E-05	0.422	20
MultiBooked_eu	10	8	5E-05	0.427	21
MultiBooked_eu	As in Dataset	8	5E-05	0.401	19
MultiBooked_ca	5	8	5E-05	0.552	35
MultiBooked_ca	10	8	5E-05	0.545	32
MultiBooked_ca	As in Dataset	8	5E-05	0.556	46
OpeNER_es	4	8	5E-05	0.572	31
OpeNER_es	8	8	5E-05	0.57	4
OpeNER_es	As in Dataset	8	5E-05	0.571	25
OpeNER_en	7	16	5E-05	0.677	24
OpeNER_en	14	16	5E-05	0.678	25
OpeNER_en	As in Dataset	16	5E-05	0.674	39
MPQA	25	8	5E-05	0.355	20
MPQA	50	16	5E-05	0.366	44
MPQA	As in Dataset	16	5E-05	0.359	42
Darmstadt_unis	25	16	5E-05	0.268	42
Darmstadt_unis	45	16	5E-05	0.277	33
Darmstadt_unis	As in Dataset	16	5E-05	0.312	36

Table 4: Null Parameter Tuning for each dataset

4.2 Experimental Setup

For the *monolingual* setting, we used the train, validation, and test splits of the same datasets. While experimenting on the *English* datasets (Darmstadt_unis, MPQA, and OpeNER_en), we use BART-base[2] as the backbone. For the *Non-English* datasets (OpeNER_es, Multibooked_eu, Multibooked_ca, and NoReC_fine), we use BART-large-MNLI [3] as the backbone. In the *cross-lingual* setting, we trained our models using the combined training data from all *English* datasets and evaluated them on the test sets of respective *Non-English* datasets (NoReC_fine not included as part of this setting). Here, we used BART-large-MNLI as the backbone for all our cross-lingual experiments.

Although predicting the intensities of sentiment polarities was not included as part of the shared task, we hypothesized that additionally learning the intensity prediction task would help the model in predicting the other entities (h, t, e, p) better in a multi-task setting. We performed additional experiments to verify our hypothesis. However, we observed little to no difference in the final results. Accordingly, we excluded intensity prediction from further consideration while performing our final experiments. We make our code repository publicly available at https://github.com/Sherlock-Jerry/SSA-SemEval.

4.3 Hyperparameter Tuning

We train all our models on Tesla P100-PCIE 16GB GPU. We perform extensive tuning experiments to

Parameters	English	Non-English
Batch Size	16	8
Learning Rate	5E-05	5E-05
Epochs	50	50
BART Model	Base	Large-MNLI
% Null	Varies with Dataset	

Table 5: Final set of hyperparameters

Dataset	Ours	Barnes et al. (2021b)
NoReC_fine	0.351	0.312
MultiBooked_eu	0.438	0.547
MultiBooked_ca	0.508	0.568
OpeNER_es	0.544	Not available
OpeNER_en	0.626	Not available
MPQA	0.327	0.188
Darmstadt_unis	0.330	0.265

Table 6: Monolingual SubTask: Test Set SG-F1 Scores.

obtain the optimal set of hyperparameters. To determine the optimal learning rate, we ran monolingual experiments on two English and Non-English datasets respectively, as elucidated in Table 3. Based on our observations, we fixed a common learning rate of $5e - 5$ for our final experiments across both the settings, monolingual as well as cross-lingual. For obtaining the optimal proportion of null instances to be used for training the final models, we perform three iterations of monolingual experiments on each dataset, each time with a different proportion of null instances used for training the models, as reported in Table 4. The final null thresholds are reported in Table 2. Table 5 summarizes the set of hyperparameters used for reporting our final results for both the subtasks. For all our experiments, the model selected according to the best F1 score on the validation data was used to evaluate on the test data.

4.4 Evaluation Metrics

Sentiment Graph F1 (SG-F1) is used to evaluate the models. *True Positive* is defined as an exact match (including polarity) at graph-level, weighted by the token-level overlap between the gold and predicted spans for holder, target, and polar expression, averaged across all three spans. *Precision* is calculated by weighting the number of correctly predicted tokens divided by the total number of predicted tokens. *Recall* is calculated by dividing the number of correctly predicted tokens by the number of gold tokens, thereby allowing for empty holders and targets which exist in the gold standard.

5 Results

5.1 Monolingual Subtask

We report the results for our monolingual experiments in Table 6 and compare them with the existing state-of-the-art (SOTA) results reported in Barnes et al. (2021b). Amongst the given datasets, our model performs the best on OpeNER_en and OpeNER_es, and has a relatively poor performance

on MPQA, Darmstadt_unis and NoReC_fine. Despite this, we comfortably outperform the existing SOTA on these datasets. On the public leaderboard hosted on CodaLab, we achieved 18th rank out of 32 entries for this task.

5.2 Crosslingual Subtask

We report the results for our crosslingual experiments in Table 7. In this paradigm, we used all the English datasets for training our model, and tested our best trained models on the test sets of the respective Non-English datasets.

Dataset	SG-F1
EN-EU (MultiBooked_eu)	0.431
EN-CA (MultiBooked_ca)	0.506
EN-ES (OpeNER_es)	0.542

Table 7: Corsslingual SubTask: Test Set SG-F1 Scores.

Here, we achieved 11th rank out of 32 entries.

5.3 Qualitative Analysis

- In the monolingual subtask, we observed that our model performs poorly on the datasets with large proportions of empty opinion tuples (null instances). As can be confirmed from Table 2, MPQA, Darmstadt_unis, and NoRec_fine have high empty tuple proportion as against OpeNER_en, and OpeNER_es with low proportion of null instances.

- We observed that for datasets having lengthy sentences, our model performs relatively poor. A comparison between the distribution of test sentence lengths and the Sentiment Graph F1 scores for each dataset is shown in Figure 3.

- We also observed annotation errors in the datasets. For instance, given the test sentence

Label Type	Source	Target	Polar Expression	Polarity
Gold	-	The size of room	reasonable	Positive
Prediction	-	The size	reasonable	Positive
Gold	-	walls	in very poor conditions	Negative
Prediction	-	walls	very poor	Negative
Gold	-	floor	in very poor conditions	Negative
Prediction	-	floor	in very poor conditions	Negative
Gold	-	ceiling	in very poor conditions	Negative
Prediction	-	ceiling	in very poor conditions	Negative

Table 8: Ground truth opinion tuples and model predictions for the sentence: "The size of room is reasonable , but floor , walls and ceiling are in very poor conditions".
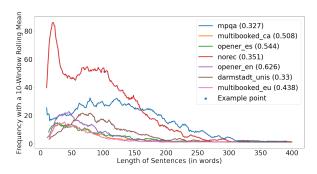

Figure 3: Comparing the distribution of test sentence lengths with best-obtained SG-F1 scores for each dataset. The "Example point" shows that there 51 sentences in the test set for NoRec_fine with length 100.

"So wonderful to see people go to work smiling and leave work still smiling and happy.", our trained generative model correctly predicts an opinion tuple with "see people go to work" as the target, and "So wonderful" as the opinion expression with a "Positive" sentiment. However, no ground truth opinion tuples are associated with the sentence.

- As reported in Table 8, we found a few instances where our model correctly predicts the necessary entities; but due to ambiguity in labelling (even at human level), we saw a mismatch. Here, our model predicts "The size" as the *target* whereas the gold standard expects "The size of room". Similar is the case with "in very poor conditions" (gold standard) versus the predicted phrase "very poor".

6 Related Work

Previous efforts on SSA have primarily focused on two approaches: sequence labeling-based (He et al., 2019), and graph-based (Barnes et al., 2021b). The corresponding scores for both these approaches are considered as baselines by the task organizers.

6.1 Sequence Labelling

In this approach, (He et al., 2019) propose a pipeline of sequence labelling and relation classification tasks. More specifically, three different sequence labellers based on BIOES tags are trained to predict the three span-based opinion entities, i.e. the holder, the target, and the polar expression. Finally, their relationship is exploited using a separate classification layer on top to predict the connecting sentiment polarity. However, such an approach inherently suffers from error propagation between the steps. Also, the inter-dependency especially between the target and the polar expression is not captured when the spans are predicted in isolation.

6.2 Dependency Graph Parsing

(Barnes et al., 2021b) have treated this task as a bilexical dependency graph prediction problem. They present two different versions of their proposed approach - (a) head-first and (b) head-final, as shown in Figure 4.
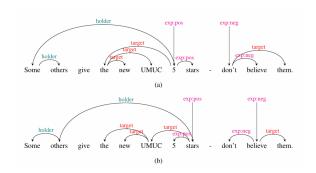

Figure 4: Dependency Graph Parsing

In both cases, the sentiment expression is considered as the root node, and the other elements are connected via arcs that represent their relationships. This approach builds upon the Dozat and Manning parser, implemented in (Kurtz et al., 2020).

7 Conclusion

Different from prior methods, we in this work present a novel generative approach to tackle the task of Structured Sentiment Analysis. We formulate the task as a structured prediction problem. Our BART-based encoder-decoder architecture is trained to predict a sequence of indices corresponding to each opinion tuple present in the text. The generated indices suitably represent the holder, target, and polar expression spans by their start and end token positions, and the sentiment polarity by its corresponding class. As part of SemEval 2022 Task 10, we participate in both monolingual and crosslingual subtasks, and achieve competitive performance on the leaderboard for both settings. In future, we would like to explore paraphrasing-based generative methods for the task.

References

Rodrigo Agerri, Montse Cuadros, Sean Gaines, and German Rigau. 2013. OpeNER: Open polarity enhanced named entity recognition. In *Sociedad Española para el Procesamiento del Lenguaje Natural*, volume 51, pages 215–218.

Jeremy Barnes, Toni Badia, and Patrik Lambert. 2018. MultiBooked: A corpus of Basque and Catalan hotel reviews annotated for aspect-level sentiment classification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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

Jeremy Barnes, Robin Kurtz, Stephan Oepen, Lilja Øvrelid, and Erik Velldal. 2021b. Structured sentiment analysis as dependency graph parsing. 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 3387–3402, Online. Association for Computational Linguistics.

Jeremy Barnes, Andrey Kutuzov, Laura Ana Maria Oberländer, Enrica Troiano, Jan Buchmann, Rodrigo Agerri, Lilja Øvrelid, Erik Velldal, and Stephan Oepen. 2022. SemEval-2022 task 10: Structured sentiment analysis. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, Seattle. Association for Computational Linguistics.

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 340–350, Online. Association for Computational Linguistics.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 504–515, Florence, Italy. Association for Computational Linguistics.

Robin Kurtz, Stephan Oepen, and Marco Kuhlmann. 2020. End-to-end negation resolution as graph parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 14–24, Online. Association for Computational Linguistics.

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*.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Rajdeep Mukherjee, Tapas Nayak, Yash Butala, Sourangshu Bhattacharya, and Pawan Goyal. 2021a. PASTE: A tagging-free decoding framework using pointer networks for aspect sentiment triplet extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9279–9291, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rajdeep Mukherjee, Shreyas Shetty, Subrata Chattopadhyay, Subhadeep Maji, Samik Datta, and Pawan Goyal. 2021b. Reproducibility, replicability and beyond: Assessing production readiness of aspect based sentiment analysis in the wild. In *Advances in Information Retrieval*, pages 92–106, Cham. Springer International Publishing.

Lilja Øvrelid, Petter Mæhlum, Jeremy Barnes, and Erik Velldal. 2020. A fine-grained sentiment dataset for Norwegian. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5025–5033, Marseille, France. European Language Resources Association.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *AAAI*.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014a. 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.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014b. 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.

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, Uppsala, Sweden. Association for Computational Linguistics.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Hang Yan, Junqi Dai, Xipeng Qiu, Zheng Zhang, et al. 2021. A unified generative framework for aspect-based sentiment analysis. *arXiv preprint arXiv:2106.04300*.