Hitachi at SemEval-2022 Task 10: Comparing Graph- and Seq2Seq-based Models Highlights Difficulty in Structured Sentiment Analysis

Gaku Morio, Hiroaki Ozaki, Atsuki Yamaguchi, and **Yasuhiro Sogawa**
Research and Development Group, Hitachi, Ltd.
Kokubunji, Tokyo, Japan
{gaku.morio.vn, hiroaki.ozaki.yu,
atsuki.yamaguchi.xn, yasuhiro.sogawa.tp}@hitachi.com

Abstract

This paper describes our participation in SemEval-2022 Task 10, a structured sentiment analysis. In this task, we have to parse opinions considering both structure- and context-dependent subjective aspects, which is different from typical dependency parsing. Some of the major parser types have recently been used for semantic and syntactic parsing, while it is still unknown which type can capture structured sentiments well due to their subjective aspects. To this end, we compared two different types of state-of-the-art parser, namely graph-based and seq2seq-based. Our in-depth analyses suggest that, even though graph-based parser generally outperforms the seq2seq-based one, with strong pre-trained language models both parsers can essentially output acceptable and reasonable predictions. The analyses highlight that the difficulty derived from subjective aspects in structured sentiment analysis remains an essential challenge.

1 Introduction

SemEval-2022 Task 10 (Barnes et al., 2022) aims at extracting structured sentiment from a given sentence. Different from other sentiment analysis tasks, structured sentiment analysis is formulated as an information extraction problem with at least three elements, namely a holder, a target and a sentiment expression. The shared task has two subtasks. In Subtask 1 (monolingual), we evaluate the performance on seven monolingual corpora, i.e., `MPQA` (Wiebe et al., 2005), `OpeNER`, `OpeNER_es` (Agerri et al., 2013), `DSu` (Toprak et al., 2010), `MultiB_ca`, `MultiB_eu` (Barnes et al., 2018) and `NoReC` (Øvrelid et al., 2020). In Subtask 2 (crosslingual), we evaluate the zero-shot prediction performance on three non-English corpora, i.e., `OpeNER_es`, `MultiB_ca` and `MultiB_eu`, generated by a model trained with an English corpus. One significant difference between syntactic parsing and structured sentiment analysis is that

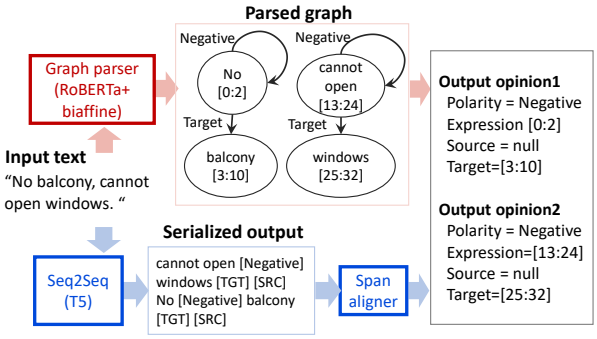

Figure 1: Overview of the two parsers: (Top) Graph and (Bottom) Seq2Seq.

the former is based on certain well-defined rules, whereas the latter forms an information structure on the basis of subjective opinions. Given the nature of the sentiment structure, Barnes et al. (2021) formulated the problem as a dependency parsing task.

Our motivation here is to throw light upon how subjective opinions affect parsing performance and how well recent strong parsing models can capture them. To this end, we compare two types of parsing model: graph- and generation-based[1] models. Though these parsers have shown competitive performance under various conditions (Oepen et al., 2020; Ozaki et al., 2020; Samuel and Straka, 2020), each of them has its own advantages and disadvantages. Graph-based parsers (McDonald et al., 2005) directly model token-to-token relations; therefore, they are suitable for modeling structured sentiment with surface anchors. On the other hand, generation-based parsers output a sequence to reconstruct the graph structure of the target meaning representation. Recent advances in deep neural networks have allowed us to generate a serialized graph directly from an input sentence by using pre-trained generation models (Ozaki et al.,

[1]We regard a transition-based parser (Dyer et al., 2015) to be a kind of seq2seq-based parser, as it generates an action sequence based on its states.

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1349 - 1359

2020; Procopio et al., 2021). Specifically, we focus on state-of-the-art seq2seq-based models like T5 (Raffel et al., 2020). Because pre-trained generation models (e.g. text summarization models) have been trained to generate a summarized statement from a given input sentence, it may be easier for the generation-based parsers to model semantic relations including subjective opinions than the graph-based ones.

In this paper, we briefly introduce our approaches together with a comparison of the two different parsers: (i) **Graph**: We use a model that jointly solves two different problems, namely span identification, and relation extraction of spans. (ii) **Seq2Seq**: We design a serialization for sentiment structures generated by our seq2seq-based parser. We simply use pre-trained text generation models and fine-tune them with the serialized sentiment structures.

Experiments showed that Graph and Seq2Seq achieved reasonable levels of performance on both subtasks. Our submitted systems are based on Graph and ranked third in both the monolingual and cross-lingual tasks. We further conducted in-depth analyses for the two parsers. Our findings are twofold:

1. Multilingual performance tends to depend more on the type of pre-trained model used than on the model architecture, while Graph outperformed Seq2Seq in non-English corpora.

2. Graph is somewhat recall-focused, whereas Seq2Seq is more precision-focused in specific corpora.

The first finding suggests that Graph with a strong pre-trained model has an advantage for multilingual training when compared to Seq2Seq. Given that mT5 (Xue et al., 2021) is not trained on any supervised tasks such as translation or summarization, it may be difficult for multilingual Seq2Seq to generate (and possibly copy) sentiment tokens from the input text. Other considerations such as the decoder architecture and training time of Seq2Seq seem to favor Graph.

Regarding the second finding, we found that Graph performs well on complex structured opinions. However, on the other side of the coin, it suggests that Graph sometimes causes over-detection (though these over detected opinions may be acceptable to humans). We found that this happens particularly for MPQA polar expressions. This could be due to the nature of MPQA, which usually includes context-dependent expressions.

As a result, we argue that it is difficult to decide which type of parser is better because: (i) the decision criteria rely on how we define the sentiment structure, e.g., structured sentiments in some corpora are semantically complex and context-dependent, while those in other corpora are not, making the corpus less context-dependent. (2) Whether to use Graph or Seq2Seq depends on whether we want to cover as much of the structured sentiments as possible or whether we want to emphasize precision. (3) In regard to metric-dependent choices, Seq2Seq is at a disadvantage in the shared task because the metric requires anchoring to the surface of the input text. We also have to decide which parser to use from various other perspectives, such as (4) whether or not we are targeting English, and (5) whether the training speed is important. These considerations make structured sentiment analysis challenging.[2]

2 Models

This section explains the Graph and Seq2Seq methods used in this work.

2.1 Graph model

We formulate the problem as a joint task of span identification and relation extraction. This approach can be classified as a graph-based approach (Dozat et al., 2017; Falenska et al., 2020), which is known to perform well in fields such as syntactic dependency parsing. Although there are various approaches to this problem, we simply use the architecture of Morio et al. (2022). The architecture generates BIO tags to predict spans using pre-trained language models such as Longformer (Beltagy et al., 2020). The span representation of each predicted span is generated by average pooling of the predicted span and subsequently fed into biaffine classifiers (Dozat and Manning, 2017) to predict relations.

Because the architecture was originally designed to predict argument structures (Lawrence and Reed, 2019), so-called argument mining, it needs a little tweaking. We thus designed a dedicated encoding for structured sentiment analysis. As shown in Figure 1 (Top), the opinions for the input text can

[2]We plan to release our code at https://github.com/hitachi-nlp/graph_parser

be represented as a graph. We represent a representative polarity expression as a root node, and its polarity as a self-loop (e.g., a *no* node and its negative self-loop). We represent the other polarity expressions, target and source via child nodes linked to the polarity nodes (e.g., *balcony* is a child node for the *no* node). Overlapping spans are divided up using the start and end indexes of the overlap, and each span can have multiple labels (e.g., *Positive_Source* is a combined label for positive and source spans). Although this graph encoding cannot fully represent structured sentiments[3], we confirmed that the data reconstructed from the graph encoding achieves about 99% F-score (so there is practically no problem.) We convert all the given datasets using this graph encoding, and given a text input, the model is trained to parse the graph.

2.2 Seq2Seq model

Seq2Seq Generation Here, we formulate the problem as a summarization task to output serialized tuples of structured sentiment. We preferably utilize pre-trained summarization language models, such as T5 (Raffel et al., 2020), and fine-tune them with the serialized tuples.

Serialization system A seq2seq model outputs serialized tuples of structured sentiment regardless of their position on the surface (see Figure 1). For example, when an input text is "No balcony, cannot open windows", we have two tuples of structured sentiment; each of them is serialized as follows:

```
No[NEG] balcony[TGT] [SRC]
cannot open[NEG] windows[TGT] [SRC]
```

We serialize a tuple into a polar expression, a target, and a holder order. The polar expression ends with `[NEG]`, `[NEU]` or `[POS]` tokens according to its polarity. The target and holder expressions end with `[TGT]` and `[SRC]`, respectively. If multiple spans are in a tuple, we concatenate them with a special separator (`[SEP]`) token. Since each expression is marked with these special tokens, we can simply concatenate all tuples without confusing them with each other.

Reconstruction from serialization Although our serialization preserves semantically sufficient information on sentiment structures, there is a piece of missing information, i.e., anchors. To reconstruct the anchor information, we utilize a word aligner based on a pre-trained language model: SimAlign (Jalili Sabet et al., 2020). Because SimAlign provides a zero-shot alignment model, we utilize it for obtaining the alignment between the input text and its serialized sentiment structure. For ease of explanation, we define *span* $s = [t_i, t_{i+1}, \ldots, t_{i+j}]$ as a single phrase appearing as a holder, a target, or a polar expression, where t is a generated token. First, we pick a single opinion; then, we extract all tokens in the opinion and form a token sequence $[t_1, \ldots, t_n]$. Second, we calculate the token alignments $[a_1, \ldots, a_n]$ between the input text and the token sequence to point out the corresponding tokens in the input, where $a > 0$ and $a \in \mathbb{N}$. Lastly, we recover the span $s = [t_i, \ldots, t_{i+j}]$ location by $[\min(a_i, \ldots, a_{i+j}), \max(a_i, \ldots, a_{i+j})]$ [4].

Furthermore, we add a heuristic procedure to improve the reconstruction accuracy. When an expression extracted from a reconstructed anchor is different from its original generated expression, we apply a greedy search to find the part where the original expression appears *as is* in the input text. If we find the part, we use an anchor that points to the part instead of the reconstructed anchor. With this heuristic, we achieved an F-score of around 97% between gold and reconstructed opinions from the serialized outputs[5].

3 Experiments

3.1 Experimental setup

Implementation details We implemented Seq2Seq and Graph with PyTorch (Paszke et al., 2019) and the Huggingface Transformers library (Wolf et al., 2020). All models were trained with a fixed number of steps (about 10,000 steps). We used a learning rate of 2e-5 for Graph and 5e-5 for Seq2Seq, with a warmup (Howard and Ruder, 2018) ratio of 0.1. The batch size was set to 16 for Graph and 32 for Seq2Seq. We set the beam width to 5 for Seq2Seq. We did not conduct hyperparameter tuning or model selection and did not use any development data during training and validation.

Pre-trained models To fully utilize the representative power of Graph and Seq2Seq, we used

[3] Our graph encoding cannot distinguish a case where one polar expression forms a single opinion with multiple targets from a case where a polar expression forms multiple opinions with a single target.

[4] Because actual anchors are character-level, we need to convert the spans from the token level to the character level.

[5] Without this heuristic, the F-score is about 91%.

Dev		DSu	OpeNER	MPQA	OpeNER_es	MultiB_eu	MultiB_ca	NoReC
		en	en	en	es	eu	ca	no
Graph	RoBERTa-large	**38.6**	**72.3**	**42.8**				
	InfoXLM-large				**71.1**	**64.7**	**71.0**	**50.7**
Seq2Seq	T5-large	38.0	69.4	42.3				
	mT5-large				66.0	61.4	67.9	48.8
Test								
Graph	RoBERTa-large	42.2	75.0	39.3				
	+ensemble (submitted)	**46.3**	**75.6**	40.2				
	InfoXLM-large				71.7	70.5	69.8	51.2
	+ensemble (submitted)				**73.2**	**71.5**	**70.9**	**53.3**
Seq2Seq	T5-large	40.5	67.1	**40.9**				
	mT5-large				65.6	66.2	65.5	48.0
Best team		49.4	76.0	44.7	72.2	73.9	72.8	52.9

Table 1: Evaluation results of Subtask 1 (monolingual) in Sentiment Graph F1 (SF_1) for the development and test sets. Graph and Seq2Seq represent graph-based and seq2seq-based parsers, respectively. We submitted the InfoXLM-large+ensemble model in the evaluation phase. Note that en=English, es=Spanish, eu=Basque, ca=Catalan, and no=Norwegian.

Dev		OpeNER_es	MultiB_eu	MultiB_ca
		en→es	en→eu	en→ca
Graph	InfoXLM-large	**62.8**	**46.2**	**62.3**
Seq2Seq	mT5-large	56.9	41.3	53.5
Test				
Graph	InfoXLM-large	61.9	51.6	60.1
	+ensemble (submitted)	**62.8**	**52.7**	**60.7**
Seq2Seq	mT5-large	57.4	44.7	53.5
Best team		64.4	63.2	64.3

Table 2: Evaluation results of Subtask 2 (crosslingual zero-shot) in SF_1. We submitted the InfoXLM-large+ensemble version in the evaluation phase.

pre-trained language models based on Transformer (Vaswani et al., 2017). For Graph, we used RoBERTa-large (Liu et al., 2019) in Subtask 1 (monolingual) and InfoXLM-large (Chi et al., 2021) in Subtask 2 (cross-lingual). RoBERTa is a well-tuned model based on BERT (Devlin et al., 2019), and it has shown state-of-the-art performance in various classification tasks. InfoXLM is a recently proposed model, which is pre-trained with contrastive learning. For Seq2Seq, we used T5-large (Raffel et al., 2020) in Subtask 1 and mT5-large (Xue et al., 2021) in Subtask 2. T5 uses a unified text-to-text framework to deal with various text-based tasks.

Submitted models In our preliminary experiments, we found that the development scores of the monolingual Graph models were slightly better than those of the Seq2Seq ones (the reasons will be discussed later). Thus, we only used Graph models for our submission. However, to discuss Graph and Seq2Seq in detail, we show the results of both Graph and Seq2Seq below.

3.2 Main results

Table 1 shows the overall results of Subtask 1 (i.e., monolingual), including the scores for the development and test data in Sentiment Graph F1 (SF_1) (Barnes et al., 2021). We tried three different seeds for each model to minimize the effects of random seeds and averaged the scores. For the ensemble methods, the scores are those from the ensemble of models with the three seeds. For reference, we also include the results of the best-performing team for the test data. Overall, Graph mostly outperformed Seq2Seq with significant differences observed in non-English corpora, such as OpeNER_es and MultiB_eu. This suggests that Graph has an advantage for multilingual training when compared to Seq2Seq. That is, while T5 is trained on summarization and its related tasks, mT5 is not. This difference might cause a disadvantage

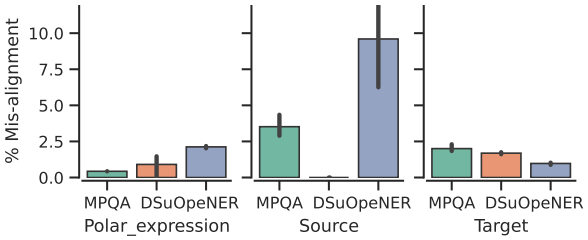

Figure 2: Misalignment ratio of Seq2Seq, where misalignment means Seq2Seq predicted correct surface tokens, but the aligned spans for the original text were incorrect.

wherein Seq2Seq generates (and possibly copies) sentiment tokens from the input text.

Although not surprising, Table 1 indicates that the ensemble models of Graph outperform the non-ensemble ones. The submitted ensemble models have comparable SF_1 to those of the best team on the test set of some corpora.

Subtask 2 In this subtask, we used `OpeNER` English for training and conducted zero-shot prediction for `OpeNER_es`, `MultiB_eu`, and `MultiB_ca`. Table 2 shows SF_1 of Subtask 2 (i.e., cross-lingual zero-shot). Overall, it seems that Graph is still a better choice than Seq2Seq. There is a significant difference in `MultiB_eu`, i.e., Seq2Seq (44.7) and Graph (51.6). This difference may be due to the pre-trained language models (i.e., InfoXLM vs. mT5) and the misalignment problem of Seq2Seq. On the other hand, Graph and Seq2Seq performed poorly compared with the top-performing team, so neither model seems suitable for the zero-shot setting.

3.3 Analysis and Discussion

This section compares Seq2Seq and Graph and points out that one is not superior to the other. Our argument is supported by an analysis of the alignment errors of Seq2Seq and the structural/semantic properties of the parsers. To simplify the discussion, we focus only on non-ensemble and monolingual models for the English corpora (i.e., `MPQA`, `DSu` and `OpeNER`).

3.3.1 Does Graph really outperform Seq2Seq?

Alignment error in Seq2Seq A major drawback of Seq2Seq is the alignment error caused by the aligner. That is, Seq2Seq can produce the correct surface tokens of the polar expression, source or target, but the aligner may align incorrect spans

Figure 2 shows the misalignment ratio (i.e., the ratio of predicted elements where the surface tokens generated by Seq2Seq are correct, but the spans generated by the aligner are incorrect). There is a certain amount of alignment error in `MPQA`, `DSu` and `OpeNER`. We can see that the ratio of `OpeNER` is larger than those of the others. We explain this in Table 3. In case #1, a polar expression *chic!* was correctly predicted, but the aligner did not include the exclamation mark. In #2, *minutes from numerous...* was correctly predicted, but the beginning word *minutes* was unfortunately not included in the span. We found that the large misalignment ratio of the source in `OpeNER` was caused by pronoun words, as illustrated in #3, where our system could not resolve which *I* (i.e., the former or the latter in the two *I* tokens) to align. Case #4 shows a similar phenomenon for the term *hate*.

If we had remedied some of the misalignments, Seq2Seq would have produced 41.8 SF_1 and 42.6 SF_1 on the test data of `DSu` and `MPQA`, respectively. These results are comparable or better than those of the Graph models shown in Table 1; thus, we can not simply conclude that Graph is better than Seq2Seq. Moreover, Seq2Seq might be the best choice for an evaluation metric that does not emphasize anchoring spans of the input sentence (which may be enough for practical purposes). On the other hand, it seems that Seq2Seq for `OpeNER` still has a significant performance gap against Graph, as shown in Table 1. We focus on this aspect below.

3.3.2 How do Graph and Seq2Seq differ from each other?

Here, we discuss the differences between Graph and Seq2Seq on the basis of structural and semantic complexity. We also discuss the limitations of the parsers.

Complexity of sentiment structure We suppose that the number of opinions in an input text can be used as a proxy metric showing the complexity of the sentiment structure. Figure 3 shows the relationship between the number of opinions in an input sentence and SF_1. Overall, the two parsers exhibit similar trends; that is, the more opinion numbers there are, the harder the prediction becomes. However, the performance of Graph seems to be less dependent on the number of opinions, while Seq2Seq generally exhibits a negative trend, especially for `OpeNER`. These results suggest that Graph is a good choice for handling complicated

#	Corpus	Text	Gold span (and its text)	Generated surface by Seq2Seq	Mis-aligned span (and its text)
1	OpeNER	hotel chic !	[6:12] (chic !)	chic!	[6:10] (chic)
2	OpeNER	It is minutes from numerous restaurants , bars , etc. and centrally located between the Prado and the Palace · · great for walking .	[6:53] (minutes from numerous restaurants, bars, etc.)	minutes from numerous restaurants, bars, etc.	[14:53] (from numerous restaurants , bars , etc.)
3	OpeNER	I was forced to stay in this area due to my business reason , but oterwise I would not suggest to come here to spend your holidys .	[75:76] (I)	I	[0:1] (I)
4	MPQA	" We do n't hate the sinner , " he says , " but we hate the sin . "	[51:55] (hate)	hate	[12:16] (hate)

Table 3: Case study of misalignments where Seq2Seq produced correct surface tokens but the reconstruction system aligned incorrect spans.

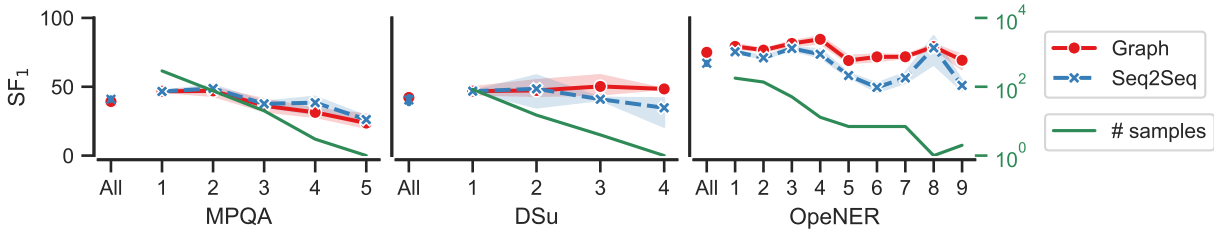

Figure 3: Relationship between the number of opinions included in a sentence (X-axis) and SF_1 (Y-axis). The green line shows the total number of samples in the evaluation set; it is evident that the larger the number of opinions is, the fewer the number of samples is. Note that *All* indicates the evaluation for full sentences, for reference.

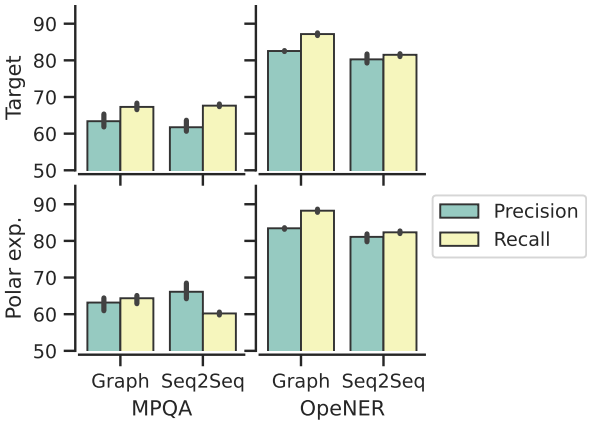

Figure 4: Precision and recall of Graph and Seq2Seq for target (top) and polar expression (bottom) outputs.

structures.

However, Graph has another disadvantage: over-detection. Figure 4 shows the precision and recall values for target and polar expression in MPQA and OpeNER. As shown, there is a possible trend that Graph is more recall-focused, and Seq2Seq is more precision-focused. In other words, Graph may over-detect opinion fragments for specific corpora. Case #1 in Table 4 shows an example of over-detection. The over-detected phrases are a *means* of "threatened" and may not be a target.

However, we find that, even for complex structures with multiple opinions, both Graph and Seq2Seq produce reasonable predictions. In case #1 in Table 4, the *means* of "threatened" is semantically a target; thus, this would be an acceptable case. Case #2 shows a long polar expression which does not appear in the gold standard, i.e., *let alone the 4 stars*, while the prediction seems to be correct in a sense. Through these observations, both Graph and Seq2Seq seem to try to output plausible outputs.

Semantic complexity Here, we begin by focusing on the difference between corpora (i.e., MPQA and OpeNER), because it is difficult to evaluate the semantic complexity directly. MPQA is annotated on the basis of the private state frame and distinguishes subjective information from material presented as fact (Wiebe et al., 2005). This makes MPQA semantically complex and context-dependent. On the other hand, OpeNER project originally focuses on lexicon creation (Agerri et al., 2013), making the corpus probably less context-dependent.

Again, let us investigate the precision and recall in Figure 4. A significant score gap between precision and recall in the figure can be found in

#	Corpus	Text	Method	Polarity	Polar exp.	Source	Target
1	MPQA	But after the Chinese side released all the US crew members , major US political figures changed their stance immediately and threatened to use human rights , trade , and Olympics hosting issues to " retaliate " against China .	Gold	Negative	threatened	major US political figures	China
			Graph	Negative	threatened	major US political figures	1. China, 2. trade , and Olympics hosting
			Seq2Seq	Negative	threatened	major US political figures	China
2	OpeNER	I would never ever come back to this hotel even if they paid me. simply it 's not worth the money , let alone the 4 stars .	Gold	Negative	would never ever come back	I	this hotel
				Negative	not worth		the money
			Graph	Negative	would never ever come back	I	this hotel
				Negative	not worth		the money
			Seq2Seq	Negative	never ever come back	I	this hotel
				Negative	not worth	I	the money
				Negative	let alone the 4 stars		it
3	MPQA	AOL would never have existed if it had been founded here , I am sure , since its employees would have been mocked into obscurity by the digerati .	Gold	Negative	mocked into obscurity	digerati	its employees
			Graph	Negative	mocked	the digerati	its employees
			Seq2Seq	⊥	⊥	⊥	⊥
4	MPQA	In the complaint , Hobeika had not yet been called by name .	Gold	Negative	the complaint		
			Graph	Negative	complaint		
			Seq2Seq	⊥	⊥		
5	DSu	I had a programming class with no lectures which is always fun for beginners to try to learn concepts without any sort of interactivity .	Gold	Negative	1.always, 2.fun for beginners	everyone	no lectures
			Graph	⊥	⊥	⊥	⊥
			Seq2Seq	⊥	⊥	⊥	⊥

Table 4: Case study of outputs by graph-based and seq2seq-based models. Magenta colored text indicates incorrect outputs. For visibility, we have omitted some of the outputs. ⊥ represents a false-negative prediction.

the polar expression of MPQA; i.e., for Seq2Seq, the polar expression's precision of MPQA is quite higher than its recall. Interestingly, the gap was not so large in the polar expression of OpeNER. We presume that this is due the semantic complexity (or context-dependent nature) of MPQA. That is, since Seq2Seq always refers to the context of the input side when generating an output, the output could be more context-dependent, and as a result, Seq2Seq may not output less-confident opinions.

Since it is difficult to test the hypothesis that Seq2Seq may not output less-confident opinions in a statistical manner, we present several case studies. Cases #3 and #4 in Table 4 show errors where Graph predicted correct or incorrect opinions but Seq2Seq did not. In #3, the term *mocked* is a negative word *as is*, and can be predicted only with a lexical perspective. However, the polar expression is located in a fictional speculation in insubstantial text, which may have confused Seq2Seq. In #4, Graph predicted *complaint* as a negative polar expression, since the complaint could be a negative lexicon *as is*, while Seq2Seq did not output any opinions. Considering the context in the entire article, it might be difficult to for Seq2Seq to determine if the complaint is an opinion. In this way, Seq2Seq might enable more context-aware predictions, but may not be suitable for structured sentiment analysis based on lexicons. Graph seems to be good at handling lexicons and context-independent phrases.

Limitation of both parsers Another semantically complex case illustrates an interesting error that neither Graph nor Seq2Seq could parse. Case #5 in Table 4 shows an irony expression that might illustrate a limitation of pre-trained models.

Which is superior? As discussed above, each model has its own advantages depending on the perspective. However, we cannot decide which is better, because the decision criteria rely on how we define the sentiment structure. MPQA defines structured sentiment on the basis of private state frames, while OpeNER focuses more on lexical

↓ Sec. per **step**		DSu	OpeNER	MPQA
		en	en	en
Graph	RoBERTa-large	**0.2**	**0.2**	**0.3**
Seq2Seq	T5-large	2.0	2.0	1.8
↓ Sec. per **epoch**				
Graph	RoBERTa-large	**34.0**	**26.9**	**92.8**
Seq2Seq	T5-large	138.6	105.3	332.1

Table 5: Training speed (in sec.) for each parser.

		SF_1
(Barnes et al., 2021)	mBERT	31.2
(Peng et al., 2021)	mBERT	31.9
PERIN NC (Samuel et al., 2022)	XLM-R-base	39.3
PERIN LE (Samuel et al., 2022)	XLM-R-base	40.4
PERIN OT (Samuel et al., 2022)	XLM-R-base	41.6
Graph (ours)	XLM-R-base	**44.8**

Table 6: Comparison with state-of-the-art methods for NoReC test data. Note that NC, LE, and OT mean the node-centric, labeled edge, and opinion-tuple variants of PERIN.

semantics. This would make a difference with syntactic parsing, which is based on a certain type of well-defined grammar, and would cause the parsing performance to be lower than that of syntactic parsers. To cover the variety of definitions of structured sentiment, an abstraction of the sentiment structure might be needed, as is done in abstract meaning representation (AMR; Banarescu et al. 2013).

3.3.3 Energy efficiency approximated by training time

Finally, we discuss the energy efficiency of Graph and Seq2Seq as an estimate of their financial and environmental impact (Strubell et al., 2019). We evaluated the approximated energy consumption in terms of the training time on NVIDIA V100 GPUs. Table 5 shows step-normalized and epoch-normalized training speed in seconds, given that a different batch size was used for Graph and Seq2Seq. As shown, Graph is usually faster than Seq2Seq on all English corpora. We suppose this is because Seq2Seq has a decoder part, which has a computational cost. The results suggest that Graph is preferable in terms of energy efficiency.

4 Related work and state-of-the-art

Sentiment analysis, such as aspect-based sentiment analysis (Chen and Qian, 2020), is a popular research area in natural language processing. Most recently, we have seen attention focusing on parsing full representations of sentiment from text, i.e., structured sentiment analysis (Barnes et al., 2021), as we have tackled in this study. Most methods (Barnes et al., 2021; Peng et al., 2021) for this task are motivated by graph-based parsers that can parse the structured sentiment using a technique similar to dependency parsing. On the other hand, the state-of-the-art graph-based parser, PERIN (Samuel et al., 2022), utilizes the idea of meaning representation parsing (Oepen et al., 2020; Samuel and Straka, 2020), which remedies the lossy dependency graphs of the previous work (Barnes et al., 2021). Our graph-based method (i.e., Graph) is one such study. The differences between Graph and the related literature are in the graph encoding method and model architecture.

To directly compare our Graph with the state-of-the-art methods, we trained Graph with XLM-RoBERTa-base (XLM-R; Conneau et al. (2020)). We tried three different seeds and averaged the scores. Because the corpora provided in the shared task were slightly modified from those used in the related studies, it is difficult to make a direct comparison. Thus, we evaluated only on the unaffected NoReC. Table 6 suggests that our Graph outperforms state-of-the-art baselines; however, we cannot conclude that our method is state-of-the-art based solely on an evaluation on NoReC. We hope that more extensive studies will clarify this situation.

On the other hand, an alternative to the traditional graph-based methods, we proposed a generation-based method (i.e., Seq2Seq) that showed promising results. Generation-based methods have been recently utilized in meaning representation parsing (Ozaki et al., 2020; Procopio et al., 2021); this framework offers more research options in terms of architecture and graph encodings for structured sentiment analysis. Our study can be positioned within this framework.

5 Conclusion

This paper showed two different parsers (i.e., graph-based and seq2seq-based parsers) for SemEval-2022 Task 10, structured sentiment analysis. The parsers were compared in various aspects such as complexity in structure and semantics. Experiments and analyses showed that both parsers output reasonable predictions, but that it is hard to decide which is better. This could be because the deci-

sion criteria rely on how the sentiment structure is defined. This makes structured sentiment analysis challenging. To deal with this difficulty, it may be helpful to apply an abstract representation of structured sentiment.

Acknowledgements

The computational resources of the AI Bridging Cloud Infrastructure (ABCI) provided by the National Institute of Advanced Industrial Science and Technology (AIST) were used. We would like to thank Dr. Masaaki Shimizu for maintenance and management of the large computational resources.

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.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

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. 2021. 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, Laura Ana Maria Oberländer, Enrica Troiano, Andrey Kutuzov, Jan Buchmann, Rodrigo Agerri, Lilja Øvrelid, and Erik Velldal. 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.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Zhuang Chen and Tieyun Qian. 2020. 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.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. 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.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China. Association for Computational Linguistics.

Agnieszka Falenska, Anders Björkelund, and Jonas Kuhn. 2020. Integrating graph-based and transition-based dependency parsers in the deep contextualized

era. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies*, pages 25–39, Online. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

John Lawrence and Chris Reed. 2019. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, and Kohsuke Yanai. 2022. End-to-end argument mining with cross-corpora multi-task learning. *Transactions of the Association for Computational Linguistics*, pages accepted, to appear.

Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajic, Daniel Hershcovich, Bin Li, Tim O'Gorman, Nianwen Xue, and Daniel Zeman. 2020. MRP 2020: The second shared task on cross-framework and cross-lingual meaning representation parsing. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 1–22, Online. Association for Computational Linguistics.

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.

Hiroaki Ozaki, Gaku Morio, Yuta Koreeda, Terufumi Morishita, and Toshinori Miyoshi. 2020. Hitachi at MRP 2020: Text-to-graph-notation transducer. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 40–52, Online. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Letian Peng, Zuchao Li, and Hai Zhao. 2021. Sparse fuzzy attention for structured sentiment analysis.

Luigi Procopio, Rocco Tripodi, and Roberto Navigli. 2021. SGL: Speaking the graph languages of semantic parsing via multilingual translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 325–337, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

David Samuel, Jeremy Barnes, Robin Kurtz, Stephan Oepen, Lilja Øvrelid, and Erik Velldal. 2022. Direct parsing to sentiment graphs. In *arXiv (to appear in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022))*.

David Samuel and Milan Straka. 2020. ÚFAL at MRP 2020: Permutation-invariant semantic parsing in PERIN. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 53–64, Online. Association for Computational Linguistics.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.