# Semantic Simplification for Sentiment Classification

**Xiaotong Jiang, Zhongqing Wang*, and Guodong Zhou**
Natural Language Processing Lab, Soochow University, Suzhou, China
devjiang@outlook.com {wangzq, gdzhou}@suda.edu.cn

## Abstract

Recent work on document-level sentiment classification has shown that the sentiment in the original text is often hard to capture, since the sentiment is usually either expressed implicitly or shifted due to the occurrences of negation and rhetorical words. To this end, we enhance the original text with a sentiment-driven simplified clause to intensify its sentiment. The simplified clause shares the same opinion with the original text but expresses the opinion much more simply. Meanwhile, we employ Abstract Meaning Representation (AMR) for generating simplified clauses, since AMR explicitly provides core semantic knowledge, and potentially offers core concepts and explicit structures of original texts. Empirical studies show the effectiveness of our proposed model over several strong baselines. The results also indicate the importance of simplified clauses for sentiment classification.

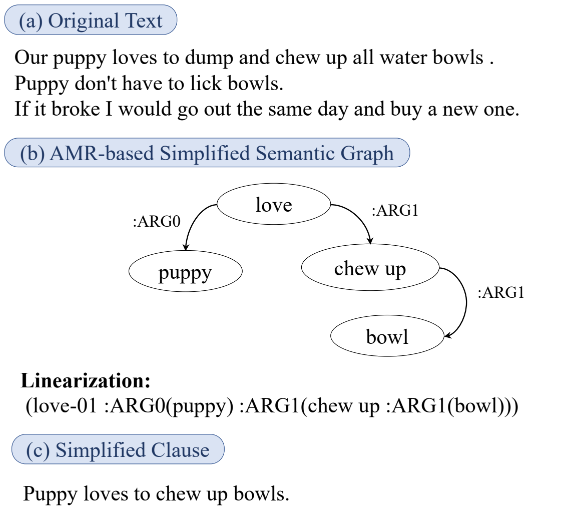


Figure 1: Example of simplified clause with AMR-based representation.

## 1 Introduction

As a critical application of natural language processing, document-level sentiment classification has received considerable attention during the last two decades with the underlying assumption that the entire text has an overall polarity.

In the literature, previous studies focus on predicting the overall sentiment from original text using either statistical (Pang et al., 2002; Wilson et al., 2005; Xia et al., 2011) or neural models (Kim, 2014; Tang et al., 2016b; Chen et al., 2020). However, as shown in Figure 1(a), the overall sentiment of the original text is often hard to capture, since the overall sentiment is usually either expressed implicitly or shifted due to the occurrences of negation and rhetorical words. To address above challenges in the original text, both attention-based (Chen et al., 2017; Amplayo et al., 2018; Tay et al., 2018) and rhetorical structure-based approaches (Li et al., 2010; Xia et al., 2015; Pröllochs et al., 2019; Yadav et al., 2021) have been proposed. Although the above pioneer studies have achieved certain success, they either heavily rely on human knowledge or suffer from the complex structure of the original text.

To tackle the above limitations, we simplify the original text to a simplified clause and employ the simplified clause for sentiment classification. As shown in Figure 1(c), a qualified simplified clause shares the same opinion with the original text and expresses the opinion much more simply. Therefore, it is much easier to detect the polarity from the simplified clause than the original text.

However, the simplified clause is hard to generate from original text, since we need to reduce the linguistic complexity of the original text, and keep the same polarity as well as the original meaning. Intuitively, such issues can be alleviated by having a structural representation of semantic information, which treats concepts as nodes and builds structural relations between nodes, making it easy to find the important and sentiment-driven content.

* Corresponding author



*Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11022–11032
December 7-11, 2022 ©2022 Association for Computational Linguistics

Explicit structures are more interpretable compared to neural representations and have been shown to be useful in many applications (Liao et al., 2018; Song et al., 2019; Xu et al., 2020; Bai et al., 2021).

In this study, we employ Abstract Meaning Representation (AMR) (Banarescu et al., 2013) for simplified clause generation in order to better exploit the semantic representation of the original text. As shown in Figure 1(b), AMR-based simplified semantic graph models the original text using rooted directed acyclic graph, which highlights its main concepts and semantic relations while abstracting away function words. It can thus potentially offer core concepts and explicit structures needed for aggregating the meaning of the original text.

Existing work on AMR parsing focuses on the sentence level. However, as shown in the right green box in Figure 2, the semantic structure of an original text contains rich cross-sentence co-reference links, and lots of duplicated and irrelevant information. To this end, we propose a simplified graph extraction algorithm to automatically derive a document-level simplified semantic graph from sentence-level AMRs, by merging co-reference links and pruning duplicate and irrelevant structures.

In summary, we firstly use a sequence-to-structure network to generate the AMR-based semantic graphs from sentences in original text. We then use a simplified graph extraction model to merge the sentence-level semantic graphs and extract a document-level simplified semantic graph. Thirdly, we employ a structure-to-sequence model to generate the simplified clause from the simplified semantic graph. Afterward, we integrate the simplified clause and original review text for sentiment classification.

Detailed evaluation shows that our model significantly advances the state-of-the-art performance on several benchmark datasets. The results also show that the simplified clause is very useful for sentiment classification, and indicates AMR is beneficial for simplified clause generation.

## 2 Related Work

In this study, we introduce two related topics of this study: document-level sentiment classification and text simplification.

### 2.1 Sentiment Classification

In the literature, various studies focus on document-level sentiment classification (Pang et al., 2002; Kim, 2014; Wu et al., 2018; Pröllochs et al., 2019; Yadav et al., 2021). However, the overall sentiment in the original text is often hard to capture. To address the above challenges, both attention-based and rhetorical structure-based approaches have been proposed.

Attention-based approaches (Wu et al., 2018; Amplayo et al., 2018; Sun et al., 2018; Zhang et al., 2019; Basiri et al., 2021) focus on capturing the important information of the original text by computing the attention weight of each word. However, existing attention-based approaches cannot leverage the syntactic structure for sentiment classification. To tackle the problem, many rhetorical structure-based approaches have been proposed. These approaches can be divided into two categories: heuristic model-based (Pröllochs et al., 2019; Yadav et al., 2021) and neural model-based (Socher et al., 2013; Baly et al., 2017; Tang et al., 2020; Zhang and Qian, 2020).

Instead of only relying on the original text to determine the overall polarity, we propose a semantic simplification model to generate the simplified clause based on the semantic representation of the original text. As shown in experiments, the generated simplified clause is obviously sentiment-driven, and beneficial for sentiment classification.

### 2.2 Text Simplification

Text Simplification is the task of reducing the complexity of the vocabulary and sentence structure of the text while retaining its original meaning. Most of the studies can be divided into two categories: lexical simplification and syntactic simplification.

Lexical simplification is the process of replacing complex words in a given sentence with simpler alternatives of equivalent meanings (Devlin and Tait, 1998; De Belder and Moens, 2010; Biran et al., 2011; Paetzold and Specia, 2017). Unlike lexical simplification, syntactic simplification seeks to identify grammatically complex text, and rewrite it to make it easier to comprehend. Early rule-based works (Aluisio and Gasperin, 2010) are limited by the difficulty in creating and validating rewrite rules. Recently, most advances (Wang et al., 2016; Bingel and Søgaard, 2016; See et al., 2017; Surya et al., 2019; Zhao et al., 2020) are based on deep neural networks, especially the neural ma-

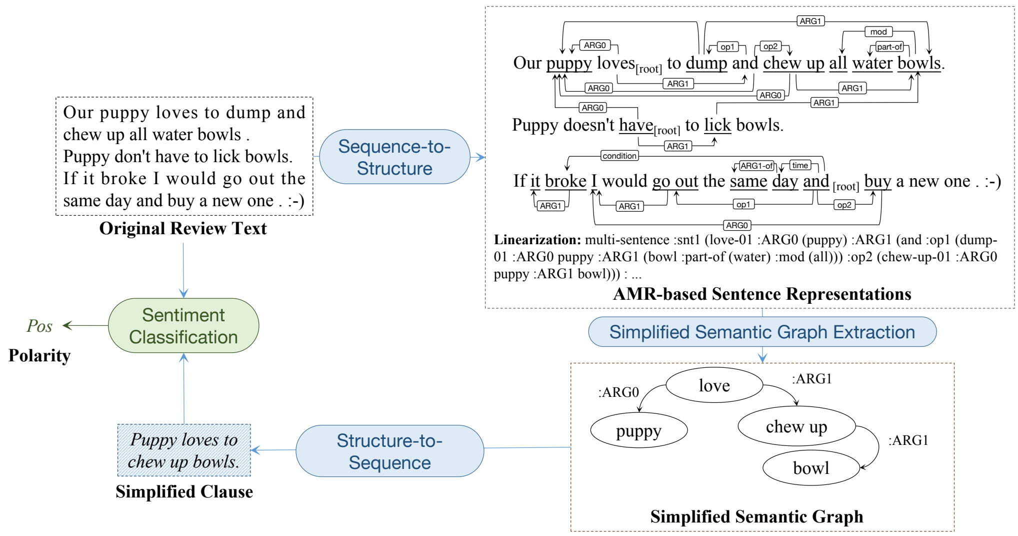


Figure 2: Overview of proposed model.

chine translation models.

The difference between the proposed semantic simplification and vanilla text simplification is that the former one pays more attention to the sentiment of the original text. Meanwhile, after semantic simplification, the simplified clause is more refined in context and more explicit in sentiment than the original text.

## 3 Method

In this study, we aim to predict the polarity of a given document with its original text and the simplified clause. As shown in Figure 2, we first employs a **sequence-to-structure network** to generate the AMR-based semantic graphs from sentences in the original text. We then use a **simplified graph extraction model** to merge the sentence-level semantic graphs and extract a document-level simplified semantic graph. Thirdly, we use a **structure-to-sequence network** to generate the simplified clause from the simplified semantic graph. Afterward, we integrate the simplified clause and original review text for sentiment classification.

In the following, we will illustrate these components of the proposed model, and then discuss the objective function and training process.

### 3.1 Sequence-to-Structure Network

We first employ a sequence-to-structure network to generate AMR graphs from each sentence in the original text. Since it is much easier to generate a sequence than generate a graph, we linearize AMR graphs to sequences. In particular, AMR graphs are first converted into AMR trees by removing variables and duplicating the co-referring nodes. Then newlines presented in an AMR tree are replaced by spaces to get a sequence (van Noord and Bos, 2017; Xu et al., 2020). The right green box in Figure 2 illustrates the linearization result of the AMR graph.

Based on the above linearization strategy, the sequence-to-structure model generates the AMR structure via a transformer-based encoder-decoder architecture (Vaswani et al., 2017). Given the token sequence $X = \{x_1, ..., x_n\}$ as input, the sequence-to-structure model outputs the linearized representation $R = \{r_1, ..., r_n\}$. To this end, the sequence-to-structure model first computes the hidden vector representation $H = \{h_1, ..., h_n\}$ of the input via a multi-layer transformer encoder:

$$H = \mathrm{Encoder}(\{x_1, ..., x_n\}) \qquad (1)$$

where each layer of Encoder is a transformer block with the multi-head attention mechanism.

After the input token sequence is encoded, the decoder predicts the output structure token-by-token with the sequential input tokens' hidden vectors. At the $i$-th step of generation, the self-attention decoder predicts the $i$-th token $y_i$ in the linearized form and decoder state $h$ as:

$$r_i, h_i^d = \mathrm{Decoder}([H; h_1^d, ..., h_{i-1}^d], r_{i-1}) \qquad (2)$$

where each layer of Decoder is a transformer block that contains self-attention with decoder state $h_i^d$ and cross-attention with encoder state $H$.

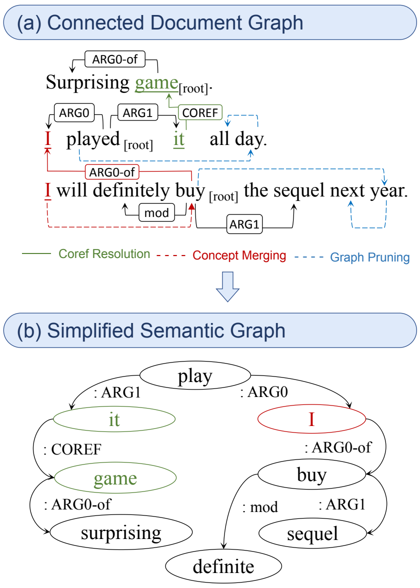


Figure 3: Example of simplified semantic graph extraction.

The generated output structured sequence starts from the start token "$\langle bos \rangle$" and ends with the end token "$\langle eos \rangle$". The conditional probability of the whole output sequence $p(R|X)$ is progressively combined by the probability of each step $p(r_i|r_{<i}, X)$:

$$p(R|X) = \prod_{i}^{n} p(r_i|r_{<i}, X) \tag{3}$$

where $r_{<i} = \{r_1, ..., r_{i-1}\}$, and $p(r_i|r_{<i}, X)$ is the probability over target vocabulary $V$ normalized by softmax.

Since all tokens in linearized representations are also natural language words, we adopt the pre-trained language model BART (Lewis et al., 2020) as our transformer-based encoder-decoder architecture. In this way, the general text generation knowledge can be directly reused.

## 3.2 Simplified Semantic Graph Extraction

After we learn the AMR-based semantic graphs of sentences in a text, we extract the document-level simplified semantic graph from these sentence-level semantic graphs. The process of simplified semantic graph extraction can be separated into two stages: document-level semantic graph construction, and graph pruning.

### Document-level Semantic Graph Construction

The semantic graph of a sentence is represented by a rooted, directed, and acyclic AMR graph (Banarescu et al., 2013), where nodes are concepts and edges are semantic relations. Given a set of sentences and their AMR graphs, we attempt to consolidate all sentence graphs to a connected document graph. As shown in Figure 3(a), we first employ a 'ROOT' node to connect the root of each sentence graph, yielding a connected document-level semantic graph.

A major challenge for understanding the document-level semantic graph is posed by pronouns (Lee et al., 2017; Kantor and Globerson, 2019; Fu et al., 2021). We thus conduct co-reference resolution using an off-to-shelf model[1] in order to identify concept nodes in sentence-level AMRs that refer to the same entity. For example, in Figure 3(a), 'game' in the first sentence, and 'it' in the second sentence refers to the same entity. We add edges labeled with 'COREF' between them to indicate their relation.

### Graph Pruning

Since there are lots of duplicate and irrelevant information in the original document-level graph, we then need to prune it into a sentiment-driven simplified semantic graph. The rules of pruning are introduced as below,

**Concept Merging.** We first perform concept merging. Graph nodes representing the same concept, determined by the surface word form, are merged to a single node in the graph. It operates on a very ad-hoc principle (van Noord and Bos, 2017): if two nodes have the same concept, the second one is actually a reference to the first one. Therefore, we replace each node that has already occurred in the AMR graph by the variable of the antecedent node. Given the example in Figure 3(a), the red dash line should be removed, since the concept 'I' already appears in the previous sentence.

**Graph Pruning.** We then need to remove the duplicate nodes in the graph. We remove nodes with the same argument and concept under the same parent. We also remove nodes that occur three times or more, no matter their parents.

Meanwhile, we remove the irrelevant information in the graph, and make sure that the graph is a sentiment-driven graph. Therefore, apart from 'ARG' and 'op' relations[2], only 'manner', 'mod', and 'polarity' relations are kept in the graph, and

[1] `https://github.com/huggingface/neuralcoref`

[2] Following PropBank conventions and Banarescu et al. (2013), 'ARG' is the frame arguments, 'op' denotes relations for lists.

we remove all the other relations. 'manner' relation denotes an action between a noun and a verb. 'mod' means modifying relation, which is always related with a noun and an adjective. 'polarity' is represented as negation logically, which expresses modals with concepts. All of these relations are basic and correlated with the sentiment of a document. We thus keep these relations to construct the sentiment-driven simplified graph. As shown in Figure 3(a), the blue dash lines should be removed, since these relations are not in the above relation set.

As shown in Figure 3(b), after graph pruning, we construct a sentiment-driven simplified semantic graph. Different from the original sentence-level graphs, which contain lots of duplicate and irrelevant information, the new document-level simplified semantic graph only consists of the key nodes and sentiment-driven relations.

### 3.3 Structure-to-Sequence Network

We then generate the simplified clause from the simplified semantic graph via the transformer-based encoder-decoder architecture (Vaswani et al., 2017) and the pre-trained language model BART (Lewis et al., 2020).

Given the input simplified semantic graph $G = \{w_1, w_2, ..., w_n\}$, which is corresponding to the original token sequence $X$, the structure-to-sequence model outputs the simplified clause $Y = \{y_1, ..., y_n\}$. Note that, we linearize the semantic graph into a sequence of nodes and edge labels using depth-first traversal of the graph.

Therefore, the structure-to-sequence model computes the hidden vector representation $H^{'}$ of the input linearized graph sequence via a multi-layer transformer encoder:

$$H^{'} = \text{Encoder}(\{w_1, ..., w_n\}) \tag{4}$$

where each layer of $\text{Encoder}$ is a transformer block with the multi-head attention mechanism.

After the input token sequence is encoded, the $\text{Decoder}$ predicts the simplified clause token-by-token with the sequential input tokens' hidden vectors using a self-attention decoder. The conditional probability of the whole output sequence $p(Y|G)$ is then progressively combined by the probability of each step $p(y_i|y_{<i}, G)$:

$$p(Y|G) = \prod_{i}^{n} p(y_i|y_{<i}, G) \tag{5}$$

where $y_{<i} = \{y_1...y_{i-1}\}$, and $p(y_i|y_{<i}, G)$ is the probability over the target vocabulary $V$ normalized by a $\text{softmax}$ layer.

### 3.4 Sentiment Classification

Finally, we employ the pre-trained language model BERT (Devlin et al., 2018) to learn the representation $\hat{H}$ from the sequence $[CLS]$ $X$ $[SEP]$ $Y$ $[SEP]$, where $X$ is the original text, $Y$ is the generated simplified clause, $[CLS]$ is BERT's special classification token, and $[SEP]$ is the special token to denote separation. We then employ a multi-layer perceptron to predict the overall polarity based on the representation $\hat{H}$,

$$H_P = \sigma(W_p^h \hat{H} + b_p^h), \tag{6}$$

$H_P$ is then used as inputs to a softmax output layer,

$$P_P = \text{softmax}(W_p H_P + B_P) \tag{7}$$

Here, $W_p^h$, $b_p^h$, $W_p$, and $B_p$ are model parameters, and $P_P$ is used to predict the overall polarity from the simplified clause and original text.

### 3.5 Objective Functions and Training

In this subsection, we show the objective functions and training process of the proposed model.

**Sentiment Simplification.** The goal is to maximize the probability of the output sentiment-driven simplified clause $Y$ given the input original text $X$. Therefore, we optimize the negative log-likelihood loss function:

$$\mathcal{L} = -\frac{1}{|\tau|} \sum_{(X,Y)\in\tau} \log p(Y|X;\theta) \tag{8}$$

where $\theta$ is the model parameters, and $(X, Y)$ is a (*original text, simplified clause*) pair in training set $\tau$, then

$$\begin{aligned} &\log p(Y|X;\theta) = \\ &= \sum_{i=1}^{n} \log p(y_i|y_{<i}, X;\theta) \end{aligned} \tag{9}$$

where $y_{<i} = \{y_1, ..., y_{i-1}\}$, and $p(y_i|y_{<i}, X; \theta)$ is calculated by the decoder.

**Sentiment Classification.** Given a token sequence $X$ from a document, and the corresponding sentiment-driven simplified clause $Y$ generated by the proposed model. Our training objective is to minimize the cross-entropy loss over a set of training examples, with a $\ell_2$-regularization term,

$$\mathcal{J} = -\sum_{i=1}^{N}\sum_{j=1}^{K} p_i \log \hat{p}_i + \frac{\lambda}{2}||\theta_p||^2 \quad (10)$$

where $p_i$ and $\hat{p}_i$ are the pre-defined and predicted sentimental labels of the original text $X$, respectively. $\theta_p$ is the set of model parameters, and $\lambda$ is a parameter for $\ell_2$-regularization.

## 4 Experimentation

In this section, we introduce the datasets used for evaluation and the baseline methods employed for comparison. We then report the experimental results conducted from different perspectives, and analyze the effectiveness of the proposed model with different factors.

### 4.1 Experimental Settings

We conduct our experiments on subsets of sentiment analysis benchmarks from Amazon Product Dataset[3]. The dataset is anonymized and contains no personal data. More specifically, we choose three domain reviews from the Amazon Product Dataset: pets supplies (Pet.), sports (Spt.), and toy (Toy.).

There are two kinds of datasets in our experiments: one is for sentiment classification, and the other is for simplified clause generation. In **sentiment classification** dataset, we randomly select 3,000 reviews for each domain, 60% reviews are used as training data, 20% reviews are used as testing data, and the remaining reviews are used as validation data. In **simplified clause generation** dataset, we select another 12,000 reviews from each domain to train the generation model. The original AMR graph of each sentence is obtained by S2S-AMR-Parser (Xu et al., 2020)[4].

We use BERT[5] and fine-tune its parameters during training the sentiment classification model. Meanwhile, we employ BART[6] and fine-tune its parameters for simplified clause generation model. We tune the parameters of our models by grid searching on the validation dataset. We select the best models by early stopping using the Accuracy results on the validation dataset. The dimension of other hidden variables of all the models is 128. The model parameters are optimized by Adam (Kingma and Ba, 2014) with a learning rate of 2e-5. The batch size is 32, and a dropout probability of 0.1 is used. Our experiments are carried out with an Nvidia GTX-1080Ti GPU.

The experimental results are obtained by averaging ten runs with the random initialization. We use scikit-learn package (Pedregosa et al., 2011) to calculate Accuracy as the evaluation metrics.

### 4.2 Main Results

Table 1 shows the results of different systems on three domains. We compare the proposed model with various strong baselines,

- *LSTM* is a basic neural model using LSTM (Hochreiter and Schmidhuber, 1997) to learn the document representation and has been widely used for sentiment classification and other NLP applications (Tang et al., 2016a; Ji et al., 2016).

- *AGLR* (Tay et al., 2018) is a lexicon-driven attention based model. It employs an attention mechanism to integrate lexicon words with long-range contextual information.

- *LexicalAT* (Xu et al., 2019) employs adversarial training with lexical information to improve the robustness of sentiment classification models.

- *RGAT* (Wang et al., 2020) extends the graph attention network to encode graphs with labeled edges. It defines a unified aspect-oriented dependency tree structure rooted at a target aspect by reshaping and pruning an ordinary parse tree.

- *CFSA* (Yang et al., 2021) generates counterfactually augmented data to raise the robustness and underlying sensitivity to the systematic bias of sentiment classification models.

- *BERT-Original* employs original text to fine-tune the BERT pre-trained language model (Devlin et al., 2018). BERT is the basic classification component in the proposed model.

- *BERT-Clause* employs the generated simplified clause to fine-tune BERT. The simplified

[3] https://nijianmo.github.io/amazon/index.html

[4] https://github.com/xdqkid/S2S-AMR-Parser

[5] $\text{BERT}_{\text{base}}$, https://huggingface.co/bert-base-uncased

[6] $\text{BART}_{\text{base}}$, https://huggingface.co/facebook/bart-base

| Method | Pet. | Toy. | Sports. | Avg. |
|---|---|---|---|---|
| LSTM | 71.7 | 75.0 | 68.9 | 71.9 |
| AGLR | 72.0 | 74.2 | 74.9 | 73.7 |
| LexicalAT | 72.4 | 80.5 | 76.3 | 76.4 |
| RGAT | 76.3 | 84.4 | 79.8 | 80.2 |
| CFSA | 75.3 | 83.9 | 80.2 | 79.8 |
| BERT-Original | 74.5 | 83.8 | 78.4 | 78.9 |
| BERT-Clause | 59.7 | 64.2 | 58.7 | 60.9 |
| Ours | 77.2 | 86.1 | 81.9 | 81.7 |

Table 1: Comparison with baselines.

| Method | Accuracy |
|---|---|
| BERT-Original | 78.9 |
| TextRank | 79.2 |
| RNNSeq2Seq | 79.5 |
| UniLM | 80.4 |
| BART | 79.8 |
| Ours | 81.7 |

Table 2: Comparison with different simplified clause generation models with average Accuracy measurement.

clause is generated by the proposed semantic simplification model.

Comparison with BERT-Original and other state-of-the-art methods, BERT-Clause achieves competitive performance. It indicates that the simplified clause is beneficial to sentiment classification. In addition, our proposed model outperforms the previous state-of-the-art methods significantly ($p < 0.05$), as the proposed model employs AMR-based semantic representation to generate the simplified clause for sentiment classification. This shows that the semantic simplification architecture is very helpful for generating the simplified clause and predicting the polarity.

### 4.3 Impact of Simplified Clause

This subsection analyzes the impact of the simplified clause with different generation models. We employ four kinds of text generation methods to generate simplified clause: TextRank (Mihalcea and Tarau, 2004) means that we employ pagerank algorithm to select the most representative sentence in document as simplified clause; RNNSeq2Seq (Bahdanau et al., 2015) is an attention-based sequence-to-sequence model, which is a representative baseline; UniLM (Dong et al., 2019) and BART(Lewis et al., 2020) are two state-of-the-art text generation models with a pre-trained language model. Furthermore, we integrate the simplified clause with the original text to fine-tune the BERT pre-trained language model for sentiment classification.

| Method | Accuracy |
|---|---|
| Ours | 81.7 |
| -AMR | 79.8 |
| -Extraction | 80.1 |
| -Concept Merging | 80.6 |
| -Graph Pruning | 81.2 |

Table 3: Impact of AMR-based semantic representation with average Accuracy measurement.

From Table 2, we can see that: 1) unsupervised TextRank method achieves acceptable results. It shows that the simplified clause is very helpful, even they are extracted with a simple unsupervised method. 2) Compared with BERT-Original, the simplified clauses which are generated by either UniLM or BART show better performance, which shows that simplified clause is much more important than original text in sentiment classification. 3) Our proposed model outperforms UniLM and BART significantly ($p < 0.05$), which indicates that the AMR-based semantic representation is very important for generating the simplified clause.

### 4.4 Impact of AMR-based Semantic Representation

As shown in Table 3, we then employ ablation experiments to analyze the impact of AMR-based semantic representation. If we totally remove AMR-based semantic representation (-AMR), it degrades the proposed semantic simplification model to a BART-based sequence-to-sequence model, and the performance drops to 79.8%. It shows that AMR can help a model to better capture the semantic representation of original text, and is beneficial to generate the simplified clause.

In addition, if we remove the simplified semantic graph extraction part (-Extraction) of the proposed model, and just employ the document-level AMR graph to generate the simplified clause, the performance drops to 80.1%. It shows that there is a lot of duplicated and irrelevant information in the document-level graph. Furthermore, we also find that both concept merging and graph pruning are beneficial to extract a sentiment-driven simplified graph. If we remove these two components, the performance drops to 80.6% and 81.2% respectively.

## 5 Analysis and Discussion

In this section, we give some analysis and discussion to show the importance of the simplified clause for sentiment classification. Note that, the results in this section are the average of all the domains.

| ID | Original Text | Simplified Clause | BART-BERT | Ours | Gold |
|---|---|---|---|---|---|
| 1 | i got simple chicken tacos ... uh delcious.. pretty cheap also.. ehh i coulda been more full after but no biggie. yummm | They are nice. | -1 | +1 | +1 |
| 2 | i went here just for kicks and ended up really liking the place. they have great food and a great atmosphere. the bands they book are just mediocre at best, but that won't stop me from coming back. | They have great food and a nice atmosphere. | -1 | +1 | +1 |
| 3 | why offer reservations if you can't seat a party within 30 minutes of their reservation time? at least the group that complained ahead of us was offered drinks ... | There were groups of people angry. | +1 | -1 | -1 |

Table 4: Examples of case study.

| Measurement | Original | Clause |
|---|---|---|
| Length | 234.8 | 15.4 |
| Vocabulary | 34,548 | 3,682 |
| Ratio of Sent. Words (%) | 8.6 | 20.8 |
| Acc. of TermCount (%) | 62.1 | 65.4 |

Table 5: Statistics of the original reviews and simplified clause.

### 5.1 Statistics of Simplified Clause

In this subsection, we give some statistics to analyze the quality of generated simplified clause compared with original text in Table 5, where 1) *Length* is the number of words in a given text (i.e., original text, generated simplified clause). 2) *Vocabulary* is the size of the vocabulary. 3) *Ratio of Sent. Words* means the percent of sentimental words in a given text. It is used to measure the sentimental richness of a given text. 4) *Acc. of TermCount* employs term counting algorithm (Turney, 2002) to predict the polarity of a given text, and uses Accuracy to measure the performance. Since the term counting algorithm only counts the sentimental term, its performance can be used to justify if the sentiment is expressed explicitly in a given text.

From Table 5, we find that: 1) the length of the simplified clause is much shorter than the original text, and the vocabulary of the simplified clause is limited. It shows that the simplified clause is much shorter and more refined than the original text. 2) Percent of sentimental words in the simplified clause is much larger than the original text. It indicates the sentimental richness of the simplified clause, and shows that the simplified clause is more informative and representative than the original text. 3) The performance of term counting on the simplified clause is much better than the original text, it shows that the sentiment in the simplified clause is expressed more explicitly, and is easier to understand than the original text.

### 5.2 Case Study

We choose three examples to illustrate the effectiveness of the proposed model compared with BART-BERT model in Table 4, where BART-BERT means that we employ BART to generate the simplified clause, and fine-tune BERT with original text and simplified clause for sentiment classification.

As shown in Table 4, the simplified clause is generated by the proposed model. From the table, we find that the proposed model predicts correct polarity based on a generated sentiment-driven simplified clause, while BART-BERT fails to predict polarity in all of these examples. From the first two examples, we find the simplified clause is much more refined and easier to predict polarity than the original text. Besides, although the third example does not contain any sentimental word, and expresses anger emotion implicitly, the proposed model can capture such anger emotion based AMR-based semantic representation, and generates the simplified clause with explicitly anger emotion.

## 6 Conclusion

In this paper, we enhance the original text with a simplified clause for document-level sentiment classification. The simplified clause shares the same opinion with the original text but expresses the opinion much more simply. Meanwhile, we employ AMR for generating the simplified clause, since AMR potentially offers core concepts and explicit structures from the original text. We then integrate the simplified clause with original text for sentiment classification. Empirical studies demonstrate that our model significantly advances the state-of-the-art performance on several benchmark datasets. The results also indicate the simplified clause is very useful for sentiment classification.

## Limitations

Although the proposed semantic simplification method achieves the best performance in sentiment classification, it needs more training time and GPU resources to learn the representations from both original text and simplified clause. In addition, it also needs external processes for AMR parsing and simplified clause generation.

## Acknowledgement

We would like to thank Mingyi Lu for her prime study on this topic, and Prof. Shoushan Li for his helpful advice and discussion during this work. Also, we would like to thank the anonymous reviewers for their insightful and valuable comments. This work is supported by the National Natural Science Foundation of China (No. 62076175, No. 61976146).

## References

Sandra Aluisio and Caroline Gasperin. 2010. Fostering digital inclusion and accessibility: the porsimples project for simplification of portuguese texts. In *NAACL HLT 2010*, pages 46–53.

Reinald Kim Amplayo, Jihyeok Kim, Sua Sung, and Seung-won Hwang. 2018. Cold-start aware user and product attention for sentiment classification. In *ACL 2018*, pages 2535–2544.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Xuefeng Bai, Yulong Chen, Linfeng Song, and Yue Zhang. 2021. Semantic representation for dialogue modeling. In *ACL 2021*, pages 4430–4445.

Ramy Baly, Hazem M. Hajj, Nizar Habash, Khaled Bashir Shaban, and Wassim El-Hajj. 2017. A sentiment treebank and morphologically enriched recursive deep models for effective sentiment analysis in arabic. *ACM Trans. Asian & Low-Resource Lang. Inf. Process.*, 16(4):23:1–23:21.

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 *ACL 2013*, pages 178–186.

Mohammad Ehsan Basiri, Shahla Nemati, Moloud Abdar, Erik Cambria, and U. Rajendra Acharya. 2021. Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis. *Future Generation Computer Systems*, 115:279–294.

Joachim Bingel and Anders Søgaard. 2016. Text simplification as tree labeling. In *ACL 2016*, pages 337–343.

Or Biran, Samuel Brody, and Noémie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *ACL 2011*, pages 496–501.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *EMNLP 2017*, pages 452–461.

Xiao Chen, Changlong Sun, Jingjing Wang, Shoushan Li, Luo Si, Min Zhang, and Guodong Zhou. 2020. Aspect sentiment classification with document-level sentiment preference modeling. In *ACL 2020*, pages 3667–3677. Association for Computational Linguistics.

Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *ACM SIGIR 2010*, pages 19–26.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Siobhan Devlin and John Tait. 1998. The use of a psycholinguistic database in the simplification of text for aphasic readers. pages 161–173.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *NIPS 2019*, pages 13042–13054.

Qiankun Fu, Linfeng Song, Wenyu Du, and Yue Zhang. 2021. End-to-end AMR coreference resolution. 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 4204–4214, Online. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. A latent variable recurrent neural network for discourse relation language models. *CoRR*, abs/1603.01913.

Ben Kantor and Amir Globerson. 2019. Coreference resolution with entity equalization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 673–677.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP 2014*, pages 1746–1751.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 188–197.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL 2020*, pages 7871–7880.

Shoushan Li, Sophia Yat Mei Lee, Ying Chen, Chu-Ren Huang, and Guodong Zhou. 2010. Sentiment classification and polarity shifting. In *COLING 2010*, pages 635–643.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract meaning representation for multi-document summarization. In *COLING 2018*, pages 1178–1190.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 404–411. ACL.

Gustavo Paetzold and Lucia Specia. 2017. Lexical simplification with neural ranking. In *EACL 2017*, pages 34–40.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *EMNLP 2002*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Nicolas Pröllochs, Stefan Feuerriegel, and Dirk Neumann. 2019. Learning interpretable negation rules via weak supervision at document level: A reinforcement learning approach. In *NAACL HLT 2019*, pages 407–413.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP 2013*, pages 1631–1642.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using AMR. *Trans. Assoc. Comput. Linguistics*, 7:19–31.

Qingying Sun, Zhongqing Wang, Qiaoming Zhu, and Guodong Zhou. 2018. Stance detection with hierarchical attention network. In *COLING 2018*, pages 2399–2409.

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *ACL 2019*, pages 2058–2068.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. Effective lstms for target-dependent sentiment classification. In *COLING 2016*, pages 3298–3307.

Duyu Tang, Bing Qin, and Ting Liu. 2016b. Aspect level sentiment classification with deep memory network. In *EMNLP 2016*, pages 214–224.

Hao Tang, Donghong Ji, Chenliang Li, and Qiji Zhou. 2020. Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In *ACL 2020*, pages 6578–6588, Online. Association for Computational Linguistics.

Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. 2018. Attentive gated lexicon reader with contrastive contextual co-attention for sentiment classification. In *EMNLP 2018*, pages 3443–3453.

Peter D. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *ACL 2002*, pages 417–424.

Rik van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *CoRR*, abs/1705.09980.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS 2017*, pages 5998–6008.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3229–3238. Association for Computational Linguistics.

Tong Wang, Ping Chen, John Rochford, and Jipeng Qiang. 2016. Text simplification using neural machine translation. In *AAAI*, volume 30.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *EMNLP HLT 2005*, pages 347–354.

Zhen Wu, Xin-Yu Dai, Cunyan Yin, Shujian Huang, and Jiajun Chen. 2018. Improving review representations with user attention and product attention for sentiment classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5989–5996. AAAI Press.

Rui Xia, Feng Xu, Chengqing Zong, Qianmu Li, Yong Qi, and Tao Li. 2015. Dual sentiment analysis: Considering two sides of one review. *IEEE Trans. Knowl. Data Eng.*, 27(8):2120–2133.

Rui Xia, Chengqing Zong, and Shoushan Li. 2011. Ensemble of feature sets and classification algorithms for sentiment classification. *Inf. Sci.*, 181(6):1138–1152.

Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2020. Improving AMR parsing with sequence-to-sequence pre-training. In *EMNLP 2020*, pages 2501–2511.

Jingjing Xu, Liang Zhao, Hanqi Yan, Qi Zeng, Yun Liang, and Xu SUN. 2019. LexicalAT: Lexical-based adversarial reinforcement training for robust sentiment classification. In *EMNLP 2019*, pages 5517–5526, Hong Kong, China. Association for Computational Linguistics.

Rohan Kumar Yadav, Lei Jiao, Ole-Christoffer Granmo, and Morten Goodwin. 2021. Human-level interpretable learning for aspect-based sentiment analysis. In *AAAI 2021*.

Linyi Yang, Jiazheng Li, Padraig Cunningham, Yue Zhang, Barry Smyth, and Ruihai Dong. 2021. Exploring the efficacy of automatically generated counterfactuals for sentiment analysis. In *ACL 2021*, pages 306–316.

Kai Zhang, Hefu Zhang, Qi Liu, Hongke Zhao, Hengshu Zhu, and Enhong Chen. 2019. Interactive attention transfer network for cross-domain sentiment classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 5773–5780. AAAI Press.

Mi Zhang and Tieyun Qian. 2020. Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In *EMNLP 2020*, pages 3540–3549.

Yanbin Zhao, Lu Chen, Zhi Chen, and Kai Yu. 2020. Semi-supervised text simplification with back-translation and asymmetric denoising autoencoders. In *AAAI 2020*, volume 34, pages 9668–9675.