Structured Sentiment Analysis as Dependency Graph Parsing

Aditya Upadhyay (2022040), Chandan Sah (2022140), Rishabh Jay (2022401)

Abstract

Structured sentiment analysis aims to extract sentiment tuples by jointly identifying holders, targets, expressions, and polarities. This paper reinterprets this task as dependency graph parsing, achieving competitive results by leveraging multilingual embeddings and neural parsing techniques.

1 Introduction

Understanding sentiment in text has grown beyond simple positive or negative classification. Structured Sentiment Analysis (SSA) captures richer sentiment information by extracting *tuples* of the form (holder, target, expression, polarity), offering nuanced insights into opinions. For instance, in "I hate the service but love the food," two distinct sentiment tuples emerge. Traditional models fall short in representing these complex, overlapping relationships, especially in multilingual or low-resource contexts. This motivates a structured parsing approach that models sentiment relations explicitly.

2 Literature Survey

Structured Sentiment Analysis (SSA) has gained traction for its ability to extract richer sentiment information beyond simple classification. Barnes et al. (2021) proposed a novel approach to SSA by framing it as a **dependency graph parsing** task, which enhances the extraction of sentiment-related tuples (e.g., holder, target, expression, polarity) by using a **biaffine dependency parser**. Their method leverages **multilingual embeddings** (mBERT) and **neural parsing techniques**, enabling effective sentiment extraction in both monolingual and cross-lingual settings.

A key contribution of their work is the integration of a **dynamic gating mechanism** in the BiLSTM encoder, allowing the model to adaptively weigh different input modalities such as word forms, POS tags, and character-level features. This dynamic approach is particularly useful in low-resource or noisy linguistic contexts. Additionally, they enhance their model with a **Graph Attention Network (GAT)**, which improves the model's ability to capture long-range dependencies and refine relational sentiment extraction.

Barnes et al. (2021) demonstrated that their approach achieves competitive results on several multilingual SSA datasets, outperforming previous methods, particularly in the **Sentiment Graph F1** metric. Their work represents a significant advancement in SSA, especially for multilingual and low-resource languages.

3 Related Work

Early sentiment analysis research primarily focused on document or sentence-level classification, aiming to determine the overall sentiment polarity of texts. Aspect-Based Sentiment Analysis (ABSA) advanced this field by targeting specific entities or aspects within texts, providing more fine-grained sentiment insights. However, ABSA often lacked structural granularity in representing the relationships between different sentiment elements.

Barnes et al. (2021) introduced Structured Sentiment Analysis (SSA) as a graph parsing task, where sentiment structures are represented as dependency graphs connecting holders, targets, and expressions. Their approach extends the biaffine dependency parser by Dozat and Manning (2018), adapting it to handle multiple sentiment roles within a unified framework. This method demonstrated strong performance across multiple languages and datasets, highlighting the effectiveness of structured parsing in sentiment analysis. For more details, refer to their work: Barnes et al., 2021.

The biaffine parser by Dozat and Manning (2018) employs deep biaffine attention mechanisms to model the relationships between words in a sentence, achieving state-of-the-art results in dependency parsing tasks. Their model utilizes a BiLSTM encoder to capture contextual information, followed by biaffine classifiers to predict head-dependent relations and their labels. This architecture has been influential in various parsing tasks. The original paper can be accessed here: Dozat and Manning, 2018.

Graph Attention Networks (GATs) have emerged as powerful tools for modeling relational data, including syntactic and semantic structures in texts. Wang et al. (2020) proposed a Relational Graph Attention Network (R-GAT) for aspect-based sentiment analysis, which incorporates syntactic dependencies to better capture the relationships between aspects and opinions. Their model demonstrates the effectiveness of leveraging graph structures in sentiment tasks. For further reading, see: Wang et al., 2020.

Multi-task learning approaches have also been explored to enhance sentiment analysis by jointly learning related tasks. For instance, some models simultaneously perform sentiment classification and opinion extraction, leveraging shared representations to improve overall performance. Such joint learning frameworks can capture interdependencies between tasks, leading to more robust models. An example of this approach is discussed in: Zhang et al., 2023.

These advancements collectively contribute to the evolution of sentiment analysis from coarse-grained classification to more nuanced and structured understanding of sentiments within texts.

4 Methodology

We adopt a graph-based neural parsing approach inspired by Dozat and Manning (2018) and refined for SSA by Barnes et al. (2021). Key steps:

- **Token Embeddings:** We use mBERT to obtain contextual embeddings. External word2vec vectors and character-level embeddings are optionally concatenated.
- **BiLSTM Encoding:** Token embeddings are passed through a BiLSTM for contextualization.
- **Biaffine Scoring:** A feedforward layer produces arc head and dependent vectors. A biaffine classifier scores token pairs for arc presence and labels.
- **Span Strategies:** We explore two span-to-token reductions: **head-first** and **head-final**, allowing multiple overlapping arcs.
- **Loss and Inference:** The model is trained using labeled arc F1 (LF1), with final graphs built from arcs with positive scores.

Baseline Model 2 (BiLSTM + MLP)

Baseline 2 simplifies the sentiment graph parsing task by framing it as a pairwise classification problem over token positions. The model comprises a BiLSTM encoder followed by two multi-layer perceptrons (MLPs): one for edge prediction and another for sentiment label classification.

Each input sentence is tokenized and encoded into a fixed-length sequence using a vocabulary constructed from the training data. The BiLSTM processes these token sequences to produce contextual embeddings. For every pair of tokens (i, j), the corresponding BiLSTM outputs $\mathbf{h}_i$ and $\mathbf{h}_j$ are concatenated to form a pair representation:

$$\mathbf{p}_{ij} = [\mathbf{h}_i \parallel \mathbf{h}_j]$$

This pair representation is passed through:

- A binary classifier (MLP) to predict the presence of a sentiment arc between tokens $i \to j$
- A multi-class classifier (MLP) to predict the sentiment label (Positive, Negative, Neutral) of the arc

The model is trained using a combination of binary cross-entropy loss for edge prediction and cross-entropy loss for label classification. Evaluation is done using Sentiment Graph

F1, comparing predicted arcs and labels to the ground truth.

Comparison with Baseline 1:

- Baseline 1 is a structured dependency parser using biaffine scoring and span-to-token reductions (head-first/head-final); Baseline 2 skips syntactic modeling and directly classifies token pairs.
- Baseline 2 does not use external linguistic features such as POS tags, lemmas, or contextual embeddings like mBERT, relying solely on learned token embeddings.
- The simplicity of Baseline 2 makes it lightweight and GPU-efficient, but it underperforms due to the lack of structured modeling and rich linguistic cues.

Dynamic Gating

To enhance the representation power of our sentiment graph parser, we introduce a dynamic gating mechanism for modality fusion within the BaseLSTM encoder. Unlike the prior implementation that simply concatenated different input features—such as word forms, POS tags, lemmas, character-level encodings, and ELMo/mBERT embeddings—our model introduces trainable scalar gates that control the relative contribution of each modality to the final representation.

Let the input sequence be of length T, and let $\mathbf{e}_i \in R^{T\times d_i}$ represent the embedding matrix for modality $i \in \{form, pos, lemma, char, elmo\}$. For each modality, we introduce a learnable scalar gate $g_i \in R$ initialized to 1.0 and optimized during training. The gated embedding is defined as:

$$\tilde{\mathbf{e}}_i = g_i \cdot \mathbf{e}_i$$

All gated embeddings are concatenated and passed through a layer normalization step:

This fused representation $\mathbf{e}_{merged}$ is then fed into the BiLSTM encoder, followed by residual projection and scoring layers.

Advantages:

- **Modality Adaptivity:** The model dynamically learns which input features are most relevant for the task and down-weights those that contribute noise or redundancy. This is particularly useful in multilingual settings where feature informativeness varies significantly across languages.
- **Improved Generalization:** Languages with rich morphology benefit more from character and lemma features, while resource-poor languages might rely more heavily on contextual embeddings. The gating mechanism naturally balances these trade-offs.
- **Training Efficiency:** Scalar gates introduce only a handful of additional parameters, making this enhancement both lightweight and effective.

Empirical Findings: Our experiments show consistent improvement in Sentiment Graph F1 scores when dynamic gating is used, both in monolingual and cross-lingual scenarios. These improvements confirm that allowing the model to attend selectively to high-utility modalities leads to more robust and scalable performance across diverse datasets.

GAT Model + Multi Head Attention

We enhance the base BiLSTM model with a Graph Attention Network (GAT) module that incorporates a multi-head attention mechanism. The enhanced module works as follows:

- **Multi-Attention GAT Layer:** The BiLSTM outputs are fed into a multi-head attention GAT which learns to capture non-local dependencies by attending to multiple aspects of the input simultaneously.
- **Layer Normalization & Feed-Forward Refinement:** After the GAT layer, outputs are refined with layer normalization and a feed-forward network, with a residual connection to retain original information.

The following diagram (Figure 1) illustrates the overall architecture of the enhanced module.

4.1 Graph Attention Networks (GAT) for Cross-lingual Learning

Graph Attention Networks (GATs) are a class of neural networks that operate on graph-structured data by assigning different weights

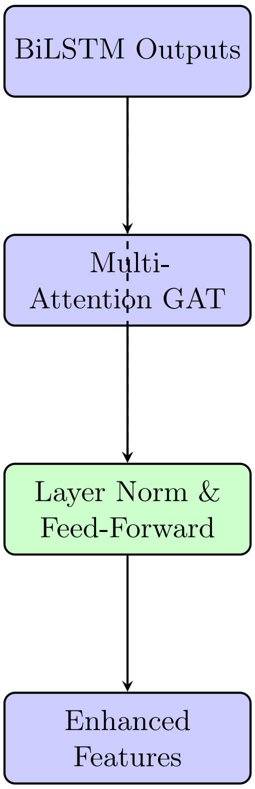

Figure 1: Architecture of the Multi-Attention GAT module. BiLSTM outputs are processed by a multi-head GAT module followed by layer normalization and a feed-forward network with residual connections, resulting in enhanced feature representations.

$$\alpha_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(a^\top [W h_i \parallel W h_j]\right)\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\mathrm{LeakyReLU}\left(a^\top [W h_i \parallel W h_k]\right)\right)}$$

Figure 2: Gat Attiention weights Computation

(attention scores) to each neighbor during message passing. In the context of cross-lingual Natural Language Inference (NLI), GATs are employed to model relational information between sentence representations across different languages.

How GAT Helps in Cross-lingual NLI: GAT captures the semantic similarity between multilingual sentence embeddings by treating each sentence as a node and constructing a graph based on pairwise cosine similarity. The attention mechanism allows the model to selectively focus on more relevant neighboring sentences (e.g., translations or semantically similar examples), thereby improving transfer learning performance.

Why It Works:

- GAT learns to propagate information from high-confidence nodes (e.g., English labeled examples) to low-resource languages.
- The attention scores help weigh contributions from multiple neighbors, reducing the noise from unrelated sentences.
- It supports label propagation, which is crucial when only a small fraction of nodes have labels.

Why It Works Even for Related Languages: Related languages (e.g., French, Spanish, Italian) tend to have similar syntactic and semantic structures. GAT leverages the shared embedding space of these languages (provided by models like XLM-R) and effectively transfers label signals through attention-weighted edges. This relational structure allows the GAT to generalize better, even when direct supervision is missing in target languages.

Our experiments show that applying this module on the BiLSTM representations improves the F1 scores by better capturing long-range dependencies and by dynamically focusing on the most relevant parts of the input during sentiment tuple extraction.

5 Dataset, Experimental Setup, and Results

5.1 Datasets

We evaluated our model on the SemEval 2022 Structured Sentiment datasets, including `mpqa`, `darmstadt_unis`, `multibooked_ca`, multibooked_eu, and norec. Each dataset is multilingual and annotated with sentiment tuples.

5.2 Setup

- Preprocessed JSON files were converted to CONLL-U format with sentiment role annotations.

- Embeddings were loaded from pretrained mBERT and zipped vector repositories (e.g., embeddings/58.zip for norec).

- Experiments were run on a single NVIDIA Tesla V100 GPU using PyTorch.

5.3 Extended GAT on Mono Models

Dataset	Sentiment Tuple F1
multibooked_ca	0.651
norec	0.359
multibooked_eu	0.678
opener_es	0.603
mpqa	0.272
opener_en	0.624
darmstadt_unis	0.309

Table 1: Evaluation Results for Extended GAT on Monolingual Models

5.4 Evaluation Metric

We used **Sentiment Graph F1**, computed by matching predicted and gold tuples on holder, target, expression, and polarity correctness.

5.5 Results Comparison

Train Dataset	Test Dataset	Baseline1	Baseline2	+ Gating	+ GAT
Monolingual Settings					
multibooked_ca	multibooked_ca_val	0.535	0.22	0.548	.651
opener_en	opener_en	0.521	0.28	0.540	.624
Cross-Lingual Settings					
opener_en	multibooked_eu	0.009	$\sim$0	0.186	.154

Table 2: Sentiment Graph F1 scores for various baselines across monolingual and cross-lingual settings. Baseline1 corresponds to the original paper. Baseline2 incorporates learnable structure embeddings. "+ Gating" adds dynamic learnable gating. "+ GAT" uses Graph Attention Networks.

5.6 Weights and biases

- **For GAT model weights and biases are in** weights

- **For Dynamic Learnable Gating Model weights and biases are in** weights

- **For Embeddings** Embeddings

6 Discussion and Observations

Our enhanced multi-head GAT module improves performance by capturing non-local dependencies and providing richer feature representations. Key observations include:

- The multi-head attention mechanism allows the model to focus on multiple aspects of sentiment simultaneously.

- The layer normalization and feed-forward refinement with residual connections help stabilize and enhance the learned representations.

- Error analysis indicates that the dynamic gating of attention focuses on subtle sentiment cues, reducing misclassification of implicit holders.

7 Conclusion and Future Work

We successfully reimplemented SSA as dependency parsing and demonstrated its effectiveness across multiple datasets. Future directions include:

- Joint training on sentiment parsing and syntactic parsing to enhance generalization.
- Semi-supervised learning using unlabeled multilingual corpora to mitigate low-resource challenges.

References

- Official SSA Repository
- Dependency Graph Codebase
- A Multimodal Coupled Graph Attention Network for Joint Traffic Event Detection and Sentiment Classification