Which Sentence Representation is More Informative: An Analysis on Text Classification

Necva Bölücü
Computer Engineering
Adana Alparslan Turkes
Science and Technology University
Adana, Turkey
nbolucu@atu.edu.tr

Burcu Can
Department of Computing Science and Mathematics
University of Stirling
Stirling, UK
burcu.can@stir.ac.uk

Abstract

Text classification is a popular and well-studied problem in Natural Language Processing. Most previous work on text classification has focused on deep neural networks such as LSTMs and CNNs. However, text classification studies using syntactic and semantic information are very limited in the literature. In this study, we propose a model using Graph Attention Network (GAT) that incorporates semantic and syntactic information as input for the text classification task. The semantic representations of UCCA and AMR are used as semantic information and the dependency tree is used as syntactic information. Extensive experimental results and in-depth analysis show that UCCA-GAT model, which is a semantic-aware model outperforms the AMR-GAT and DEP-GAT, which are semantic and syntax-aware models respectively. We also provide a comprehensive analysis of the proposed model to understand the limitations of the representations for the problem.

1 Introduction

The text classification problem has been widely studied in the literature (Yao et al., 2019; Malekzadeh et al., 2021) in the field of Natural Language Processing (NLP).

The text classification problem has been recently used as a downstream task in SentEval (Conneau and Kiela, 2018), a toolkit for evaluating sentence representations. In the literature, studies on Semantic Textual Similarity (STS) (Reimers et al., 2019; Gao et al., 2021) have used the text classification to evaluate the sentence embeddings learned by their proposed models using the datasets provided by the SentEval toolkit (Conneau and Kiela, 2018).

For text classification, traditional deep learning models such as Long Short-Term Memory (LSTM) Networks (Hochreiter and Schmidhuber, 1997) and Convolutional Neural Networks (CNN) (Kim, 2014) have been adopted. These deep learning models capture the local semantic and syntactic information by using the input as a sequence of words but they ignore the semantic and syntactic information of the input (Peng et al., 2018). Recently, Graph Neural Networks (GNNs) (Battaglia et al., 2018; Cai et al., 2018) have been used for text classification (Yao et al., 2019; Malekzadeh et al., 2021), sequence labeling (Marcheggiani and Titov, 2017; Zhang et al., 2018a), and question answering (Song et al., 2018; De Cao et al., 2019).

In dependency parsing the aim is to find a tree that represents dependencies between words in a sentence. On the contrary, semantic parsing maps a text to its formal representation that provides an abstraction of its meaning. There has been a recent increase in the studies that propose various neural network architectures such as tree-LSTM (Takase et al., 2016), Heterogeneous Graph Transformer (Li et al., 2020), and Transformer (Xie et al., 2021) that integrate semantic and syntactic information. GNN models that integrate external representations into deep learning models referred to as semantic and syntax-aware models, are the well-studied models in the literature for various NLP problems such as Neural Machine Translation (NMT) (Bastings et al., 2017) and text classification (Elbasani and Kim, 2022). These models have gained attention because they are capable of capturing information over long distances, especially between discontinuous constituents (Wang and Li, 2022).

In this study, we analyzed the impact of semantic and syntactic representations within Graph Attention Networks (GAT), particularly for the text classification problem. We used the dataset provided by SentEval toolkit (Conneau and Kiela, 2018). We constructed the GAT model by integrating Abstract Meaning Representation (AMR) (Banarescu et al., 2013) and the Universal Conceptual Cognitive Annotation (UCCA) (Abend and Rappoport, 2013) as graph-based semantic representations and the dependency tree as syntactic representation. Since the size of the datasets in SentEval toolkit (Con-

Proceedings of the Seventh International Conference on Dependency Linguistics (Depling, GURT/SyntaxFest 2023), pages 9 - 21

neau and Kiela, 2018) is different, we evaluated the results of our proposed model with the studies that use the SentEval toolkit (Conneau and Kiela, 2018). [1]

The rest of the paper is organized as follows. Section 2 reviews similar semantic and syntax-aware models. Section 3 describes our methodology for addressing the text classification problem using semantic and syntactic parser models. Section 4 presents our experimental results along with a detailed analysis of the proposed models. Finally, Section 5 concludes the paper with insights on the impact of the semantic and syntactic information on the classification problem.

2 Related Work

In addition to the traditional neural networks that simply rely on neural language models, semantic and syntax-aware models have been recently used effectively in NLP problems such as text classification (Ahmed et al., 2018; Huang et al., 2020; Liang et al., 2022; Elbasani and Kim, 2022), natural language generation (Guo et al., 2021), question answering (Schlichtkrull et al., 2020), semantic role labeling (SRL) (Schlichtkrull et al., 2020; Mohammadshahi and Henderson, 2021), reading comprehension (Sachan and Xing, 2016; Galitsky, 2020), text summarization (Takase et al., 2016; Dohare and Karnick, 2017), language modelling (Zhang et al., 2020), and machine translation (Qin and Liang, 2020; Slobodkin et al., 2021; Nguyen et al., 2021; Li and Flanigan, 2022).

Dependency trees usually provide sufficient syntactic information in various NLP tasks (Huang et al., 2020; Liang et al., 2022; Guo et al., 2021) and improve the performance of the models considerably. As for the external resource of semantic information, the most popular semantic representation is the AMR (Hardy and Vlachos, 2018; Elbasani and Kim, 2022; Kouris et al., 2022).

In particular, GNNs (Bastings et al., 2017; Marcheggiani and Titov, 2019; Schlichtkrull et al., 2020; Guo et al., 2021; Elbasani and Kim, 2022) have been used as models into which syntactic and semantic information are easily integrated. In addition to GNNs, Transformers have also been used to integrate such external resources such as syntax-aware word representation (SAWR) (Xie et al., 2021), syntax-aware local attention (SLA) (Li et al., 2020), syntax-graph guided self-attention (SGSA) (Gong et al., 2022), Scene-Aware Self-Attention (SASA), and Scene-Aware Cross-Attention (SACrA) head (Slobodkin et al., 2021). Last but not least, the Heterogeneous Graph Transformer (Hu et al., 2020), a customized version of the Transformer (Vaswani et al., 2017), has been recently introduced as a model with semantic AMR information (Yao et al., 2020).

3 Methodology

In this section, we describe the proposed semantic- and syntax-aware GAT models that integrate semantic and syntactic information as external resources into the model. First, we explain the preprocessing step that is performed to convert the text into the required form to be processed by the GAT model.

3.1 Preprocessing

GAT models use adjacency and feature matrices that are extracted from graphs as input. There are several approaches to transform a text into a graph, such as digitizing text (Hamid et al., 2020), statistical methods (PMI, TF-IDF) (Yao et al., 2019), dependency trees (Zhang et al., 2018b) or semantic graphs (AMR) (Elbasani and Kim, 2022).

In this study, we use dependency trees and semantic graphs. Here we explain the preprocessing step along with the parser model that is used to convert datasets into dependency trees and semantic graphs, as well as the details of extracting adjacency and feature matrices from graphs and trees.

Converting datasets into graphs/trees The parser models that are employed to extract the graphs and trees from the datasets are described below:

- **UCCA Semantic Parser** We use the self-attentive semantic parser model by Bölücü and Can (2021) to extract the UCCA-based semantic representations. The model is based on a graph-based approach with an encoder-decoder architecture, where the encoder is a Transformer (Vaswani et al., 2017) with 2 MLP classifiers and the decoder corresponds to the CYK algorithm (Chappelier and Rajman, 1998) that generates a constituency tree with the maximum score using the per-span scores obtained from the transformer encoder.

[1]The code is publicly available at `https://github.com/adalin16/depling-GAT`

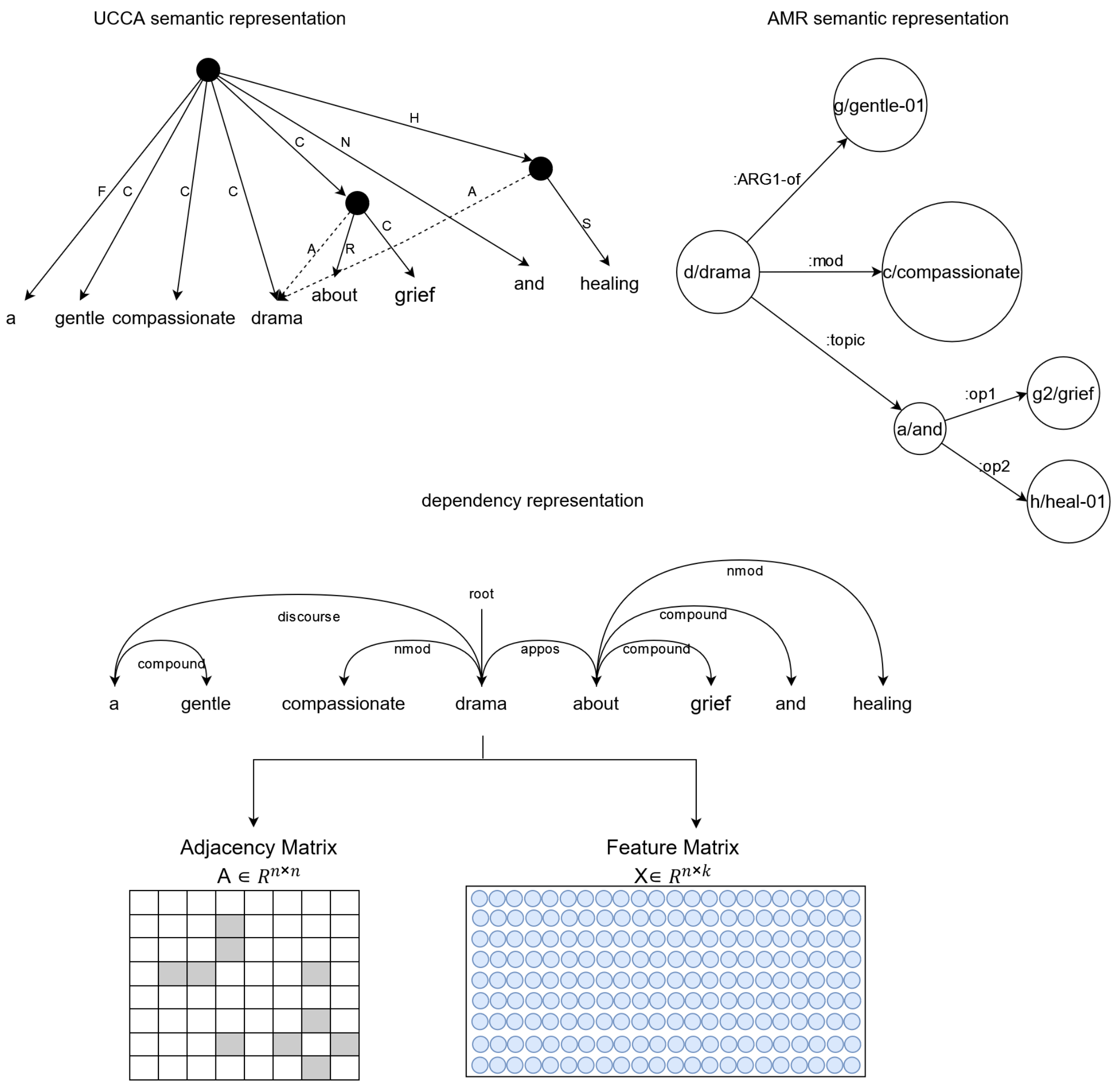


Figure 1: UCCA, AMR semantic graphs, and the dependency trees along with the feature and adjacency matrices that are used as input to the GAT model are illustrated for the example phrase *"a gentle compassionate drama about grief and healing"* from the MR dataset (Pang and Lee, 2005). The gray color in the matrix represents the value of 1 and the white color represents the value of 0. Each row in the feature matrix corresponds to the pre-trained word embedding of a node in the graph/tree.

- **AMR Semantic Parser** As an AMR semantic parser, we use the T5 parser (Roberts et al., 2020). The model is based on a language model that is fine-tuned on English. The model is integrated into the spaCy library (Honnibal and Montani, 2017) and is called AMRLib[2].

- **Dependency Parser** We use the Deep Biaffine dependency parser model Dozat and Manning (2016) to extract the dependency trees. The model is based on a graph-based approach where BiLSTM with biaffine classifiers is used as an encoder and MST is used as a decoder that generates dependency trees from predicted arcs and labels in the encoder. We use the model[3] integrated within the Stanza library (Qi et al., 2020).

Extracting adjacency and feature matrices from graphs/trees Since the inputs of the proposed model are adjacency and feature matrices, we extracted the matrices from graphs and trees. The semantic representations of UCCA and AMR are based on DAG, and the dependency trees are represented by trees. We followed the same proce-

[2]https://spacy.io/universe/project/amrlib

[3]https://stanfordnlp.github.io/stanza/depparse.html

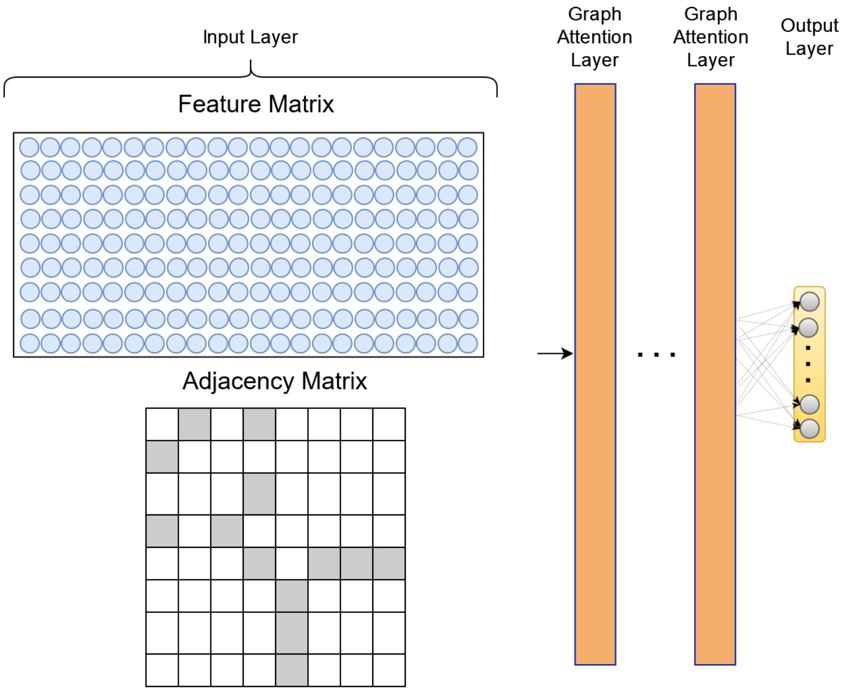


Figure 2: Overview of the GAT model along with its input in the form of a feature and adjacency matrix. The matrices correspond to semantic and syntactic information in the form of a UCCA or an AMR graph, or a dependency tree.

Dataset	Train	Dev	Test
Movie Review (MR) (Pang and Lee, 2005)	10,662	train in k-fold	test in k-fold
Customer Review (CR) (Hu and Liu, 2004)	3,770	train in k-fold	test in k-fold
Subjectivity / Objectivity (SUBJ) (Pang and Lee, 2004)	10,000	train in k-fold	test in k-fold
Multi-Perspective Question and Answering (MPQA) (Wiebe et al., 2005)	10,606	train in k-fold	test in k-fold
Stanford Sentiment Analysis 2 (SST-2) (Socher et al., 2013)	67,349	872	1,821
Text Retrieval Conference (TREC) (Voorhees and Tice, 2000)	5,452	train in k-fold	500
The Microsoft Research Paraphrase Corpus (MRPC) (Dolan et al., 2004)	4,726	train in k-fold	1,725

Table 1: The details of the datasets given in the downstream tasks in SentEval toolkit.

dure for all of the representations considering all as graphs.

For a given graph $G = (V, E)$, V is the set of nodes and E is the set of labeled edges (UCCA - edges, AMR - relations between nodes, dependency tree - dependency relations). We extracted:

- the feature matrix X ($n \times k$, where n is the number of nodes (UCCA - terminal and non-terminal nodes, AMR - words, dependency tree - words except the `ROOT` node) in the graph and k is the embedding dimension),
- the adjacency matrix A ($n \times n$, where n is the number of nodes in the graph), which is not trainable.

For the feature matrix, we used pre-trained word embeddings (BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019)) for nodes (UCCA - terminal nodes, AMR - words, dependency tree - words) and a randomly generated embedding with the same embedding dimension of the pre-trained word embeddings for non-terminal nodes in UCCA.

UCCA, AMR, and dependency tree representations of the phrase *"a gentle compassionate drama about grief and healing"* from the Movie Review (MR) dataset (Pang and Lee, 2005) with extracted adjacency and feature matrices are given in Figure 1.

3.2 Graph Attention Network

In order to incorporate external semantic information, we adopted Graph Attention Networks (GAT) (Veličković et al., 2017) that are based on self-attention layers. We used GATs for the text classification problem since they provide a straightforward method to utilise semantic information in the form of a semantic graph (UCCA/AMR) or a dependency tree. The overview of the model is given in Figure 2.

GNN models have different types of updating mechanisms for nodes. The basic version of updating, as applied in this study, updates each node i in the l-th layer, H^{l+1} as follows:

$$H^{l+1} = \sigma(AH^l W^l) \tag{1}$$

where $\sigma(\cdot)$ refers to ReLU non-linear activation function, A is the adjacency matrix, W^l is the attention weights in the l-th layer. H^l is the feature matrix of the l-th layer ($H^0 = X$, where X is the feature matrix extracted from a semantic graph or a dependency tree) where l is a hyperparameter that needs to be finetuned for the graph.

We fed the output of the node in the final layer into the output layer that applies the softmax function to generate the output class of a given text either as a binary or a multi-class classification:

$$Z = softmax(H^o) \tag{2}$$

where H^o is the feature matrix of the final GAT layer.

4 Experiments and Results

4.1 Datasets

We evaluated the model on 7 downstream tasks given in the SentEval toolkit (Conneau and Kiela, 2018). The details of the datasets are given in Table 1.

4.2 Experimental Setting

We used PyTorch 3.7 to implement the model. We used cross-entropy loss for both binary and multi-class classification. The Adam (Kingma and Ba, 2014) was used as the optimizer in all models with $\epsilon = 1e - 8$, and the default max grad norm for gradient clipping.

We used the monolingual (BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019)) and multilingual pre-trained language models (M-BERT (Devlin et al., 2019), XLM-R (Conneau and Lample, 2019), XLM-R-large (Conneau et al., 2020)) in order to build the feature matrices as described in Section 3.1. All hyperparameters along with their values are given in Appendix A.

We evaluated the models applied to binary and multi-class classification problems using the SentEval toolkit (Conneau and Kiela, 2018). We used accuracy metric in all downstream tasks and reported Precision, Recall, and F1 for a detailed analysis of the class-wise results for TREC.

4.3 Results

The results obtained from the semantic and syntax-aware GAT models (UCCA-GAT, AMR-GAT, and Dep-GAT) on 7 datasets in SentEval toolkit (Conneau and Kiela, 2018) along with the state-of-the-art results are given in Table 2. The results show that the performance of the GAT models is slightly behind the state-of-the-art results (Cer et al., 2018; Gao et al., 2021; Reimers et al., 2019). The main reason is that these models learn sentence embeddings and then apply the learned sentence embeddings to the downstream tasks (Reimers et al., 2019; Gao et al., 2021). Here, the main aim is to investigate the external usage of semantic and syntactic information without performing separate learning for sentence embeddings but solely relying on the existing semantic and syntactic information. Therefore, we only compare the performance of the semantic- and syntax-aware GAT models with each other for 7 downstream tasks. The results show that the UCCA-GAT model performs better than the AMR-GAT and the Dep-GAT models. The analysis of the adjacency matrices extracted from the AMR semantic parser and the UCCA semantic parser shows that the relations such as “about", “like", “of", etc. are defined as concepts and used as edge labels instead of nodes in the AMR representation. Since our models use the nodes without edge labels, the model misses the concepts that might give a clue about the target class. This also leads to sparse adjacency matrices for AMR graphs compared to other adjacency matrices extracted from UCCA graphs and dependency trees.

We analyse the class-wise results obtained from the three models using the TREC dataset (Voorhees and Tice, 2000) (multi-class classification problem). The results are given in Table 3. It can be clearly seen that UCCA-GAT is particularly good at predicting the classes “num" and “loc", since the number of relations in the UCCA graphs is higher in these classes than in other classes. The performance of AMR-GAT is worse than the other models (UCCA-GAT, Dep-GAT) because we lose the relations represented as labels in the AMR semantic representation and we used only the nodes in the semantic and syntactic representations in the preprocessing step during the extraction of the adjacency matrices for the AMR-GAT model. The Dep-

	Our proposed models							
	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC	Avg.
UCCA-GAT	82.04	83.37	90.38	87.29	89.35	81.92	73.50	83.98
AMR-GAT	81.55	81.11	88.98	83.94	85.83	79.65	72.87	83.42
Dep-GAT	80.66	81.62	89.10	85.76	88.03	81.06	75.25	83.07
	State-of-the-art							
BERT-CLS embedding ♡	78.68	84.85	94.21	88.23	84.13	**91.40**	71.13	84.66
BiLSTM ♢	81.1	86.3	92.4	90.2	-	-	-	-
Universal Sentence Encoder ♣	80.09	85.19	93.98	86.70	86.38	93.2	70.14	85.10
SimCSE-BERT$_{base}$ ♠	83.64	89.43	94.39	89.86	88.96	89.60	**76.00**	87.41
SBERT-NLI-large ♡	**84.88**	**90.07**	**94.52**	**90.33**	**90.66**	87.4	75.94	**87.69**

Table 2: Accuracy results of the downstream tasks using the proposed models and the other state-of-the-art models. The highest scores are given in bold. (♣ results from (Cer et al., 2018); ♠ results from (Gao et al., 2021); ♡ results from (Reimers et al., 2019); ♢ results from (Conneau et al., 2017))

	UCCA-GAT			AMR-GAT			Dep-GAT		
Class	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
num	0.97	0.89	**0.93**	0.90	0.84	0.87	0.91	0.82	0.87
loc	0.86	0.79	**0.83**	0.86	0.78	0.82	0.82	0.80	0.81
hum	0.80	0.80	0.80	0.74	0.80	0.77	0.77	0.85	**0.81**
desc	0.85	0.83	0.84	0.82	0.83	0.82	0.87	0.87	**0.87**
enty	0.70	0.85	0.77	0.77	0.83	0.80	0.81	0.87	**0.84**
abbr	0.86	0.67	**0.75**	0.64	0.78	0.70	0.67	0.67	0.67
avg.	0.84	0.81	0.82	0.79	0.81	0.80	0.81	0.81	0.81

Table 3: Class-wise results on the TREC dataset (Voorhees and Tice, 2000)

PLM	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC
	Monolingual Embeddings						
BERT	78.33	79.15	87.80	82.78	85.01	80.40	69.68
RoBERTa	80.16	79.89	89.11	**87.29**	**89.35**	79.11	72.52
XLNet	74.62	75.99	83.15	77.46	80.56	76.82	67.71
	Multilingual Embeddings						
M-BERT	79.27	81.94	88.15	83.11	83.14	81.00	72.35
XLM-R	**82.04**	82.23	89.48	84.76	85.01	**81.92**	72.93
XLM-R-large	78.78	**83.37**	**90.38**	85.82	87.59	81.42	**73.50**

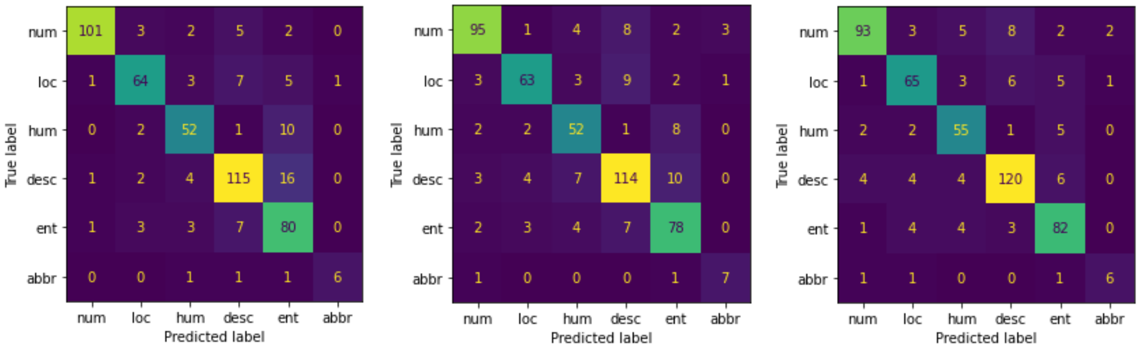
Table 4: Accuracy results obtained with monolingual and multilingual embeddings in UCCA-GAT model. The best values are in bold.

GAT model achieves better overall results since the dependency trees can capture long-distance information. The only class that Dep-GAT cannot capture is “abbr", compared to the success achieved with other classes in the TREC dataset (Voorhees and Tice, 2000).

Figure 3 illustrates the confusion matrices of the semantic and syntax-aware GAT models for the TREC dataset. The results show that the UCCA-GAT model predicts the class “num" better than other models. In addition, the Dep-GAT model is better at predicting the class “desc". For all models, there is a general confusion between the classes “desc" and “ent".

We also analyse the models deeply in terms of the impact of the layers and embeddings.

- **Embeddings** We present an analysis of the pre-trained language models used in the extraction of feature matrix X from UCCA, AMR, and dependency



(a) UCCA-GAT (b) AMR-GAT (c) Dep-GAT

Figure 3: Confusion matrices of the semantic and syntax-aware GAT models on TREC dataset (Voorhees and Tice, 2000)

PLM	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC
Monolingual Embeddings							
BERT	77.68	**81.11**	83.98	83.11	82.48	75.61	70.43
RoBERTa	**81.55**	79.44	85.44	83.44	**85.83**	**79.65**	70.78
XLNet	72.64	72.12	82.56	78.15	79.68	71.95	68.87
Multilingual Embeddings							
M-BERT	78.77	79.71	87.45	82.17	83.91	76.42	71.19
XLM-R	79.49	79.28	**88.98**	83.56	84.46	78.20	**72.87**
XLM-R-large	80.10	80.08	87.95	**83.94**	85.01	78.62	72.35

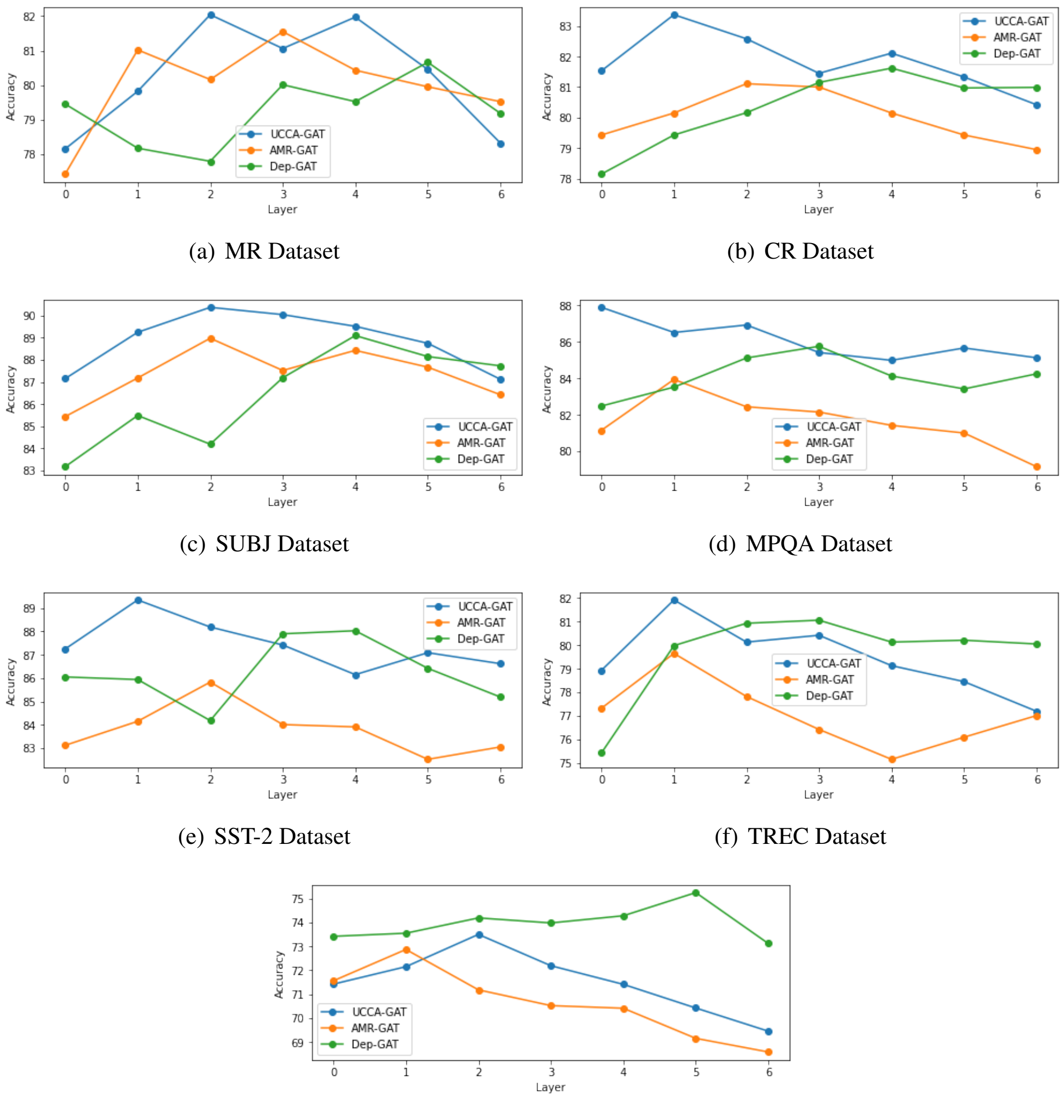
Table 5: Accuracy results obtained with monolingual and multilingual embeddings in AMR-GAT model. The best values are in bold.

PLM	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC
Monolingual Embeddings							
BERT	77.30	79.50	86.43	82.99	83.64	78.80	70.78
RoBERTa	78.95	**80.11**	89.10	83.14	**88.03**	79.62	71.59
XLNet	72.45	74.40	82.47	78.04	81.38	75.27	69.51
Multilingual Embeddings							
M-BERT	79.39	79.55	84.69	82.64	84.51	79.89	73.51
XLM-R	80.19	81.62	87.59	83.84	85.78	**81.06**	74.09
XLM-R-large	**80.66**	81.14	**88.11**	**85.76**	86.49	79.49	**75.25**

Table 6: Accuracy results obtained with monolingual and multilingual embeddings in Dep-GAT model. The best values are in bold.

tree. We used BERT (Devlin et al., 2019) (`bert-base-cased`), RoBERTa (Liu et al., 2019) (`roberta-base`), and XLNet (Yang et al., 2019) (`xlnet-base-cased`) monolingual embeddings with base variants consisting of 768 hidden dimensions, whereas we used multilingual version of BERT (M-BERT) (Devlin et al., 2019), and RoBERTa (XLM-R) (Conneau and Lample, 2019) and its large version (XLM-R-large) (Conneau and Lample, 2019).

The results obtained using monolingual and multilingual pre-trained embeddings are given in Table 4, 5 and 6 for UCCA-GAT, AMR-GAT, and Dep-GAT respectively. The re-



(a) MR Dataset

(b) CR Dataset

(c) SUBJ Dataset

(d) MPQA Dataset

(e) SST-2 Dataset

(f) TREC Dataset

(g) MRPC Dataset

Figure 4: Accuracy scores based on the number of layers in the proposed models.

sults show that multilingual embeddings are more effective for both proposed semantic and syntax-aware models. In monolingual embeddings, the results obtained from the models RoBERTa pre-trained word embeddings are higher than that of the others (BERT, XLNet).

- **Impact of the layers** We also analyse the impact of the number of layers in the proposed models (UCCA-GAT, AMR-GAT, Dep-GAT) on the performance of the models. We perform the experiments with embeddings with which we obtained the best results. We vary the number of the layers from 1 to 7 and report the results in Figure 4 for all datasets with UCCA-GAT, AMR-GAT, and Dep-GAT models. The results show that the syntax-aware model (Dep-GAT) learns in deeper layers, and semantic-aware models (UCCA-GAT and AMR-GAT) tend to learn in shallow layers or in the middle layers. The previous studies already show that syntactic features are encoded in the shallow layers and semantic features are encoded in the deeper layers of the pre-trained language models (Conneau et al., 2018; Jawahar et al., 2019), and here we also obtained better results with deeper layers in the syntax-aware model and with shallow layers in the semantic-aware models (UCCA-GAT and AMR-GAT).

5 Conclusion

Semantic and syntax-aware models have recently been proposed for various NLP problems, that especially require long-distance information, especially between discontinuous constituents, in addition to the local information captured by sequential models. In this paper, we propose a graph neural network model that incorporates semantic and syntactic information for the text classification task. To the best of our knowledge, this is the first study that compares semantic and syntactic information used in a graph neural network, specifically for the task of text classification. We present a detailed analysis of the results, showing that the UCCA semantic information improves the performance of the classification model compared to syntactic information (i.e. dependency tree). However, we were not able to obtain similar results with the model using the AMR semantic representation. This shows that the preprocessing step to convert the graph into adjacency and feature matrices is a very important step in GNN models.

As future work, we plan to improve the preprocessing step to obtain more informative adjacency and feature matrices.

References

Omri Abend and Ari Rappoport. 2013. UCCA: A Semantics-based Grammatical Annotation Scheme. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*, pages 1–12.

Usman Ahmed, Lubna Zafar, Faiza Qayyum, and Muhammad Arshad Islam. 2018. Irony Detector at SemEval-2018 Task 3: Irony Detection in English Tweets using Word Graph. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 581–586.

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.

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph Convolutional Encoders for Syntax-aware Neural Machine Translation. *CoRR*, abs/1704.04675.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, H. Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani, Kelsey R. Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew M. Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. 2018. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261.

Necva Bölücü and Burcu Can. 2021. Self-Attentive Constituency Parsing for UCCA-based Semantic Parsing.

Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. 2018. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. *CoRR*, abs/1803.11175.

J-C Chappelier and Martin Rajman. 1998. A generalized CYK algorithm for parsing stochastic CFG. In *Proc. of 1st Workshop on Tabulation in Parsing and Deduction (TAPD'98)*, CONF, pages 133–137.

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.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $ &!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. *Advances in neural information processing systems*, 32.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question Answering by Reasoning Across Documents with Graph Convolutional Networks. In *Proceedings of NAACL-HLT*, pages 2306–2317.

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.

Shibhansh Dohare and Harish Karnick. 2017. Text Summarization using Abstract Meaning Representation. *CoRR*, abs/1706.01678.

William B Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Ermal Elbasani and Jeong-Dong Kim. 2022. AMR-CNN: Abstract Meaning Representation with Convolution Neural Network for Toxic Content Detection. *Journal of Web Engineering*, pages 677–692.

Boris Galitsky. 2020. Employing Abstract Meaning Representation to Lay the Last-Mile Toward Reading Comprehension. In *Artificial Intelligence for Customer Relationship Management*, pages 57–86. Springer.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *CoRR*, abs/2104.08821.

Longchao Gong, Yan Li, Junjun Guo, Zhengtao Yu, and Shengxiang Gao. 2022. Enhancing low-resource neural machine translation with syntax-graph guided self-attention. *Knowledge-Based Systems*, 246:108615.

Qipeng Guo, Xipeng Qiu, Xiangyang Xue, and Zheng Zhang. 2021. Syntax-guided text generation via graph neural network. *Science China Information Sciences*, 64(5):1–10.

Abdullah Hamid, Nasrullah Sheikh, Naina Said, Kashif Ahmad, Asma Gul, Laiq Hassan, and Ala I. Al-Fuqaha. 2020. Fake News Detection in Social Media using Graph Neural Networks and NLP techniques: A COVID-19 use-case. *CoRR*, abs/2012.07517.

Hardy and Andreas Vlachos. 2018. Guided Neural Language Generation for Abstractive Summarization using Abstract Meaning Representation. *CoRR*, abs/1808.09160.

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

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*, pages 2704–2710.

Lianzhe Huang, Xin Sun, Sujian Li, Linhao Zhang, and Houfeng Wang. 2020. Syntax-aware graph attention network for aspect-level sentiment classification. In *Proceedings of the 28th international conference on computational linguistics*, pages 799–810.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization.

Panagiotis Kouris, Georgios Alexandridis, and Andreas Stafylopatis. 2022. Text summarization based on semantic graphs: An abstract meaning representation graph-to-text deep learning approach.

Changmao Li and Jeffrey Flanigan. 2022. Improving Neural Machine Translation with the Abstract Meaning Representation by Combining Graph and Sequence Transformers. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*, pages 12–21.

Zhongli Li, Qingyu Zhou, Chao Li, Ke Xu, and Yunbo Cao. 2020. Improving BERT with Syntax-aware Local Attention. *CoRR*, abs/2012.15150.

Shuo Liang, Wei Wei, Xian-Ling Mao, Fei Wang, and Zhiyong He. 2022. BiSyn-GAT: Bi-Syntax Aware Graph Attention Network for Aspect-based Sentiment Analysis. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics.

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. *CoRR*, abs/1907.11692.

Masoud Malekzadeh, Parisa Hajibabaee, Maryam Heidari, Samira Zad, Ozlem Uzuner, and James H Jones. 2021. Review of graph neural network in text classification. In *2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 0084–0091. IEEE.

Diego Marcheggiani and Ivan Titov. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515.

Diego Marcheggiani and Ivan Titov. 2019. Graph Convolutions over Constituent Trees for Syntax-Aware Semantic Role Labeling. *CoRR*, abs/1909.09814.

Alireza Mohammadshahi and James Henderson. 2021. Syntax-Aware Graph-to-Graph Transformer for Semantic Role Labelling. *CoRR*, abs/2104.07704.

Long HB Nguyen, Viet H Pham, and Dien Dinh. 2021. Improving neural machine translation with AMR semantic graphs. *Mathematical Problems in Engineering*, 2021.

Bo Pang and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278.

Bo Pang and Lillian Lee. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 world wide web conference*, pages 1063–1072.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. *CoRR*, abs/2003.07082.

Ying Qin and Ye Liang. 2020. Semantic Analysis and Evaluation of Translation Based on Abstract Meaning Representation. In *International Conference on Web Information Systems and Applications*, pages 268–275. Springer.

Nils Reimers, Iryna Gurevych, Nils Reimers, Iryna Gurevych, Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model? *CoRR*, abs/2002.08910.

Mrinmaya Sachan and Eric Xing. 2016. Machine comprehension using rich semantic representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 486–492.

Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. 2020. Interpreting Graph Neural Networks for NLP With Differentiable Edge Masking. *CoRR*, abs/2010.00577.

Aviv Slobodkin, Leshem Choshen, and Omri Abend. 2021. Semantics-aware Attention Improves Neural Machine Translation. *CoRR*, abs/2110.06920.

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 *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Linfeng Song, Zhiguo Wang, Mo Yu, Yue Zhang, Radu Florian, and Daniel Gildea. 2018. Exploring Graph-structured Passage Representation for Multi-hop Reading Comprehension with Graph Neural Networks. *CoRR*, abs/1809.02040.

Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on abstract meaning representation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1054–1059.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207.

Haitao Wang and Fangbing Li. 2022. A text classification method based on LSTM and graph attention network. *Connection Science*, 34(1):2466–2480.

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

Defeng Xie, Jianmin Ji, Jiafei Xu, and Ran Ji. 2021. Combining Improvements for Exploiting Dependency Trees in Neural Semantic Parsing. In *Pacific Rim International Conference on Artificial Intelligence*, pages 58–72. Springer.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Advances in neural information processing systems*, 32.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377.

Shaowei Yao, Tianming Wang, and Xiaojun Wan. 2020. Heterogeneous graph transformer for graph-to-sequence learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7145–7154.

Yue Zhang, Qi Liu, and Linfeng Song. 2018a. Sentence-state LSTM for text representation. *CoRR*, abs/1805.02474.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018b. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. *CoRR*, abs/1809.10185.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020. Semantics-aware BERT for language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9628–9635.

A Hyperparameter Values

Table 7, 8, and 9 list the hyperparameter values used in the UCCA-GAT, AMR-GAT and Dep-GAT models, respectively, for downstream tasks.

Parameters	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC
weight decay	0.2	0.1	0.2	0.2	0.2	0.1	0.1
batch size	1	1	1	1	1	1	1
learning rate	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
dropout rate	0.1	0.1	0.1	0.2	0.1	0.1	0.1
number of hidden	800	800	800	800	400	800	800
number of head	2	1	2	2	4	1	1

Table 7: Hyperparameters used for the UCCA-GAT for downstream tasks in experiments

Parameters	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC
weight decay	0.1	0.1	0.2	0.1	0.2	0.1	0.1
batch size	1	1	1	1	1	1	1
learning rate	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
dropout rate	0.2	0.1	0.2	0.1	0.2	0.1	0.1
number of hidden	800	400	800	800	800	400	800
number of head	2	1	2	2	4	1	1

Table 8: Hyperparameters used for the AMR-GAT for downstream tasks in experiments

Parameters	MR	CR	SUBJ	MPQA	SST-2	TREC	MRPC
weight decay	0.1	0.1	0.2	0.1	0.2	0.1	0.1
batch size	1	1	1	1	1	1	1
learning rate	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5
dropout rate	0.1	0.1	0.2	0.1	0.2	0.1	0.1
number of hidden	800	400	800	800	800	400	800
number of head	2	1	2	2	4	1	1

Table 9: Hyperparameters used for the Dep-GAT for downstream tasks in experiments