Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 6 Issue 01
2023

Research Article
https://doi.org/10.56979/601/2023

Sentiment Analysis in Movie Reviews Using Knowledge Graph Embeddings and Deep Learning Classification

Khalid Hussain[1], and Imran Ihsan[1*]

[1]Department of Creative Technologies, FCAI, Air University, Islamabad, Pakistan.
[*]Corresponding Author: Imran Ihsan. Email: imranihsan@gmail.com

Received: August 12, 2023 **Accepted:** December 01, 2023 **Published:** December 05, 2023

Abstract: Movie reviews are a good source of deciding whether the movie is worth the time or not, but going through all the reviews manually is a laborious and time-wasting effort. Automatic analysis of movie reviews can help in the reduction of this human effort. To create this automatic analysis, various techniques are available. However, the most promising technique is Sentiment Analysis. Sentiment Analysis can classify a movie review as positive or negative. Traditionally, Sentiment Analysis of movie reviews is performed using different Machine Learning or Deep Learning models. But with the advent of Knowledge Graphs, some experiments are available that are using Knowledge Graph and Knowledge Graph Embedding approaches for Sentiment Analysis. This research attempts to integrate two techniques: Deep Learning and Knowledge Graph Embedding to get the sentiments of the Text. The process includes knowledge graph embedding in Large Movie Review Dataset as an input to the Transformer-based model BERT. In the end, a comparison is done with BERT itself and two-hybrid Deep Learning model. For experiments, we are using the Large Movie Review Dataset. The dataset is equally distributed into positive and negative classes.

Keywords: Senitment Analysis; Knowledge Graph; Knowledge Graph Embedding; Bert.

1. Introduction

Sentiment Analysis (SA) is a technique of text analysis to extract methodically, identify, study effective states, and quantify personal information [1]. SA is to get the polarity of the data, generally in 2 classes: positive and negative. SA is a rapidly expanding area of natural language processing. SA approaches are unquestionably one of the most critical grounds in decision-making. Generally, SA is performed using textual data like surveys, reviews, customer feedback, and many more. SA's primary purpose for any business is to understand the customer's needs [2]. SA depends upon data, it may be via any digital platform, or it can be in the form of surveys, opinion pools, or any other way. Businesses can increase profit by knowing customers' honest opinions [3]. In the case of digital platforms, commonly known as social media platforms, people post their reviews on various websites and discuss their opinions (sentiments). These sentiments are essential in making decisions regarding any business or product. This approach can be most important for companies that work with vast amounts of data to launch their products successfully [4]. SA is usually used for polarity, but it can also be used for detecting a few other things like emotions, feelings, intentions, and urgency [5].

Commonly SA is performed through Deep-Learning or Machine-Learning approaches. Deep Learning models have shown better results as compared to Machine-Learning. Therefore, Deep learning has played a vital role in SA. Most Deep learning models learn features from data mining through embeddings. These embeddings are vector representations of the text. Moreover, these days in Deep Learning, there is a new model called Transformers which are pre-trained models. These models have an architecture that transforms any sequence by using two parts: An encoder and Decoder[6]. Transformers use an attention mechanism that allows them to provide context to any position of the input sequence [7]. Thus, Transformers work the same as many other RNNs yet differ because, unlike any RNN, Transformers do not start from the beginning of input. It starts according to the context [8]. The most commonly used Transformers are BERT, RoBERTa, DistilBERT.

Besides Deep Learning models, SA can be performed using Knowledge Graphs. Knowledge graph is basically a semantic relationship between two entities. Knowledge Graph has a unique way of gathering structured knowledge from texts and images for semantic analysis [9]. Knowledge graphs' way of representing information has made a lot of impossible tasks possible, like, recommendation systems, information retrieval, and question answering. Knowledge graphs were used for SA by Google for the first time. Google's KG is used to disambiguate and identify entities in the text, which is used for searching [10]. Every Google user has experienced the potential of Google's KG. Whenever we search for something on Google, we see relevant searches based on the context of the information we had just searched [11].

Knowledge Graph can help us get the embeddings as well. These embeddings are called Knowledge Graph Embeddings (KGE). KGE techniques effectively convert high-dimensional scattered graphs into low-dimensional, continuous, and dense vector spaces. These methods perform all the above tasks while keeping most of the graph-structure properties [12]. KGE preserves structural information, which means it converts entities and their relationships into vector space. KGE models establish different score functions. Score functions measure the distance between 2 entities regarding their relation type in embedding space. They are usually trained using KGE to reduce the distance between entities and their relationships.

Deep learning models require embedded inputs formulated through embedding techniques. The most used embedding technique can be said a word-embedding technique called Word2Vec. What word2Vec does is it simply transforms a word into a vector without realizing that word's sentiment polarity. In other words, Word2Vec doesn't get the sentiment of the text. Transformer-based models have already overcome this limitation because they are pre-trained models for thousands of features. However, we believe that Knowledge Graph can also help because we know that KGE can overcome problems such as homonymy and polysemy. Although KGE is an active research area [13]. There is a lot of research gap concerning Knowledge Graph Embeddings for SA. Some published work can be seen in section 2. However, most of these experiments have used the Twitter dataset for experiments with hardly any experiment on the Large Movie Reviews Dataset. The goal of this research is to check how well the Knowledge Graph Embedding technique performs along with Transformer based Deep Learning model for Sentiment Analysis on Large Movie Review Dataset.

The flow of the paper is: Section, 1 Introduction, this section includes the general introduction of the concepts, models, and algorithms used for SA covering different Machine Learning, Deep Learning, and Knowledge Graph approaches. Following is Section 2, That's Related Work. The latter Section 3, is Methodology. In this section, we have gone through our Methodology. Then comes Section 4 Experiments, here Experiments are explained. later section 5 is Results and their comparison. Here, we have described the results. Finally, the last Section 6 is the Conclusion.

2. Related Work

In this section, we have provided related work based on Sequential-based Deep Learning, Transformer-based Deep Learning, and Knowledge Graph based models for Sentiment Analysis.

2.1. Sequential-based Deep Learning

Traditionally researchers have worked on many machine-learning algorithms. However, as technology advances, simple machine learning algorithms are either being omitted due to their limitations or merged with deep learning algorithms for better accuracy and performance. Nevertheless, nowadays, most researchers use Deep Learning Techniques to build their models. So far, in the past few decades, LSTM has been the preferred architecture of researchers for SA. We will shed light on a few LSTM models, [14] have shown that when a document is divided into two parts, its sentiment classification's performance can be in- creased, for this they have proposed" Attention-based Hierarchical LSTM" with an accuracy of 91.51. [15] have proposed a training strategy that improves BiL- STM's performance. They trained BiLSTM with cross-entropy loss (max: likelihood), which proved to increase the overall performance of the BiLSTM model. [16] have compared the famous Deep Leaning techniques like CNN, LSTM, and LSTM-CNN on the

Large Movie Reviews Dataset and have shown that CNN outperforms. [17] have even compared LSTM's activation functions and have shown that some unpopular activation functions like Elliott, Softsign, and Modified Elliott perform better than the popular ones. [18] have proposed a Hybrid CNN-LSTM model for SA of the Large Movie Reviews Dataset. This model gets the word-by-word semantics along with their long-term dependencies. [19] have diminished the limitation of the LSTM model by acquiring and allocating attention from both backward and forward layers with an accuracy of 90.91%. [20] they have proposed a model that can auto-tune its hyper-parameters, which is Domain independent and can handle big social data. They have used CNN in the pooling layer, and LSTM's use is to capture extended dependency between words in a sentence and has achieved an accuracy of about 94.96. [21] Proposed self-attention mechanism this is done by joining various feature vectors with the BiLSTM model's output. Thus, giving various weights to different words results in obtaining contextual information.

2.2. Transformer-based Deep Learning

Although LSTM has been used a lot in the past, most researchers are using Transformer-based models these days. As its name suggests, the transformer model transforms any text sequence using Encoder and Decoder. Transformers use attention mechanism [7] to provide position-wise context. [22] have introduced a model that detects the interactions between features. They have ap- plied this approach with Machine Learning, Deep Learning, and Transformer models, i.e., LSTM, CNN, and BERT. This model-agnostic method HEDGE used word embeddings making top-down explanations through detecting feature interactions. BERT has achieved the highest accuracy among other models, i.e, 93%.

2.3. Knowledge Graph

As far as Knowledge graphs are concerned, very few models have been proposed for SA with the Knowledge Graph. [23] have proposed a model ConvE. This model predicts the missing links in a graph and uses many layers of non-linear features and embeddings to construct a knowledge graph. [24] have proven that Knowledge Graphs can be unified with popular Deep Learning algorithms, like LSTM and BiLSTM. Their proposed approach has an F-measure of 0.88 and 0.75, respectively. More research needs to be done on Knowledge Graph Em- bedding for SA. [13] have surveyed different Knowledge Graph Embedding algorithms like TransE, TransR, TransH, RASCAL, ComplEx, Etc. [25] their model catches the sentiment features with the help of KGE and external knowledge to augment-semantic information. [26] they have got aspect features and have pro- posed a new model that has used Knowledge Graph Embedding. Their model integrates external knowledge, semantics, and syntax on Aspect Level Sentiment Classification. [27] they have proposed a model that got the Dependency of Relation through Embedding, and this model can fully utilize the syntax information through a well-designed Dependency Relation Embedded Graph Convolutional Network. [28] Amazon-reviews, and they got domain invariant features and proved that the domain invariant features can be learned using a Graph- Convolutional auto-encoder. Their model KinGDOM ("Knowledge-Guided Do- main adaptation for SA") has an accuracy of 84.0.

3. Methodology

The flow diagram of the proposed model can be seen in Figure 1. Our primary goal is to find the efficiency of the Knowledge Graph Embedding technique along with Deep Learning mode. For this purpose, we have compared our model with 3 other models. We have explained all the steps in detail below.

3.1. Dataset

IMDB Large Movie Review Dataset is used. This dataset contains 50000 reviews with two classes (positive and negative) and this data has a 50:50 ratio of the polarity, which means that this dataset is equally distributed between its classes. This dataset contains a vocabulary of around 15000 words.

3.2. Preprocessing

As our dataset is human-written, it is bound to contain noise. By noise, we refer to short words, emojis, special symbols, punctuations, a little bit of HTML syntax, and many words with abbreviations. That's why we need to clean our data by preprocessing it. Data preprocessing is altering or dropping data before its usage to increase performance.

3.3. Knowledge Graph

The name "knowledge graph" was coined since this material is typically kept in graph databases and shown as a graph structure. Basically, a Knowledge Graph is a semantic relationship between two-entities.

It consists of triples(h,t,r). We got the Triples by getting the Entities and their Relationships.

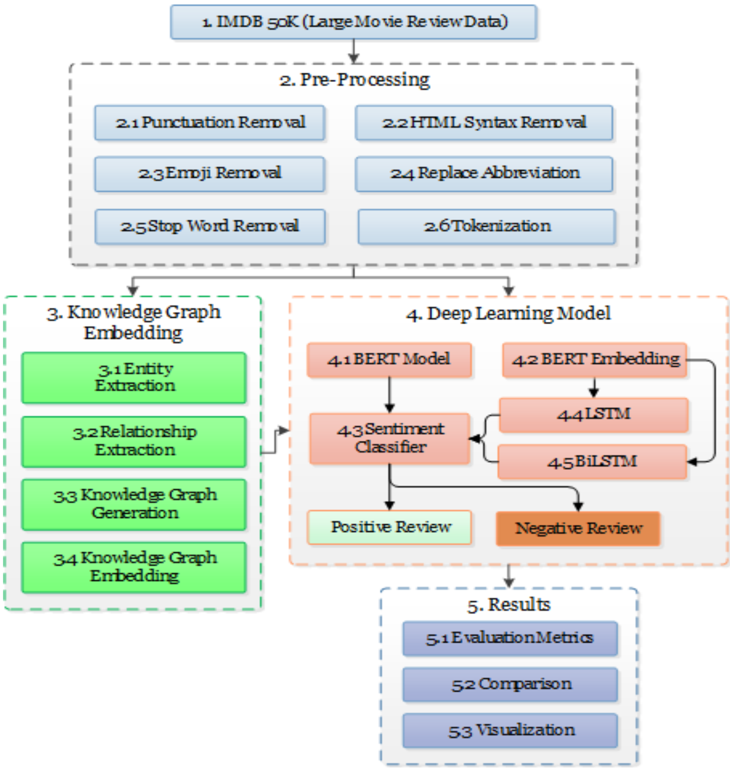

Figure 1. Flow Diagram of the model

3.4. Knowledge Graph Embedding

We have used the TransE method to compute Knowledge Graph Embeddings. TransE represents Relations and Entities as vectors in the same embedding space of the Rd dimension. This model represents Relationships by interpreting translating operations on learned low-dimensional embeddings of the knowledge graph Entities rather than directly on the graph structure. When a subject, relation, and Object, "fact" is true in TransE, the object entity's embedding should be close to the subject entity's embedding plus a vector representing the relationship type. This means that in embedding space translation vector embeds entities together that have little distance relation(r). The learned embedding should only ensure that Subject + Relation = Object if the "fact" is correct and that Subject + Relation is distant from the Object if the "fact" is false.

3.5. Deep Learning Model

We have chosen BERT as our base model. Because it's a pre-trained model and this model has achieved excellent performance for Sentiment Analysis. BERT has an advantage since it creates word representations that are influenced dynamically by the words surrounding them. In this step of our methodology, we have performed Sentiment Analysis. SA is performed on the models, i.e., KGE + BERT, BERT, BERT embedding + LSTM and BERT embedding + BiLSTM. The later models are used for comparison purposes.

4. Experiments

4.1. Dataset Division

The dataset was divided into two separate groups that contain 80% and 20% of the total no of reviews in our dataset. Simply put 80% is used for training and 20% for testing. As our dataset described above is equally distributed with high polarity.

4.2. Data Preprocessing

We have used multiple libraries to perform our preprocessing task. For regular expressions, regex is used. Regex is a tool to deal with regular expressions. It provides massive help in cleaning the data. Cleaning processes like Removing Stop Words, Removing Emojis, Removing Punctuation, Removing HTML syntax, removing emojis, removing emojis, and replacing abbreviations are per- formed. After cleaning the dataset, we performed Tokenization. Tokenization breaks the text into small parts known as tokens.

4.3. Knowledge Graph

To create a Knowledge Graph we need triples which are basically Entities and Relations between them. We got the Entities by getting the subject and objects from the dataset using the spaCY library. We got the Relationship between Entities by using the predicate rule. Finally, we have the triples to create a Knowledge Graph and it can be seen in Figure 2.

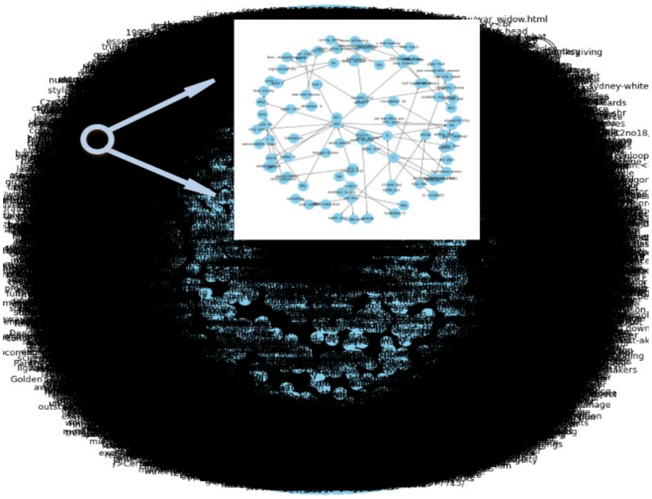

Figure 2. Knowledge Graph

4.4. Knowledge Graph Embedding

After the creation of the Knowledge Graph, we got the Embeddings through the Knowledge Graph Embedding technique called TransE. Selected method TransE is used to assign vectors to node types and nodes. For our triples (Subject, Object, and Relation), we have reduced the distance between the subject vector's translation along the relation vector and the object vector by using TransE. Sample of KGE can be seen in Figure 3

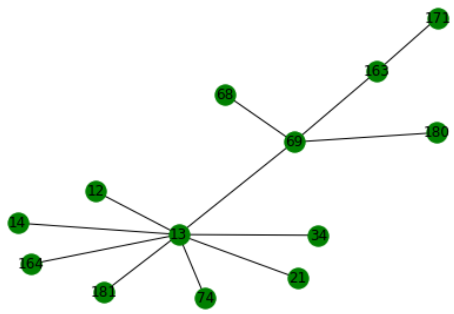

Figure 3. Sample of Knowledge Graph Embedding

4.5. Deep Learning Model

In this step, we have performed Sentiment Classification multiple times. At first is for our hybrid model i,e, KGE + BERT, and the rest are different Deep Learning models with different embeddings(BERT, BERT embedding + LSTM, and BERT embedding + BiLSTM) to compare the performance of our model.

4.5.1. Knowledge Graph Embedding + BERT

For this model, we have passed the triples as an input to BERT and we have added the Knowledge Graph Embedding technique TransE in BERT's Embedding layer before its own Token Embedding. After that, we divided our dataset for training, testing, and validation. Later we added a Classification layer on top of the outputs of the Trans- former and used the Softmax Classifier in the last layer. The weights de- cided are explained in Table 1. The results can be seen in Section 5.1.

Table 1. KGE + BERT hyperparameters

Model	KGE + BERT
Batch-Size	16
Drop Out	0.1
Epochs	5
Optimizer	AdamW
Activation Function	Softmax

4.5.2.BERT

After preprocessing, we move to the Deep Learning model. We have used the transformer-based approach BERT to perform SA on our preprocessed data without Knowledge Graph Embedding. After getting the data, BERT performs 2 embeddings, Token Embedding and Contextual Embeddings (Positional and Sentence). After performing Embedding we divided our dataset for training, testing, and validation. Later we added a Classification layer on top of the outputs of the Transformer and used the Softmax activation function in the last layer. The weights decided are explained in Table 2.

Table 2. BERT's hyperparameters

Model	BERT
Type	BERT Large
Max-Length	300
Drop Out	0.1
Batch-Size	16
Epochs	10
Optimizer	AdamW
Activation Function	Softmax

4.5.3. Bert Embedding + LSTM and Bert Embedding + BiLSTM

From the BERT model, we got the embeddings. We have used these Embeddings as input into LSTM and BiLSTM. After that, we divided our dataset for training, testing, and validation. Later we added a Classification layer on top of the outputs of the Transformer and used the Softmax Classifier in the last layer. The weights decided are explained in Table 3.

Table 3. BERT Embedding + LSTM and BERT Embedding + BiLSTM hyperparameters

Model	LSTM & BiLSTM
Embeddings	BERT Embedding
Max-Length	300
Drop Out	0.1
Batch-Size	12
Epochs	10
Optimizer	AdamW
Activation Function	Softmax

5. Results

In this section, the results are described. First, our proposed model KGE + BERT then, base model BERT's result for sentiment analysis is described.

5.1. KGE + BERT

After using KGE with BERT and fine-tuning the final model, its results can be seen in table 4. The average accuracy achieved is 54%. For positive sentiments, the precision, recall, and f1-scores are 54%, 61% and 57% respectively. For negative sentiments, the precision, recall, and f1-scores are 55%, 48% and 51% respectively.

Table 4. KGE + BERT Results

	P	R	F1-Score
Negative	55	48	51
Positive	54	61	57
marco avg	54	54	54
weighted avg	54	54	54

5.2. BERT

The overall performance of BERT is 94%. For positive sentiments, the precision, recall, and f1-scores are 95%, 93% and 94% respectively. For negative sentiments, the precision, recall, and f1-scores are 93%, 95% and 94% respectively. Bert Embedding + LSTM results are explained here. The average accuracy achieved is 91%.

Table 5. Bert Results

	P	R	F1-Score
Negative	93	95	94
Positive	95	93	94
marco avg	94	94	94
weighted avg	94	94	94

5.3. BERT Embedding + LSTM

For positive sentiments, the precision, recall, and f1-scores are 88%, 94%, and 91% respectively. For negative sentiments, the precision, recall, and f1-scores are 90%, 90%, and 90% respectively.

Table 6. Bert Embedding + LSTM Results

	P	R	F1-Score
Negative	94	88	91
Positive	88	94	91
marco avg	91	91	91
weighted avg	91	91	91

5.4. BERT Embedding + BiLSTM

The average accuracy achieved is 92%. For positive sentiments, the precision, recall, and f1-scores are 91%, 94% and 92% respectively. For negative sentiments, the precision, recall, and f1-scores are 94%, 91% and 92% respectively.

Table 7. Bert Embedding + BiLSTM Results

	P	R	F1-Score
Negative	94	91	92
Positive	91	94	92
marco avg	92	92	92
weighted avg	92	92	92

6. Conclusion.

In this work, we proposed a model that joins syntax and semantics for Sentiment Analysis using Large Movie Review Dataset. Focusing on successfully getting the syntax and semantics, we have created a Knowledge Graph of our dataset to get the embeddings through the Knowledge Graph Embedding technique called TransE. Furthermore, we have fused our Knowledge Graph Embeddings with the Transformer-based model BERT to perform Sentiment Analysis. Finally, we have used Softmax Classifier to get the Classification. For future work: we can improve the performance of our model by incorporating it with any movie reviews related to ontology to enrich our Knowledge Graph with more astounding entities and their relationships. As far as this model's application use, we can use it for Product Analysis by discovering the public's opinion about any product (new/old). It can also be used for Market Research by analyzing the different marketing strategies used in the industry and help in reaching the target audience.

References

1. B. Liu, “Sentiment analysis and opinion mining,” Synthesis lectures on human language technologies, vol. 5, no. 1, pp. 1–167, 2012.
2. “Sentiment analysis.” [Online]. Available: https://monkeylearn.com/ sentiment-analysis/
3. “What is sentiment analysis and what are its benefits for busi- ness? — litslink blog.” 2022. [Online]. Available: https://litslink.com/blog/ what-is-sentiment-analysis-and-how-can-you-apply-it-to-benefit-your-business
4. “Top 4 types of sentiment analysis where to use — upgrad blog.” 2022. [Online]. Available: https://www.upgrad.com/blog/types-of-sentiment-analysis/
5. “Types of sentiment analysis and how brands perform them,” 2022. [Online]. Available: https://www.analyticsinsight.net/ types-of-sentiment-analysis-and-how-brands-perform-them/
6. “what is a transformer?.an introduction to transformers and.— by maxime — inside machine learning” 2022. [Online]. Available: https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04
7. A. Vaswani, ““attention is all you need,”,” Advances in Neural Information Pro- cessing Systems, vol. 2017, p. 5999–6009, 2017.
8. ““sentiment analysis — comprehensive beginners guide — thematic — thematic.”,” 2022. [Online]. Available: https://getthematic.com/sentiment-analysis/
9. D. B. T. C. F. Lovera, Y. Cardinale and M. N. Homsi, ““deep learning enhanced with graph knowledge for sentiment analysis,”,” 2021. [Online]. Available: http://ceur-ws.org
10. ““google launches the knowledge graph- 2012 - wordlift blog.”,” 2022. [Online]. Available: https://wordlift.io/blog/en/entity/ google-launches-knowledge-graph-2012/
11. ““what is google’s knowledge graph?”,” 2022. [Online]. Available: https://yoast.com/google-knowledge-graph/
12. M. Xu, “Understanding graph embedding methods and their applications,” SIAM Review, vol. 63, no. 4, pp. 825–853, 2021.
13. S. Choudhary, T. Luthra, A. Mittal, and R. Singh, “A survey of knowledge graph embedding and their applications,” arXiv preprint arXiv:2107.07842, 2021.
14. B. Wang and B. Fan, “Attention-based hierarchical lstm model for document sentiment classification,” vol. 435, no. 1, 2018.
15. M. Z. D. S. Sachan and R. Salakhutdinov, “Revisiting lstm networks for semi- supervised text classification via mixed objective function,” p. 6940–6948, 2019.
16. M. l. S. J. Gao, J. Lanchantin and Y. Qi, “Black-box generation of adversarial text sequences to evade deep learning classifiers,” pp. 50–56, 2018.
17. H. M. A. Farzad and H. Hassanpour, “A comparative performance analysis of different activation functions in lstm networks for classification,” vol. 31, no. 07, p. 2507–2521, 2019.
18. B. R. A. U. Rehman, A. K. Malik and W. Ali, “A hybrid cnn-lstm model for improving accuracy of movie reviews sentiment analysis,” vol. 78, no. 18, p. 26597–26613, 2019.
19. X. Z. Y. Wang, H. Wang, “An attention-aware bidirectional multi-residual recur- rent neural network (abmrnn): A study about better short-term text classification,” vol. 8, p. 3582–3586, 2020.
20. S. K. R. R. K. Behera, M. Jena and S. Misra, “Co-lstm: Convolutional lstm model for sentiment analysis in social big data,” vol. 58, no. 1, p. 102435, 2021.
21. M. T. W. Li, F. Qi and Z. Yu, “Bidirectional lstm with self-attention mechanism and multi-channel features for sentiment classification,” vol. 387, p. 63–77, 2020.
22. G. Z. H. Chen and Y. Ji, “Generating hierarchical explanations on text classification via feature interaction detection,” p. 5578–5593, 2020.
23. T. Dettmers, “Convolutional 2d knowledge graph embeddings”,” 2017. [Online]. Available: http://arxiv.org/abs/1707.01476
24. Y. C. C. F. A. Lovera and M. N. Homsi, “Sentiment analysis in twitter based on knowledge graph and deep learning classification,” vol. 10, no. 22, 2021.
25. Q. Zhong, L. Ding, J. Liu, B. Du, H. Jin, and D. Tao, “Knowledge graph augmented network towards multiview representation learning for aspect-based sentiment analysis,” arXiv preprint arXiv:2201.04831, 2022.
26. H. Yu, G. Lu, Q. Cai, and Y. Xue, “A kge based knowledge enhancing method for aspect-level sentiment classification,” Mathematics, vol. 10, no. 20, p. 3908, 2022.
27. Y. Liang, F. Meng, J. Zhang, Y. Chen, J. Xu, and J. Zhou, “A dependency syn- tactic knowledge augmented interactive architecture for end-to-end aspect-based sentiment analysis,” Neurocomputing, vol. 454, pp. 291–302, 2021.
28. D. Ghosal, D. Hazarika, A. Roy, N. Majumder, R. Mihalcea, and S. Poria, “Kingdom: Knowledge-guided domain adaptation for sentiment analysis,” arXiv preprint arXiv:2005.00791, 2020.