THROUGH THE WORDS OF VIEWERS: USING COMMENT-CONTENT ENTANGLED NETWORK FOR HUMOR IMPRESSION RECOGNITION

Huan-Yu Chen, Yun-Shao Lin, Chi-Chun Lee

Department of Electrical Engineering, National Tsing Hua University, Taiwan
MOST Joint Research Center for AI Technology and All Vista Healthcare, Taiwan

2021 IEEE Spoken Language Technology Workshop (SLT) | 978-1-7281-7066-4/20/$31.00 ©2021 IEEE | DOI: 10.1109/SLT48900.2021.9383564

ABSTRACT

Research into understanding humor has been investigated over centuries. It has recently attracted various technical effort in computing humor automatically from data, especially for humor in speech. Comprehension on the same speech and the ability to realize a humor event vary depending on each individual audience's background and experience. Most previous works on automatic humor detection or impression recognition mainly model the produced textual content only without considering audience responses. We collect a corpus of TED Talks including audience comments for each of the presented TED speech. We propose a novel network architecture that considers the natural entanglement between speech transcripts and user's online feedbacks as an integrative graph structure, where the content speech and online feedbacks are nodes where the edges are connected though their common words. Our model achieves 61.2% of accuracy in a three-class classification on humor impression recognition on TED talks; our experiments further demonstrate viewers comments are essential in improving the recognition tasks, and a joint content-comment modeling achieves the best recognition.

Index Terms— viewer comments, humor recognition, entangled network, speech content

1. INTRODUCTION

Humor plays an important role in human's social communication, e.g., positive humor styles and social competence have been shown to be positively correlated [1]. Research into seeking the reasons behind humor and laughter has dated back in the Classical period of Ancient Greece [2] [3]. There are three main theories of humor in modern literature [4]. The superiority theory states that people laugh at others misfortune with their superiority in their background [5]; the relief theory states that laughter and humor are used to release psychological tension and overcome social inhibitions; the incongruous theory indicates enjoyable events that are unexpected and violate our usual mental pattern are considered humorous [6]. There is a continuous scientific effort into unpacking what constitute a humor impression from both perspectives of production and perception. Interpretation (perception) of humor depends highly on an individual's mental state and experience. This effect is especially critical when planning on delivering a funny or humorous speech when the interpretation of the same content can vary depending on each audience's viewpoint. The clear evident, e.g., audience laughter, of humor events further underscores the importance of investigating the *audience* response in understanding humor.

Computational models have also been developed in computing humor using data often with an aim at advancing human-computer interface design [7], i.e., making computer more user-friendly, cleverly interact with humans, improving user experiences [8]. Many prior works have focused on automatically detecting humor from the produced content, specially focusing on the textual content. For example, Mihalcea and Strapparava collect a large one-liner dataset from the web and design stylistic features to detect humor [9]; Davidov et al. recognize sarcasm on tweets and product reviews by comparing surface word patterns [7]; Chen and Soo use convolution neural network on various text-based datasets [10], while Weller and Seppi use Transformer-based model [11] [12]; Chandrasekaran et al. focus on visual humor that is to predict how funny a scene is [13]. Most prior computational works on detecting humor use textual information concentrate on data of short sentences or dialogues; very few works have considered content such as public speech, sitcoms, and movies.

Our goal in this work is to detect viewer's rating of humor impression on public speech, specifically TED talks, using textual data. Aside from working on a new medium of humor messages, we propose to recognize humor impression by integrating both the produced speech content, i.e., transcript, and viewers responses, i.e., comments left online, using a novel graph network learning architecture. Online comments can be seen as a perceptive record from the user perspective after one experiences the media content. Many past works have utilized these online data in a variety of predictive tasks; for example, Lei et al. build a social-rating based recommender system by applying sentiment analysis on comments and reviews [14]; Jamali and Rangwala use the number and the length of comments to predict popularity scores on their service [15]. Specifically, we propose a comment-content entangled network that is composed of two sub graphs to perform a three-

class humor impression rating classification tasks. One of the sub graphs is from the TED talk transcript and another one is from the online comments, where the two graphs are further linked through their common words. We use 1618 talks from the top viewed TED Talks videos, where the task is defined base on the proportion of viewers that rates a given presentation funny. Our framework achieves 61.2% accuracy on this task, which is a 3.36% improvement over conventional content based textual features. We additionally examine the predicted humor level for the words in the TED talk corpus.

The rest of this paper is organized as follows: section 2 demonstrates the proposed framework, the dataset, and the method on features extraction; section 3 shows the details of our experiment setup and results; we summarize the proposed method and future work in section 4.

2. RESEARCH METHODOLOGY

2.1. TED Talks Dataset

TED is a global community under the mission of spreading ideas, looking for deeper understanding of the world. Under the slogan of "ideas worth spreading", conferences on inspiring ideas shared from a diversity of people are held all around the world. TED.com is an online archive that collects videos of profound and influential talks from those events, and viewers that are willing to engage in ideas with each other gravitate to this website that creates a large online community. Users can leave comments on the talks and exchange views with others in the discussion forum. Instead of simple like and dislike, users can also rate the talks with fourteen different impression keywords, i.e., beautiful, confusing, courageous, fascinating, funny, informative, ingenious, inspiring, jaw-dropping, long-winded, obnoxious, ok, persuasive and unconvincing to describe impressions on talks.

In this work, we collect the most viewed 1800 talks on the TED website. We download the videos, transcripts, comments and vote counts for all ratings of each talk. Other meta attributes like video length, tags and description are also included. Only the top-level comments are considered as the viewer's comment to the video, other discussion under others' comments are not taken into consideration in this work. Some talks which are not in English are eliminated, and other contents that are not in the form of speech such as music and magic performances are also manually removed. A total of 1618 talks remain in our dataset. Table 1 gives a summary of word statistics for the content and comment portion of this dataset.

Table 1. Statistics of words in the TED Talk dataset.

Avg. #words in each talk	Content	2024.9
	Comment	9063.2
Avg. unique #words in each talk	Content	577.7
	Comment	1584.3
Unique #words	Content	47071
	Comment	179680
Unique #words in Content and Comment		190745
Avg. #comments		110.4

2.2. Label Definition

Every keyword has its corresponding number of votes. Since the raw number of votes would correlate highly to the view count, we first perform normalization between the votes of different keywords for each single video to obtain the voting proportion of each keyword, i.e., the percentage of each keyword voted among all keywords. We focus on the keyword "funny" in this work. By comparing the percentage of "funny" in the dataset, the samples are split into "low", "medium", and "high" categories using cutoff point of 33% quantile and 66% quantile. The label can intuitively be interpreted as the extent of an aggregated degree of humor impression from the TED talk online viewing community.

2.3. Text Graph Convolutional Network

Graph convolutional network, proposed by Kipf and Welling [16], can efficiently model the information propagation on graphs and also learn representations of nodes. A single layer of graph convolution can aggregate the information from one-hop neighbors for each node. Considering a graph $G = (V, E)$ with N nodes $v_i \in V$, edges between nodes $(v_i, v_j) \in E$, a self-looped adjacency matrix $A \in \mathbb{R}^{N \times N}$, and a degree matrix $D_{ii} = \sum_j A_{ij}$, we describe a two-layer GCN for node classification as $f(X, A)$ on the data $X \in \mathbb{R}^{N \times C}$ with N nodes of C-dimensional feature vectors and the adjacency matrix A. The inference of the model can be written as

$$Z = f(X, A) = \text{softmax}(\hat{A}\text{ReLu}(\hat{A}XW^{(0)})W^{(1)}) \quad (1)$$

where $\hat{A}$ is the normalized symmetric adjacency matrix defined as $\hat{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ and $W^{(0)}, W^{(1)}$ are the learnable weight matrix for feature transformation in the first and the second layer, respectively. The objective function of a classification task of label Y is the masked cross-entropy error that only evaluated on the labeled training samples $\mathcal{Y}_L$.

$$L = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} ln Z_{lf} \quad (2)$$

Yao et al. further extend this framework to text based classification, termed as Text GCN [17]. The text graph is built with word nodes, document nodes, word-to-word edges, and word-to-document edges. The edge weights of the text graph is designed based on the dataset's global word co-occurrence

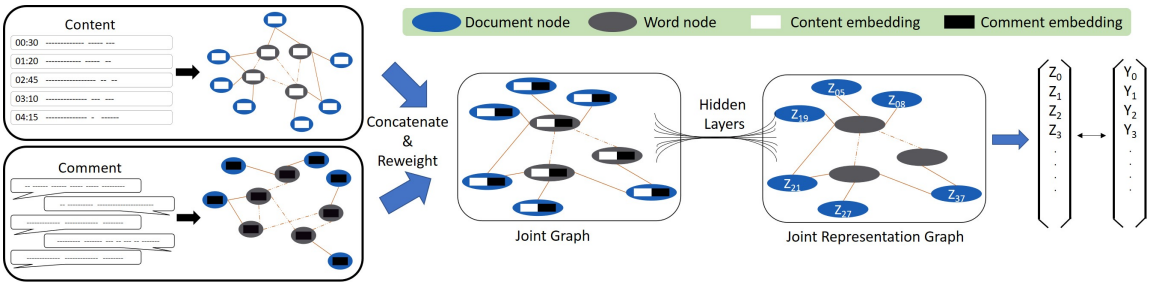

Fig. 1. The schematic of our proposed framework. We first build the content graph and comment graph. A graph includes document nodes, word nodes, and edges between nodes. We use BERT embedding initialization for node representation and custom edge weights are point-wise mutual information or TF-IDF based on type of the node. Then, the two graphs are merged into a joint embedding graph by concatenating the node embeddings and re-weighting the edges. This joint graph is fed into the graph convolutional network for training and evaluation of the given labels. The loss of classification task is the cross-entropy between last layer as decision score Z_i and label Y_i, for a document node i.

as follows:

$$A_{ij} = \begin{cases} \text{PMI}(i,j) & i,j \text{ are words, PMI}(i,j) > 0 \\ \text{TF-IDF}_{ij} & i \text{ is word and } j \text{ is document} \\ 1 & i = j \\ 0 & otherwise \end{cases} \tag{3}$$

where PMI stands for point-wise mutual information defined as:

$$\text{PMI}(i,j) = log\frac{p(i,j)}{p(i)p(j)} \tag{4}$$

where $p(i)$, $p(j)$, and $p(i,j)$ are the probability of appearance in the corpus for word i, word j, or both words, respectively. By splitting a document into sliding windows, the probability of appearance is defined as the number of windows containing the word divided by total number of sliding windows.

In this work, we utilize Text GCN as our main building block. We propose a joint graph that combines both content and comment representations as shown in Fig. 1. By feeding a single joint graph into graph convolutional network, we can improve the node embedding by considering both speakers' and viewers' perspective. First, the content graph and the comment graph are built separately, where each structure is similar to a TextGCN. For content graph, the transcript for each TED talk is defined as one document. We only take top 2000 most occurring words after removing stopwords with spaCy toolkit [18]. For comment graph, the whole comment section under one talk is treated as a document. The word nodes are also limited to the top 2000 words of most occurrence in all comments. Since the number of unique vocabulary words in comment space is much larger than the counterpart in content space, we take the same amount of words from both spaces. Intuitively, we want to equalize the influence from both spaces when performing graph combination and also to reduce graph size and complexity. After building both graphs, we further merge them into a joint graph by combining nodes and edges. The document nodes are merged by directly concatenating the textual feature embeddings from both spaces. The word nodes are chosen either through union or intersection of the two sets of word nodes. The feature vector of the merged word nodes are also vector concatenation from the two spaces. However, when encountering out of vocabulary issues on one side, we fill the representation with random normal vector of the same length. The edges are merged using average weight from two sides if both connectivity exists from content graph and comment graph.

2.4. Feature Extraction

In terms of text representation, we take a different approach from the one proposed in [17], i.e., using identity matrix which treats each node as orthogonal one-hot vectors. Since the relation between content and comment might not be unrelated, we select the pretrained BERT model provided by huggingface [19] for better initialized textual representation. BERT model would encode a single sentence into a sequence of word vectors. We use the word vector of each word as a building block and aggregate the lower representations into upper level by averaging the sequence of vectors. Hence, a sentence embedding is obtained by averaging the word embedding sequence, and a paragraph embedding is computed as averaging over sentence embeddings. The required four sets of feature vectors in our framework are word representation and document representation in both the content and comment space.

The content information is extracted from transcript of each talk, which is treated as a single paragraph. The content representation would be an aggregation through word embeddings and sentence embeddings to a single paragraph embedding. A single comment from a user is defined as a paragraph. Therefore, the whole comment section is treated as an upper level of paragraphs, which is composed of multiple indepen-

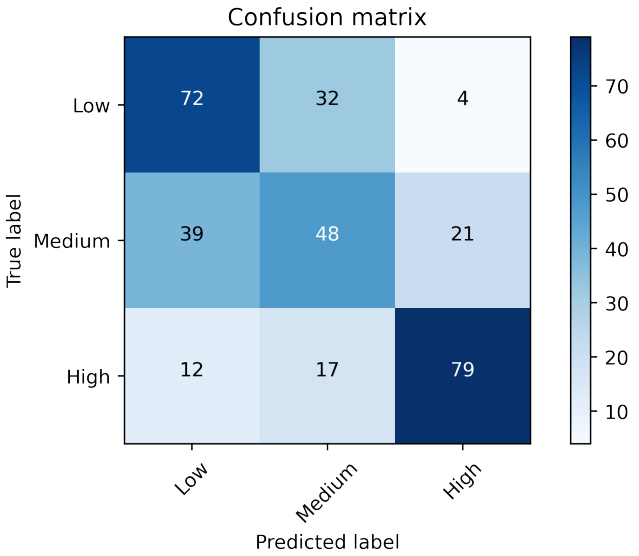

Fig. 2. The confusion matrix of the classification result using TextGCN(Union).

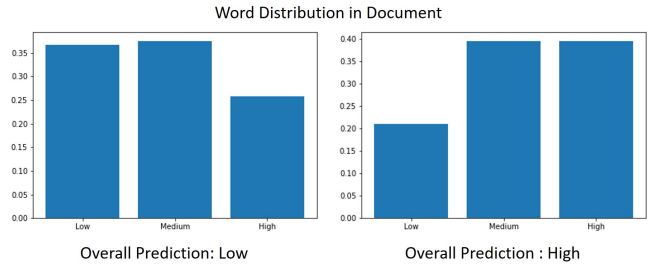

Fig. 3. The distribution of word prediction in each document sample from low and high class. We examine the word distribution in documents and their portion in "low" and "high" class. There are larger portion of words that are predicted as "low". The similar phenomenon occurs in the document predicted as "high", which has more words predicted as "high".

dent comments. The final representation of the whole comment section is the mean of paragraph embedding from each comment. In case of the word representation, it is the average of word vectors from the collection of multiple word appearances in the content corpus and comment corpus, respectively.

3. EXPERIMENT SETUP AND RESULTS

3.1. Experiment Setup

In this work, we perform a three-class classification on keyword "funny" in the TED dataset for humor impression recognition. The architecture of our network includes two graph convolution layers followed with a linear layer with dimension of $\{128, 16, 3\}$. ReLu activation function and batch-normalization are applied between graph convolution layers. We first split the samples into training and testing set with a 80% - 20% split. The hyper-parameters are optimized with a 5-fold cross validation within the training set. We then evaluate the model on the testing set, reporting accuracy as the final metric. The initial learning rate is set as 0.0001 with a scheduler decaying the learning rate by 0.25 every 500 epochs. We train the model using a total of 1000 epochs.

We compare our proposed method in with various baseline and other prior works listed below:

- Text Embedding
  - **BERT**: An encoding vector with length of 768 using "bert-base-uncased" pretrained model from huggingface is used as our text features [11] [19].
  - **Infersent**: A sentence encoding vector with length of 4800 that combines both uni- and bi-directional pretrained model outputs is used our text features [20].
  - **OneHot(all words)**: One-Hot encoding method used in [17], where each node is represented as an independent dimension, is used as our text features. We use all the words as word nodes.
  - **OneHot(2000 words)**: Similar to OneHot(all words), but we only take first 2000 words of top appearance in this setup as our text features.
- Recognition Model
  - **SVC**: Linear support vector classifier is used as a conventional model baseline for classification.
  - **TextGCN**: A base framework described in section 2 is used as a state-of-the-art comparison framework.
  - **TED Projection**: A framework that jointly leverage content and comment by performing intra-class projection from content to comment space, which is proposed in Chen et al. [21].
  - **TextGCN(Intersection)**: The framework proposed in this work. When joining two graphs of content and comment, we take intersection of unique words between the two spaces instead of union.
  - **TextGCN(Union)**: The framework proposed in this work. When joining two graphs of content and comment, we take union of unique words of the two spaces.

Table 2. The results of models considering content, comment, or both by concatenating. The reported numbers are accuracy.

		Content	Comment	Both(concat)
SVC	BERT	55.6%	54.6%	59.9%
SVC	Infersent	55.9%	58.3%	59.3%
TextGCN	BERT	57.9%	54.1%	-
TextGCN	Infersent	57.1%	55.2%	-
TextGCN	OneHot(2000 words)	53.7%	54.3%	-
TextGCN	OneHot(all words)	57.4%	58.0%	-
TED Projection	Infersent	-	-	59.0%
TextGCN(Intersection)	BERT	-	-	57.8%
TextGCN(Union)	BERT	-	-	**61.2%**

3.2. Experiment Results

Table 2 is the classification results. The columns with header "Content" and "Comment" are the results considering either content space or comment space solely. We first examine the models using content text data only. TextGCN has about the same performance with different feature representations methods and achieves +2% improvement compared to SVC. The accuracy drops dramatically when using one-hot encoded features on only 2000 words of top appearances, and this phenomenon also happens in the comment space. This indicates that with a single space text network, TextGCN requires the entire set of word instead of only high occurring words to take advantage of relation between word and document. The same phenomenon, while less obvious, is still evident when using the comment space text features only. Furthermore, when comparing the accuracy obtained from using content and comment separately, we observe that the performance is slightly better when using comment space. This indicates that viewer's comments would have a stronger connection to the voted impression of funniness, which means the viewpoint from the audience are intuitively more reflective of the impression.

The rightmost column in Table 2 shows the results obtained by using our proposed comment-content entangled text network. Generally, we see that the performances obtained by combining features from both spaces outperform content space or comment space alone. When taking intersection of words, i.e. the words appeared in both sides, to build the joint graph, the model does not show any improvement even with features from both spaces. When we include the words not in the content side, i.e., union of the words from both content and comment space, the proposed framework achieves the best accuracy of 61.2% in this three-class classification task. Figure 2 shows the confusion matrix of the TextGCN(Union). Our experiment demonstrates a key observation that additional information from comments can provide a perspective from viewers that is distinct from the speaker-planned content, and these two information do complement each other in this humor classification task when jointly consider the relationship between the two perspectives.

3.3. Analysis on Word Distribution

Since our proposed use of graph convolution network naturally perform a semi-supervised learning, we can simultaneously examine the prediction outcomes not only on document nodes but also on the *word* nodes to help investigate the underlying working mechanism of the model intuitively. Unlike document nodes which have labels based on votes, the word nodes are unlabeled and serve as an information propagation bridge between each other. As shown in Fig. 3, we select one of documents predicted as "low" and "high" class, and all the words within the selected document can also be predicted using the trained model. An intuitive observation is that when a document is predicted as "low", it would contain more words predicted as being in the"low" class. "High" words also hold higher portion in a "high" document. This demonstrates the mechanism using graph convolutional network to aggregate/propagate information between different *levels* of lexical information (e.g., words to document or vice versa). We further show actual comments with highlighted words from the dataset in Fig. 4. Those words in red are the words predicted as "high", those in gray are words out of vocabulary (not in the 2000 words), and words in black are in "medium" and "low". While the prediction on words seems to be noisy mostly due to the word-level prediction where the label is placed on the whole talk, the model does successfully capture several some keywords related to humor such as "funny", "entertaining", "joke", and "humor" etc.

4. CONCLUSIONS AND FUTURE WORKS

In this work, we propose a comment-content entangled graph network that models the textual data of both content of TED talks and online viewer's comment for these talks, i.e., representing perspectives of speakers and audiences. We propose to leverage this joint content-comment modeling by integrating perspective of audiences through addition of a graph of

Fig. 4. Some sample sentences from semi-supervised learning on word nodes. The words in red are predicted as "high", the words in gray are those out of vocabulary, and the remaining words in black are in class of "low" and "medium". Despite being noisy, the model captures key funny and humor related words.

social comments into the conventional text-based computational framework (i.e., usually operated only on the content alone). We evaluate our framework in a three-class humor impression recognition tasks using the TED talks dataset. Our framework also outperforms other recent work about impression recognition on the same TED Talks dataset. To best of our knowledge, this is one of the first work that jointly models the content and user feedback with a text-based graph structural framework to perform overall impression recognition that also provide intuitive insights on how viewers respond to the content down to the *word-level*. For future work, since the word and document embeddings involve several steps of aggregation, here we set all the aggregation function as averaging; this hierarchical nature of aggregation should be further modeled to derive a better representational approach. We also plan on investigating other types of media content, e.g., movies, and jointly unpack these heterogeneous types of media to understand its relationship to the audience feedbacks in better predicting the overall impression ratings of these media data.

5. REFERENCES

[1] Jeremy A. Yip and Rod A. Martin, "Sense of humor, emotional intelligence, and social competence," in *Journal of Research in Personality 40*, pp. 1202–1208. 2006.

[2] Aristotle and R. McKeon, *The Basic Works of Aristotle*, Modern Library, 2001.

[3] Platoand E. Hamilton and H. Cairns., *The Collected Dialogues of Plato, Including the Letters*, Pantheon Books, 1961.

[4] Moniek Buijzen and Patti M. Valkenburg, "Developing a typology of humor in audiovisual media," in *MEDIA PSYCHOLOGY*, pp. 147–167. Lawrence Erlbaum Associates, Inc, 2004.

[5] M.P. Mulder and Antinus Nijholt, *Humour Research: State of Art*, vol. 02 of *CTIT Technical Report Series*, Centre for Telematics and Information Technology (CTIT), Netherlands, 9 2002, Imported from CTIT.

[6] John Morreall, "Philosophy of humor," in *The Stanford Encyclopedia of Philosophy*, Edward N. Zalta, Ed. Metaphysics Research Lab, Stanford University, winter 2016 edition, 2016.

[7] Dmitry Davidov, Oren Tsur, and Ari Rappoport, "Semi-supervised recognition of sarcasm in twitter and amazon," in *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, 2010, pp. 107–116.

[8] Julia M. Taylor, "Ontology-based view of natural language meaning: the case of humor detection," in *Journal of Ambient Intelligence and Humanized Computing*, 09 2010, pp. 221–234.

[9] Rada Mihalcea and Carlo Strapparava, "Making computers laugh:investigations in automatic humor recognition," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. 2005, p. 531–538, Association for Computational Linguistics.

[10] Peng-Yu Chen and Von-Wun Soo, "Humor recognition using deep learning," in *NAACL-HLT 2018*, 2018, pp. 113–117.

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

[12] Orion Weller and Kevin Seppi, "Humor detection: A transformer gets the last laugh," in *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp. 3621–3625.

[13] Arjun Chandrasekaran, Ashwin K. Vijayakumar, Stanislaw Antol, Mohit Bansal, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh, "We are humor beings: Understanding and predicting visual humor," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] X. Lei, X. Qian, and G. Zhao, "Rating prediction based on social sentiment from textual reviews," *IEEE Transactions on Multimedia*, vol. 18, no. 9, pp. 1910–1921, 2016.

[15] Salman Jamali and Huzefa Rangwala, "Digging digg: Comment mining, popularity prediction, and social network analysis," *2009 International Conference on Web Information Systems and Mining, WISM 2009*, 11 2009.

[16] Thomas N. Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[17] Liang Yao, Chengsheng Mao, and Yuan Luo, "Graph convolutional networks for text classification," in *33rd AAAI Conference on Artificial Intelligence (AAAI-19)*, 2018, pp. 7370–7377.

[18] Matthew Honnibal and Ines Montani, "spacy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017.

[19] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, vol. abs/1910.03771, 2019.

[20] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, September 2017, pp. 670–680, Association for Computational Linguistics.

[21] Huan-Yu Chen, Yun-Shao Lin, and Chi-Chun Lee, "Through the eyes of viewers: A comment-enhanced media content representation for ted talks impression recognition," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2019, pp. 414–418.