

Graph Capsule Aggregation for Unaligned Multimodal Sequences

Jianfeng Wu
wujf36@mail2.sysu.edu.cn
Sun Yat-sen University
Guangzhou, Guangdong, China

Sijie Mai
maisj@mail2.sysu.edu.cn
Sun Yat-sen University
Guangzhou, Guangdong, China

Haifeng Hu*
huhaif@mail2.sysu.edu.cn
Sun Yat-sen University
Guangzhou, Guangdong, China

ABSTRACT

Humans express their opinions and emotions through multiple modalities which mainly consist of textual, acoustic and visual modalities. Prior works on multimodal sentiment analysis mostly apply Recurrent Neural Network (RNN) to model aligned multimodal sequences. However, it is impractical to align multimodal sequences due to different sample rates for different modalities. Moreover, RNN is prone to the issues of gradient vanishing or exploding and it has limited capacity of learning long-range dependency which is the major obstacle to model unaligned multimodal sequences. In this paper, we introduce **Graph Capsule Aggregation** (GraphCAGE) to model unaligned multimodal sequences with graph-based neural model and Capsule Network. By converting sequence data into graph, the previously mentioned problems of RNN are avoided. In addition, the aggregation capability of Capsule Network and the graph-based structure enable our model to be interpretable and better solve the problem of long-range dependency. Experimental results suggest that GraphCAGE achieves state-of-the-art performance on two benchmark datasets with representations refined by Capsule Network and interpretation provided.

CCS CONCEPTS

• Information systems → Multimedia streaming.

KEYWORDS

Multimodal Sentiment Analysis; Unaligned Multimodal Sequences; Graph Capsule Aggregation; Long-range Dependency

ACM Reference Format:

Jianfeng Wu, Sijie Mai, and Haifeng Hu. 2021. Graph Capsule Aggregation for Unaligned Multimodal Sequences. In *Proceedings of the 2021 International Conference on Multimodal Interaction (ICMI '21)*, October 18–22, 2021, Montréal, QC, Canada. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3462244.3479931>

*Haifeng Hu is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICMI '21, October 18–22, 2021, Montréal, QC, Canada

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8481-0/21/10...\$15.00

<https://doi.org/10.1145/3462244.3479931>

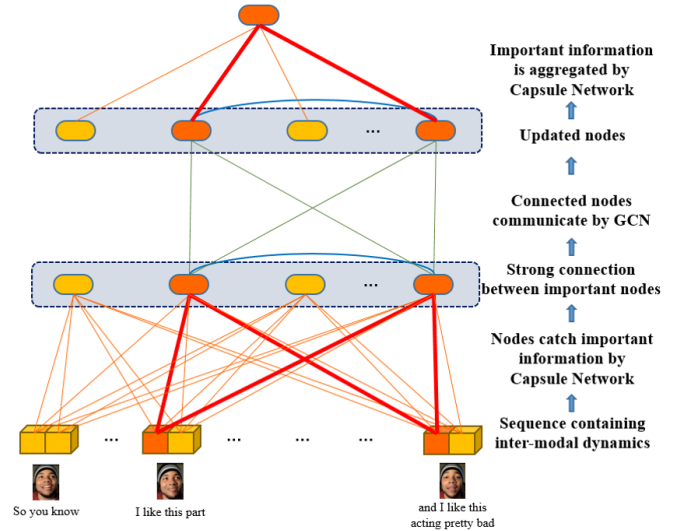


Figure 1: Illustration of the aggregation capability of the GraphCAGE. The color of the links depend on the values of routing coefficients. Red means large value and important information such as the word "like", whereas orange means common information. Blue links indicate edges between nodes. Note that our model can pay attention to critical information from different time steps although they are far from each other.

1 INTRODUCTION

Humans analyze sentiment by the rich information from spoken words, facial attributes and tone of voice, which correspond to textual, visual and acoustic modalities, respectively [13, 17]. It is natural that multimodal sources provide more reliable information for a model to predict sentiment labels. Nevertheless, there are two fundamental challenges for multimodal sentiment analysis. One is the "unaligned" nature of multimodal sequences. For instance, streams from audio and vision are created by receptors using different receiving frequency. As a result, successfully inferring long-range dependency is the key to tackle the issue of "unaligned" nature. The other challenge is how to effectively and efficiently model the long sequences. As common methods to model sequences, RNN and its variants are susceptible to gradient vanishing or exploding and have high time complexity due to their recurrent nature [16]. Therefore, it is critical to propose a model which can process sequential data appropriately without recurrent architecture.

Existing models commonly implement forced word-alignment before training[5, 15, 20, 25, 34, 36] to solve the problem of “unaligned” nature, which aligns the visual and acoustic features to the resolution of words before inputting them into model. However, such word-alignment [32] is time-consuming and not feasible because it requires detailed meta-information about the datasets. Moreover, it may lead to inadequate interactions between modalities as the interactions are not limited to the span of one word. Therefore, the issue of long-range dependency still exists. In addition, owing to heavy reliance on RNN, previous models are usually difficult to train and require plenty of time to infer. Recently, some transformer-based models[9, 24, 37] which can compute in parallel in the time dimension have been proposed to avoid problems of RNN and better explore long-range dependency. Nevertheless, they fail to obtain highly expressive and refined representation of sequences because transformer [27] is a sequence model which cannot sufficiently fuse information from all time steps.

In this paper, we propose an end-to-end model called **Graph Capsule Aggregation** (GraphCAGE) that can compute in parallel in the time dimension by converting unaligned multimodal sequential data into graphs and explicitly learn long-range dependency by the aggregation capability of Capsule Network and graph-base neural model. GraphCAGE consists of two stages: graph construction and graph aggregation. The former first implements modality fusion by cross-modal transformer, then applies Dynamic Routing of Capsule Network and self-attention to create nodes and edges, respectively. This module can significantly solve the problem of long-range dependency because the nodes can proportionally absorb information from every time step by routing mechanism. The latter stage combines Graph Convolutional Network (GCN) with Capsule Network to further aggregate information from nodes and finally produces high-level and refined representation of the graph. We illustrate the aggregation capability of our model in Figure 1. Additionally, routing mechanism equips GraphCAGE with interpretability because we are able to observe the values of routing coefficients to figure out the contributions from different elements. We will discuss the interpretability in Section 4.4.3.

In brief, the main contributions of this work are listed below:

- We propose a novel architecture called GraphCAGE to model unaligned multimodal sequences. GraphCAGE applies Dynamic Routing of Capsule Network to construct node, which enables the model to process longer sequence with stronger ability of learning long-range dependency. Taking advantage of aggregation capability of Capsule Network, GraphCAGE produces high-expressive representations of graphs without any loss of information.
- With sequences transformed into graphs, GraphCAGE can model sequence without RNN, which prevents gradient vanishing or exploding during training. Moreover, computing in parallel greatly boosts efficiency in inferring time.
- Applying Capsule network in node construction and graph aggregation, GraphCAGE is interpretable owing to routing mechanism. With larger routing coefficients indicating greater contribution, we can figure out what information our model focuses on to make predictions.

- The proposed GraphCAGE model achieves state-of-the-art performance on two widely-used datasets. In addition, the extensive experiments in Section 4.4.2 and Section 4.4.3 on routing coefficients demonstrate that our model explicitly explores long-range dependency with interpretation provided.

2 RELATED WORKS

2.1 Human Multimodal Language Analysis

Multimodal language learning aims at learning representations from multimodal sequences including textual, visual and acoustic modalities[14, 25]. A lot of previous studies[1, 3, 21, 34–36] regard RNN such as LSTM and GRU as the default architecture for sequence modeling and they focus on exploring intra- and inter-modal dynamics for word-aligned multimodal sequences. For example, Zadeh et al. propose Memory Fusion Network which is constructed by LSTMs and gated memory network to explore view-specific and cross-view interactions[34]. In [35], Multi-attention Recurrent Network is composed of LSTMs and multi-attention block in order to model both dynamics above. With RNN being the main modules, they are confronted with the problems of training and long inferring time. Recently, [16, 24, 30] propose alternative networks to model unaligned multimodal sequences. Tsai et al.[24] use cross-modal transformer and self-attention transformer to learn long-range dependency. However, the temporal information is collected by self-attention transformer which is a sequence model, implying that fusion among different time steps is not sufficient. In contrast, our proposed GraphCAGE replaces the self-attention transformer with graph-based model which produces more refined and high-level representations of sequences. In [16] and [30], sequences are transformed into graphs and GCN is applied to learn long-range dependency, which not only avoid the problems of RNN but also successfully model unaligned multimodal sequences. Nevertheless, they implement graph pooling and edge pruning to drop some nodes in order to obtain the final representation of graph, leading to information loss. In contrast, GraphCAGE effectively retains all information with Capsule Network which applies Dynamic Routing instead of pooling to aggregate features.

2.2 Capsule Network

Capsule Network is first proposed in [22] and is improved in [7] and [12], which is designed for image features extraction. In general, Capsule Network can not only effectively fuse information from numerous elements into highly expressive representations without information loss, but also reveal the contributions from different elements to the representations by routing mechanism. In [22], the authors claim that pooling will destroy the robustness of the model because some valuable features are ignored by pooling layer. In order to retain these features, pooling layer is replaced with Dynamic Routing for the transmission of information between layers, bringing the benefit of no information loss. In [26], the proposed Multimodal Routing is designed based on Capsule Network and provides both local and global interpretation, verifying the fact that Dynamic Routing of Capsule Network can equip model with interpretability. Inspired by Dynamic Routing, our proposed GraphCAGE uses Capsule Network to construct node from features

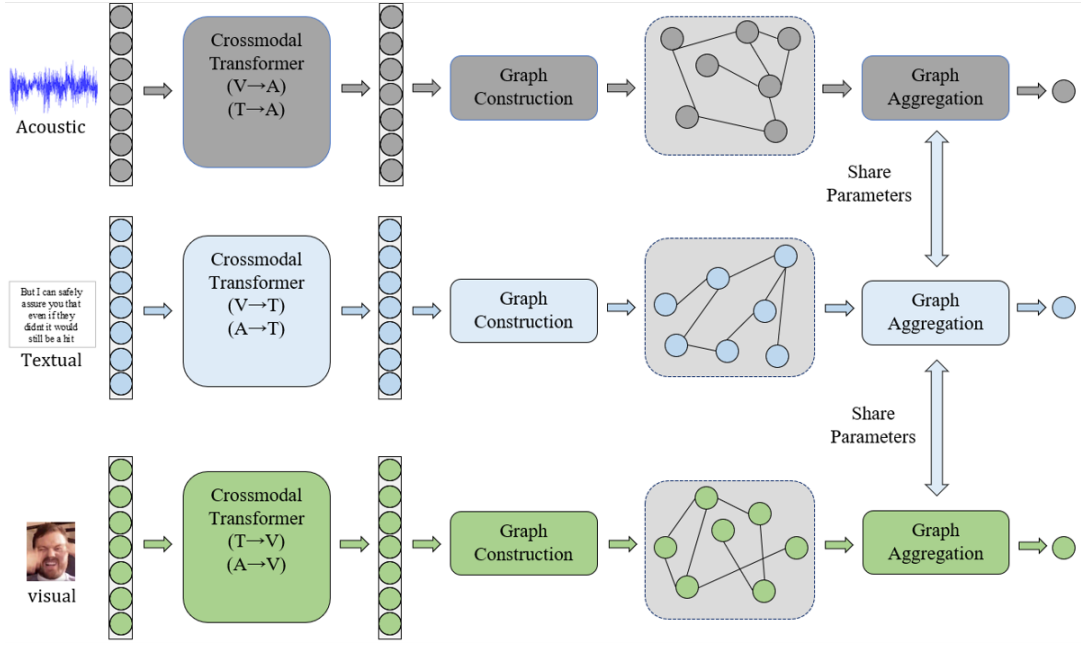


Figure 2: The Schematic Diagram of our proposed GraphCAGE.

containing inter-modal dynamics. In addition, the final representations of graphs are also created by Capsule Network. As a result of efficient transmission of information and great aggregation capability of Capsule Network, our GraphCAGE can effectively learn long-range dependency and explicitly model unaligned multimodal sequences with interpretation ability provided and no information loss.

2.3 Graph Neural Network

As graph-structured data is widely used in many research fields, a series of Graph Neural Networks (GNN) have been introduced in recent years [18, 23, 29, 39]. Among them, Graph Convolutional Network (GCN) [11] is the most popular because of its superior performance on various tasks. Informed by the fact that GCN can effectively aggregate information of related nodes, we apply GCN to integrate related nodes which contain information from various time steps. By this way, the issue of long-range dependency is solved because even the information from two distant time steps can directly communicate with each other. In most cases, the final representation of a graph is obtained by graph pooling [6, 16, 31]. Similarly, in order to obtain high-level graph representation, edge pruning[30] is usually applied in each GCN layer. However, pooling and pruning may rudely drop some important nodes, leading to the loss of information. As we conduct Dynamic Routing of Capsule Network instead of pooling or pruning after GCN, our proposed GraphCAGE model produces high-level and refined representations of sequences without the loss of information.

3 PROPOSED METHOD

In this section, we elaborate our proposed GraphCAGE with its diagram illustrated in Figure 2. Our GraphCAGE consists of two

stages including graph construction and graph aggregation. In the first stage, multimodal sequences are transformed into graphs with nodes and edges created by Capsule Network and self-attention respectively, which enables our model to compute in parallel in the time dimension. In the second stage, each graph is condensed into a representative vector via Graph Convolutional Network (GCN) and Capsule Network. Fundamentally, the Capsule Network in the first stage integrates information of every time step into each node, then the GCN and the Capsule Network in the second stage further aggregate information of nodes, which equips our model with excellent capability of learning long-range dependency.

3.1 Graph Construction

To construct a graph, we need to first create nodes from sequence, then define edges based on these created nodes. All the nodes and edges comprise the graph which contains sufficient information about sentiment and long-range dependency.

3.1.1 Node definition. In order to create node containing information of interactions between different modalities, we first input features of textual, acoustic and visual modalities into cross-modal transformers¹[24]:

$$\begin{aligned} Z^t &= CT^{v \rightarrow t}(X^t, X^v) \oplus CT^{a \rightarrow t}(X^t, X^a) \\ Z^a &= CT^{t \rightarrow a}(X^a, X^t) \oplus CT^{v \rightarrow a}(X^a, X^v) \\ Z^v &= CT^{t \rightarrow v}(X^v, X^t) \oplus CT^{a \rightarrow v}(X^v, X^a) \end{aligned} \quad (1)$$

where $X^{\{t,a,v\}} \in \mathbb{R}^{d^{\{t,a,v\}} \times T^{\{t,a,v\}}}$ denotes the inputted unimodal sequence with $d^{\{t,a,v\}}$ being the dimensionality of features and

¹More detail about cross-modal transformer can be found in the link <https://github.com/kenford953/GraphCAGE>

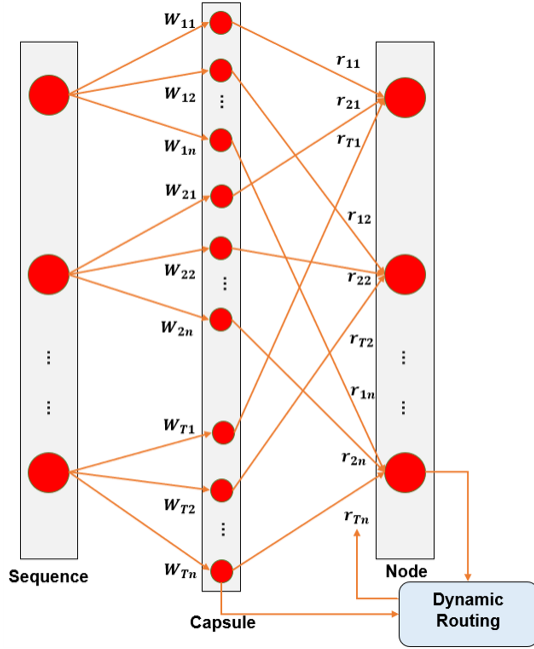


Figure 3: The Schematic Diagram of Node Definition. Due to conciseness, routing mechanism is only presented for r_{Tn} . In fact, every routing coefficient is updated at every iteration by routing mechanism.

$T^{\{t,a,v\}}$ being the sequence length. $CT^{\alpha \rightarrow \beta}$ is the cross-modal transformer translating α modality into β modality with \oplus being the operation of concatenation. For conciseness, we denote $m \in \{t, a, v\}$ as a specific modality in the rest of this paper. The outputs $Z^m \in \mathbb{R}^{d \times T^m}$ contain inter-modal dynamics but long-range dependency is still understudied because the output of cross-modal transformer is still a sequence which requires adequate fusion at the time dimension to explore the interactions between distant time steps. Capsule Network is an excellent model to figure out relations among various elements. Therefore, we apply Capsule Network to construct node from the output sequence Z^m in order to properly fuse information from a large number of time steps. We illustrate the node definition in Figure 3. As shown in Figure 3, we first create capsules as:

$$Caps_{i,j}^m = W_{i,j}^m Z_i^m \quad (2)$$

where $Z_i^m \in \mathbb{R}^d$ denotes the features of the i^{th} time step of sequence Z^m with $W_{i,j}^m \in \mathbb{R}^{d_c \times d}$ being the trainable parameters. $Caps_{i,j}^m \in \mathbb{R}^{d_c}$ means the capsule from the i^{th} time step and it is used for constructing the j^{th} node. Then, we define nodes based on these capsules and Dynamic Routing as Algorithm 1 shows. Specifically, a node is defined by the weighted sum of corresponding capsules as shown below:

$$N_j^m = \sum_i Caps_{i,j}^m \times r_{i,j}^m \quad (3)$$

where N_j^m denotes the embedding of the j^{th} node and $r_{i,j}^m$ is the routing coefficient assigned to capsule $Caps_{i,j}^m$. It is worth noting that for a total of p iterations, all routing coefficients are normalized

Algorithm 1 Node Definition By Dynamic Routing

Input: capsules $Caps_{i,j}^m$

Output: nodes N_j^m

- 1: Initialize all routing coefficients to zero as $b_{i,j}^m = 0$
 - 2: **for** p iterations **do**
 - 3: Normalize all routing coefficients as Eq. 4
 - 4: Create node N_j^m as Eq. 3
 - 5: Update all routing coefficients as Eq. 5
 - 6: **return** nodes N_j^m
-

by softmax and updated based on inner product between the embeddings of capsule and node in every iteration step. The equations for updating $r_{i,j}^m$ are shown as below:

$$r_{i,j}^m = \frac{\exp(b_{i,j}^m)}{\sum_j \exp(b_{i,j}^m)} \quad (4)$$

$$b_{i,j}^m \leftarrow b_{i,j}^m + Caps_{i,j}^m \odot N_j^m \quad (5)$$

where $b_{i,j}^m$ means the routing coefficient before normalization, which is initialized to zero before iteration begins. \odot denotes the operation of inner product. By comparing the values of routing coefficients, we can understand how much information from a specific time step flows into a node, which provides interpretation. With Capsule Network applied to construct node, our model can effectively learn long-range dependency, because nodes contain information from the whole range of sequence and more informative time steps will be assigned larger routing coefficients.

3.1.2 Edge definition. After node construction, edges are created by the self-attention mechanism over the nodes:

$$A^m = f\left(\frac{(W_q^m N^m)^T (W_k^m N^m)}{d_c}\right) \quad (6)$$

where $A^m \in \mathbb{R}^{n \times n}$ is the adjacency matrix and $N^m \in \mathbb{R}^{d_c \times n}$ denotes the overall node embeddings with n being the number of nodes. $W_q^m, W_k^m \in \mathbb{R}^{d_c \times d_c}$ are learnable parameters. f is the nonlinear activation function which is set to ReLU, and T means the matrix transpose operation. With ReLU as our activation function, the negative links between nodes can be effectively filtered out [16] (a negative link implies that a direct connection between these two nodes is not necessary).

It is worth noting that Capsule Network has a large number of trainable parameters. As a result, we apply L2 Regularization on these parameters to alleviate overfitting as:

$$L_{reg} = \lambda \left(\sum_i \sum_j \|W_{i,j}^t\|^2 + \sum_i \sum_j \|W_{i,j}^a\|^2 + \sum_i \sum_j \|W_{i,j}^v\|^2 \right) \quad (7)$$

where λ is a hyper-parameter which reflects the importance of the loss function. Therefore, during training, the total loss function is the Mean Absolute Error (MAE) plus L_{reg} .

As we finish constructing nodes and edges, a graph which contains rich inter-modal dynamics and reliable connections between related nodes has been created. Our graph construction method is informed by recent graph-based architectures for sequential data, but is distinct from all of them in the method of node construction.

For example, in [16] and [30], the authors define node and edge based on multimodal sequences processed only by Feed-Forward-Network, which causes that the created graph is not highly expressive because the node embedding is not built on high-level features. Moreover, they regard every time step as a node and only depend on GCN to learn long-range dependency, leading to insufficient learning. Contrary to them, our model first uses cross-modal transformer to obtain high-level features which contain inter-modal dynamics, then constructs node based on these features by Capsule Network which enables node to properly gain information from a great quantity of capsules. Note that the number of nodes here is significantly fewer than the length of the input sequence. By this way, each node is built on various time steps and the created graph is highly expressive and also is easier to be processed because of a small number of nodes. In the next stage, we illustrate how we conduct message passing between nodes and extract high-level representation from the graph.

3.2 Graph Aggregation

In most cases, representation of graph is extracted by GCN followed by graph pooling or edge pruning to dump some redundant nodes. However, it is hard to avoid dropping valuable nodes which causes the loss of information. To prevent this problem, we retain GCN due to its excellent capability of exchanging information among nodes, and replace pooling or pruning with Capsule Network to prevent information from being lost. The graph aggregation consists of inner-loop and outer-loop. The relationship between inner-loop and outer-loop can be explained in this way: in every iteration of outer-loop, all iterations of inner-loop will be performed. As for the proposed method, Graph convolution is performed in outer-loop and the Dynamic Routing is performed in the inner-loop. So, in every iteration of graph convolution, we will perform p iterations of Dynamic Routing to obtain a graph representation. The equations for the k^{th} iteration of graph convolution are shown below:

$$\begin{aligned} n^{m,k} &= W^k N^{m,k-1} (A^m + I) \\ N^{m,k} &= f(W_o^k n^{m,k}) \end{aligned} \quad (8)$$

where $N^{m,k}$ denotes the node embedding at the k^{th} iteration and $N^{m,0}$ is the output node embedding in the graph construction stage ($1 \leq k \leq 2$). I denotes the identity matrix which is used to perform self-loop operation and f is chosen to be the \tanh activation function. Note that W^k and W_o^k have no superscripts m because we share all weights for three modalities in the graph aggregation stage. When all nodes are updated, we generate the final representation of the graph at the k^{th} iteration using Capsule network. The Capsule network consists of p iterations (i.e., the inner-loop iteration) to update the routing coefficients of the nodes. The equation is shown as below:

$$R^{m,k} = CapsNet(N^{m,k}) \quad (9)$$

where $R^{m,k} \in \mathbb{R}^{d_c}$ denotes the final representation at the k^{th} iteration and the details of *CapsNet* are shown in Algorithm 2. Specifically, Dynamic Routing (i.e., the inner-loop) contains normalization of routing coefficients, construction of representation and update

Algorithm 2 Capsule Network for Graph Aggregation

Input: node embedding $N^{m,k}$

Output: representation $R^{m,k}$

- 1: Create capsules for each node as $Caps_j^{m,k} = W_j^k N_j^{m,k}$
 - 2: Initialize all routing coefficients as $b_j^{m,k} = 0$
 - 3: **for** p iterations **do**
 - 4: Normalize all routing coefficients as Eq. 10
 - 5: Create representation $R^{m,k}$ as Eq. 11
 - 6: Update all routing coefficients as Eq. 12
 - 7: **return** representation $R^{m,k}$
-

of routing coefficients as shown below:

$$r_j^{m,k} = \frac{\exp(b_j^{m,k})}{\sum_j \exp(b_j^{m,k})} \quad (10)$$

$$R^{m,k} = \sum_j Caps_j^{m,k} \times r_j^{m,k} \quad (11)$$

$$b_j^{m,k} \leftarrow b_j^{m,k} + Caps_j^{m,k} \odot R^{m,k} \quad (12)$$

where $Caps_j^{m,k}$ means the capsule created by the j^{th} node at the k^{th} graph convolution iteration (see Algorithm 2). Note that different from the graph construction stage, each node only owns one capsule at each graph convolution iteration so only one subscript j is enough for denoting the capsule.

As stated above, graph convolution enables related nodes communicate with each other and update node embedding, which helps our model further learn long-range dependency because nodes contain information from related time steps. Moreover, intra-modal dynamics are explored effectively because nodes are from two identical modalities. Finishing updating the nodes, Capsule Network is applied to aggregate all the nodes into a highly expressive representation with complete information transmission. More importantly, the highly expressive representation proportionally absorbs information from all nodes by Dynamic Routing, where larger routing coefficient will be assigned if information of the node is more valuable. By this way, interpretation is provided, indicating which node contributes most to the final representation. In contrast, many graph-based architectures roughly drop nodes by pooling or pruning to obtain the final representation, leading to the loss of information. In addition, interpretation of their models depends on the edges between related nodes, which reflect relations among different elements. But the contribution to prediction is not interpretable.

As intra- and inter-modal dynamics are effectively explored and long-range dependency is explicitly learned, we concatenate the graph representations of all the modalities at each iteration k and apply fully-connected layers to predict sentiment labels.

4 EXPERIMENTS

In this section, we evaluate our proposed model GraphCAGE on two frequently-used datasets: CMU-MOSI[38] and CMU-MOSEI[36]. We first show details about the datasets, baseline models, experimental settings, and then present the results with comparison

Table 1: Performance of GraphCAGE on two benchmark datasets. The bold means the best performance. We put asterisk behind the result by our model which is not the best but close to SOTA(<1%).

Models	CMU-MOSI					CMU-MOSEI				
	Acc7	Acc2	F1	MAE	Corr	Acc7	Acc2	F1	MAE	Corr
Recurrent Models										
CTC+EF-LSTM	32.2	73.7	73.5	1.038	0.594	41.7	65.3	76.0	0.799	0.625
CTC+LF-LSTM	31.3	74.5	74.3	1.042	0.608	48.0	79.5	79.6	0.632	0.650
CTC+TFN	32.4	77.9	75.0	1.040	0.616	49.3	79.5	78.9	0.613	0.673
CTC+MFN	30.9	77.7	75.5	1.032	0.627	49.1	80.6	80.0	0.612	0.687
Parallel Computing Models										
MuT	35.3	80.6	79.3	0.972	0.681	49.0	80.1	80.9	0.630	0.664
Multimodal Graph	32.1	80.6	80.5	0.933	0.684	49.7	81.4	81.7	0.608	0.675
MTAG	31.9	80.5	80.4	0.941	0.692	48.2	79.1	75.9	0.645	0.614
GraphCAGE	35.4	82.1	82.1	0.933	0.684*	48.9*	81.7	81.8	0.609*	0.670

Table 2: Comparison with RNN-based models about inferring time on CMU-MOSI test set. Note that the inferring time is calculated based on the whole test set, which contains 686 video clips. the batch size and the environment are the same for all models.

Models	Inferring Time (s)
CTC+LF-LSTM	8.926
CTC+TFN	6.146
CTC+RAVEN	19.369
GraphCAGE	3.813

Table 3: An ablation study on the benefit of GraphCAGE’s Capsule Network using unaligned CMU-MOSI.

Models	Acc2	F1
Graph Construction without Capsule Network	76.1	76.7
Graph Aggregation with GAT	75.9	77.0
Graph Aggregation with mean pooling	77.0	77.0
Graph Aggregation with LSTM	79.0	79.2
GraphCAGE	82.1	82.1

among GraphCAGE and other baseline models. The remaining part of this section are illustrations about long-range dependency and interpretability.

4.1 Datasets

CMU-MOSI is a popular dataset for multimodal sentiment analysis which contains 2199 video clips. Each video clip is labeled with a real number within $[-3, +3]$ which reflects the sentiment intensity, where +3 means strongly positive sentiment and -3 means strongly negative sentiment. In accordance with most prior works, various metrics are reported including 7-class classification accuracy (Acc_7), binary classification accuracy (Acc_2), Mean Absolute Error (MAE), F1 score and the correlation of the model’s prediction with humans.

The total numbers of video clips for training set, validation set and testing set are 1284, 229 and 686, respectively.

CMU-MOSEI consists of 22856 video clips and we use 16326, 1871 and 4659 segments as training, validation and testing set. The reported metrics and sentiment label are the same as those of CMU-MOSI.

4.2 Baseline Models

We separate baseline models into two groups including recurrent models and parallel computing models.

Recurrent models include Early Fusion LSTM (EF-LSTM), Late Fusion LSTM (LF-LSTM), Tensor Fusion Network (TFN)[33] and Memory Fusion Network (MFN)[34]. EF-LSTM and LF-LSTM simply concatenate features at input and output level, which apply LSTM[8] to extract features and infer prediction. As stated in [33], these approaches fail to explore intra- and inter-modal dynamics due to simple concatenation. TFN effectively explores both dynamics with outer product adopted to learn joint representation of three modalities. MFN depends on systems of LSTM to learn interactions among modalities. However, EF-LSTM and MFN are word-level fusion methods which study aligned multimodal sequences and thus we combine connectionist temporal classification (CTC)[4] with them to process the unaligned sequences. The CTC module we use comprises two components: alignment predictor and the CTC loss. The alignment predictor is chosen as a recurrent networks. We train the alignment predictor while minimizing the CTC loss. Then, we multiply the probability outputs from the alignment predictor to source signals. The recurrent natures of the above models bring about some disadvantages including gradient vanishing or exploding, long inferring time and insufficient learning for long-time dependency.

Parallel computing models include Multimodal Transformer (MuT)[24], Multimodal Graph[16] and Modal-Temporal Attention Graph (MTAG)[30], which disuse RNN to better explore long-range dependency within multimodal sequences. MuT extends Transformer network[27] to model unaligned multimodal sequences by cross-modal transformer. Nevertheless, it utilizes self-attention transformer to integrate information from different time steps, which causes inadequate fusion at the time dimension because

self-attention transformer is a sequence-to-sequence model and cannot fuse sequences at the time dimension. Multimodal Graph and MTAG both creatively adapt GCN to explore long-range dependency with problems of RNN avoided. However, they are confronted with information loss because of the operations of pooling and pruning.

4.3 Experimental details

Our model is developed on Pytorch and we choose Mean Absolute Error (MAE) as loss function for sentiment prediction task on CMU-MOSI and CMU-MOSEI datasets. Note that the total loss during training is MAE plus L2 Regularization loss. The optimizer is RMSprop and all hyper-parameters are selected by grid search. The textual, acoustic and visual features are extracted by GloVe word embedding[19], COVAREP[2] and Facet[10] respectively, with more details in <https://github.com/A2Zadeh/CMU-MultimodalSDK>. We specify the hyper-parameters and the features in our github link².

4.4 Results and Discussions

The overall results are shown in Table 1 which indicates that our model outperforms both recurrent and parallel computing models on most of the metrics for two popular datasets. In general, based on the results that parallel computing models achieve better performance than recurrent models, we can infer that it is practical to apply model without recurrent structure to multimodal sequence modeling.

Comparing with recurrent models, GraphCAGE outperforms them by a considerable margin which implies that our model processes sequential data better than canonical RNN. Low performance on unaligned sequences by RNN-based models verifies the incompetence of recurrent network to model excessively long sequence which requires strong capability of learning long-range dependency. With aggregation capability of Capsule Network and the graph-based structure, GraphCAGE can effectively link remote but related time steps, which contributes to the explicit exploration of long-range dependency. Moreover, as Table 2 shows, the inferring time of our model is significantly reduced, demonstrating the high efficiency of our model which can compute in parallel in the time dimension.

As for parallel computing models, GraphCAGE achieves the best performance due to substantially longer memory and efficient transmission of information. Specifically, because GCN and Capsule Network in graph aggregation stage can realize more sufficient fusion at the time dimension than self-attention transformer, GraphCAGE explicitly explores long-range dependency and outperforms MulT. In addition, with Capsule Network applied to transmit information, our model achieves better performance than MTAG and Multimodal Graph which have shortcoming about the loss of information.

4.4.1 Ablation Study. In order to verify the effectiveness of our graph construction and graph aggregation stages, we conduct ablation study on CMU-MOSI dataset as Table 3 shows. Generally, the absence of Capsule Network in both stages of our model leads to drastic decline on performance, which indicates that they are

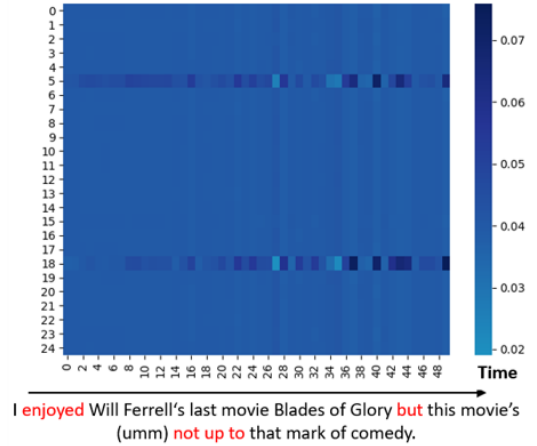


Figure 4: Visualization for routing coefficients of Capsule Network in graph construction stage. The element of the i^{th} column and the j^{th} row represents the value of routing coefficient r_{ij} which is the same as r_{ij} in Figure 3. Note that the values in each column are the routing coefficients from each time step. The presented routing coefficients are from textual modality for the sake of clear interpretation.

critical to improve the ability of learning long-range dependency and enable our GraphCAGE to better model unaligned multimodal sequences.

For model without Capsule Network in graph construction, we directly define each node embedding as the feature of each time step and edges are constructed by self-attention. Apparently, each node only contains information from one time step, which causes insufficient learning for long-range dependency. Moreover, owing to the long sequence length, the number of nodes is excessively large. As a result, the latter GCN and Capsule Network are hard to figure out the relations among these nodes. In contrast, our model first condenses information from sequence into a moderate number of nodes by Capsule Network, then models their relations by later layers, which improves the capability of linking remote but related time steps.

For graph aggregation without Capsule Network, we retain the GCN part and design three aggregation methods including Graph Attention Network (GAT)[28], mean pooling and LSTM to replace the Capsule Network. Note that GAT applies attention mechanism to aggregate nodes and achieves excellent performance on various node classification tasks. However, based on the lower performance, we argue that GAT is not suitable for our model because we need to decode the nodes to predict a label rather than classify them. As for mean pooling, the final representation is the average of the embeddings of all nodes. Obviously, mean pooling is too simple to obtain highly expressive graph representation and it will cause the loss of information. For LSTM, it is slightly better than mean pooling because of more learnable parameters. However, the final representation is the last element of the output sequence. As a result, the input order may heavily affect the performance and we cannot figure out the best order because information of a node changes

²The code for our model and details about hyper-parameters and features can be found in <https://github.com/kenford953/GraphCAGE>

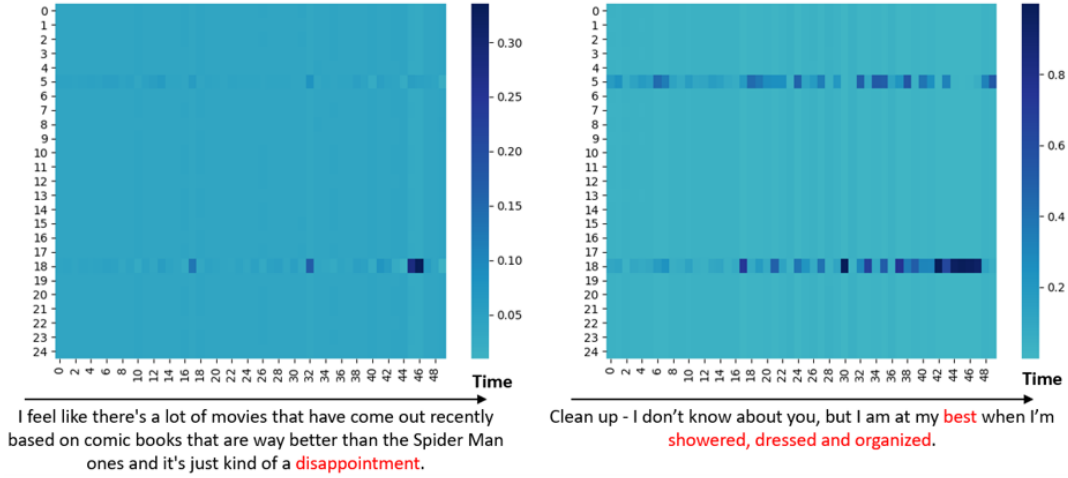


Figure 5: Visualization for routing coefficients of Capsule Network in Graph Construction stage. Similarly, The element represents the value of a routing coefficient and all routing coefficients come from textual modality. The left is a negative example whereas the right is a positive one.

dynamically. In conclusion, applying Capsule Network to aggregate information of nodes is more suitable than other frequently-used aggregation methods because the final representation is refined by absorbing more important information by Dynamic Routing.

4.4.2 Discussion of Long-range Dependency. As we stated above, because of the adaptation of Capsule Network, GraphCAGE is skilled at modeling long sequences which requires excellent capability of learning long-range dependency. To present this ability in detail, as shown in Figure 4, we find an example of CMU-MOSEI and observe its routing coefficients in graph construction stage which reflect how much the model pays attention to specific information. Specifically, the sentiment of this example is obviously negative because of the word "but" and the phrase "not up to" in the last part of the sentence. However, some models with weak ability of learning long-range dependency may predict positive for this example based on the word "enjoyed" in the front part of the sentence. In contrast, our model attends to both parts of the sentence and pays more attention to the last part with larger routing coefficients assigned. Moreover, we found that the information prefers to flow into the fifth and eighteenth nodes which communicate by GCN and are integrated by Capsule Network later. Presumably it is because the distance between these two nodes is moderate which prevents our model from overly focusing on specific part of the sequence and the later GCN and Capsule Network enable our model to figure out the relations among important parts of the sentence. So we believe that even if the exact sentiment requires contextual information, our model can correctly predict sentiment with excellent capability of connecting remote but related elements.

4.4.3 Discussion of Interpretability. Interpretation helps us to figure out how the model comes to a prediction from a large number of time steps, which is useful for improving performance on different datasets. To provide interpretation, we adapt Capsule Network into

our model where the routing coefficients reflect how much information from the corresponding time step flows into the next layer. As shown in Figure 5, we observe the values of routing coefficients from textual modality of two examples with different sentiments. For the left example, information of the word "disappointment" is highlighted by the largest routing coefficient, indicating our model predicts the negative sentiment mostly depending on it. As for the right example, our model successfully catches the important positive words "best", "showered", "dressed" and "organized" by assigning larger routing coefficients to them. Based on the analysis above, we can safely draw a conclusion that GraphCAGE actually understands which element leads to specific sentiment and it provides interpretation for us to find out what information contributes to the prediction.

5 CONCLUSION

In this paper, we develop a model called GraphCAGE for multimodal sentiment analysis using unaligned multimodal sequences which are too long to model by recurrent networks. For the purpose of explicitly learning long-range dependency, our model adapts Capsule Network to transmit information and applies Graph Convolutional Network to explore relations among different time steps, which can avoid the loss of information and contributes to the interpretation. Moreover, modeling sequences with graph-based structure instead of RNN prevents various problems like gradient vanishing or exploding. Extensive experiments with routing coefficients verify the effectiveness of the adaptation of Capsule Network and GCN. Experiments on two popular datasets show that GraphCAGE achieves SOTA performance on modeling unaligned multimodal sequences.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant 62076262.

REFERENCES

- [1] Wenliang Dai, Zihan Liu, Tiezheng Yu, and Pascale Fung. 2020. Modality-Transferable Emotion Embeddings for Low-Resource Multimodal Emotion Recognition. *CoRR abs/2009.09629* (2020). arXiv:2009.09629 <https://arxiv.org/abs/2009.09629>
- [2] Gilles Degottex, John Kane, Thomas Drugman, Tuomo Raitio, and Stefan Scherer. 2014. COVAREP: A Collaborative Voice Analysis Repository for Speech Technologies. In *ICASSP*. 960–964.
- [3] Deepanway Ghosal, Md Shad Akhtar, Dushyant Chauhan, Soujanya Poria, Asif Ekbal, and Pushpak Bhattacharyya. 2018. Contextual Inter-modal Attention for Multi-modal Sentiment Analysis. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 3454–3466. <https://doi.org/10.18653/v1/D18-1382>
- [4] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*. 369–376.
- [5] Yue Gu, Kangning Yang, Shiyu Fu, Shuhong Chen, Xinyu Li, and Ivan Marsic. 2018. Multimodal Affective Analysis Using Hierarchical Attention Strategy with Word-Level Alignment. In *ACL*. 2225–2235.
- [6] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.
- [7] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. 2018. Matrix capsules with EM routing. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=HJWLFGRb>
- [8] S Hochreiter and J Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.
- [9] Jian Huang, Jianhua Tao, Bin Liu, Zheng Lian, and Mingyue Niu. 2020. Multimodal Transformer Fusion for Continuous Emotion Recognition. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 3507–3511. <https://doi.org/10.1109/ICASSP40776.2020.9053762>
- [10] iMotions 2017. 2017. iMotions. *Facial expression analysis* (2017).
- [11] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. In *ICLR*.
- [12] Adam R. Kosiorek, Sara Sabour, Yee Whye Teh, and Geoffrey E. Hinton. 2019. Stacked Capsule Autoencoders. arXiv:1906.06818 [stat.ML]
- [13] Bruno Latour, K. R. Gibson, and T. Ingold. 1994. Tools, Language and Cognition in Human Evolution. *Man* 29, 2 (1994), 486.
- [14] Paul Pu Liang, Ziyin Liu, Amir Zadeh, and Louis Philippe Morency. 2018. Multimodal Language Analysis with Recurrent Multistage Fusion. In *EMNLP*. 150–161.
- [15] S. Mai, H. Hu, J. Xu, and S. Xing. 2020. Multi-Fusion Residual Memory Network for Multimodal Human Sentiment Comprehension. *IEEE Transactions on Affective Computing* (2020), 1–1.
- [16] Sijie Mai, Songlong Xing, Jiaxuan He, Ying Zeng, and Haifeng Hu. 2020. Analyzing Unaligned Multimodal Sequence via Graph Convolution and Graph Pooling Fusion. *CoRR abs/2011.13572* (2020). arXiv:2011.13572 <https://arxiv.org/abs/2011.13572>
- [17] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, and D. McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.
- [18] A. Micheli. 2009. Neural Network for Graphs: A Contextual Constructive Approach. *IEEE Transactions on Neural Networks* 20, 3 (2009), p.498–511.
- [19] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. <http://www.aclweb.org/anthology/D14-1162>
- [20] Hai Pham, Paul Pu Liang, Thomas Manzini, Louis Philippe Morency, and Poczós Barnabás. 2019. Found in Translation: Learning Robust Joint Representations by Cyclic Translations Between Modalities. In *AAAI*. 6892–6899.
- [21] Hai Pham, Manzini Thomos, Liang Paul Pu, and Poczós Barnabás. 2018. Seq2Seq2Sentiment: Multimodal Sequence to Sequence Models for Sentiment Analysis. In *ACL 2018 Grand Challenge and Workshop on Human Multimodal Language*.
- [22] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing Between Capsules. *CoRR abs/1710.09829* (2017). arXiv:1710.09829 <http://arxiv.org/abs/1710.09829>
- [23] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* 20, 1 (2009), 61–80.
- [24] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Multimodal Transformer for Unaligned Multimodal Language Sequences. In *ACL*. 6558–6569.
- [25] Yao Hung Hubert Tsai, Paul Pu Liang, Amir Zadeh, Louis Philippe Morency, and Ruslan Salakhutdinov. 2019. Learning Factorized Multimodal Representations. In *ICLR*.
- [26] Yao-Hung Hubert Tsai, Martin Q. Ma, Muqiao Yang, Ruslan Salakhutdinov, and Louis-Philippe Morency. arXiv preprint arXiv:2001.08735, 2020. Multimodal Routing: Improving Local and Global Interpretability of Multimodal Language Analysis. arXiv:arXiv:2004.14198
- [27] 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 *NIPS*. 5998–6008.
- [28] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.
- [29] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks?. In *ICLR*.
- [30] Jianing Yang, Yongxin Wang, Ruitao Yi, Yuying Zhu, Azaan Rehman, Amir Zadeh, Soujanya Poria, and Louis-Philippe Morency. 2020. MTGAT: Multimodal Temporal Graph Attention Networks for Unaligned Human Multimodal Language Sequences. *CoRR abs/2010.11985* (2020). arXiv:2010.11985 <https://arxiv.org/abs/2010.11985>
- [31] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*. 4800–4810.
- [32] Jiahong Yuan and Mark Liberman. 2008. Speaker identification on the SCOTUS corpus. *Acoustical Society of America Journal* 123 (2008), 3878. <https://doi.org/10.1121/1.2935783>
- [33] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis Philippe Morency. 2017. Tensor Fusion Network for Multimodal Sentiment Analysis. In *EMNLP*. 1114–1125.
- [34] Amir Zadeh, Paul Pu Liang, Navonil Mazumder, Soujanya Poria, Erik Cambria, and Louis Philippe Morency. 2018. Memory Fusion Network for Multi-view Sequential Learning. In *AAAI*. 5634–5641.
- [35] Amir Zadeh, Paul Pu Liang, Soujanya Poria, Prateek Vij, Erik Cambria, and Louis Philippe Morency. 2018. Multi-attention Recurrent Network for Human Communication Comprehension. In *AAAI*. 5642–5649.
- [36] Amir Zadeh, Paul Pu Liang, Jonathan Vanbriesen, Soujanya Poria, Edmund Tong, Erik Cambria, Minghai Chen, and Louis Philippe Morency. 2018. Multimodal Language Analysis in the Wild: CMU-MOSEI Dataset and Interpretable Dynamic Fusion Graph. In *ACL*. 2236–2246.
- [37] A. Zadeh, C. Mao, K. Shi, Y. Zhang, P. P. Liang, S. Poria, and L. P. Morency. 2019. Factorized Multimodal Transformer for Multimodal Sequential Learning. (2019).
- [38] Amir Zadeh, Rowan Zellers, Eli Pincus, and Louis Philippe Morency. 2016. MOSI: Multimodal Corpus of Sentiment Intensity and Subjectivity Analysis in Online Opinion Videos. *IEEE Intelligent Systems* 31, 6 (2016), 82–88.
- [39] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*. 5165–5175.