Neural Networks 143 (2021) 345–354

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Graph routing between capsules

Yang Li [a], Wei Zhao [c], Erik Cambria [b,*], Suhang Wang [d], Steffen Eger [c]

[a] Northwestern Polytechnical University, China
[b] Nanyang Technological University, Singapore
[c] Technical University of Darmstadt, Germany
[d] Pennsylvania State University, USA

ARTICLE INFO

Article history:
Received 29 February 2020
Received in revised form 12 June 2021
Accepted 17 June 2021
Available online 23 June 2021

Keywords:
Capsule neural network
Text classification
Routing
Graph routing

ABSTRACT

Routing methods in capsule networks often learn a hierarchical relationship for capsules in successive layers, but the intra-relation between capsules in the same layer is less studied, while this intra-relation is a key factor for the semantic understanding in text data. Therefore, in this paper, we introduce a new capsule network with graph routing to learn both relationships, where capsules in each layer are treated as the nodes of a graph. We investigate strategies to yield adjacency and degree matrix with three different distances from a layer of capsules, and propose the graph routing mechanism between those capsules. We validate our approach on five text classification datasets, and our findings suggest that the approach combining bottom-up routing and top-down attention performs the best. Such an approach demonstrates generalization capability across datasets. Compared to the state-of-the-art routing methods, the improvements in accuracy in the five datasets we used were 0.82, 0.39, 0.07, 1.01, and 0.02, respectively.

1. Introduction

Since capsule networks were proposed, they have been successfully applied in image processing (Hinton, Krizhevsky, & Wang, 2011; Sabour, Frosst, & Hinton, 2017). In recent years, a lot of works have appeared that also adopted capsule networks for text classification. For example, Zhao et al. (2018) investigate the effectiveness of capsule networks in small data settings, while Ren and Lu (2018) introduce a variant of capsule networks with deep compositional code learning in text classification. The success of capsule networks mainly relies on a bottom-up mechanism, namely routing by agreement (Sabour et al., 2017; Sabour, Frosst, & Hinton, 2018), which routes low-level capsules to high-level capsules in order to learn hierarchical relationships between layers. This mechanism helps capsule networks capture spatial features well.

However, there are problems in applying capsule networks to text data. Generally, these routing algorithms do not account for the intra-relationships between capsules in each layer, while these intra-relationships exist in text data usually. For example, in the sentence "The battery has a long life" with a positive sentiment, there is a part-whole relationship between the words "battery" and "life", a modification relationship between the words "long" and "life" and etc. These intra-relationships, often ignored by previous routing methods, help understanding the positive sentiment. It is imperative to leverage implicit knowledge in the words' semantic context to learn hierarchical and intra-relationships spontaneously across layers (Akhtar, Ekbal, & Cambria, 2020; Cambria, Li, Xing, Poria, & Kwok, 2020; Chikersal, Poria, Cambria, Gelbukh, & Siong, 2015).

In this paper, we study the novel problem of simultaneously exploring these two relationships (intra-relationship and hierarchical relationship) interactively in the routing context. In essence, we need to solve several challenges: (i) how to evaluate the relationship between capsules in the same layer; (ii) how to further improve the quality of the learned features in text classification; and (iii) how to evaluate the quality of the relationships that is learned from the capsules. Graph convolutional networks (GCN) (Kipf & Welling, 2017) have demonstrated good effects in the graph feature learning (Pan et al., 2019; Wu et al., 2020). And we believe this method also takes effect in the capsules feature gathering. In an attempt to solve these challenges, we treat the capsule as a node in a graph and propose a new graph routing mechanism that learns the intra-relationships with GCN. In general, our proposed graph routing learns the intra-relationship by aggregating information from other capsules in the same layers, and learns the hierarchical relationship with routing by agreement across different layers. Thus, the differences between proposed graph routing and existing routing are:

* Corresponding author.
E-mail addresses: liyangnpu@nwpu.edu.cn (Y. Li), zhao@aiphes.tu-darmstadt.de (W. Zhao), cambria@ntu.edu.sg (E. Cambria), szw494@psu.edu (S. Wang), eger@aiphes.tu-darmstadt.de (S. Eger).

https://doi.org/10.1016/j.neunet.2021.06.018
0893-6080/

- Our graph routing method pays attention to the intra-relationship learning with graph neural network, while existing routing methods do not;
- Our graph routing method applies the attention mechanism in the feature strength, and existing routing methods do not;

In this paper, the relationship between different capsules is evaluated by the Wasserstein distance (WD) effectively, and a new normalization trick is proposed to approximate the adjacency matrix well. Also, implicit information in text data is identified easily with information aggregation and routing agreement. Compared with state-of-the-art models, the performance in three of the five datasets improved by more than 0.3 points in accuracy. Therefore, the challenges mentioned before are solved well with the proposed graph routing, and the main contributions of this paper are:

- We investigate, for the first time, intra-relationships between capsules by leveraging words' semantic context where we treat capsules in each layer as nodes in a graph.
- We introduce a new routing algorithm combining bottom-up routing and top-down attention and learn hierarchical and intra-relationships spontaneously.
- Extensive experiments show that our proposed routing algorithm performs better than existing routing methods.

2. Preliminary

When capsule networks were proposed (Hinton et al., 2011), they were mainly applied in image processing. Multiple capsules were required to be consistent in one detection. However, the intra-relationship between capsules cannot be ignored when processing text data. To our best knowledge, there are no works about intra-relationship learning in the routing. Before introducing our graph routing, we will make a preliminary introduction about the classic routing. The symbols that this paper uses are listed in Table 1.

Since dynamic routing was proposed by Sabour et al. (2017), it has been treated as the standard routing method. Let the capsule vector in the first layer be $\mathbf{u}_i$. Before routing, a transformation procedure is applied to encode spatial relationships between local features and global features:

$$\hat{u}_{j|i} = W^D \mathbf{u}_i$$

where W^D is an affine transformation matrix. Generally, routing decides how to send the capsule vector $\mathbf{u}_i$ to the second layer. This is controlled by a weight variable c_{ij}, which is multiplied with the corresponding value $\hat{u}_{j|i}$:

$$\mathbf{s}_j = \sum_{i=1}^{n} c_{ij}\hat{u}_{j|i}$$

The squash function is then applied to ensure that the norm of $\mathbf{s}_j$ is bounded by 1, while the direction of $\mathbf{s}_j$ should not be changed. The squash function has the form

$$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1+\|\mathbf{s}_j\|^2}\frac{\mathbf{s}_j}{\|\mathbf{s}_j\|}$$

Generally, c_{ij} is a non-negative scalar, and the sum of all weights c_{ij} in the first layer is 1. It is obtained from the softmax function:

$$c_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}$$

Different from attention models, routing is a bottom up method in feature gathering, and this is illustrated in following equation which is the across layer operation with i being the capsules from the first layer and j being the capsules from the second layer:

$$e_{ij} = e_{ij} + \hat{u}_{j|i}\mathbf{v}_j$$

e_{ij} is the coupling coefficient, and the general routing method is using the clustering method to find the probability in mapping different capsules to related capsules across layers. However, as opposed to related works, we investigate relationships between capsules in the same layer.

3. Model architecture

To make a fair comparison with other routing mechanisms, we are using the same architecture proposed by Zhao et al. (2018). The model architecture components are four layers which are the N-gram convolutional layer (NCL), the primary capsule layer (PCL), the routing layer (RL), and the representation layer. The architecture is basically the same as Capsule-A in Zhao et al. (2018), which is depicted in Fig. 1.

3.1. N-gram Convolutional Layer (NCL)

After getting the input from the word embedding, each document is represented by $\mathbf{x} \in R^{L\times D}$, where L denotes the document length (padded where necessary), D is the word embedding length. Then the input will be fed into the NCL to detect features at different positions. The process is depicted in Eq. (1),

$$g_i^a = f_1(W^a \cdot \mathbf{x}_{i:i+K-s} + b_1) \tag{1}$$

where K denotes the N-gram size, s is the stride width, $W^a \in R^{K\times D}$ is the filter during the feature detection with convolutional operation, $b_1 \in R$ is a bias term, and f_1 denotes the activation function (i.e., ReLU) in that layer. After the filtering with a certain filter W^a over the $\mathbf{x}_{i:i+K-s}$, we get a feature value $g_i^a \in R$. All of those features form the feature map $\mathbf{g}^a = [g_1, g_2, \ldots, g_{L-K+s}]$ with $\mathbf{g}^a \in R^{L-K+s}$ from the NCL. The number of the filter is set to B_1. Hence, the output of the NCL is $\mathbf{g}$ which is rearranged as

$$\mathbf{g} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_{B_1}] \in R^{(L-K+s)\times B_1}$$

3.2. Primary Capsule Layer (PCL)

Then $\mathbf{g}$ is fed into the PCL which is composed of the convolutional operation. In this layer, each row in $\mathbf{g}$ convolves with a filter $W^b \in R^{B_1\times d}$, where d denotes the capsule dimension. This is depicted in Eq. (2).

$$p_i = f_2(W^b \cdot \mathbf{g}_i + b_2) \tag{2}$$

where f_2 is the squash function for the output vector, b_2 is a capsule bias term. As in the NCL, there are B_2 filters in total. Therefore, the generated features are rearranged as

$$\mathbf{p} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{B_2}] \in R^{(L-K+s)\times B_2\times d}$$

Extensive computational resources are required when there is a large document as input. Hence, the capsule compression is conducted to condense the capsule number to a smaller one. The condensed capsule u_i is computed as:

$$u_i = \sum_j w_j \mathbf{p}_j \in R^d \tag{3}$$

where w_j is the parameter needed to be learned.

In the next step, the transformation matrix W_{ij} is utilized to generate the prediction vector $\hat{u}_{j|i} \in R^d$ (the parent capsule j) from its child capsule u_i, where N is the number of parent capsules in

Table 1
The list of symbols involved in this paper.

$\mathbf{x}$	The representation of a document	I_N	The identity matrix
$\mathbf{u_i}$	The first layer capsule vector in the routing	$\mathbf{s_j}$	The second layer capsule vector in the routing
$\mathbf{v_j}$	The output from the routing	c_{ij}	The weight variable
W^D	The affine transformation matrix	e_{ij}	The coupling coefficient
W_i	The parameters of the neural networks	$A, \tilde{A}$	The adjacency matrix
D	The degree matrix in the graph	$f_{att}(\cdot)$	The feed-forward function
d^w, d^e, d^c	The Wasserstein distance, euclidean distance and cosine distance	g_i^a	The output from N-gram convolutional layer
h_j	The output from GCN	m_j	The output from f_{att}
W^a, W^b	The filter in the convolutional layer	p_i	The output from primary capsule layer
K	The size of N-gram	s	The stride size
b_1, b_2	The bias term	B_1, B_2	The number of the filter
L	The document length	d	The capsule dimension
E	The capsule number after the routing	$f_1(\cdot)$	The activate function (i.e. ReLU)
C	The class number	$f_2(\cdot)$	The squash function
α_j	The attention value		

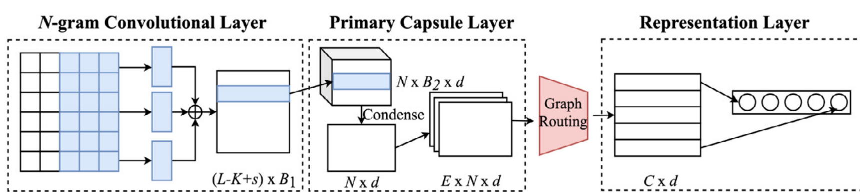

Fig. 1. The architecture of the model.

the last step which is $(L - K + s)$. This vector can be computed as follows:

$$\hat{u}_{j|i} = W_{ij}u_i + b_{j|i} \in R^d \tag{4}$$

where $b_{j|i}$ denotes the capsule bias term. This step helps learn child–parent relationships for the capsule networks as argued in Sabour et al. (2017). Therefore, the feature map generated from the primary capsule layer is $\mathbf{u} \in R^{E\times N\times d}$, where E is the capsule number after the matrix transformation.

Then the graph routing algorithm is utilized to get voting results, which can be depicted as:

$$v = \text{Graph-Routing}(\hat{u}) \tag{5}$$

where $\hat{u}$ is the composition of all of the child capsules, and $v \in R^{C\times d}$ denotes all of the parent-capsules, where C denotes the class number. Details are referred to Section 4.

3.3. Representation layer

Representation layer is the last layer which gets the input v from the graph routing. The final capsule number C is the same as the class number of the document. The vector normalization is utilized to get the class probability $\hat{p} = \|v\| \in R^C$.

4. Graph routing

To make the routing simple and effective, each capsule is treated as a node in a graph. Meanings of sentences are expressed by the composition of different phrases, and each capsule only captures a certain phrase during the feature selection. To obtain a comprehensive understanding of a sentence, the relationship between different capsules must be learned. Actually, there should be connections between nodes as it will be helpful for reaching an agreement during the routing. Inspired by GCN (Kipf & Welling, 2017), which elaborates the relationships among the nodes in a graph, we learn intra-correlation with the graph convolutional operation. The structure of graph routing is shown in Fig. 2.

The GCN procedure is shown in the red box in Fig. 2, based on an undirected complete graph that is composed of the capsules in the lower layer. There are usually more than three iterations in a routing. Therefore, in order to make the structure efficient and simple, a single GCN layer is used in each iteration to gather intra-relationship within the nodes. If the number of iterations is set to 3, we will have a routing network with three layers of GCN. The weight a_{ij} in the edge is the intra-relationship between the two connected capsules i and j. Generally, if two connected capsules are semantically close, there will be a large value for the weight a_{ij}. And if there is no semantic connection between these two capsules, there will be a small value for weight a_{ij}. The way to get the intra-relationship between two capsules will be discussed in the following subsection. In general, these three distance measurements are used first, coupled with normalization trick in Section 4.2, when evaluating the relationship between capsules. The attention mechanism in Section 4.3 involved is the same as that in Bahdanau, Cho, and Bengio (2014).

4.1. Relationship between capsules

In the common GCN (Kipf & Welling, 2017), the relationship between two nodes is expressed by the non-negative value in the adjacency matrix A. However, it is the prior known information for A which is predefined by the existing data. The convolutional operation is conducted in the Fourier domain to get the hidden feature, which is expressed in Eq. (6) (Hammond, Vandergheynst, & Gribonval, 2011).

$$W_i \star u_i \approx W_i\tilde{A}u_i \tag{6}$$

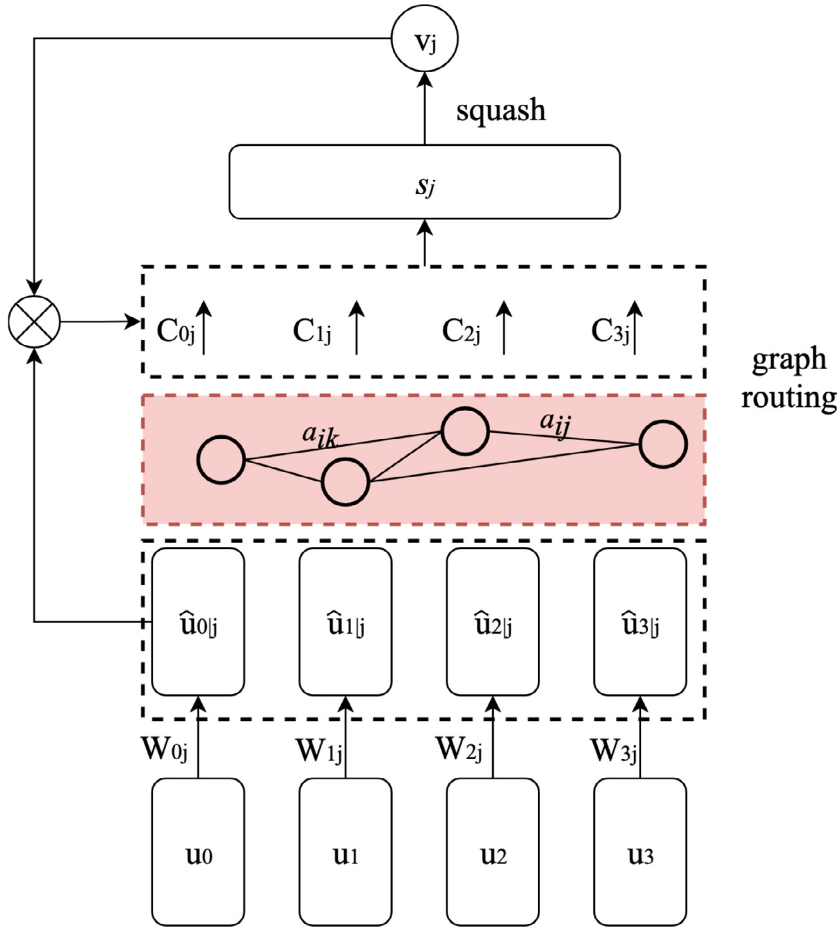

Fig. 2. The structure of the graph routing.

where W_i is the parameter of the network, and $\tilde{A}$ is from Eq. (7)

$$\tilde{A} = I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \tag{7}$$

where $A \in R^{N \times N}$ is the adjacency matrix, N is the capsule number, and D is the degree matrix, with $D_{ii} = \sum_j A_{ij}$.

After filtering by the convolutional kernel, u_i will aggregate more relevant information for the classification. From Eq. (6), we can see that if one capsule is related to another capsule in layer l, the corresponding value in the adjacency matrix will be big. The relationship matrix is one of the adjacency matrices, with the correlation value as the connection strength. Generally, the adjacency matrix A is known information where its values are binary or real to express the distance between two nodes.

In our case, it is difficult to evaluate the relationship between two capsules as this information is unknown before routing. Furthermore, the capsule vector is a hidden representation which lacks detailed semantic meaning. Therefore, to overcome this challenge, we use different distance measurements to find the relationship among capsules semantically and effectively. In this paper, we explore three distance measurements, which are Wasserstein distance (WD), Euclidean distance (ED), and cosine similarity (CS). We argue that if two vectors (for example, vector y_i and vector y_j) express a similar meaning, they will be semantically close with a large relationship value in the adjacency matrix.

4.1.1. Wasserstein Distance (WD)

Wasserstein Distance is usually used in generative models (Li, Pan, Wang, Yang, & Cambria, 2018) due to its ability in measuring the distance between probability distributions. The capsule vector can be treated as the probability of a certain attribute that is captured. Therefore, WD which is calculated as $d^w_{ij} = \inf E[\|y_i - y_j\|_p]^{1/p}$ can be applied in capsule vector relationship evaluation, where inf denotes infimum, i.e., the greatest lower bound. In this paper, p is set to 1. To transfer the Wasserstein distance d^w to the intra-relationship, $a_{ij} = -d^w_{ij}$ is applied. Generally, if two capsules are semantically close, the value of the intra relationship will be big.

4.1.2. Euclidean Distance (ED)

Euclidean Distance is the straight-line distance between two vectors in Euclidean space, and it is calculated as $d^e_{ij} = \sqrt{(y_i - y_j) \cdot (y_i - y_j)}$. To transfer the Euclidean distance d^e to the intra relationship, $a_{ij} = -d^e_{ij}$ is also applied. Similarly, if two capsules are semantically close, the value of the intra relationship will be big.

4.1.3. Cosine Similarity (CS)

Cosine Similarity is another popular measurement between two non-zero vectors, and it can be calculated as $d^c_{ij} = \frac{y_i \cdot y_j}{\|y_i\| \|y_j\|}$. To ensure the distance to itself is zero which is same as the case in ED and WD, the intra relationship between two capsules is expressed as $a_{ij} = d^c_{ij} - 1$.

4.2. Normalization trick

Generally, the renormalization trick is used in Eq. (7), which is $I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \rightarrow \hat{D}^{-\frac{1}{2}} M \hat{D}^{-\frac{1}{2}}$, with $M = A + I_N$ and $\hat{D}_{ii} =$

$\sum_j M_{ij}$. However, if we use ED and WD as the measurement for intra relationship, the values are negative, which makes it difficult to apply the general normalization trick to get the approximate adjacency matrix A because it requires all of the values being non-negative. To overcome this issue, a more general normalization trick for Eq. (7) is proposed in this paper, which is given in Eq. (8),

$$\tilde{A} \rightarrow \mathrm{softmax}(A) + I_N \tag{8}$$

Each capsule treats itself as the centroid of the clustering by adding a one-hot vector. $\tilde{A}$ has eigenvalues in the range [0, 1] as the distance to itself is 0. And this avoids the gradients exploding/vanishing during the neural network optimization. It is an instance-level normalization for the capsule's relationship learning. Different from the renormalization trick mentioned before, the semantic distance is normalized in the view of the current capsule by using the softmax function in the same row. It can be treated as a variant of random Laplacian matrix normalization. This matches the true situation in which the importance of the word is different when the centroid word is different. Taking "The battery has a long life" as an example, in the view of the word "The", the word "battery" will have a high value as "The" is the definite article for "battery". However, in the view of the word "battery", "long life" will play a more important role to express the positive sentiment than the word "The". The ablation study shows the effectiveness of the proposed method.

4.3. With attention

To further improve the quality of the learned features from our graph routing, and guarantee the learned features are useful, the attention mechanism is applied. Graph routing is a bottom-up method which highly depends on lower layer features. If there is no clear cluster center, it will be difficult to do the feature selection. Different from the routing mechanism, attention mechanism is a top-down method which helps the routing to select useful relationships by considering the context information in the same layer. This guarantees the quality of the gathered relationship. The attention is applied to the aggregated output h_j from GCN which is described in Eq. (9).

$$\begin{aligned} m_j &\leftarrow f_{att}(h_j) \\ \alpha_j &= \frac{\exp m_j}{\sum_j \exp m_j} \end{aligned} \tag{9}$$

Here, f_{att} is the feed-forward function to get the attention weight. The attention score α_j is then applied to the learned feature h_j that is got from GCN. The proposed algorithm is in Algorithm 1. It is the same framework as dynamic routing, except for GCN and attention mechanisms. As mentioned earlier, in a routing iteration, it is a single-layer GCN for the feature aggregation.

Algorithm 1 The Algorithm of the Graph Routing

Require: Capsules u_i
1: **for** l layers **do**
2: Initialize $e_{ij} \leftarrow 0$
3: Get the adjacency matrix A about capsules in layer l.
4: Get $\tilde{A}$ with Eq. (8).
5: $h_j \leftarrow u_{ij}^l \tilde{A}_i W_{ij}$
6: $m_j \leftarrow f_{att}(h_j)$
7: $\alpha_j = \mathrm{softmax}(m_j)$
8: $o_j \leftarrow \alpha_j h_j$
9: **for** r routing number **do**
10: All capsule i in layer l:
11: $c_{ij} \leftarrow \mathrm{softmax}(e_{ij}^l)$
12: All capsule j in layer $(l+1)$:
13: $s_j \leftarrow \sum_i c_{ij} o_j$
14: $v_j \leftarrow \mathrm{squash}(s_j)$
15: All capsule i in layer l and capsule j in layer $(l+1)$:
16: $e_{ij}^{l+1} \leftarrow e_{ij}^l + u_{ij}^l \cdot v_j$
17: **end for**
18: **return** v_j
19: **end for**

In the lines 3–4, the $\tilde{A}$ is obtained from the adjacency matrix A row by row. Lines 5–17 are the routing details about the graph routing. In lines 5–8, GCN and attention are applied. In the line 11, the weight scalar c_{ij} is calculated from e_{ij} in layer l. In the line 14, the squash function is applied to get the output value v_j. In the line 16, e_{ij} in layer $l+1$ is calculated.

5. Experiments

In this section, we conduct experiments to evaluate the effectiveness of the proposed graph routing. Specifically, we conduct experiments of document classification to validate the effectiveness of the proposed routing in text classification. Then, we validate the hierarchical relationship learning with semantic consistency, and validate intra relationship learning with adjacency matrix illustration. Next, the ablation study is conducted to validate the effectiveness of the proposed normalization trick. To show the generalization of the proposed routing, we also compare to BERT (Devlin, Chang, Lee, & Toutanova, 2018). Next, the case study is illustrated to see the effectiveness of the proposed routing. Finally, we conduct a parameter analysis.

Table 2
The details of the datasets.

Datasets	Train	Test	Classes	Avg-Docs	Max-Docs	Vocab Size
Amazon Clothing	250k	25k	5	68.0	4355	81 836
Amazon Beauty	180k	15k	5	99.2	4494	80 935
Emotion	15k	5k	13	65.7	167	30 716
Yelp	650k	50k	5	118.4	1463	293 402
AG news	120k	7.6k	4	39.3	196	19 637

5.1. Experimental settings

5.1.1. Datasets

We adopt five benchmark datasets for document classification to evaluate the effectiveness of the proposed framework. Amazon-Clothing and Amazon-Beauty are two review datasets that we select,[1] where each document is labeled with a rating ranging from [1, 5]. Emotion dataset[2] contains 13 emotion categories, namely: {Love, Empty, Relief, Anger, Surprise, Neutral, Happiness, Sadness, Fun, Enthusiasm, worry, boredom, hate}. Yelp[3] is another review dataset we use. The label range in Yelp dataset is also [1, 5]. AG News (Conneau, Schwenk, Barrault, & Lecun, 2017) is a news dataset where each document belongs one of the four classes, i.e., world, sports, business, and technology . The statistics of those datasets are listed in Table 2.

Avg-Docs denotes the average word number in the document, while Max-Docs is the maximum word number. Vocab size is the number of the word that the corpus contains.

1 http://jmcauley.ucsd.edu/data/amazon/
2 https://data.world/crowdflower/sentiment-analysis-in-text
3 http://yelp.com/dataset

Table 3
The classification results with different routing methods over those five datasets. § means results citing from Zhao et al. (2018). Leaky DR is the abbreviation of leaky dynamic routing that is proposed in Zhao et al. (2018).

Routing Methods	Amazon Clothing	Amazon Beauty	Microsoft Emotion	Yelp	AG News
Transformer	61.87%	62.50%	31.07%	64.61%	89.16%
EM Routing	60.57%	60.65%	31.04%	62.51%	–
Dynamic Routing	62.87%	63.19%	30.82%	63.65%	89.02%
Leaky DR (A)	62.92%	63.47%	31.11%	64.08%	92.10%§
Leaky DR (B)	–	–	–	–	92.60%§
KDE Routing	62.94%	63.95%	31.83%	64.45%	90.08%
GCN	62.86%	63.47%	30.87%	63.35%	89.61%
Graph Routing(CS)	62.80%	63.81%	31.40%	64.69%	92.30%
Graph Routing(ED)	63.19%	64.03%	31.55%	64.61%	91.91%
Graph Routing(WD)	63.44%	64.13%	31.61%	65.33%	**92.62%**
With Attention	**63.76%**	**64.34%**	**31.90%**	**65.62%**	92.60%

5.1.2. Implementation details

We use TensorFlow in python to implement our model. In all of the experiments, all word embeddings are initialized randomly with 300-dimensional vectors. The capsule number is set to 50, the out channel number from NCL and PCL is 64, the stride for the convolutional operation is 2. The batch size is set to 32, and the learning rate is 5e-5 during the model training with the Adam optimization algorithm (Kingma & Ba, 2014). All of the experimental results are averaged over 5 runs.

5.2. Baseline methods

In the experiments, the compared routing methods include:

- **EM Routing:** EM routing is proposed in Sabour et al. (2018). It is good at spatial feature extraction, and usually used in image classification.
- **Dynamic Routing:** Dynamic routing is another popular routing method which is proposed in Sabour et al. (2017), and also usually used in image classification.
- **Leaky Dynamic Routing:** Leaky dynamic routing is proposed in Zhao et al. (2018) and achieves state of the art results in sentiment classification.
- **KDE Routing:** KDE routing is another robust routing model proposed in Zhao, Peng, Eger, Cambria, and Yang (2019). This model is good at the multi-label classification and has a good generalization.
- **GCN:** To make the ablation study about the GCN, we only utilize GCN in the routing part. Here, the adjacency matrix is calculated with WD.
- **Transformer:** The transformer-based method is very strong baseline model. It is added on top of the model to replace the routing part while keeping other parts identical.

To make a fair comparison, all of the compared methods share the same parameters with the same architecture described in Section 3.

5.3. Document classification

The results with different routing methods are listed in Table 3. The bold number in the table is the best result in a dataset.

The word in the bracket behind the graph routing represents the intra relationship measurement used in the adjacency matrix. In the table, we can see that our proposed graph routing achieves competitive performance. In particular, graph routing with different intra relationships substantially outperforms the transformer, and there is a noticeable margin on all the experimental datasets. Also, graph routing with different intra relationships obtains competitive results against existing routing methods, such as dynamic routing, leaky dynamic routing and EM routing, especially in the graph routing with WD which achieves the best results on all of the datasets compared with other intra relationship measurements. Therefore, we can conclude that the graph routing with WD has obtained a good quality of the relationship between different capsules. There is a slight improvement for the leaky dynamic routing by using Leaky-Softmax (Sabour et al., 2017) to replace the standard softmax function in strengthening the relationship between a child and parent capsules. If we only apply GCN to map the capsule vector to the final output by aggregating the learned features directly, the improvement compared with existing routing methods is limited. By combining with the attention mechanism, our graph routing with WD has a further improvement with the best precision on most of the datasets compared with other routing methods. That is to say, with the help of the attention mechanism, graph routing becomes more robust and effective.

Table 4
The semantic consistency between different layer output.

Routing Methods	NCL	PCL	RL
Dynamic Routing	25.72%	25.39%	85.82%
Leaky Dynamic Routing	21.82%	25.05%	85.39%
KDE Routing	24.65%	25.56%	86.30%
Graph Routing	24.85%	25.70%	87.55%

5.4. Semantic consistency

One important feature of our graph routing method is that it can learn the hierarchical and intra relationship between capsules at the same time. In this subsection, the effectiveness in the hierarchical relationship learning is validated. However, it is difficult to evaluate this relationship directly as there is no specific meaning in the capsule, even after the graph routing. However, if this relationship between different capsules is well learned in a layer, the capsule distribution in that layer will correlate with the final output as well. That is to say, there will be a high semantic consistency between the current capsule layer and the final output if the hierarchical relationship is well learned. Thus, semantic consistency between different layers is calculated. The final output denotes $v \in R^{C\times d}$, where C is the class number, the output in each layer is transferred to $p \in R^{m\times d}$. Each vector in p will be labeled with the class number by finding the closest vector in v. Then semantic consistency is calculated as the percentage of times that the vector is in the right class. And the vectors from the NCL, PCL, and RL are considered. Then the outputs from those three layers will compute with the final output to get the semantic consistency. This experiment is conducted on the AG News dataset, and the results are listed in Table 4.

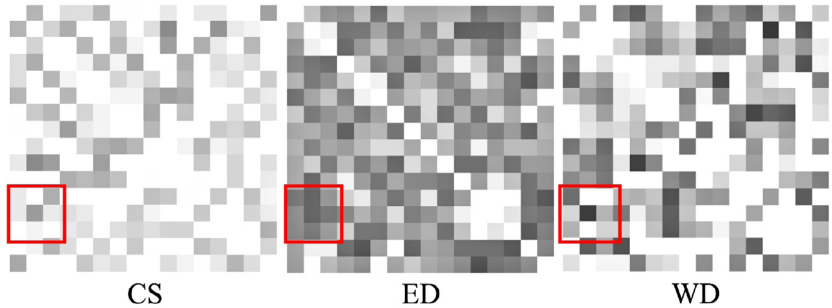

Fig. 3. The relationship among capsules that is learned with different measurements. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

From Table 4, we can see that there is no difference in column NCL and PCL compared with existing routing methods. There are four categories about the document in the AG News. Therefore, only about 25% of the vectors from layer NCL and PCL remain relevant to the final output. However, after the routing, the number that is related to the final output increases sharply, especially in the case of graph routing. That is to say that the hierarchical relationship between different capsules is learned well after the graph routing.

5.4.1. Different relationship measurements

There are different intra relationship measurements to get the adjacency matrix between the capsules. From Table 3, we can see that the graph routing with WD achieves the best results on all datasets compared with CS and ED. That is to say that WD is an efficient method in distance evaluation. To explain the goodness of the WD, the adjacency matrices with different measurements over the same document are illustrated in Fig. 3.

From the color blocks in the red box which is selected randomly from the same location in the three cases, we can see that the ED is noisier than the other two measurements, and there is a clear value in WD. On the contrary, the value in CS is blurred. That is to say that WD has a good evaluation for the intra relationship.

5.5. Ablation study

To analyze the effectiveness of the new normalization trick we proposed in Section 4.2, the ablation validation compared with the renormalization in Eq. (7) is conducted. As there is no negative value for the adjacency matrix, the exponential function is applied to the ED and WD. Therefore, $a_{ij} = \exp(-d_{ij}^{e})$ in the case of ED, and $a_{ij} = \exp(-d_{ij}^{w})$ in the case of WD. CS is kept the same due to its range being [0, 1]. Except for the normalization trick over $\tilde{A}$ is different, all the other parts of the model are kept the same. The results are showed in Fig. 4.

From the figure, we can see that our normalization trick can achieve better results compared with the normal one, especially in the case of WD and ED. When using CS as the relationship measurement, there is no difference between those two normalization tricks as our normalization trick can be treated as a variant of random Laplacian matrix normalization.

To validate the importance of the intra-relations that is learned by the graph routing, we replace $\tilde{A}$ that is learned from Eq. (7) with the identity matrix, and mask the intra-relations that is learned among the capsules in the same layer. The result is pictured in the red line named 'Without A' in Fig. 4. From the red line we can see that, all these three different distance metrics are not working under such condition. That is to say, without intra-relations, the performance of our graph routing degrades dramatically. This validates the importance of the intra-relations that is learned by our graph routing.

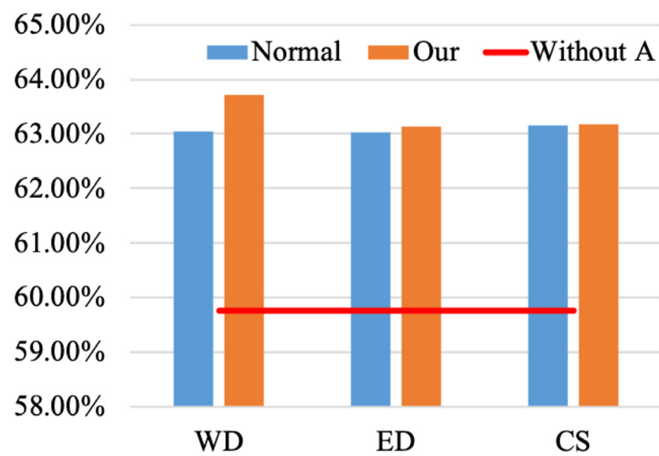

Fig. 4. The classification results with different normalization trick in GCN. y-axis denotes the classification result, and the red line named 'Without A' is the result of $\tilde{A}$ being replaced by the identity matrix.

Table 5
Comparisons with different pooling out methods in BERT.

	Max	Avg	Sum	Our Routing
5 layers	36.13%	37.26%	33.08%	**37.41%**
12 layers	36.43%	37.00%	31.77%	**37.03%**

5.5.1. With BERT

BERT (Devlin et al., 2018) is so popular that we cannot ignore it. To make fair comparisons, we set "output all encoded layers" to *True* in the pre-trained BERT[4] model, which has been trained on massive data and computing resources. Before sending these results (12 layers in total, we also select last 5 layers as another example) to the classifier, there are four ways to produce the output, which are max pooling out (Max), average pooling out (Avg), summation (Sum) and our routing method. The results on the Microsoft emotion dataset are reported in Table 5.

We can see that with the help of BERT, our graph routing achieves the best results compared with other pooling methods both in the case of 5 layers and 12 layers. From this result, we can conclude that our graph routing has a good generalization when applied in a new model.

5.6. Case study

In this subsection, we show in Fig. 5 the effectiveness of our model in category information extraction. Each word embedding is labeled by the vector from the graph routing.

4 https://github.com/google-research/bert

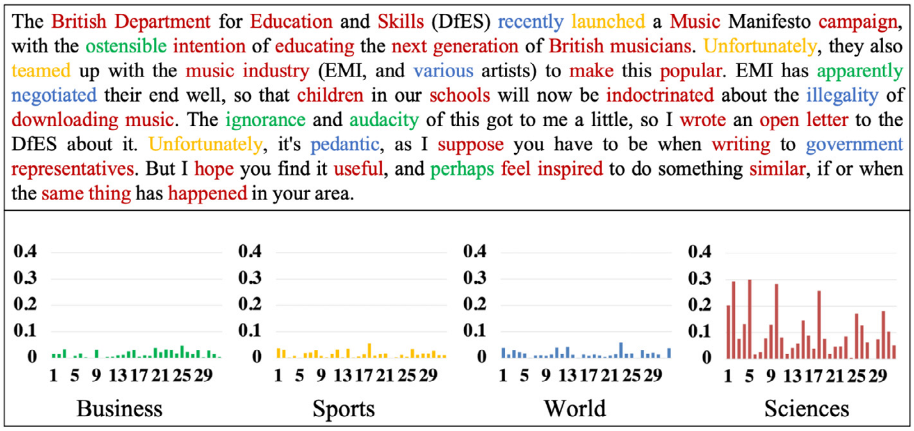

Fig. 5. The figure above is the distribution of the word. Words in red are the science category, in blue are the world, in yellow are the sport and in green are the business. The figures below are the category vector distribution after the squash operation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

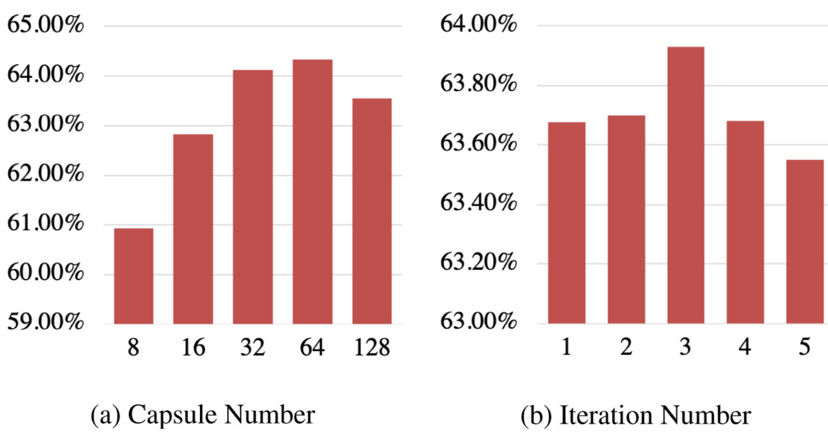

Fig. 6. The parameters analysis about the capsule number. y-axis denotes the classification result.

We can see that most of the words are labeled as the science category after the optimization, and those words are closely related with the sentence topic. Also, from the output vector that is shown in the text below, we can see that the values in science are bigger compared with other categories. This example shows the effectiveness of our routing in the semantic relationships extraction.

5.7. Parameter analysis

In this subsection, two parameters of the routing, iteration number and the capsule dimension, are discussed. Results are in Fig. 6. In Fig. 6(a), when the capsule dimension is set to 64, it achieves the best result. When the capsule dimension is 256, the server shows the OOM error due to the huge amount of memory occupation. There is no improvement compared with the dimension of 64 when the dimension is 128. Therefore, it is the best choice to set the dimension to 64. In Fig. 6(b), when the iteration number is set to 3, it achieves the best result. As the iteration number increases, there is no improvement in performance.

We know that graph routing has more steps than existing routing methods, for example, it takes time to compute the adjacency matrix with WD. To validate its computing efficiency, a comparison of the number of parameters and the time cost is shown in Table 6. From the table, we can see that our graph routing has a smaller parameter number than dynamic routing and leaky dynamic routing. Although the time cost is slightly larger than dynamic routing and leaky dynamic routing, it is still competitive considering the number of parameters.

Table 6
The parameter number and time cost of different routing methods. s denotes second.

Routing methods	#Parameter	Time
EM Routing	31.69 MB	0.1587 s
KDE Routing	31.68 MB	0.1439 s
Leaky DR (A)	69.07 MB	0.2634 s
Dynamic Routing	69.07 MB	0.2632 s
Graph Routing with Attention	40.09 MB	0.2655 s

6. Related works

In this work, we applied the GCN in the routing between the capsules. Therefore, the related works about the routing between capsules will be introduced. Also, our work relates with document classification with capsule network which will be introduced at the same time.

6.1. Routing between capsules

Since the concept of "capsule" neural networks (Hinton et al., 2011) was firstly proposed, it has become a hot research topic due to its ability in improving the representational limitation of CNNs in which pooling operation causes the information losing. The transformation matrices between different capsules make the capsule network capture the part-whole relationships. Together with the routing-by-agreement (Sabour et al., 2017, 2018), capsules neural network has achieved promising results on MNIST data.

It is a bottom-up way for the dynamic routing and the EM routing proposed in Sabour et al. (2017, 2018) when clustering the vector or matrix together. There are different variants of dynamic routing. Leaky softmax is applied by replacing the softmax function to strengthen the relationship between a child and parent capsules (Zhao et al., 2018). Task routing algorithm was proposed by applying the clustering procedure in the task level (Xiao, Zhang, Chen, Wang, & Jin, 2018). Different from the general dynamic routing, task routing introduced a new coupling coefficient c_{ij}^k by subdividing it to the task k. Xi, Bing, and Jin (2017) gave empirical results over the best parameters selection in the capsule network with dynamic routing. Especially the routing iterations during the routing procedure affects the performance. Chen and Crandall (2018) believed that dynamic routing is not well integrated into the training procedure, especially for the iteration number which needs to be decided manually. Thus, there is a bottleneck for the dynamic routing due to its expense of computation during the routing. Therefore, Zhang, Zhou, and Wu (2018) proposed weighted Kernel Density Estimation (KDE) to accelerate the routing. In Zhao et al. (2019), an adaptive KDE routing algorithm was proposed to make the routing decide the iteration number automatically, which gives a further optimization for the routing. The empirical results demonstrate its effectiveness on different classification tasks. Apart from KDE, K-means clustering method was also validated for its effectiveness during routing (Ren & Lu, 2018). It is a bottom-up way for the attention, while routing is a top-down way. There are also researches about combining attention with the routing together in the capsule network (Choi, Seo, Im, & Kang, 2019).

6.2. Capsule network for text classification

Since capsule networks were proposed, most works are on image classification (Chen & Crandall, 2018; Sabour et al., 2017, 2018; Xi et al., 2017), while recurrent neural networks (RNNs) are more popular in text data processing (Chaturvedi, Ong, Tsang, Welsch, & Cambria, 2016; Chen, Ye, Cambria, Chen, & Xing, 2017; Li, 2018; Li et al., 2017). In Gong, Qiu, Wang, and Huang (2018), Wang, Sun, Han, Liu, and Zhu (2018), RNN-Capsule was proposed for sentiment analysis. In Ren and Lu (2018), together with the proposed compositional coding mechanism, bidirectional Gated Recurrent Units (GRU) were adopted for text classification. Based on the RNN, the attention model was applied in the capsule network for aspect-level sentiment analysis (Wang, Sun, Huang, & Zhu, 2019).

Sabour et al. (2017) applied the convolutional layer, primary capsule layer in their model similarly as in image classification. MCapsNet stacked the convolutional network, primary capsule layer, and representation layer together to do multi-task text classification (Xiao et al., 2018). Different from the MCapsNet, Zhao et al. (2018) paralleled three such architecture together with filter window of 3, 4, 5 in the convolutional layer. All of those works show the effectiveness of such architectures in text feature extraction.

7. Conclusion & future works

It is the first time to treat the capsule as a node in a graph, and our graph routing applies GCN to explore the relationship between capsules in the same layer. Together with the attention mechanism, graph routing also remedies the disadvantage of the routing which highly depends on the lower level features. Furthermore, different distances between capsules are discussed. Empirical results show the effectiveness of the proposed model: The performance of three out of five datasets is improved by more than 0.3 compared with state-of-the-art models. In the future works, we will explore a more effective graph neural network in the intra-relationship learning within the capsules, and find out a more efficient method for feature learning of text data using capsule networks. Also we will explore and design a new attention mechanism to accommodate our graph routing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research is supported by the Agency for Science, Technology and Research (A*STAR), Singapore under its AME Programmatic Funding Scheme (Project #A18A2b0046).

References

Akhtar, Md Shad, Ekbal, Asif, & Cambria, Erik (2020). How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble. *IEEE Computational Intelligence Magazine*, *15*(1), 64–75.

Bahdanau, Dzmitry, Cho, Kyunghyun, & Bengio, Yoshua (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Cambria, Erik, Li, Yang, Xing, Frank, Poria, Soujanya, & Kwok, Kenneth (2020). SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis. In *CIKM* (pp. 105–114).

Chaturvedi, Iti, Ong, Yew-Soon, Tsang, Ivor, Welsch, Roy, & Cambria, Erik (2016). Learning word dependencies in text by means of a deep recurrent belief network. *Knowledge-Based Systems*, *108*, 144–154.

Chen, Zhenhua, & Crandall, David (2018). Generalized capsule networks with trainable routing procedure. arXiv preprint arXiv:1808.08692.

Chen, Guibin, Ye, Deheng, Cambria, Erik, Chen, Jieshan, & Xing, Zhenchang (2017). Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In *IJCNN* (pp. 2377–2383).

Chikersal, Prerna, Poria, Soujanya, Cambria, Erik, Gelbukh, Alexander, & Siong, Chng Eng (2015). Modelling public sentiment in Twitter: Using linguistic patterns to enhance supervised learning. In *Computational linguistics and intelligent text processing* (pp. 49–65). Springer.

Choi, Jaewoong, Seo, Hyun, Im, Suii, & Kang, Myungju (2019). Attention routing between capsules.

Conneau, Alexis, Schwenk, Holger, Barrault, Loïc, & Lecun, Yann (2017). Very deep convolutional networks for text classification. In *15th conference of the european chapter of the association for computational linguistic*.

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, & Toutanova, Kristina (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Gong, Jingjing, Qiu, Xipeng, Wang, Shaojing, & Huang, Xuanjing (2018). Information aggregation via dynamic routing for sequence encoding. arXiv preprint arXiv:1806.01501.

Hammond, David K., Vandergheynst, Pierre, & Gribonval, Rémi (2011). Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, *30*(2), 129–150.

Hinton, Geoffrey E., Krizhevsky, Alex, & Wang, Sida D. (2011). Transforming auto-encoders. In *International conference on artificial neural networks* (pp. 44–51). Springer.

Kingma, Diederik P., & Ba, Jimmy (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kipf, Thomas N., & Welling, Max (2017). Semi-supervised classification with graph convolutional networks. In *ICLR*.

Li, Yang (2018). *Research on short text sentiment analysis and its applications* (Ph.D. thesis), ShanXi Province, China: Northwestern Polytechnical University.

Li, Yang, Pan, Quan, Wang, Suhang, Yang, Tao, & Cambria, Erik (2018). A generative model for category text generation. *Information Sciences*, *450*, 301–315.

Li, Yang, Pan, Quan, Yang, Tao, Wang, Suhang, Tang, Jiliang, & Cambria, Erik (2017). Learning word representations for sentiment analysis. *Cognitive Computation*, *9*(6), 843–851.

Pan, Shirui, Hu, Ruiqi, Fung, Sai-fu, Long, Guodong, Jiang, Jing, & Zhang, Chengqi (2019). Learning graph embedding with adversarial training methods. *IEEE Transactions on Cybernetics*, *50*(6), 2475–2487.

Ren, Hao, & Lu, Hong (2018). Compositional coding capsule network with k-means routing for text classification. arXiv preprint arXiv:1810.09177.

Sabour, Sara, Frosst, Nicholas, & Hinton, Geoffrey E. (2017). Dynamic routing between capsules. In *Advances in neural information processing systems* (pp. 3856–3866).

Sabour, Sara, Frosst, Nicholas, & Hinton, G. (2018). Matrix capsules with EM routing. In *6th international conference on learning representations*.

Wang, Yequan, Sun, Aixin, Han, Jialong, Liu, Ying, & Zhu, Xiaoyan (2018). Sentiment analysis by capsules. In *Proceedings of the 2018 world wide web conference* (pp. 1165–1174). International World Wide Web Conferences Steering Committee.

Wang, Yequan, Sun, Aixin, Huang, Minlie, & Zhu, Xiaoyan (2019). Aspect-level sentiment analysis using AS-capsules. In *The world wide web conference* (pp. 2033–2044). ACM.

Wu, Zonghan, Pan, Shirui, Chen, Fengwen, Long, Guodong, Zhang, Chengqi, & Philip, S. Yu (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Xi, Edgar, Bing, Selina, & Jin, Yang (2017). Capsule network performance on complex data. arXiv preprint arXiv:1712.03480.

Xiao, Liqiang, Zhang, Honglun, Chen, Wenqing, Wang, Yongkun, & Jin, Yaohui (2018). Mcapsnet: Capsule network for text with multi-task learning. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 4565–4574).

Zhang, Suofei, Zhou, Quan, & Wu, Xiaofu (2018). Fast dynamic routing based on weighted kernel density estimation. In *International symposium on artificial intelligence and robotics* (pp. 301–309). Springer.

Zhao, Wei, Peng, Haiyun, Eger, Steffen, Cambria, Erik, & Yang, Min (2019). Towards scalable and reliable capsule networks for challenging NLP applications. In *ACL* (pp. 1549–1559).

Zhao, Wei, Ye, Jianbo, Yang, Min, Lei, Zeyang, Zhang, Suofei, & Zhao, Zhou (2018). Investigating capsule networks with dynamic routing for text classification. arXiv preprint arXiv:1804.00538.