Multimedia Tools and Applications (2023) 82:8665–8690
https://doi.org/10.1007/s11042-021-11471-1

1209: RECENT ADVANCES ON SOCIAL MEDIA ANALYTICS AND MULTIMEDIA SYSTEMS: ISSUES AND CHALLENGES

Enhanced sentiment extraction architecture for social media content analysis using capsule networks

P. Demotte[1] · K. Wijegunarathna[1] · D. Meedeniya[1] · I. Perera[1]

Received: 30 November 2020 / Revised: 26 June 2021 / Accepted: 19 August 2021 /
Published online: 16 September 2021

Abstract
Recent research has produced efficient algorithms based on deep learning for text-based analytics. Such architectures could be readily applied to text-based social media content analysis. The deep learning techniques, which require comparatively fewer resources for language modeling, can be effectively used to process social media content data that change regularly. Convolutional Neural networks and recurrent neural networks based approaches have reported prominent performance in this domain, yet their limitations make them sub-optimal. Capsule networks sufficiently warrant their applicability in language modelling tasks as a promising technique beyond their initial usage of image classification. This study proposes an approach based on capsule networks for social media content analysis, especially for Twitter. We empirically show that our approach is optimal even without the use of any linguistic resources. The proposed architectures produced an accuracy of 86.87% for the Twitter Sentiment Gold dataset and an accuracy of 82.04% for the CrowdFlower US Airline dataset, indicating state-of-the-art performance. Hence, the research findings indicate noteworthy accuracy enhancement for text processing within social media content analysis.

Keywords Deep learning · Capsule networks · Twitter · Sentiment analysis · Social media content analysis

1 Introduction

With the recent rapid growth in information and communication technologies, social media has become a major form of human interaction. This has created a large pool of data collection for researchers to analyze data, infer trends, and make suggestions. Sentiment analysis is a process of systematic computational analysis of opinions, sentiments, and expressions in the text, and plays a vital role in analyzing user opinions [20]. Twitter is one of the

✉ D. Meedeniya
dulanim@cse.mrt.ac.lk

[1] Department of Computer Science and Engineering, University of Moratuwa, Moratuwa, Sri Lanka

growing social media networks with over 330 million users. Tweets are uniquely suited for sentiment analysis to infer further knowledge because of their brevity and precise nature. Tweets are limited to a maximum length of 280 characters though statistically, only 1% of the tweets reach this prescribed limit [26].

Recent research in text processing and sentiment analysis using deep learning has reported state-of-the-art results considering many aspects of language modeling through neural language representation [40]. Such techniques include Convolutional neural network (CNN) and Recurrent neural network (RNN) such as Long Short-Term Memory (LSTM) networks and Bi-LSTMs [35]. However, these techniques have inherent limitations that degrade performance in social media content analysis [15]. In contrast, CNNs that have produced better performance, are also have issues of information loss due to the max-pooling operation used between the convolutional layers [31]. In addition, recently introduced attention-based models such as Transformer-based strategies show better performance. However, these techniques require more language resources and computational power, which hinders their ability to be readily used in many language modeling tasks [2, 6, 9, 23, 43].

In this research, we have explored capsule networks. Capsule networks were originally invented for image analysis [28, 31] and recently introduced for natural language processing tasks [37]. We use capsule network based approach for social media analysis to extract sentiments present in the textual content. Capsules-based architectures have produced competitive results [14, 37], especially compared to CNN-based approaches [39]. Capsules within capsule networks encode information about the objects within the data as a vector representation. This elevates the ability of the capsules to capture the exact order pose of the information for the background information in many natural language processing (NLP) tasks. Moreover, routing procedures introduced under capsule networks [31] mitigate the information loss seen in pooling strategies in CNNs. Therefore, we empirically evaluate shallow capsules, capsules with static routing, and capsules with dynamic routing against the CNN-based Twitter sentiment classification procedures to set a new benchmark for Twitter sentiment analysis using capsule networks. Furthermore, the capsule networks presented in this study show improved accuracies compared to the baseline models for all datasets used for the experiment. Also, the capsule networks are lightweight and easy to train. Shallow capsule networks with static routing produce optimal performances considering the short sequential nature of the Twitter text.

The rest of the paper is structured as follows. Section 2 discusses the related work and the usage of capsule networks in NLP. The capsule networks based architectures are discussed in detail under the model architectures in Sect. 3 along with the static routing and dynamic routing algorithms. Sections 4 and 5 describe the implementation process and the result analysis, respectively. Section 6 discusses the lessons learned with the novel contribution of the proposed study and Sect. 7 concludes the paper.

2 Related work

Apart from the traditional statistical methods, many tools and solutions of social media content analysis, particularly in Twitter data, incorporate machine learning techniques. Advanced techniques for Twitter data classification utilize n-gram features as local spatial patterns and sequential information to achieve sophisticated language modeling procedures. Such strategies include CNNs and RNNs such as LSTM [40], which model the text classification tasks beyond the boundaries of the meaning of words. Therefore, low

dimensional features such as word embeddings [24, 25] are favorably used to extract convoluted syntactic and semantic features of the text content from social media.

Among the related studies, Liao et al. have used a basic form of CNNs, with a single hidden layer, for sentiment analysis of Twitter data [19]. Jianqiang et al. have proposed the use of convolutional neural networks for Twitter sentiment analysis using their Global Vectors for word representation (GloVe) Deep Convolutional Neural Network (DCNN) model [11]. This GloVe-DCNN model has shown improvements over Bag of Words (BoW) with Support vector machines (SVM), BoW with Logistic Regression (LR), and GloVe with LR models. In addition, Johnson and Zang have proposed the use of CNNs for high dimensional text classifications in the broader topic of sentiment analysis [12] prior to Jianqian et al. Although they have achieved significant improvements, their models were complex and expensive to train. The study by Cai has compared the performance of very deep convolutional neural networks (VD-CNN) [5] with Google's pre-trained Bidirectional Encoder Representations from Transformers (BERT) architecture [8]. The results have placed the VD-CNNs below BERT's state-of-the-art performance; however, the VD-CNN's architecture is comparatively simpler and cheaper to train than BERT [8, 29] and BERT variants such as RoBERTa (A Robustly Optimized BERT Pretraining Approach) [21], Albert (A Lite Bert) [17] and DistilBERT (distilled version of BERT) [34].

Successively, the possibility of using ensemble architectures for Twitter sentiment analysis has been researched with a range of techniques. Twitter sentiment analysis using an ensemble of several traditional approaches like Naive Bayes, Random Forest, SVM and Logistic Regression has proven more accurate than the models that have used individually [33]. Ensembles of traditional machine learning approaches and novel deep learning techniques have also been proposed by Araque et al. [3]. They have tried both an ensemble of several sentiment classifiers trained with different kinds of features and an ensemble of features, where the combination is made at the feature level.

The multimodality of Twitter presents a new dimension to the challenges in sentiment analysis. Twitter enables users to express themselves using images and GIF videos, which are combined with text. While most studies tackle sentiment analysis with one modality, Kumar and Garg's have attempted to analyze tweets consisting of both infographic and typographic data [16]. Their study has addressed the multimodal sentiment analysis of tweets. For textual sentiment analysis, they have applied a hybrid approach of lexicon and machine learning techniques. Various other neural networks such as Skip-grams and denoising autoencoders have also been tested for multimodal Twitter data for sentiment analysis [4].

Sentiment analysis of Twitter data has been harnessed in several research studies showcasing its applications. For instance, recent research on COVID-19 related tweets and social media content harvested by filtering #COVID related keywords produced some intriguing insights into the reactions of the masses to pandemic-related restrictions and government interventions [10, 18]. The study by Imran et al. [10], has addressed how countries from the same region show high correlations among them except for Norway and Sweden, mainly due to the different approaches taken by their respective governments. The Stanford CoreNLP [22] tool has also been used in building a system that analyses tweets in real-time to predict stock market fluctuations [7]. The proposed system attempts to predict the stock market prices of several reputed companies by the sentiments of the tweets that have mentioned the company names. Signal or spike detection in Twitter data is another interesting area of research in Social media analysis. Spikes in tweets in the form of hashtags, frequently mentioned keywords, sentiments of tweets and volume of tweets can be used to infer trends and make useful predictions into the future.

As a related study, Nazir et al. [27], have proposed the use of three viable algorithms to detect spikes in tweets. They have assessed the spikes in tweets, while showing the use of integrating a Gaussian algorithm and a threshold algorithm that provides better results on the real-time data.

When considering the domain of text classifications and language modeling tasks, regardless of certain advancements that were produced by the LSTMs and Bi-LSTMs in neural language representation, their intrinsically sequential nature of modeling strategies has led to several limitations. While vanishing gradient problem hinders encoding longer sequences within the learning approach [13], LSTMS and Bi-LSTMs also endure from a computational bottleneck with the sequential information processing [41]. CNNs overcome this computational bottleneck by providing parallelization within convolutional filters. While the CNNs produced better results compared to LSTMs in text classification [39] yet endure the information loss due to the pooling strategy when representing deep neural language representation [31].

Table 1 summarizes the techniques used by some of the related studies. Most of the studies have used techniques such as Artificial recurrent neural network (RNN), Deep LSTM, and different embedding methods such as BoW, GloVe and Embeddings proposed from language model (ELMo) and BERT. Among these, several studies have used GloVe word embedding.

The capsule networks have produced state-of-the-art results with the dynamic routing procedure proposed by Sabour et al. [31]. The intention behind the capsule strategy was to represent the features of objects within the data as vector representation to identify the exact order or pose of the information. The dynamic routing procedure reduces the information loss of CNNs due to max-pooling and elevates the advancement of the part-to-whole relationship between capsules for deeper capsule representation in classification tasks. Rajasegaran et al. [30], have proposed an optimized strategy to eliminate high computation cost and vanishing gradient problem of deeper capsules by applying 3D convolution with capsule strategy. This method reduced 68% of parameters while producing state-of-the-art results in the domain of capsules.

Inspired by the capsule network architecture, Wang et al. [36] have applied capsules for sentiment classifications with the combination of RNNs, which produced the best results at that time. In another study, Yang et al. [37] have conducted an empirical experiment of capsule networks with dynamic routing to validate the utilization of capsule networks for text classification. The implementation of different variations of capsule architectures as capsule-A and capsule-B for binary and multi-class text categorization with a dynamic routing process, have produced optimal performances in text classification. Another, dynamic routing based Siamese architecture with a twin capsule network and a fully connected network has proposed by Abeysinghe et al. [1]. They have shown that the use of capsule layers-based Siamese network reduces the information loss in CNNs and allows train the model with a smaller number of parameters and datasets, while achieving on par performance with CNNs. With even deeper analysis, Kim et al. [14] have produced an approach based on static routing between capsules depicting the use of capsules for text classification. This method has addressed the limitations of capsule networks with dynamic routing due to the variations of text with background noise, as opposed to the corresponding image classification tasks. We explore the use of capsule networks with static and dynamic routing methods to obtain higher accuracies for the sentiment extractions from social media text content. Thereby, setting a new standard for benchmark in sentiment analysis of Twitter data using deep learning architectures with low resource setting.

Table 1 Summary of techniques used by related studies

Description	CNN	RNN based LSTM	Word embedding (Context-free)	Contextual embedding	Transformer based embedding	Ensemble model (ML)
Analyze sentiments using contextual embedding [26]		BiLSTM	GloVe	ELMo	BERT	
Twitter sentiment analysis without word embedding [19]	X					
A sentiment analysis model with word embeddings and word sentiment polarity score [11]	X		GloVe			
Binary classification approach to sentiment analysis of tweets [5]	X		GloVe		BERT	
A weighted ensemble model to analyze tweet sentiments [33]			BoW			X
Multimodal sentiment analysis to identify sentiment polarity of tweets with text, image or infographics [16]	X					
Analyze the sentiment polarity, reactions among cultures [10]		X	FastText GloVe			
Real-time sentiment analysis of emotional tweets to predict stocTwitter [7]		X				
A model to capture tweets semantics and sentiments [38]	X	X				

3 Model architectures

The proposed model in this study uses shallow capsule networks, deep capsule networks and ensemble deep capsule networks on top of the CNNs intending to enhance the classification strategy. The scalar representation of CNNs based language modeling tasks is replaced with vector representation of capsules to identify the exact order or pose of the information. Penetrating deeper with capsules, an additional routing mechanism is introduced to map the low-level capsules to the high-level capsules. This technique was used to enhance the pooling strategy in CNNs [31], which results in information loss. This section describes the baseline CNN structure, main capsule-based layers on top of the CNN structure, dynamic routing and static routing strategies between capsules, and the task of Twitter sentiment analysis with the proposed capsule architecture.

3.1 Convolutional neural network (CNN)

For the sentiment analysis tasks using CNN-based techniques, the text representation of Twitter data content is fed into the CNN using pre-trained word vectors. Therefore, each word in tweets is considered as a word vector. Let, a tweet consist of n words with k-dimensional word vectors. The feature map specific for a tweet could be considered as a map obtained through the concatenation of word vectors governed by Eq. (1). Here, $x_i \in R^k$ refers to the word vector of the *i-th* word of the input tweet and $\oplus$ refers to the concatenation operator of the word vectors. Therefore, the concatenated word-vectors form a $n\times$ d dimensional feature map which will be used as the input features for the CNN.

$$\text{feature map consisted of word vectors } = \mathrm{x}_1 \oplus \mathrm{x}_2 \oplus \cdots \oplus \mathrm{x}_\mathrm{i} \oplus \ldots \oplus \mathrm{x}_\mathrm{n} \tag{1}$$

The convolution operations extract n-gram features from a context window, where a filter is applied on top of the context window. Let the context window be $x_{i:i+l} \in R^{l\times k}$, where the context window consists of l number of word-vectors concatenated with each other and i is the starting index of the context window. A filter $H \in R^{l\times k}$ is applied on top of the corresponding context window to extract a feature $f_i \in R$. This process is governed by Eq. (2), where $\circ$ represents the element-wise matrix multiplication, b denotes a biased term and g represents an activation function (ReLU or tanh) for extracted features.

$$\mathrm{f_i} = \mathrm{g}(\mathrm{H} \circ \mathrm{x_{i:i+l}} + \mathrm{b}) \tag{2}$$

The considered filter convolves with each possible context window,$\mathrm{CW} \in \{\mathrm{x_{1:1+l}}, \mathrm{x_{2:l+1}}, \ldots\ldots\ldots, \mathrm{x_{n-l+1:\,n}}\}$. This extracts the number of features governed by Eq. (3). Here, d_{in} is the input dimension for the convolution operation (concatenated word vectors) and d_{out} is the resulting number of features after the convolution. The padding is kept as 0 and strides as 1 for the convolution process in our experiments.

$$\mathrm{d_{out}} = \frac{\mathrm{d_{in}} - \text{ Kernel Size } + (2 \times \text{ Padding})}{\text{Stride}} + 1 \tag{3}$$

This procedure generates $(n - l + 1)$ sized feature column. We can use the max-pooling operation on top of the extracted features to highlight the most significant feature in the extracted feature set as, $f_{max} = max\{f_i\}$. Consequently, the N number of features could be generated with N number of filters. For Twitter sentiment analysis tasks, these extracted

features could be combined with a fully connected neural network using the softmax or sigmoid activated dense layers based on the requirements of the task.

3.2 Capsule layers and routing algorithms

Vanilla capsule networks, built solely upon convolutions [37], mainly include three varieties of layers based on the task specificity, namely primary capsules, convolutional capsules and text capsules. We have evaluated combinations of different variations of these layers. Moreover, we used both dynamic and static routing between capsules. These routing procedures are established instead of pooling operations in CNNs, to obtain better performances in feature extraction and computational processing. Compared to the pooling operations in CNNs such as max-pooling and average pooling, the dynamic routing procedure does not discard the information of a specific region that describes the precise position of an entity within the considered region [1, 31]. As per the intuition behind pooling, the most significant and average feature of a given region represents that the considered region in max-pooling and average pooling, respectively. Thus, pooling does not encode the exact order or pose of the information that explains the precise position of an entity within the data. The dynamic routing algorithm proposes a novel strategy to map low-level capsules to high-level capsules in a hierarchical manner based on a matrix multiplication operation, where the exact pose or order of information within the capsules are preserved.

3.2.1 Primary capsules

We represent the objects within the data as the vector representation of capsules instead of the scalar representation of the CNNs using the following process. The generated feature columns are concatenated to obtain a feature map as in Eq. (4), instead of applying pooling operations on the extracted features by N filters. The feature map $M \in R^{(n-l+1)\times N}$ includes feature columns extracted by N filters and $m_i \in R^{(n-l+1)}$ represents the feature column extracted by $i-th$ filter.

$$\text{Feature map (M)} = \{\mathrm{m}_1, \mathrm{m}_2, \mathrm{m}_3, ..., \mathrm{m}_\mathrm{i}, ..., \mathrm{m}_\mathrm{N}\} \tag{4}$$

In order to obtain the primary capsules based on the extracted features by the CNN, a matrix multiplication operation is carried out. We instantiate a capsule $c_i \in R^d$ as d -dimensional vectors. A matrix filter $W_i \in R^{N\times d}$ is multiplied with concatenated feature columns M, given in Eq. (4). This procedure results in a column list of capsules $c \in R^{(n-l+1)\times d}$ computed as given in Eq. (5), where b_1 represents the bias term and f represents the squash function.

$$\mathrm{c} = \mathrm{f}\,(\mathrm{W_i M} + \mathrm{b_1}) \tag{5}$$

Moreover, with p number of matrix filters, a map of capsules $C \in R^{(n-l+1)\times p\times d}$ generated with $(n-l+1)\times p$ number of capsules. The squash function is stated in Eq. 11, which converts each capsule's length between a value 0 and 1. Therefore, the length of a capsule could be considered as the probability of the existence of an entity within capsules such as syntactic and semantic information of text or sentiment category of given data.

3.2.2 Convolutional capsules

In this layer, the capsules are mapped to a local region of the layer below to facilitate the ability of capsules to identify local spatial patterns quite effectively. We assume that a local region with size $(m \times p)$ in the layer below (primary capsule layer) is mapped to the convolutional capsule layer. Therefore, capsules in that region compute matrix multiplication operations to learn child-to-parent relations between low-level capsules and high-level capsules. For the matrix multiplication operation, a weight matrix $W^c \in R^{E \times d \times d}$ is used, where, E denotes the number of capsules in a convolutional capsule layer. Given a child capsule, a parent capsule is generated according to Eq. (6). Here, $\hat{u}_{j|i}$ is the convolution capsule generated, u_i is the local region $(m \times p)$ for a given child capsule in lower-layer, W_j^c is the $j-th$ matrix in the matrix tensor W^c and $\hat{b}_{j|i}$ is the bias term for $\hat{u}_{j|i}$ convolution capsule generation for a given lower layer capsule u_i. Consequently, $(n-l-m+2) \times E$ number of d-dimensional convolutional capsules are generated using this procedure.

$$\hat{u}_{j|i} = W_j^c u_i + \hat{b}_{j|i} \tag{6}$$

3.2.3 Sentiment capsules

The sentiment capsule layer is designed as the final layers of capsule architectures. This layer mainly consists of capsules for each target sentiment category to represent classification tasks. Therefore, the capsules in this layer are generated based on the matrix multiplication to learn child-to-parent relationships. To obtain the sentiment capsules based on the layer below, all capsules in that layer are flattened into the list of capsules and multiplied by the transformation matrix $W^d \in R^{U \times d \times d}$ as in Eq. (6), where U denotes the number of sentiment capsules for the corresponding task and d is the instantiated parameter for the dimension of capsules. The capsules in the sentiment capsule layers have the length or the norm of the vector representation denoting the probability of the existence of the target sentiment category. Thus, these probabilities were used to extract the sentiment of a given sequence of text.

3.2.4 Child-to-parent relationship

Routing by agreement algorithms is initially designed as a strategy to learn the child-to-parent relationship between capsules incrementally, by mitigating the issues of the pooling strategies used in CNNs, to map low-level features to high-level features in Deep CNNs [31]. Also, Kim et al. [14] have suggested that static routing procedures are better at handling variability of background information of text than the dynamic routing procedures that are proposed by Yang et al. [37]. In this study, we empirically evaluated these routing algorithms for capsule networks for Twitter sentiment analysis.

3.2.5 Dynamic routing between capsules

The main purpose of the dynamic routing algorithm is to establish a non-linear map between child capsules to parent capsules iteratively, to send child capsules to its most relevant parent capsules by ensuring that the child-to-parent relationship is correctly

established. Therefore, using this process each child-capsule can learn its potential parent to be mapped incrementally varying the connection strength between child-to-parent. This procedure elevates the issues due to the pooling strategy used in CNNs. Generally, pooling strategies result in information loss due to the neglect of surrounding features of the most significant features [1]. Dynamic routing further elevates vector representation of capsules considering essential background information, especially for text-based classification tasks [31]. Algorithm 1 describes the dynamic routing between two capsule layers. First, we initialize the log prior probabilities b_{ij}, between each capsule i in the layer below and each capsule j in the layer above, as stated in Eq. (7) that corresponds to line-3 of Algorithm 1. These log prior probabilities b_{ij}, represent the connection strength between a pair of child and parent capsules.

$$b_{ij} \leftarrow 0 \tag{7}$$

Algorithm 1: Dynamic Routing Algorithm

1. $procedure\ Routing\ (\ \hat{u}_{j|i}\ ,r\ ,l\ )$:
2. $for\ all\ capsules\ i\ in\ layer\ l\ and\ j\ in\ layer\ l+1\ do$:
3. $\quad b_{ij} \leftarrow 0$
4. $for\ r\ iterations\ do$:
5. $\quad for\ all\ capsules\ i\ in\ layer\ l\ do$:
6. $\quad\quad c_i \leftarrow softmax\ (\ b_i\ )$
7. $\quad for\ all\ capsules\ j\ in\ layer\ (l\ +1)\ do$:
8. $\quad\quad s_j \leftarrow \sum_i \quad c_{ij} * \hat{u}_{j|i}$
9. $\quad for\ all\ capsules\ i\ in\ layer\ l\ and\ j\ in\ layer\ (l\ +\ 1)\ do$:
10. $\quad\quad b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} * s_j$
11. $for\ all\ capsules\ j\ in\ layer\ l+1\ do$:
12. $\quad\quad v_j \leftarrow squash\ (\ s_j\ )$
13. $return\ v_j$

Secondly, the log prior probabilities are learnt incrementally within the iterative learning procedure as shown in line-4 of Algorithm 1. The connection strength of a child-capsule for all parent-capsules in the layer above is calculated based on the softmax function to indicate the probability of sending the information represented in the child-capsule to each of the parent capsules as shown in line-6 of Algorithm 1. This process is governed by Eq. (8). Here, c_{ij} represents the coupling coefficient between capsule i in the layer below and capsule j in the layer above, and *exp* denotes the exponentiation function. The proposed strategy based on the softmax function calculates all coupling coefficients between a capsule in the layer below and every capsule in the layer above for routing purposes.

$$c_{ij} \leftarrow \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \tag{8}$$

Moreover, the routing procedure computes the capsules in the layer above using coupling coefficients and predicted capsules, which are retrieved during the matrix transformation process as described in Sects. 3.2.2 and 3.2.3. This process represents by line-8 of Algorithm 1 and governs by Eq. (9). Here, s_j denotes the computed capsules for the layer above. The connection strength c_{ij} represents the coupling coefficient between capsule i in the layer below and capsule j in the layer above. As mentioned in Eq. (6), $\widehat{u}_{j|i}$ is the generated convolution capsule.

$$s_j \leftarrow \sum_i c_{ij} * \widehat{u}_{j|i} \tag{9}$$

Also, as shown in line-10 of Algorithm 1 and Eq. (10), the log prior probabilities b_{ij}, between capsule i in the layer below and capsule j in the layer above are updated iteratively by considering the similarity between the predicted capsule $\widehat{u}_{j|i}$ and computed capsule s_j within the routing procedure.

$$b_{ij} \leftarrow b_{ij} + (\widehat{u}_{j|i} * s_j) \tag{10}$$

In our proposed model, the squash non-linearity is applied for each computed capsule s_j after the iterative updating process, which hinders the degradation of instantiated parameters of capsules within the iterative process [31]. The squash function is applied to each computed capsule s_j as in Eq. (11), which corresponds to line-12 of Algorithm 1. Here, $||s_j||$ denotes the standard norm for capsule s_j. The length vector v_j represents the probability of the existence of objects with a capsule. Therefore, the final layers of the capsule architectures are designed to represent the tweet category existence probability within the length of the capsules.

$$v_j \leftarrow \frac{||s_j||^2 s_j}{(1 + ||s_j|^2)||s_j||^2} \tag{11}$$

3.2.6 Static routing between capsules

The text-based classification tasks have higher variability of background information compared to image processing tasks [14]. As suggested by Kim et al. [14], the text-based tasks are considered under a static routing process that eliminates different variations of routing between child-to-parent based on spatial patterns, without considering the whole context of the text. Thus, the capsules in the layer below will only be mapped to their parent's capsules in the layer above, using a matrix transformation governed by Eqs. (12) and (13).

$$s_j = \sum_i W_{ij} h_i \tag{12}$$

$$v_j = \text{squash}(s_j) \tag{13}$$

Here, $W_{ij} \in R^{M \times N}$ is the transformation matrix that transforms the capsules i in the layer below to capsules j in the layer above. M is the dimension of the capsules to be generated in the layer above and N is the number of capsules in h_i that denotes the capsules in the

layer below. Then, the squash function shown in Eq. (11) is applied to obtain the vectors with the length of the vector v_j, as the probability of the existence of an entity within a capsule.

3.2.7 Loss function

We classify Twitter data using a separate margin loss function to identify the location of a given category in each sentiment capsule. Here, we utilize the length of the capsule to represent the probability of the existence of a given sentiment category with a sentiment capsule. Equation (14) is used to derive the marginal loss for sentiment capsules [31]. If the tweet category exists within the text capsule, then $Ts = 1$, otherwise it is set to 0. The values m^+ and m^- are set as 0.9 and 0.1 accordingly. After several experiments, we have set the down-weighting coefficient λ to 0.25 that gives the optimal performance. This down-weighting coefficient reduces the initial learning of sentiment capsules for tweet sentiment categories that are not present within those sentiment capsules. The total loss is simply taken as the sum of the losses for all sentiment capsules.

$$\text{Margin Loss } = \mathrm{T_s}\ \max(0, \mathrm{m}^+ - ||\mathrm{v_s}||)^2 + \lambda(1 - \mathrm{T_s})\ \max\ (0, ||\mathrm{v_s}|| - \mathrm{m}^-)^2 \qquad (14)$$

3.3 Capsule network architectures

3.3.1 Shallow capsule network

The proposed solution uses two types of shallow capsule networks as illustrated in Fig. 1. These capsule networks include two capsule layers namely, primary capsules and sentiment capsules followed by the word embedding layer and the convolutional layer. The convolutional layer is employed specifically to extract n-gram information from the text. Primary capsules are generated by considering the feature maps obtained through the CNN layer. The number of capsules in the final capsule layer or the sentiment capsule layer is equal to the target number of sentiment categories. Thus, sentiment capsules represent the sentimental features of the text that are utilized to classify the text into sentiment classes. As the routing procedure between capsules, both dynamic routing and static routing have experimented.

In a neural network perspective, the word embedding layer consists of n number of k-dimensional vectors where the ultimate input feature map represents $n \times k$ dimensionality. For shallow capsule networks, this feature map is fed to a CNN layer, where

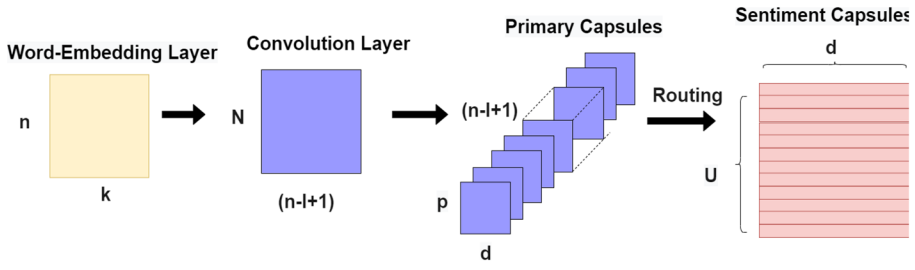

Fig. 1 Shallow capsule Network

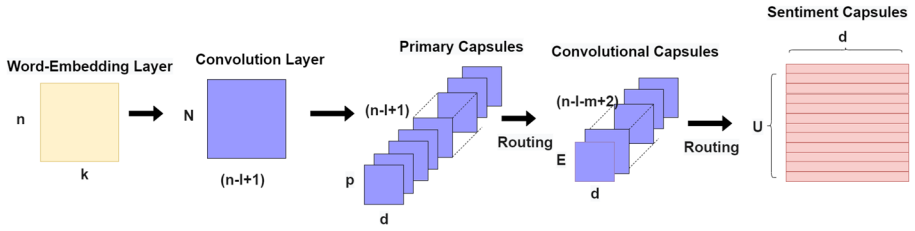

Fig. 2 Deep capsule Network

N number of $l \times k$ filters are utilized to extract n-gram features from the text. Thus, the CNN layer generates N number of feature columns that are $(n - l + 1)$ in size. Furthermore, considering the procedure in Sect. 3.2.1, a map of capsules $C \in R^{(n-l+1) \times p \times d}$ generated as the primary capsules. These capsules with routing procedures described in Sects. 3.2.5 and 3.2.6 generate sentiment capsules with $u \times d$ in dimensionality, where u is the number of target sentiment categories and d indicates the dimensionality of sentiment capsules.

3.3.2 Deep capsule network

The deep capsule network architecture combines all three capsule layers: primary capsule layer, convolutional capsule layer, and sentiment capsule layer followed by the word embedding layer and the convolutional layer, as shown in Fig. 2. The convolutional layer is employed to extract n-gram information from the text as in the shallow capsule networks. Primary capsules are generated by considering the feature maps generated through the CNN layer. The convolutional capsules are generated by considering the dynamic routing procedure in Sect. 3.2.5. The significance of convolutional capsules can elaborate as the ability to relate local features within the text since the local regions of primary capsules are mapped to the convolutional capsules as indicated in Sect. 3.2.2. Moreover, as in the shallow capsule networks, the number of capsules in the final capsule layer or the sentiment capsule layer is equal to the target number of sentiment categories. These sentiment capsules are generated based on convolutional capsules and the dynamic routing procedure.

From an architectural perspective, deep capsule networks only have one additional capsule layer namely the convolutional capsule layer. The word embedding layer, which is the initial layer of the network consists of a feature map that represents $n \times k$ dimensionality. This feature map is fed to a CNN layer where N number of $l \times k$ filters was utilized to extract n-gram features from the text as in shallow capsule networks. The resultant $N \times (n - l + 1)$ feature map is utilized to generate primary capsules as indicated in Sect. 3.2.1. Ultimately a map of capsules $C \in R^{(n-l+1) \times p \times d}$ generated as the primary capsules. These capsules with dynamic routing procedures as described in Sect. 3.2.5 generate convolutional capsules with $(n - l - m + 2) \times E \times d$ in dimensionality. As the final capsule layer, sentiment capsules are generated for the sentiment classification purpose. The dynamic routing procedure with convolutional capsules was utilized to construct the sentiment capsules. These capsules are $u \times d$ in dimensionality, where u is the number of target sentiment categories and d represents the dimensionality of sentiment capsules.

3.3.3 Ensemble capsule network

Generally, the ensemble capsule networks have produced prominent performances in text classification tasks [37]. Therefore, we evaluated an ensemble capsule network for Twitter data sentiment classification with the dynamic routing algorithm. As illustrated in Fig. 3, the ensemble capsule network consists of three layers namely the primary capsule layer, convolutional capsule layer, and sentiment capsule layer. Three separate deep capsule networks consisting of these layers were utilized to extract different variations of n-grams features from Twitter data. In the final sentiment capsule layer, the generated capsules were average pooled considering three capsule networks for the classification purpose.

4 System Methodology

4.1 Datasets

We used two widely used and publicly available Twitter datasets as follows:

lower-alpha CrowdFlower US Airline dataset - this dataset is released by CrowdFlower and has a total of 14,640 tweets related to six major US Airlines: American airline, United airline, US Airways, Southwest airline, Delta airline, and Virgin airline. Each of these tweets is tagged as positive, negative, or neutral tweets.

1 The Stanford Twitter Sentiment Gold (STSGd) dataset – this dataset is created by Saif et al. [32]. There are 2034 tweets and manually annotated as negative or positive on the agreement of three annotators.

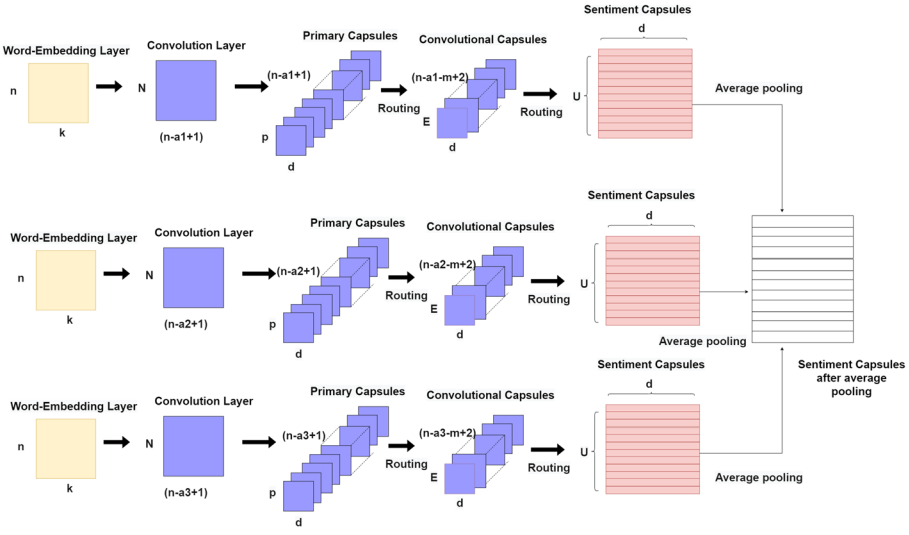

Fig. 3 Ensemble capsule Network

The summary statistics of each dataset are shown in Table 2. For the experiment with the CrowdFlower US Airline dataset, the vocabulary $|V|$ is consisted of 26841 tokens and the maximum tweet length was 36. For the STSGd dataset, the vocabulary size $|V|$ was set to 8470 and the maximum tweet length was set to 31.

4.2 Model implementation

4.2.1 Word-embeddings

In the deep learning era, the models trained using pre-trained word-embeddings have reported state-of-the-art performance even without using linguistic resources. This is because the pre-trained word-embeddings can extract syntactic and semantic information of a given token in a language-independent manner. The proposed Twitter sentiment analysis task uses two types of word-embeddings models as follows. We empirically evaluate the usage of the two embeddings based on the task specificity on Twitter data sentiment analysis.

- GloVe 300-dimensional word-vectors trained Common Crawl corpus with 840 billion tokens with a vocabulary of 2.2 million tokens.
- Glove Twitter 200-dimensional word-vectors pre-trained on 2 billion tweets, 27 billion tokens and vocabulary of 1.2 million tokens that are Twitter specific.

4.2.2 Baseline models

The baseline model for the STSGd dataset is derived from the study presented by Jianqiang et al. [11]. They have used a radial basis function (RBF) kernel SVM and an LR model using unigram and bigram features consists of BoW. In addition, they have used the same models with additional word sentiment polarity features, Twitter-specific features and word vector features with GloVe. The DCNN using GloVe word embeddings is considered as a basis for the proposed capsule network in our study. Further, we have used 10-fold cross-validation as the evaluation metrics of our approach using the CrowdFlower US Airline dataset.

4.2.3 Data pre-processing

Generally, Twitter content includes high noise due to non-dictionary terms, ill-formed language structure, and grammatical mistakes. Therefore, the following procedures are used to reduce the noise within the data.

Table 2 Statistics for datasets used under experiments

Dataset	Positive	Negative	Neutral	Total tweets	$\|V\|$	Max tweet-length
CrowdFlower US Airline	2363	9178	3099	14640	26841	36
Stanford Twitter Sentiment Gold (STSGd)	632	1402	-	2034	8470	31

- Removed the special characters within the tweets that do not carry any specific information about the tweet category.
- Removed the URLs and links within the tweets as they do not carry any sentiment-specific information.

4.2.4 Sentiment extraction

Generally, the activities of the neurons within capsules in a capsule network represent an entity within data in its exact order or pose and with certain other properties, using a vector representation of capsules. As the final layer of our deep learning architecture, we use sentiment capsules to represent the sentiment categories in a sentiment analysis task. Since this considers the number of sentiment categories, we use three and two sentiment capsules for the CrowdFlower US Airline dataset and the Stanford Twitter Sentiment Gold dataset, respectively, which correspond to the number of sentiment classes in each dataset. Furthermore, the length of a sentiment capsule or the norm of vector representation of the capsule represents the probability of the existence of sentiment category within the capsule. Thus, these probabilities are used to extract the sentiment of a given sequence of text.

4.2.5 Classification model

We evaluated the Twitter-based sentiment classification model for each dataset by varying the components of the models as follows. This process enables to measure model performance empirically, by showing the effectiveness of capsule-based architectures for Twitter analysis.

- Four main model architectures are used as shallow capsule network with static routing, shallow capsule network with dynamic routing, deep capsule network with dynamic routing, and ensemble capsule network with dynamic routing.
- Each model is fed with both Twitter-specific 200-dimensional word-embeddings and 300-dimensional common crawl corpus-based GloVe word-embeddings.

We have used the Adam optimizer for the optimization process with exponential learning rate decay. The models are trained on Google Colab with Tensorflow as the implementation platform. The optimal hyperparameters for the models in the STSGd dataset are indicated in Table 3. For each model training, the learning rate was set to $1e-3$, and the learning rate decay was set to 0.95. Max tweet length is defined as the tweet length to be fed to the models as input embedding dimension, considering the variations of the datasets. The evaluation is based on the 10-fold cross-validation approach.

As given in Table 3, each capsule architecture with dynamic routing processes utilizes three iterations for dynamic routing procedure to enhance the child-to-parent relationship between capsules. The number of convolutional filters in the initial layer of each model is indicated in the column of the number of filters. The ensemble capsule network utilizes three filter sizes in the initial convolutional layers to structure ensemble architecture as shown in the filter sizes column of Table 3. All other models use the filter size of three to extract n-gram features from the convolutional layer. Additionally, the dimension of capsules for each layer is indicated layer-wise. While shallow capsule networks have two layers, deep capsule layers networks have three layers of capsules, respectively. Here, |C| indicates the number of capsules in layer-wise for each layer of the network. Further, the

Table 3 Optimal Hyper-parameters for each model for Twitter Sentiment Glod dataset

Model	Routing iterations	Batch size	No. of filters	Filter sizes	|C|	Capsule dimensions
GloVe Twitter + shallow capsule network with static routing	-	32	256	3	32/2	16/16
GloVe Twitter + Shallow capsule network with dynamic routing	3	16	256	3	32/2	16/16
GloVe Twitter + Deep Capsule network	3	32	256	3	32/16/2	16/16/16
GloVe Twitter + Ensemble deep capsule network	3	32	256	3, 4, 5	32/16/2	16/16/16
GloVe + shallow capsule network with static routing	-	32	256	3	32/2	16/16
GloVe + Shallow capsule network with dynamic routing	3	16	256	3	32/2	16/16
GloVe + Deep Capsule network	3	32	256	3	32/16/2	16/16/16
GloVe + Ensemble deep capsule network	3	32	256	3, 4, 5	32/16/2	16/16/16

Table 4 Experimental results for Twitter Sentiment Gold (STSGd) dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1(%)
BoW + SVM [11]	68.79	74.68	55.76	44.05
BoW + LR [11]	75.86	72.36	74.64	65.34
GloVe + SVM [11]	80.07	75.09	78.08	69.21
GloVe + DCNN [11]	85.97	82.75	82.61	82.65
GloVe Twitter + shallow capsule network with static routing	86.53	86.45	86.51	86.19
GloVe Twitter + shallow capsule network with dynamic routing	86.12	85.96	86.12	85.85
GloVe Twitter + deep capsule network	78.76	76.67	78.76	75.87
GloVe Twitter + ensemble deep capsule network	80.60	80.83	80.60	80.36
GloVe + shallow capsule network with static routing	86.74	**87.11**	**86.87**	**86.79**
GloVe + shallow capsule network with dynamic routing	**86.87**	86.86	86.87	86.70
GloVe + deep capsule network	83.18	83.21	83.18	82.45
GloVe + ensemble deep capsule network	83.30	83.32	83.50	82.64

same hyper-parameters are used for the CrowdFlower US Airline dataset, where the final sentiment capsule layer is configured with three capsules by considering the sentiment categories namely positive, negative, and neutral.

5 Result evaluation and analysis

We used accuracy, precision, recall, and F1 score as the evaluation metrics for the STSGd dataset. Since the CrowdFlower US Airline dataset includes multi-class classification tasks, weighted evaluation metrics are used for the evaluation. The 10-fold cross-validation is used for each experiment. The classification results for the STSGd dataset and CrowdFlower US Airline dataset are given in Tables 4 and 5, respectively.

For the results obtained for the STSGd dataset, all the trials with capsule networks outperformed existing baseline techniques. This could be justified as the ability of capsule networks to handle language syntactic and semantic information quite effectively utilizing

Table 5 Experimental results for CrowdFlower US Airline dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1(%)
GloVe Twitter + shallow capsule network with static routing	80.19	79.62	80.19	79.78
GloVe Twitter + shallow capsule network with dynamic routing	80.53	80.27	80.53	78.82
GloVe Twitter + deep capsule network	72.27	69.30	72.27	70.80
GloVe Twitter + ensemble deep capsule network	74.11	73.07	74.11	72.17
GloVe + shallow capsule network with static routing	**82.04**	**81.31**	**82.04**	**81.11**
GloVe + shallow capsule network with dynamic routing	80.26	79.56	80.26	79.79
GloVe + deep capsule network	76.16	75.34	76.16	75.60
GloVe + ensemble deep capsule network	76.76	76.39	76.76	76.35

vector representation of the capsules. Capsules-based strategies further demonstrate the capability of handling background information of text which validates the optimal results in text-based sentiment analysis tasks. Since tweets are based on short texts, shallow capsule networks reported optimal performance for deep capsule-based architectures. Shallow capsule networks with static and dynamic routing have produced competitive results, while static routing based technique has slightly higher performance. Therefore, as stated by Kim et al. [14], this could be elaborated as the ability to identify the variability of background information of text using static routing. Compared to the image classification task, text-based classification tasks do not depend on the exact order of words like the objects within images. Thus, the static routing can be optimistic when using child-to-parent links among layers of the capsule network.

Moreover, the experiments were conducted with two varieties of input embeddings obtained through the Stanford GloVe project. The 300-dimensional GloVe embeddings trained on large common crawl corpus reported better performance compared to the 200-dimensional GloVe embeddings trained on Twitter-specific data. This observation could be justified by the fact that generic GloVe embeddings have learned deep semantic structure compared to the Twitter-specific GloVe embeddings, which carry information only for a specific domain.

In particular, the deep capsule architectures perform slightly lower compared to shallow capsule networks. This observation could be expected because the tweets are based on text with shorter sequences, hence lesser information contained with the text-based tweets. Since the number of the learnable parameters in deep capsule network-based architectures are much higher than shallow capsule networks, short sequences of text prevent proper language modeling with deep capsule architectures compared to shallow capsule architectures. To further validate our proposed architecture, the models were evaluated against the CrowdFlower US Airline dataset.

As shown by the results for the CrowdFlower US Airline dataset in Table 5, the performance of the shallow capsule network with static routing guarantees optimal performance. Therefore, for Twitter-based sentiment analysis tasks, shallow capsule networks could be effectively employed to capture Twitter-specific syntactic and semantic relations for sentiment analysis tasks. Since the existing approaches for the CrowdFlower US Airline dataset do not validate the performance based on 10-fold cross-validation, they are not reported under this experiment. Optimistically, shallow capsule networks could be introduced as a lightweight model compared to BERT-like models, which are more resource-intensive of both linguistic resources and computational power. Therefore, capsule-networks-based models could be used as a replacement for BERT-like models with competitive results for Twitter-based content analysis.

Moreover, to evaluate the model performance with respect to the number of training epochs for the STSGd dataset, a separate experiment was carried out. The dataset was divided into train, validation, and test set based on the 8:1:1 ratio. The performance was evaluated using the accuracy metric and the results are illustrated in Fig. 4. Here, four shallow capsule networks were experimented with based on Twitter-specific GloVe embeddings and GloVe embeddings trained on common crawl corpus. The shallow capsule networks trained on GloVe embeddings with common crawl corpus consistently outperformed the shallow capsule networks trained on Twitter-specific GloVe embeddings. Interestingly, these shallow capsule networks produce the best accuracy within three or four iterations, indicating the effectiveness of the model architecture for low resource consumption in neural language modeling tasks.

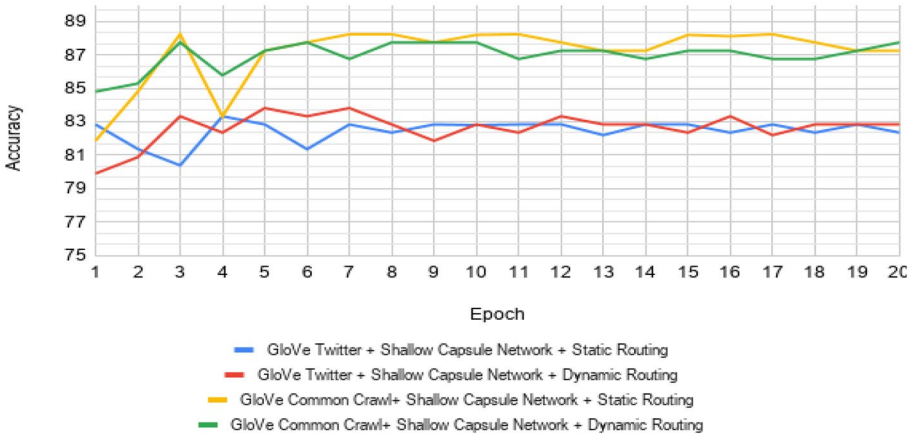

Fig. 4 Epoch wise model results for shallow capsule networks STSGd dataset

Furthermore, to understand the intuition behind the capsule network as a lightweight model compared to existing highly resource-intensive BERT-like language models, the number of total parameters and trainable parameters used for each learning model are shown in Tables 6 and 7, respectively.

Accordingly, compared to the BERT-Base model with 110M parameters and the BERT-Large model with 340M parameters [8], the largest model under this experiment only had 38M total number of parameters with only 14M trainable parameters. Therefore, the capsule networks could be effectively utilized for tasks with low language resources, competitively compared to BERT-like models with less computational resource setups. Furthermore, the tests conducted using BERT pre-training tasks for low resourced languages such as Romanian [9, 23], Arabic [2, 43] and Filipino [6], have shown sub-optimal results compared to the large corpus of English pre-trained data used for BERT-like models with pre-training procedures. Thus, the ideology behind capsule networks could increase the usage of capsule networks in low-resource language domains, where the language resources are not sufficient for the pretraining procedure of BERT-like models.

6 Discussion

6.1 Study contributions and lessons learned

The research findings indicate that capsule networks cab be effectively used for text classification tasks without using any linguistic resources. This would enable the research exploration of text processing and classification under a low resource setup without compromising the accuracy or effectiveness of the task. Therefore, it is evident that the capsule networks provide equal or better results as per the current state-of-the-art. The outcome of this research based on capsule networks under limited resources with fewer parameters and computational power still demonstrates sufficiently competitive results against highly resource-intensive BERT-like models [8]. The increased training and inference costs associated with the transformer models can limit the applicability of BERT-like models for a

Table 6 Number of total parameters in each capsule network based model (in millions)

Dataset/ Embeddings	Shallow capsule network with static routing	Shallow capsule network with dynamic routing	Deep capsule network with dynamic routing	Ensemble capsule network
STSGd dataset + GloVe Twitter	5.6M	5.6M	5.7M	17.1M
STSGd dataset + GloVe	6.5M	6.5M	6.7M	19.9M
CrowdFlower US Airline data + GloVe Twitter	10M	10M	10.1M	30.3M
CrowdFlower US Airline data + GloVe	12.7M	12.7M	12.8M	38M

Table 7 Number of trainable parameters in each capsule network based model (in millions)

Dataset/ Embeddings	Shallow capsule network with static routing	Shallow capsule network with dynamic routing	Deep capsule network with dynamic routing	Ensemble capsule network
STSGd dataset + GloVe Twitter	3.9M	3.9M	4M	12M
STSGd dataset + GloVe	4M	4M	4.1M	12.3M
CrowdFlower US Airline data + GloVe Twitter	4.6M	4.6M	4.7M	14M
CrowdFlower US Airline data + GloVe	4.7M	4.7M	4.8M	14M

Table 8 Accuracy comparisons with the related studies

Study	Twitter Dataset	Dataset size	Used techniques	Accuracy
[42]	OMD	1250	Quantum Language Model (QLM) and Density matrix	62.9%
	SSTd	2113		61.8%
[19]	STSGd	2034	CNN without word embedding	68%
[13]	OMD	1250	LSTM	78.06%
	Sentiment140	3061	CNN	71.85%
[33]	Sentiment140	1600000	Weighted ensemble classifier using Naïve Bayes (NB), Random forest (RF), Support vector machine (SVM), Logistic regression (LR).	75.81%
	Health Care Reform (HCR)	888		73.68%
	First GOP debate	10729		85.83%
	Twitter sentiment analysis	99989		74.67%
[5]	Sentiment140	1600000	Deep CNN + BerT	80.7%
[10]	Sentiment140	1600000	LSTM + FastText	82.4%
	Emotional Tweets	21051	LSTM + GloVe	81.9%
[11]	STSTd	359	Deep CNN	87.62%
	SE2014	5892		85.82%
	STSGd	2034		85.97%
	SED	2648		87.39%
	SSTd	3326		81.36%
Proposed study	STSGd	2034	Capsule network + GloVe	86.87%
	CrowdFlower US Airline	14640		82.04%

given text analysis task. Although BERT-like models give more context, their processing capabilities gets compromised in situations where these models are difficult to apply.

According to the obtained results, our proposed capsule network-based approach is reasonably accurate and contextually rich at comparable levels, though it does not require more resources due to the lightweight architecture with fewer parameters. This is useful for processing tasks that require model re-training with shorter lead times to release new inference, such as edge or real-time sentiment extraction. The pre-trained GloVe word-vectors based on a Twitter-specific corpus, which contains 27 billion tokens and a vocabulary of 1.2 million tokens, were tested with our proposed architectures. Although Twitter-specific word-vectors could capture syntactic and semantic relationships in tweets by considering the context of the domain, the models trained with more generic GloVe word-vectors from common crawl corpus outperformed the models with Twitter-specific pre-trained word-embeddings. This is because of the deep feature extraction of generic GloVe word-vectors, which were trained with the largest common crawl corpus of 840 billion tokens and vocabulary of 2.2 million tokens. Therefore, the GloVe word-vectors trained on common crawl corpus could be effectively used to identify sentiment categories within tweets. The variability of the tokens within the Twitter data could be effectively managed with generic GloVe embeddings since the common crawl corpus includes data from most of the domains within the Twitter-based textual representations.

Moreover, it is possible to have numerous variations of models as shallow capsules with static and dynamic routing methods. Also, the use of deep capsule networks with dynamic routing and ensemble capsule networks could be recommended for better accuracy in the Twitter data processing. Shallow capsule networks with static routing produced promising results for the datasets used in this research. The effectiveness of shallow capsule networks could be described as the ability to capture syntactic and semantic relationships of tweets as short sequences of text. The static routing algorithm elevates child-to-parent relationships in a specific way for text-based classification tasks. It handles background information of text quite effectively, preventing the drawbacks caused by the background noise of text.

6.2 Comparison with the existing studies

Table 8 shows a comparison of the proposed solution with the existing studies in terms of the used datasets, techniques, and the obtained accuracies. The existing studies are based on several Twitter databases such as Obama-McCain Debate (OMD), Sentiment Strength Twitter Dataset (SS-Tweet), Stanford Twitter Sentiment Test (STSTd), SemEval2014 Task9 (SE2014), Stanford Twitter Sentiment Gold (STSGd), Sentiment Evaluation (SED), Sentiment Strength Twitter (SSTd) and STS-Gold Twitter dataset. Accordingly, the proposed approach has shown the highest accuracy of 86.87% for the STSGd dataset using 300-dimensional common crawl Glove word-embeddings and shallow capsule network with dynamic routing. The highest accuracy of 82.04% was reported utilizing 300-dimensional common crawl Glove word-embeddings and shallow capsule network with static routing for the CrowdFlower US Airline dataset.

The novel contribution and usefulness of the proposed approach compared to the existing studies based on capsule networks can be highlighted. The capsule-based architectures could be effectively used as a replacement for CNN-based deep learning architectures due to the vector representations of features instead of scalar feature representation of CNNs. The vector representation of features in capsules effectively handles the background

information of the text. Moreover, highly resource-intensive models like BERT could be replaced with capsule-based techniques, since capsule architectures could produce competitive results in low resource domains for BERT models as suggested in Sect. 5.

6.3 Open challenges and future research directions

It is challenging to process short sequences of social media text content with varying context and background information. Static routing could be more effective over dynamic routing algorithms for short sequences when handling the variability of background information. Pre-processing of Tweets can be applied to improve the model's performance due to the noise of special characters and web URLs, which do not carry any sentiment information within a tweet. A possible future research extension would be to explore the use of Attention-based capsule networks with dynamic routing for relation extraction as part of sentiment analysis and text content processing with social media data. Moreover, contextual embeddings could be integrated with capsule-based techniques, since most of the deep learning techniques have reported promising performances utilizing this strategy.

7 Conclusion

This research explored the use of capsule networks in social media text content analysis with natural language processing. The proposed strategy aimed at sentiment analysis of Twitter-based data utilizing a variety of capsule networks. Twitter-specific and generic GloVe embeddings were used in shallow and deep capsule networks together with static and dynamic routing for sentiment analysis of tweets. A notable achievement in this research is the higher level of accuracy over the existing sentiment analysis methods used in social media content, thereby setting a new benchmark standard for Twitter data analysis with capsule networks. The classification results support the use of shallow capsule networks with static routing for optimal performance. Moreover, it produced state-of-the-art results considering the relatively shorter sequences of texts in tweets. For the CrowdFlower US Airline dataset, the shallow capsule network with static routing produced an optimal accuracy of 82.04%, while the highest accuracy of 86.67% for the Stanford Twitter Sentiment Gold dataset was reported by shallow capsule networks with dynamic routing.

Furthermore, considering the lightweight nature of the capsule networks, they are useful for low resource languages where the BERT-like models could not be utilized due to a lack of language resources for pre-training procedures. Thus, we have proven a novel methodology to analyze social media text content in resource-constrained setups such as edge processing, where the capsule networks of the analysis model can be deployed. This will revolutionize social media content analysis as the proposed capsule network-based distributed processing architecture can easily rely upon portable devices and nodes, which can open the pathway to real-time sentiment analysis at the edge of the processing channel. This study concludes that the introduction of capsule networks into state-of-the-art text processing and natural language processing methods has shown impressive performance and potential in the research area of Twitter sentiment analysis.

Funding The research is not funded.

Availability of data and material Public dataset

Declarations

Conflicts of interest There are no conflicts of interests

References

1. Abeysinghe C, Perera I, Meedeniya, DA (2021) Capsule networks for character recognition in low resource languages. In Machine Vision Inspection Systems, Volume 2: Machine Learning-Based Approaches, Malarvel, M., Nayak, S. R., Pattnaik, P. K., & Panda, S. N. (Eds.), Ch.2, 23–46, John Wiley & Sons Inc
2. Abuzayed A, Al-Khalifa H (2021) Sarcasm and sentiment detection in Arabic tweets using BERT-based models and data augmentation. In Proceedings of the 6th Arabic Natural Language Processing Workshop 312–317
3. Araque O, Corcuera-Platas I, Sánchez-Rada JF, Iglesias CA (2017) Enhancing deep learning sentiment analysis with ensemble techniques in social applications. Expert Syst Appl 77:236–246
4. Baecchi C, Uricchio T, Bertini M, Del Bimbo A (2016) A multimodal feature learning approach for sentiment analysis of social network multimedia. Multimed Tools and Appl 75(5):2507–2525
5. Cai M (2018) Sentiment analysis of tweets using deep neural architectures. In Proceedings of the 32nd Conference on Neural Information Processing Systems 1–8
6. Cruz JCB, Cheng C (2020) Establishing baselines for text classification in low-resource languages. arXiv preprint arXiv:2005.02068
7. Das S, Behera RK, Rath SK (2018) Real-time sentiment analysis of Twitter streaming data for stock prediction. Proc Comput Sci 132:956–964
8. Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 1:4171–4186
9. Dumitrescu SD, Avram AM, Pyysalo S (2020) The birth of Romanian BERT. arXiv preprint arXiv:2009.08712
10. Imran AS, Daudpota SM, Kastrati Z, Batra R (2020) Cross-cultural polarity and emotion detection using sentiment analysis and deep learning on COVID-19 related tweets. IEEE Access 8:181074–181090
11. Jianqiang Z, Xiaolin G, Xuejun Z (2018) Deep convolution neural networks for Twitter sentiment analysis. IEEE Access 6:23253–23260
12. Johnson R, Zhang T (2015) Effective use of word order for text categorization with convolutional neural networks. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 103–112
13. Kag A, Zhang Z, Saligrama V (2019) RNNs incrementally evolving on an equilibrium manifold: A panacea for vanishing and exploding gradients?. In Proceedings of the International Conference on Learning Representations 1–24
14. Kim J, Jang S, Park E, Choi S (2020) Text classification using capsules. Neurocomputing 376:214–221
15. Koehn P, Knowles R (2017) Six challenges for neural machine translation. In Proceedings of the First Workshop on Neural Machine Translation 28–39
16. Kumar A, Garg G (2019) Sentiment analysis of multimodal Twitter data. Multimed Tools Appl 78(17):24103–24119
17. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R (2019) ALBERT: A lite BERT for self-supervised learning of language representations. In International Conference on Learning Representations 1–17
18. Lenadora DS, Gamage GSW, Haputhanthri HDI, Meedeniya D, Perera I (2020) Exploratory Analysis of a cocial media network in Sri Lanka during the COVID-19 virus outbreak. arXiv preprint arXiv:2006.07855
19. Liao S, Wang J, Yu R, Sato K, Cheng Z (2017) CNN for situations understanding based on sentiment analysis of Twitter data. Proc Comput Sci 111:376–381
20. Liu B (2010) Sentiment analysis and subjectivity. Handb Nat Lang Process 2:627–666
21. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2020) Roberta: A robustly optimized BERT pretraining approach. In Proceedings of the 28th International Conference on Computational Linguistics 6626–6637

22. Manning CD, Surdeanu M, Bauer J, Finkel JR, Bethard S, McClosky D (2014) The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations 55–60
23. Masala M, Ruseti S, Dascalu M (2020) RoBERT–A Romanian BERT model. In Proceedings of the 28th International Conference on Computational Linguistics 6626–6637
24. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in neural information processing systems 3111–3119
25. Mikolov T, Grave É, Bojanowski P, Puhrsch C, Joulin A (2018) Advances in pre-training distributed word representations. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation 1–4
26. Naseem U, Razzak I, Musial K, Imran M (2020) Transformer based deep intelligent contextual embedding for Twitter sentiment analysis. Future Gener Comput Syst 113:58–69
27. Nazir F, Ghazanfar MA, Maqsood M, Aadil F, Rho S, Mehmood I (2019) Social media signal detection using tweets volume, hashtag, and sentiment analysis. Multimed Tools Appl 78(3):3553–3586
28. Patrick MK, Adekoya AF, Mighty AA, Edward BY (2019) Capsule networks–a survey. J King Saud Univ Comput Inf Sci 1–16
29. Pires T, Schlinger E, Garrette D (2019) How multilingual is multilingual BERT?. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics 4996–5001
30. Rajasegaran J, Jayasundara V, Jayasekara S, Jayasekara H, Seneviratne S, Rodrigo R (2019) Deepcaps: Going deeper with capsule networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 10725–10733
31. Sabour S, Frosst N, Hinton GE (2017) Dynamic routing between capsules. In Proceedings of the 31st Conference on Neural Information Processing Systems 3856–3866
32. Saif H, Fernandez M, He Y, Alani H (2013) Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset, the STS-Gold. In Proceedings of the 1st International Workshop on Emotion and Sentiment in Social and Expressive Media: Approaches and Perspectives from AI 1–13
33. Saleena N (2018) An ensemble classification system for Twitter sentiment analysis. Proc Comput Sci 132:937–946
34. Sanh V, Debut L, Chaumond J, Wolf T (2019) DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108
35. Wang X, Jiang W, Luo Z (2016) Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In Proceedings of the 26th International Conference on Computational Linguistics 2428–2437
36. Wang Y, Sun A, Han J, Liu Y, Zhu X (2018) Sentiment analysis by capsules. In Proceedings of the 2018 World Wide Web Conference 1165–1174
37. Yang M, Zhao W, Ye J, Lei Z, Zhao Z, Zhang S (2018) Investigating capsule networks with dynamic routing for text classification. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing 3110–3119
38. Zhang Y, Song D, Zhang P, Li X, Wang P (2019) A quantum-inspired sentiment representation model for Twitter sentiment analysis. Appl Intell 49(8):3093–3108
39. Zhang Y, Wallace BC (2017) A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In Proceedings of the Eighth International Joint Conference on Natural Language Processing 253–263
40. Zhang L, Wang S, Liu B (2018a) Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8(4):e1253
41. Zhang Y, Liu Q, Song L (2018b) Sentence-state LSTM for text representation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics 1:317–327
42. Zhang Y, Song D, Li X, Zhang P (2018c) Unsupervised sentiment analysis of Twitter posts using density matrix representation. In Proceedings of the European Conference on Information Retrieval 316–329
43. Zhang C, Abdul-Mageed M (2019) BERT-based Arabic social media author profiling. arXiv preprint arXiv:1909.04181