Contextual Semantics using Hierarchical Attention Network for Sentiment Classification in Social Internet-of-Things

Akshi Kumar
Department of Computer Science & Engineering, Delhi Technological University, New Delhi, India

Abstract: Social networks can permeate intelligence to aid autonomous decision making by enhancing the service needs and communication among object peers in a Social Internet of Things (SIoT). But the user-generated data has multiple layers of meaning which necessitate AI-driven solutions such as sentiment analysis to handle the dynamics. As people express opinions in complex ways and use rhetorical devices like sarcasm, irony, and implication etc., considering only the lexical content can be misleading. Moreover, intra-textual and sub-sentential reversals, topic drift, negation can further misrepresent sentiment, fostering the need to recognize and incorporate contextual semantics for increasing the sentiment classification accuracy. This research evaluates the use of hierarchical attention network (HAN) to classify sentiments in real-time Twitter data. HAN allows differential contribution of various parts of tweet (tweet-sentence-word) to its essential meaning as it introduces two attentive mechanisms and the context-dependent importance of the parts of tweet are considered when constructing the representation of the document. The model is evaluated on two benchmark datasets and compares favorably to state-of-the-art approaches giving an effective solution to tweet-level analysis of sentiments in SIoT.

Keywords: Artificial intelligence, Sentiment analysis, Social IoT, Deep learning, Social media

1. Introduction

Social networks define a ubiquitous complex system which comprises various mathematical models and theories to study its behaviour, and to explain and predict its dynamics. These networks are sometimes called "relationship networks," as these help people and organizations connect online to share information and ideas. Concurrently, the pervasiveness of Internet of Things (IoT) devices make them the focal point of the consumer internet economy, as the range of connected devices is now greater than ever. Indeed, in the recent years, there is a considerable upsurge in the use of mobile services, sensors, and application data to better understand user behaviour and to gain tangible benefits from the IoT. Social IoT (SIoT) has emerged as a new paradigm to leverage social users' behaviors or opinions for enhancing the functionality of the IoT devices [1]. Built on the unison of social networks and IoT, SIoT creates social networks in which things are nodes that interact and establish social links (relationships) with each other similar to the way humans do to achieve a common goal. Based on the fact that connected things exceptionally affects the way we relate and connect with others, SIoT converts the 'smart' things to 'social' things. The advantages of utilizing the social relationships within the IoT include simplification in the navigability of a dynamic network with billions of objects, robustness in the management of the trustworthiness of objects when providing information and services and efficiency in the dynamic discovery of services and information [2]. Certainly, the social network characteristics offer the indispensable information for human's activities and behaviours, which can be well utilized by the SIoT networks to enhance functionality, usability, consistency and connectivity of IoT networks. Further, it can permeate intelligence and contextual semantics to aid autonomous decision making, enhance service needs and communication among object peers. Fig.1 depicts the SIoT framework.

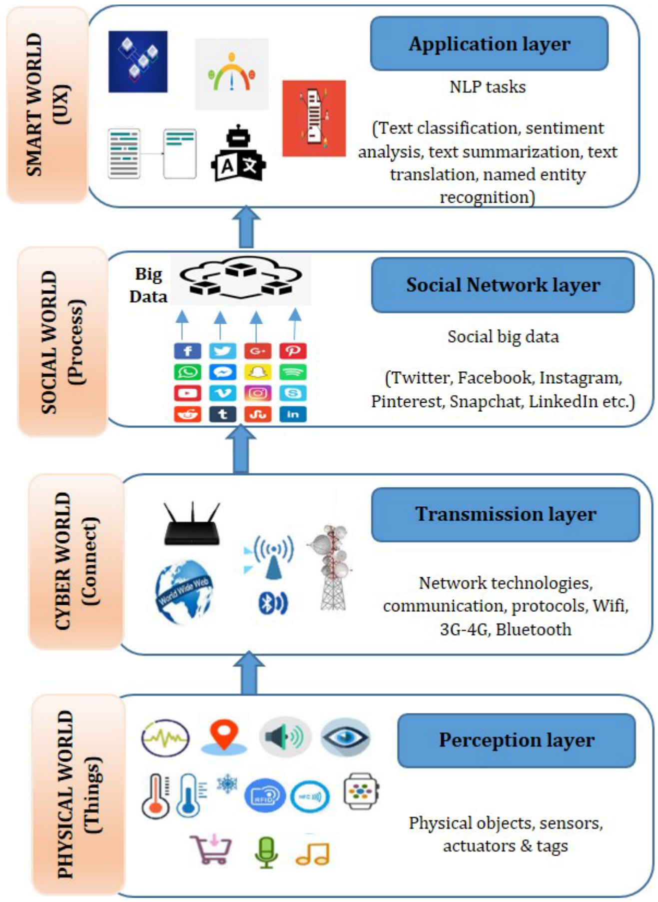

Fig.1. SIoT framework.

In SIoT, social networks have a massive data input, and a significant means to manage, process and implement that data. While massive textual, aural, visual, temporal and spatial information is generated on social media, it's calculable that the bulk of it comes from unstructured sources such as posts, likes, tweets, views, comments, favourites, etc. [3]. Moreover, the data now flows not only from the social network users to and from connected things (sensors, wearables etc.) but also from user to user. The users' act as human sensors and the intelligence framework evolves to a cyber-physical-social system. Thus, it is imperative to analyze this huge amount of data with diagnostic, predictive and prescriptive analytics tools which observe, learn and offer insights, suggestions and even automated actions.

The analysis of human data is becoming pervasive in technology solutions. Understanding the more subtle meaning of what someone is saying, can steer towards valuable user-related services. Natural language understanding and artificial intelligence (AI) emerge as the most appropriate computational paradigms to provide embedded intelligence and have been effectively utilized in innumerable domains such as text classification, question-answering systems and named entity recognition amongst others. These facilitate contextual understanding and allow personalization of products and services for customers. On the social networks, sentiment analysis (SA) has proven valuable in user-related services as a pragmatic tool for public opinion extraction and analysis in SIoT. SA addresses the dynamics of complex socio-affective applications by automatically ascertaining the polarity of opinion expressed by user towards entities in a chunk of text or review. It offers valuable information to users or smart devices to make decisions and gain better value for user-related services within the SoIT framework. Therefore, mining this invaluable and opinion-rich information from social networks can simulate collective intelligence processing of social behaviour data that can serve SIoT, thus making the value chain more efficient.

Deep learning models have achieved state-of-the-art results when applied to various natural language processing (NLP) tasks with their power to adapt to new problems and perform automatic feature engineering [4]. Convolutional neural networks (CNNs), conventionally modelled for a variety of computer vision tasks such as image recognition and object classification, have been successfully adapted for NLP tasks [5]. Recurrent Neural Networks (RNNs) have also proven useful in NLP tasks to process sequential data (time-stamped) where a word's meaning in the sentence significantly relies on the context [6]. Further, both LSTM (Long Short Term Memory) and GRU (Gate Recurrent Unit) have been introduced with the goal of tracking long-term dependencies effectively while handling the vanishing/exploding gradient problems of Vanilla RNN. A typical extension of LSTMs, the Bi-directional LSTMs (Bi-LSTMs) have also been applied to sequence classification problems to improve the accuracy of models. Attention-based deep neural architectures facilitate sequential reasoning by processing subsets of the input in a sequential manner. Attention mechanisms can considerably improve the performance of RNNs. Yang et al. [7] developed a hierarchical attention network (HAN) for document classification that outperformed both RNNs and CNNs. HAN attempts to classify a document based on the knowledge it can infer about the document from its constituent parts, that is, the sentences and words that make up the document and includes an attention mechanism that is able to find the most important words and sentences in a document while taking the context into consideration. The 'hierarchical' in HAN comes from the design that this knowledge is built hierarchically, starting from using the words in a sentence and followed by using the sentences in a document. In this paper, we utilize a similar architecture to classify sentiments in real-time data.

Polarity detection is not always straightforward as opinion words change polarity depending on the context of use. For example, the sentence *"I am happy"* is positive and *"I am not happy"* is negative whereas *"It's never the case that I am not happy"* is again positive. Quite clearly, the word *'happy'* shifts polarities and requires contextual semantics to determine the correct polarity. Moreover, the current admissible post length on Twitter is 280 characters which allows multiple sentences to be included in a single tweet or long story expressed as tweet threads (multi-part tweet). Using HAN for analyzing sentiments in tweet (multi-sentence as well as multi-part) tries to find a solution for the following problems:

- Not every word in a sentence and every sentence in a tweet are equally important to understand the main message of a tweet.
- The changing meaning of a word depending on the context needs to be taken into consideration. For example, the meaning of the word *"pretty"* can change depending on the way it is used: *"She is wearing a pretty dress"* vs. *"The weather is pretty bad".*

Hence to handle these complexities in language, HAN with deep contextualized language representations from ELMo [8] is used. ELMo analyses words within the context that they are used. Also, as it is character-based, it allows the model to form representations of out-of-vocabulary words. Using HAN allows differential contribution of various parts of tweet (tweet-sentence-word) to its essential meaning by introducing two attentive mechanisms and the context-dependent importance of these parts is considered when constructing the representation of the document. The model compares favourably to state-of-the-art approaches and gives an effective solution to tweet-level analysis of sentiments in SIoT.

The organization of paper is as follows: The next section briefs about the related work within the domain of sentiment analysis in SIoT followed by the description of the HAN model in section 3. Section 4 presents the details of model performance and conclusion is given in section 5.

2. Preliminaries

The Scherer's typology of affective states [9] postulates a generic framework to cognize sentiments. The typology characterizes the ubiquitous and constant part of the human experience and categorizes the affective states into five types based on coherence and time course. Figure 2 outlines the five types of affective states as defined by Scherer et al.

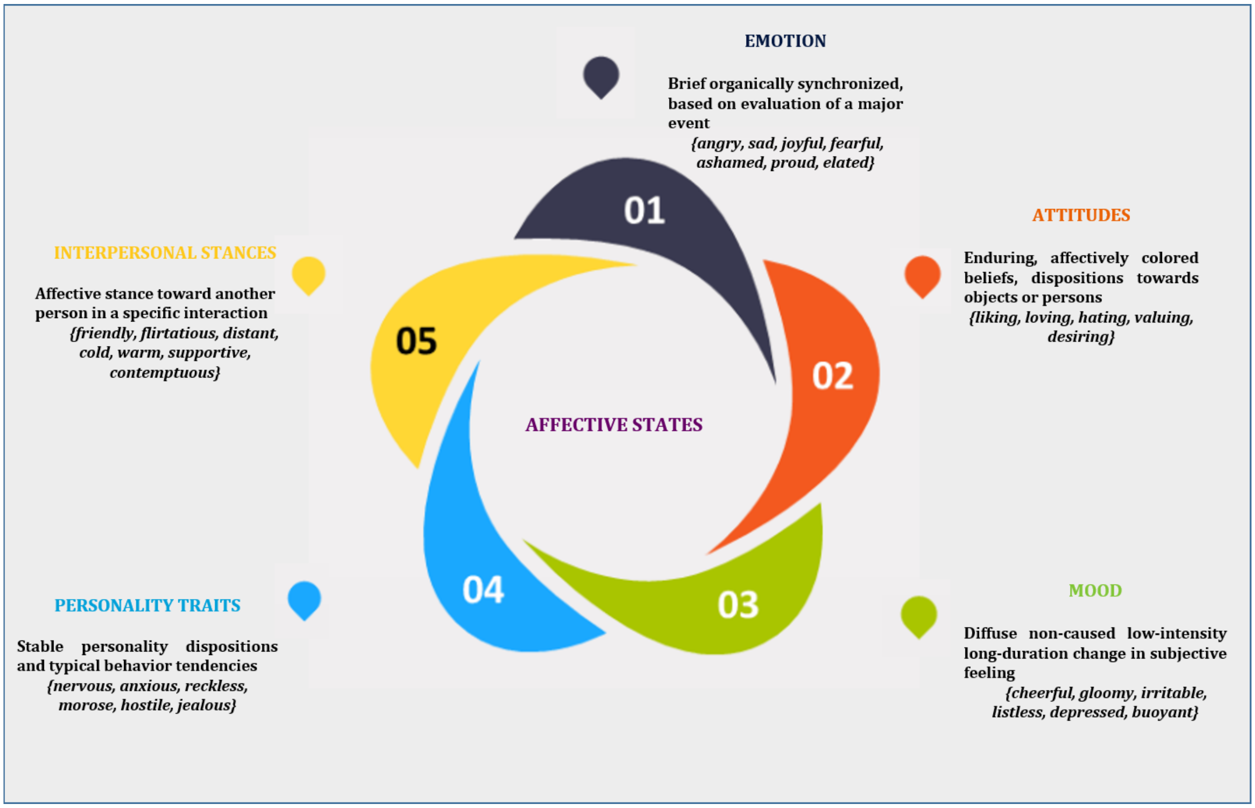

Fig. 2. Scherer's typology of affective states

Sentiment analysis is the detection of attitudes, defined using a 4-tuple {Holder, Target, Type, Text}, where the holder is the source of attitude, target is the aspect of attitude, type defines the commonly weighted polarity type such as positive, negative or neutral and text is the piece of information containing the attitude (sentence or entire document). In recent times the 'text' component has compounded as multimedia text owing to the extensive use of various semiotic modalities (aural, visual, and textual) on social media. The task of SA can range from a simplistic classification of text attitude as positive or negative to a more advanced fine-grain emotion analysis and complex attitude type detection.

One of the early attempts to analyze sentiments on Twitter made use of lexicon-based approaches such as SentiWordNet [10]. These approaches are further categorized as dictionary-based and corpus-based [11]. Classical ML-based approaches such as Naïve bayes, maximum entropy classifier, Markov blanket classifier and support vector machine etc. have also been extensively used [12-14]. All these approaches require structured data and learning from labeled features. Good features are considered as the backbone for any learning model, and good feature creation often needs adequate domain knowledge, creativity and time. As a result, new computational methodologies such as swarm-based algorithms have also been used for finding optimal feature set for improving the performance of the sentiment classifier in terms of predictive accuracy and result comprehensibility

[15-16]. But classical ML-based approaches are often plagued with inaccuracies for language processing since languages do not have a fixed set of rules. They are dynamic and a host of interdependent factors like usage, context, constructs, semantics, discourse etc. play a role in determining the underlying meaning of text. Deep learning (DL) models have emerged as strategic player in understanding the nuanced interpretations of language text and have been used for sentiment classification. Pertinent studies report the used of various DL models such as convolutional neural networks (CNN), long short term memory (LSTM), Bi-directional LSTM (Bi-LSTM) with attention etc. The models have been evaluated on various datasets such as IMDb movie review dataset, Yelp restaurant review dataset, Stanford sentiment treebank (SSTb), Amazon review dataset, SemEval-2013, 2014, 2016, 2017 datasets, Sentiment140 and Stanford Twitter sentiment-gold (STS-G) dataset. A detailed review has been given by Dang et al. [17]

In this work, the proposed model has been evaluated on SemEval-2017 and STS-Gold datasets, therefore the related work on these two is presented. The best performing system on the SemEval 2017 task dataset is the one described by Nguyen et al. [18] in 2020. The authors proposed a BERTweet model which achieved a macro-average recall of 73.2. There were two winners [19, 20] of the SemEval 2017 Task 4A with a macro-average recall of 68.1. Cliché [19] used CNNs and LSTMs network to produce a state-of-the-art Twitter sentiment classifier whereas Baziotis et al. [20] used deep LSTM with two kinds of attention mechanism.

In 2014, Saif et al. [21] had put forward the SentiCircle on the benchmark STS-gold Dataset and reported an accuracy of 80.3. Various hybrid models (Ml +lexicon based) have been proposed by using SentiCircle (SC) approach with SentiWordNet (SWN) [10], MPQA [22] and Thelwall lexicon [23] using median method and the accuracies achieved are about 69%, 76%, and 79% respectively. SentiStrength (SS) put forward by Thelwall et al. [24] reported an accuracy of 82.1.In 2017, Jin et al. [25] proposed a proposed a deep belief network with delta rule based on restricted Boltzmann machines (RBM) for sentence- level classification English Text and evaluated the model on STS-G dataset. The model reported an accuracy of 83.55 using 1-hidden layer, 83.05 using 2-hidden layers and 83.10 using 3-hidden layers. In 2018, Jianqiang et al. [26] used Deep CNN and achieved accuracy of 86. Kumar et al. [3] proposed a textual sentiment analytics model which combined deep CNN with SentiCircles and achieved an improved accuracy of 88%. Islam et al. [27] reported a multi-channel CNN model to investigate sentiment analysis on standard datasets. The MC-CNN model reported an accuracy of 90.6%

3. Model Description

To learn the text-based contextual features HAN classifier is used. The HAN classification model was put forward by Yang et al. [7] in 2017 for the purpose of classification of documents. HAN attempts to classify a document based on the knowledge it can infer about the document from its constituent parts, that is, the sentences and words that make up the document. It is based on the notion that a document has a hierarchical structure, i.e., words form various sentences, and then those different sentences form the complete document. It takes this hierarchy into consideration while formulating the vector representation of the document. In order to accurately encapsulate the unerring thought of the document, the HAN model employs two distinct levels of attention mechanisms, one at the word level, and the other at the sentence level. Different parts of a document contribute distinctively towards its true meaning. These unique contributions of the several disparate parts of the document along with the context-dependent significance of these parts are taken into account while formulating the vector representation of the document. The HAN model constitutes of an embedding layer, attention layers and encoders which together aids the model in comprehending the textual features. The encoders are responsible for the extraction of relevant context. The attention layers evaluate how relevant a sequence of tokens is with reference to the document. The framework of HAN is made up of five distinct layers, namely, the embedding layer, word sequence encoder, word-level attention layer, sentence encoder and sentence-level attention layer. The next generation ELMo (Embeddings from Language Models) word embedding [8] is used as the word vector learning technique to provide the classifier with the learned representations of the text. The word vectors generated by the embedding layer are fed to the word encoder which generates the annotations of those words. These annotations are then filtered by the attention mechanism based on their contribution to the essential meaning of the sentence. This whole process is repeated once again at the sentence level where the inputs are sentence vectors. Thus, the HAN classifier tackles each document at the word level and then again at the sentence level, finally forming a document vector. This document vector is then passed through a softmax activation function to generate the output. Figure 3 depicts the HAN architecture used for sentiment classification.

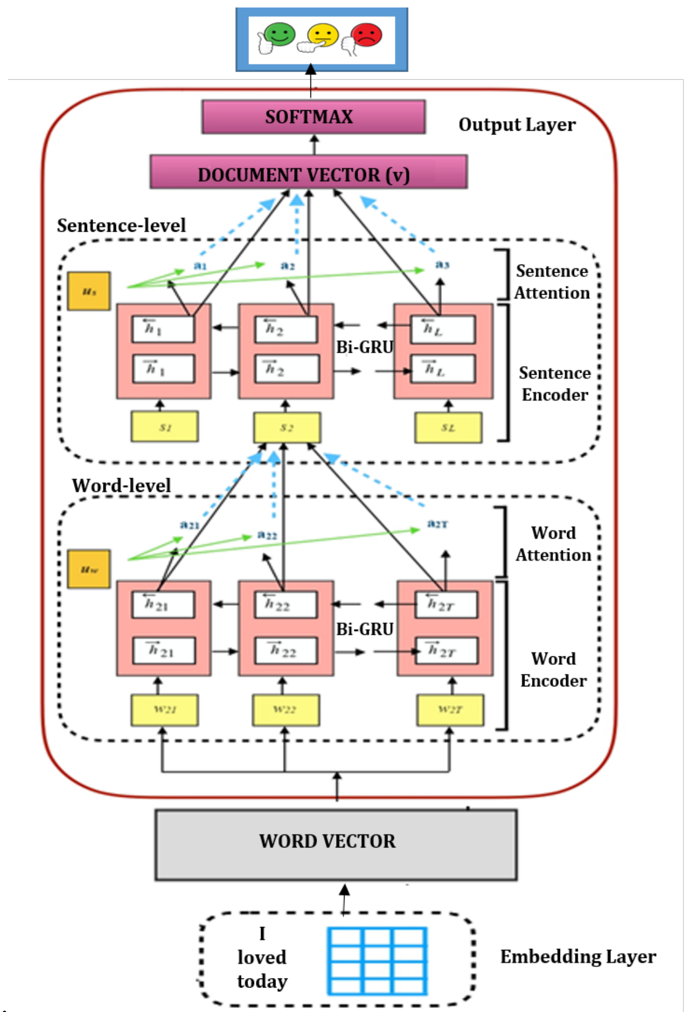

Fig.3. HAN Architecture

3.1. Embedding Layer

Word embeddings extract meaning from the text while preserving the contextual similarity of the words meaning that words with similar context end up having similar vector representations. It essentially provides us with a mapping of the learned representations of all the words in a document. In the text that is input, each word is encoded into an integer. Therefore, a unique integer represents each word of the input text. The embedding layer is initialized with random weights, and it learns the embedding for all of the words present in the training set. In this work, the ELMo 5.5B word embeddings model [8] is used to produce the vector representation for the words. ELMo was preferred over the conventional models such as Word2Vec, fastText or GloVe, because ELMo provided with contextualized word representations, which essentially means that the vector representation generated by ELMo for each word, entirely depends on the context in which the word has been used. It is possible for the exact same word to have two distinct vector representations depending on the context in which it is used. ELMo produces vectors on-the-spot by feeding words into the deep learning model instead of having a dictionary of words and their matching vectors, as is the case with conventional word embedding models. Additionally, ELMo representations are purely character-based, which permits our model to generate representations for words that are absent from the training set.

3.2. Encoder

A GRU based sequence encoder is used in HAN. The GRU uses a gating mechanism without employing any separate cells for memory to track the sequences of the state. It comprises of the following two gates: the update gate z_t and the reset gate r_t. The reset gate regulates the amount of contribution provided by the previous state to the current state. The update state z_t defines the extent of the past information to be added and also of the new information to be added to the current state. Both, the reset gate r_t and the update gate z_t jointly control the updating of information in the current state.

At any time, t, the new state h_t is calculated as given in (1):

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \qquad (1)$$

where h_{t-1} stands for the previous state, and $\tilde{h}$ stands for the current state.

The update state z_t is computed as given in (2)

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \qquad (2)$$

where x_t stands for the sequence vector at any time t.

The candidate state $\tilde{h}_t$ is computed as given in (3):

$$\tilde{h}_t = \tanh(W_h x_t + r_t \odot (U_h h_{t-1}) + b_h), \qquad (3)$$

where r_t stands for the reset gate.

The reset gate is computed as using (4)

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \qquad (4)$$

- **Word Encoder**

  A mapping of discrete variables to a vector of continuous numbers is termed as an embedding matrix. Neural network embeddings prove to be utilitarian as they are able to lower the dimensionality of categorical variables and pertinently represent categories in the transformed space. Provided a sentence with words w_{it}, $t \in [0, T]$, it starts by embedding the words to vectors with the help of an embedding matrix W_e, $x_{it} = W_e w_{it}$

  The information is summarized from both the directions, namely, forward and backward, using a bidirectional GRU in order to obtain annotations of the words. The forward GRU $\vec{f}$ reads the sentence s_i from w_{i1} to w_{iT} as given in (5).

$$\overrightarrow{h}_{it} = \overrightarrow{GRU}(x_{it}), t \in [1,T], \quad (5)$$

The backward GRU $\overleftarrow{f}$ reads the sentence s_i from w_{iT} to W_{i1}. The concatenation of the backward hidden state and the forward hidden state helps us in acquiring the annotation of the word $w_{it,}$ i.e., $h_{it} = [\overrightarrow{h}_{it}, \overleftarrow{h}_{it}]$ as given in (6).

$$\overleftarrow{h}_{it} = \overleftarrow{GRU}(x_{it}), t \in [T,1]. \quad (6)$$

- **Sentence Encoder**
  Analogous to a word encoder, the sentence encoder is utilized to obtain the document vector from the given sentence vectors s_i. A bidirectional GRU is employed for encoding the sentence as given in (7) and (8):

$$\overrightarrow{h}_i = \overrightarrow{GRU}(s_i), i \in [1,L], \quad (7)$$

$$\overleftarrow{h}_i = \overleftarrow{GRU}(s_i), t \in [L,1], \quad (8)$$

To get the annotation of a sentence *i* both the backward hidden state h_j and the forward hidden state h_i, are concatenated, i.e., $h_i = [\overrightarrow{h}_i, \overleftarrow{h}_i]$.

3.3. Attention Layer

The hierarchical attention layers are used for the document level classification. Suppose the document has L sentences $s_{i,}$ and each sentence s_i contains T_i words. The notion of 'attention' is based on the fact the words in a sentence do not contribute equally to its meaning. Similarly, not all sentences contribute equally towards the overall meaning of the document (Fig.4).

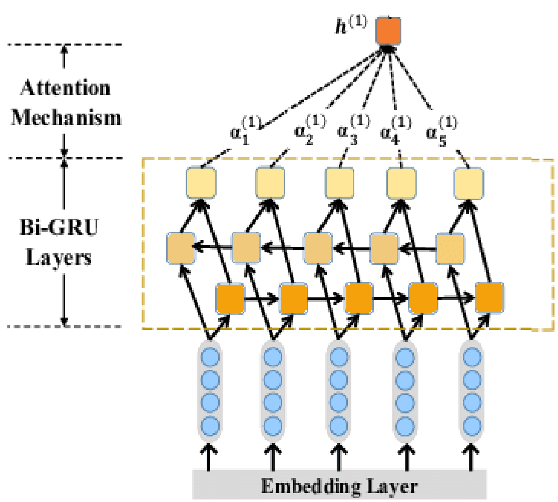

Fig.4. GRU-based attention mechanism

The word-level and sentence-level attention mechanism are described next.

- **Word Attention Layer**

In trying to understand the meaning of the sentences, it becomes clear that all the words present in a sentence do not contribute equally to its meaning. Therefore, to find out those words which are key to the meaning of the sentences, the word attention layer is used. The extracted words, then, form the sentence vector. To get the hidden representation of the h_{it}, u_{it} is calculated using a one-layer MLP using (9).

$$u_{it} = \tanh(W_w h_{it} + b_w) \quad (9)$$

The normalized importance a_{it} is then generated through a sigmoid function to calculate the significance of the word as a similitude of u_{it} with a word-level context vector u_w using (10).

$$a_{it} = \frac{\exp(u_{it}^T u_w)}{\Sigma_t \exp(u_{it}^T u_w)} \quad (10)$$

After that, the weighted sum of the word annotations based on their weights is computed which corresponds to the sentence vector s_i as given in (11).

$$s_i = \sum_t a_{it} h_{it}. \quad (12)$$

- **Sentence Attention Layer**

Since all the sentences in the document are not important to understand the meaning of the document, it is necessary to extract such sentences that are of more importance compared to the others. To get the hidden representation of h_i, u_i is calculated using a one-layer MLP as given in (12).

$$u_i = \tanh(W_s h_i + b_s) \quad (12)$$

The normalized importance a_i is then generated through a sigmoid function to calculate the significance of the sentence as a similitude of u_i with a sentence-level context vector u_s using (13).

$$a_i = \frac{\exp(u_i^T u_s)}{\Sigma_i \exp(u_i^T u_s)} \quad (13)$$

After that, the document vector $v_{i,}$ is computed, which is the weighted sum of the sentence annotations depending on their weights. The summarization of all the information gathered from all the sentences of a document is present in the document vector $v_{i.}$ as given in (14).

$$v_i = \sum_i a_i h_i. \quad (14)$$

3.4. Document Classification

As this research analyses sentiment in tweets and tweets can be a single sentence, multiple sentence or multi-part, we consider tweet as the document. Therefore, to finally classify the document into one of the three classes is done using the document vector v. The document vector v passes through the last layer, i.e., the output layer, which employs a softmax function which outputs the probability distribution over all classes.

4. Model Performance

4.1. Datasets & Experimental set-up

Two datasets were used to evaluate the performance of HAN. These were the Stanford Twitter Sentiment Gold Standard (STS-Gold) [21, 28] and the SemEval-2017 Task 4A datasets [29]. The SemEval-2017 dataset consist of English tweets annotated for three sentiment classes, namely positive, negative or neutral categories. The STS-Gold dataset consists of English tweets annotated for five sentiment classes, namely positive, negative, neutral, mixed or others. The 'mixed' and 'other' class ignored to ensure the same number of classes in both the dataset. The dataset statistics are given in table 1.

Table 1. Dataset Statistics

Datasets	Positive	Negative	Neutral	Mixed	Other	Total
SemEval-2017 Task 4A [29]	2352	3811	5742	Φ	Φ	11905
STS-Gold [21]	632	1402	77	90	4	2205

* Only Tweet IDs provided by organizers and therefore some tweets were not available for download due to edited privacy or removed.

Figure 5 depicts the category percentage for both the datasets.

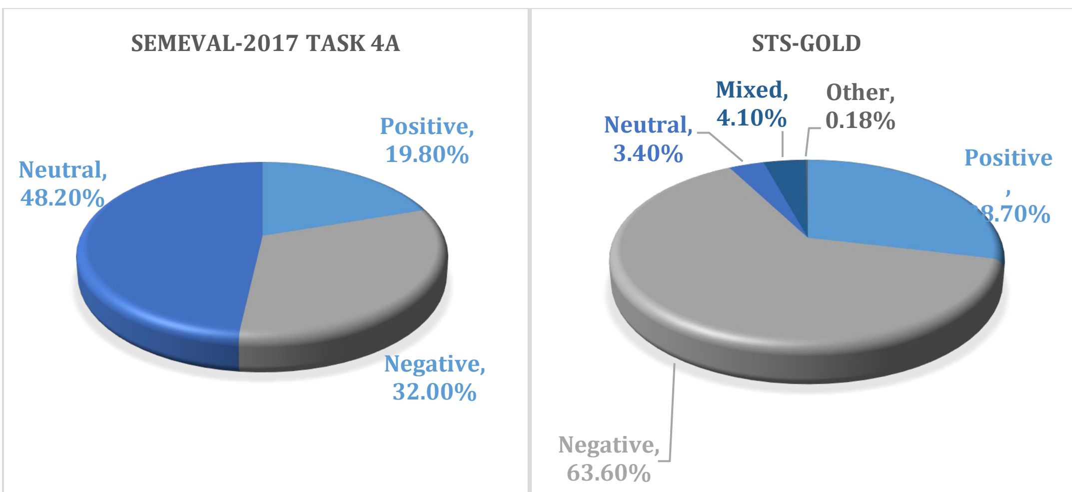

Fig. 5. Category % for SemEval2017 and STS-Gold datasets

Data preprocessing is done where words are lemmatized and stop words are filtered out. The tweets are then tokenized. As discussed for vectorization of tokens, ELMo 5.5B word embedding is used.

4.2 Baselines and Evaluation

To ensure that the HAN achieves state-of-the-art performance, we compared its effectiveness to the existing best reported performance on both the datasets. The metrics used for evaluation were average recall, $F1^{PN}$ and Macro F-score. The best performing system on the SemEval 2017 task 4A dataset is the one described in [18] which achieved a macro-average recall of 73.2. There were two winners [19, 20] of the SemEval 2017 Task A with a macro-average recall of 68.1. Interestingly, HAN achieves superlative results than the winners of task and comparable results to the state-of-the-art (SOTA). Table 3 depicts the comparison of SemEval-2017 sentiment analysis task 4A.

Table 3. Performance of HAN on SemEval2017-Task4A

Model	AvgRec	$F1^{PN}$	Acc
BB_twtr [19] (Leader SemEval-2017, TaskA)	68.1	68.5	65.8
DataStories [20] (Leader SemEval-2017, TaskA)	68.1	67.7	65.1
State-of-the-art [18]	73.2	72.8	71.7
HAN	**73.1**	72.0	**71.7**

The best performing model on STS-Gold dataset is given by Islam et al. [27] in 2019 where the authors use a multi-channel convolutional neural CNN for Twitter emotion and sentiment recognition and report an accuracy of 90.7. The HAN model outperforms the SOTA on STS-Gold by achieving a superior accuracy of 94.6. Table 4 depicts the comparison of STS-Gold dataset.

Table 4. Performance of HAN on STS-Gold

Model	P	R	F1	Acc
Thelwall-Lexicon [23]	79.3	88.0	79.6	82.3
SentiStrength [24]	79.5	77.9	78.6	82.1
SentiCircle with Pivot [31]	Φ	Φ	77.5	80.3
SentiWordNet +Gradient Boosting [32]	Φ	Φ	Φ	82.21
Deep CNN [26]	82.8	82.6	82.7	86.0
Deep CNN + SentiCircle [3]	Φ	Φ	Φ	88.0
MC-CNN [27]	90.0	88.3	88.9	90.7
HAN	**92.8**	**90.2**	**92.0**	**94.6**

We take a closer look at the predictions made by HAN with the help of heatmap and mosaic plot visualization for both the datasets as shown in fig. 6 & 7. The rows show the actual class of a repetition and columns show the classifier's prediction of the respective confusion matrix.

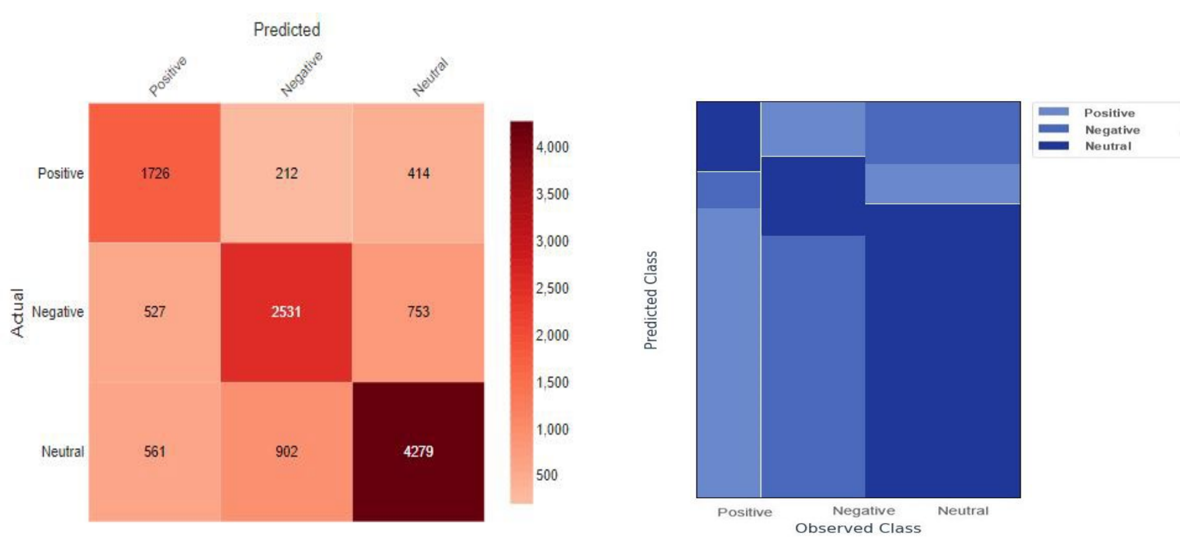

Fig. 6. Heatmap and mosaic plot of SemEval-2017 Task 4A dataset

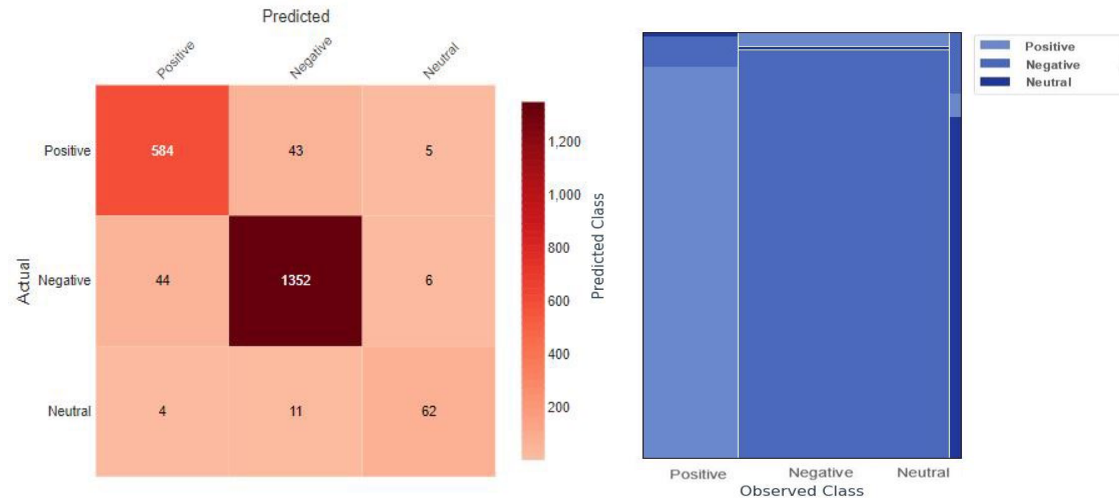

Fig. 7. Heatmap and mosaic plot of STS-Gold dataset

5. Conclusion

Undeniably, social intelligence on social networks is continuously increasing but the heterogeneity of content makes it hard to analyse and understand sentiments. Considerable research on sentiment analysis has concentrated on feature engineering as generating inputs for a learning model which recognizes the context, tone, and previous indications of sentiment can eventually improve the accuracy and comprehend a better overall sense of what actually is being said. This work reports and evaluates the HAN model with ELMo deep contextualized embeddings for sentiment classification in tweets. That is, ELMo is used for word vector learning to provide the classifier with the learned representations of the text and the classifier utilizes a bi-directional GRU with attention mechanisms at both word and sentence level. The model is evaluated on SemEval-2017 and STS-Gold datasets and respective accuracy of 71.7% and 94.6% is reported. Consequently, an AI-driven social media mining model is proffered which accomplishes sentence-level sentiment classification task to enhance user-services in the SIoT framework.

As a future work, the model will be extended to multimodal text. Also, as the cultural miscellanies, geographically limited trending topic hash-tags, access to aboriginal language keyboards and conversational comfort in native language exemplify the variability and size of user-generated content on social networks, multilinguality compounds the linguistic challenges of SA and form a vital direction of research.

References

1. Roopa, M. S., Pattar, S., Buyya, R., Venugopal, K. R., Iyengar, S. S., & Patnaik, L. M. (2019). Social Internet of Things (SIoT): Foundations, thrust areas, systematic review and future directions. *Computer Communications*, *139*, 32-57.
2. Jiang, N., Tian, F., Li, J., Yuan, X., & Zheng, J. (2020). MAN: mutual attention neural networks model for aspect-level sentiment classification in SIoT. *IEEE Internet of Things Journal*, *7*(4), 2901-2913.
3. Kumar, A., Srinivasan, K., Cheng, W. H., & Zomaya, A. Y. (2020). Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data. *Information Processing & Management*, *57*(1), 102141.
4. Young T, Hazarika D, Poria S, Cambria E. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence magazine*. 2018 Jul 20; 13(3):55-75.
5. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J, Chen T. Recent advances in convolutional neural networks. *Pattern Recognition*. 2018 May 1; 77:354-77.

6. Otter DW, Medina JR, Kalita JK. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*. 2020 Apr 21.
7. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification. *In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies 2016* Jun (pp. 1480-1489).
8. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L. Deep contextualized word representations*. arXiv preprint arXiv*:1802.05365. 2018 Feb 15.
9. Scherer, K., Dan, E., & Flykt, A. (2006). What determines a feeling's position in affective space? A case for appraisal. *Cognition & Emotion*, *20*(1), 92-113.
10. Baccianella, S., Esuli, A., & Sebastiani, F. (2010, May). Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec* (Vol. 10, No. 2010, pp. 2200-2204).
11. Kumar, A., & Sebastian, T. M. (2012). Sentiment analysis: A perspective on its past, present and future. *International Journal of Intelligent Systems and Applications*, *4*(10), 1-14.
12. Kumar, A., & Sebastian, T. M. (2012). Sentiment analysis on twitter. *International Journal of Computer Science Issues (IJCSI)*, *9*(4), 372.
13. Kumar, A., & Jaiswal, A. (2017). Empirical study of twitter and tumblr for sentiment analysis using soft computing techniques. In *Proceedings of the world congress on engineering and computer science* (Vol. 1, pp. 1-5).
14. Kumar, A., & Jaiswal, A. (2020). Systematic literature review of sentiment analysis on Twitter using soft computing techniques. *Concurrency and Computation: Practice and Experience*, *32*(1), e5107.
15. Kumar, A., & Jaiswal, A. (2019). Swarm intelligence based optimal feature selection for enhanced predictive sentiment accuracy on twitter. *Multimedia Tools and Applications*, *78*(20), 29529-29553.
16. Kumar A. & Jaiswal, A. (2020). A Deep Swarm-Optimized Model for leveraging Industrial Data Analytics in Cognitive Manufacturing, in IEEE Transactions on Industrial Informatics doi: 10.1109/TII.2020.3005532.
17. Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study. *Electronics*, *9*(3), 483.
18. Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. BERTweet: A pre-trained language model for English Tweets. arXiv preprint arXiv:2005.10200 (2020).
19. Cliché, M. (2017). BB twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs. SemEval-2017: 573–580.
20. Baziotis C., Pelekis N. & Doulkeridis C. (2017). DataStories at SemEval-2017 Task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis, *Proc. 11th Int. Workshop on Semantic Evaluations*, pp. 747-754, Aug. 2017.
21. Saif, H., Fernandez, M., He, Y., & Alani, H. (2013). Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset, the STS-Gold.
22. Wilson, T., Wiebe, J., & Hoffmann, P. (2005, October). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing* (pp. 347-354).
23. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., & Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the American society for information science and technology*, *61*(12), 2544-2558.
24. Thelwall, M., Buckley, K., & Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, *63*(1), 163-173.
25. Jin, Y., Zhang, H., & Du, D. (2016, November). Improving deep belief networks via delta rule for sentiment classification. In *2016 IEEE 28th international conference on tools with artificial intelligence (ICTAI)* (pp. 410-414). IEEE.
26. Jianqiang, Z., Xiaolin, G., & Xuejun, Z. (2018). Deep convolution neural networks for twitter sentiment analysis. *IEEE Access*, *6*, 23253-23260.
27. Islam, J., Mercer, R. E., & Xiao, L. (2019, June). Multi-channel convolutional neural network for twitter emotion and sentiment recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 1355-1365).
28. Saif, H., He, Y., Fernandez, M., & Alani, H. (2016). Contextual semantics for sentiment analysis of Twitter. *Information Processing & Management*, *52*(1), 5-19.

29. Rosenthal, S., Farra, N., & Nakov, P. (2017, August). SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)* (pp. 502-518).
30. Saif, H., He, Y., Fernandez, M., & Alani, H. (2014, May). Adapting sentiment lexicons using contextual semantics for sentiment analysis of twitter. In *European Semantic Web Conference* (pp. 54-63). Springer, Cham.
31. Saif, H., Fernandez, M., He, Y., & Alani, H. (2014, May). Senticircles for contextual and conceptual semantic sentiment analysis of twitter. In *European Semantic Web Conference* (pp. 83-98). Springer, Cham.
32. Kumar, A., & Garg, G. (2019). Sentiment analysis of multimodal twitter data. *Multimedia Tools and Applications*, *78*(17), 24103-24119.