Natural Language Processing for intent classification: an application to the classification of emotions and dialog act

Pierre Personnat[*]
ENSAE Paris
pierre.personnat@ensae.fr

Alexis Vignard[*†]
ENSAE Paris
alexis.vignard@ensae.fr

Abstract

This paper forcuses on the implementation of a Natural Language Processing (NLP) model that performs contextual emotion classification on a dialogue dataset sourced from the popular TV show "Friends". The primary goal is to account for the context of each sentence in the conversations through a self-attention mechanism. To achieve this objective, the authors propose constructing a hierarchical attention network that mimics the structure of a corpus of dialogues, which has a hierarchical structure between the conversation, the sentences within it, and the words in each sentence. The aim is to capture a dual context, namely, that the importance of words varies depending on the sentence or conversation, and not every sentence in a conversation carries equal importance in the context of the dialogue. Therefore, the meaning and, consequently, the label of an utterance will vary depending on its location within the conversation.

1 Introduction

The emergence of large-scale language model such as ChatGPT has marked a significant paradigm shift in the field of natural language processing. These language models are capable of learning from vast amounts of textual data, allowing them to capture linguistic patterns and structures at an unprecedented level of complexity.

However, identifying the underlying emotions [8, 6] in a sentence can be a challenging task, especially when considering its context. The same sentence can convey completely different emotions [16, 4] depending on the preceding and subsequent sentences. This challenge is particularly evident in movie or TV show dialogues. In order to overcome this challenge, it is essential not only to have labeled sentences, but also to know the order and context in which they were spoken. This requires a supervised sequential labeling approach, where each conversation is numbered and each sentence, or utterance, is placed within a specific context and order.

This being said, we will build a hierarchical deep neural network model that hierarchically encodes each conversation, coupled with a self-attention layer that optimally gives the right weight to each word in each sentence depending on the context of the conversation. This self-attention mechanism makes it possible to process words and sentences with different meanings depending on the context in which they are spoken.

	Utterance	Emotion
Rachel:	I've been so crazy ...	sadness
Ross:	I know.	neutral
Rachel	**Yeah.**	**neutral**
Rachel:	Does it still hurt?	sadness
Ross:	**Yeah.**	**sadness**
Phoebe:	It's warm.	neutral
Joey:	**Yeah.**	**joy**

Table 1: Example of the same Utterance in different conversations

[*] stands for equal contribution

[†] https://github.com/AlexisVignard-hub/Intent-classification-NLP-2023

2 Problem framing

The task of Emotion/Sentiment classification takes a dataset D, where conversations C are a sequence of utterances $U = \{u_1, u_2, \ldots, u_n\}$. Each utterance is a sequence of words, such that $u_i = \{w_1^i, w_2^i, \ldots, w_n^i\}$. For a conversation j, we have a sequence of labels of emotion matching every utterances $Y^j = \{y_1^j, y_2^j, \ldots, y_n^j\}$.

In order to make the conversations and the sentences suitable for the application of a NLP model, we need to ensure that all conversations and sentences respect a certain structure. To do so, we denote T_x the maximum number of words in one sentence, and U_τ the maximum number of sentences in one conversation.

First of all, the words within the Utterances are embedded using FasText[1], which is an open-source library developed by Facebook AI Research (FAIR). It is an extension of the word2vec algorithm that is specifically designed for learning word embeddings. FastText's approach involves breaking words down into sub-word units, such as character n-grams, and creating a vector representation for each of these sub-word units. By adding up the vectors of all the sub-word units that form a word, FastText can create a word vector that captures the meaning of a word, even when it includes rare or unfamiliar character combinations.

Secondly, our Hierarchical Attention Network (HAN) aims at building a conversation representation that would help predicting the labels for each utterance by first building a sentence representation for each sentence in the conversation. This sentence representation is built from an encoding phase of the words contained in the sentence, coming from the output of a Recurrent Neural Network (RNN) that deals with the order of the words and respect the sequence, but slightly modified by a self attention layer. We will use Gated Recurrent Units (GRU) to perform this first encoding.

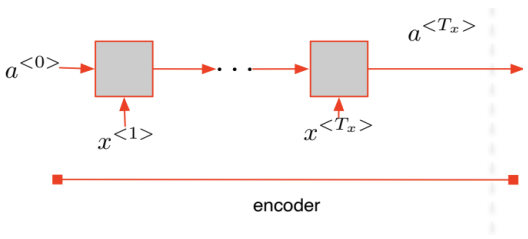

Figure 1: Encoding scheme of words in RNN

In this representation, each $x^{<i>}$ is one word of the sequence and $a^{<T_x>}$ is the final sentence encoding. At each time t of the sentence (or t^{th} word), the hidden state a^{t-1} is transmitted to the next grey box (a GRU) that takes as input a^{t-1} and the t^{th} word $x^{<t>}$. A GRU uses a gates mechanism in order to track the encoding of previous states: a relevance gate Γ_r and an update gate Γ_u to control how past and new information are passed to the next state.

he relevance gate Γ_r controls how much the previous hidden state contributes to the new candidate state and follows the equation:

$$\Gamma_r = \sigma(W^r[a^{<t-1>}, x^{<t>}] + b^r)$$

Then, the candidate cell value is :

$$\tilde{c}^{<t>} = \tanh\big(W^c[\Gamma_r \odot a^{<t>}, x^{<t>}] + b^c\big)$$

In addition, the update gate defined as follow $\Gamma_u = \sigma(W^u[a^{<t-1>}, x^{<t>}] + b^u)$ determines how much past information is kept and how much of the candidate state information is added.

Finally, the output memory cell is:

$$c^{<t>} = (1 - \Gamma_u) \odot a^{<t-1>} + \Gamma_u \odot \tilde{c}^{<t>}$$

Then the output memory cell is the hidden state and $a^{<t>} = c^{<t>}$.

This architecture summarizes a one-directional encoding but a bi-directional one works on the same manner with a forward GRU taking the previous information and the backward GRU taking the future information into account. The final word hidden state is defined as the concatenation of those two hidden informations:

$$A^{<t>} = [\overrightarrow{c^{<t>}}, \overleftarrow{c^{<t>}}]$$

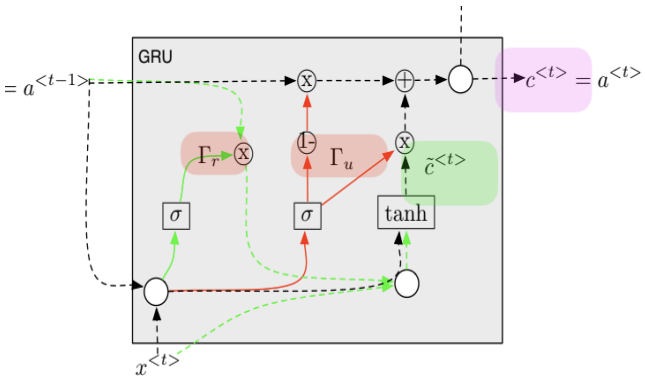

Figure 2: Description of a GRU

Once this final encoding of words in a sentence is computed, we could consider all the information is summarized in a single vector representation. Indeed $a^{<T_x>}$ is supposed to memorize all the informations contained in the words of the sentence in order to have a sentence representation.

However, we want to pay attention to the context of the words in the sentence, but also to the overall conversation context. Indeed, we remember that not all words contribute equally to the representation of the sentence meaning depending on the context. Therefore, we introduce a context aware self-attention mechanism that takes as input firstly the words hidden states of each time and secondly the previous sentence hidden state and therefore computes a weight of importance for each word in the sentence.

We want to replace the τ^{th} sentence vectorisation $c = a^{<T_x>}$ by a local version c^{τ} which is computed as a weighted sum of the encoding hidden states. This leaves us at how to compute those weights.

In order to compute the attention weights, we first pass all the words hidden states and the $(\tau - 1)^{th}$ sentence hidden state through a linear layer followed by a tanh activation: let's call $e^{<\tau,t>}$ the output of this layer for word t of the sentence τ (more precisly $e^{<\tau,t>} = \tanh\big(W^e A^{<t>} + W^s s^{<\tau-1>} + b^e\big)$, $s^{<\tau-1>}$ being defined later as the $(\tau-1)^{th}$ sentence representation). We then use a softmax function and a context word vector to compute definitively those weights:

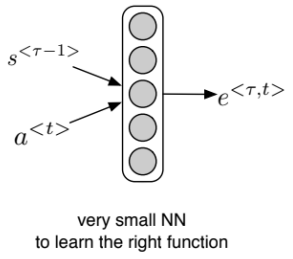

Figure 3: Computation of alignment weights

$$\alpha^{<\tau,t>} = \frac{\exp\big(e^{<\tau,t>} e^{<\tau>}\big)}{\sum_{i=1}^{T_x} \exp\big(e^{<\tau,t>} e^{<\tau>}\big)}$$

The context word vector of sentence τ, $e^{<\tau>}$, is seen as an alignement model between the words of the sentence and its label and summarizes the most informative words. It is randomly initialized and then learnt during the training.
Finally, the τ^{th} sentence representation is $s^{\tau} = \sum_{i=1}^{T_x} \alpha^{<\tau,t>} A^{<t>}$ with $\sum_{i=1}^{T_x} \alpha^{<\tau,t>} = 1$.

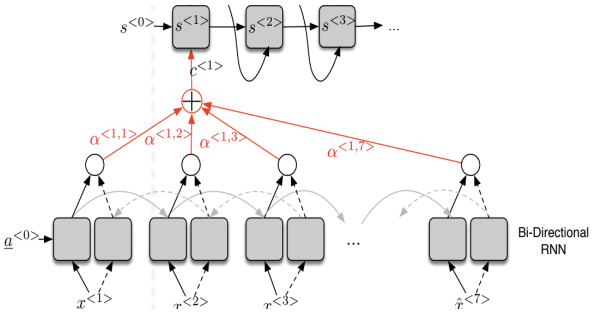

Figure 4: Attention mechanism between word encoding and sentence representation

We now have a contextual sentence representation s^{τ} for each sentence τ.

Then we encode those sentences representations across their unique conversation in order to establish a conversation vectorisation. In order to do so, we take back the structure of the RNN layer with the bi-directional GRU described earlier, and we feed it with the past and future sentence representations at each time (which means for each sentence). One obtains a contextual encoded sentence hidden state defined as $S^{\tau} = [\overrightarrow{b^{<\tau>}}, \overleftarrow{b^{<\tau>}}]$ with $\overrightarrow{b^{<\tau>}} = \overrightarrow{GRU}(s^{\tau})$ and $\overleftarrow{b^{<\tau>}} = \overleftarrow{GRU}(s^{\tau})$.

This bi-directional GRU allows to consider the whole structure of the conversation with sentences not having the same role and importance in a conversation.

Having now all the whole conversation representation by concatenating each hidden sentence vector, we add a classification layer with inputs this conversation representation (U_{τ} elements) and outputs, for each of these elements, a vector of 7 probabilities to belong to each of the emotion's label. This layer is a fully-connected layer of a typycal neural neutwork and consists in a simple linear interpolation of the inputs, followed by a softmax activation for each encoded input.

This layer makes the bridge between the encoded representation of the conversation, itself containing the encoded representation of the sentence, computed from the encoding and attention mechanism of the words that it contains, and each utterance emotion's label.

The issue spotted with this last layer is that each label is decoded independently of the other labels of the same sentence.

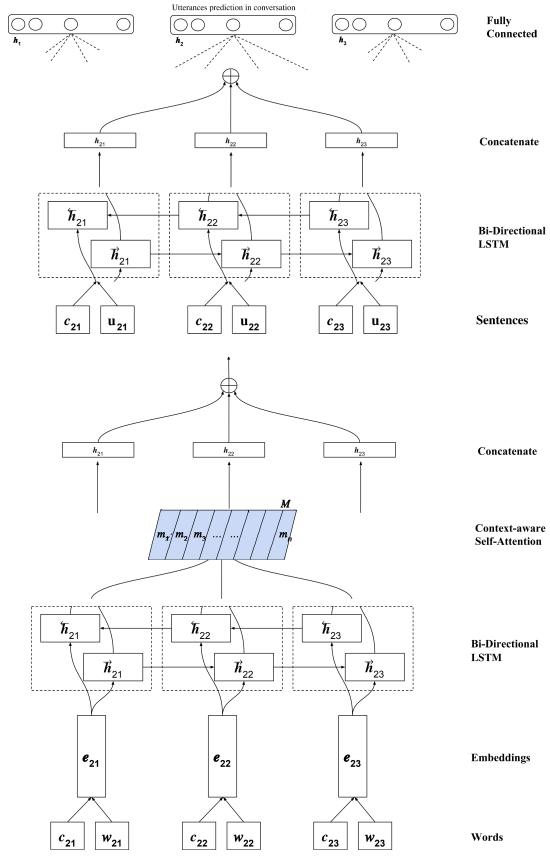

Figure 5: Model architecture

3 Experiments Protocol

3.1 Related works

Early works already dealt with the EmotionLine dataset. *An Emotion Corpus of Multi-Party Conversations*[14], provides an approach to model emotion recognition using not only textual language, but collection of audiovisual conversations between multiple speakers designed for use in training and testing emotion recognition models. They introduced for the first time the EmotionLine dataset and explained how the annotation of utterances had been done. The paper's findings suggest that the EmotionLine dataset can be a valuable resource for future research in emotion recognition and multimodal analysis.

Hierarchical Pre-training for Sequence Labelling in Spoken Dialog[2] provides a clear and didactic explanation of the double encoding scheme within the words and the sentences.

Dialogue Act Classification with Context-Aware Self-Attention[15] explains how to manage the implementation of different layers on those networks: from the words embedding, the words and sentences RNN encoding with GRU, and specially the context aware attention mechanism. The figure 5 describing the architecture of our neural network was largely inspired by this paper.

Hierarchical Attention Networks for Document Classification[18] provides guidelines on the understanding and the implementation of the GRU layers and the multiple details within an attention layer, from the linear interpolation of the encoding, to the computation of the self attention weights.

3.2 Dataset

3.2.1 DailyDialog dataset

First of all, in order to check that our proposed models works correctly, we are going to test it on the same dataset used in class, which is already preprocessed, a less complex and a more reliable one than our EmotionLine Dataset. The DailyDialog dataset contains multiple conversations from human written daily conversations with only two parties involved and no speaker information.
This dataset offers labelled data on emotions (neutral, happy, surprise, sad, anger, disgust, and fear), but also on dialog acts (inform, question, directive, commissive).

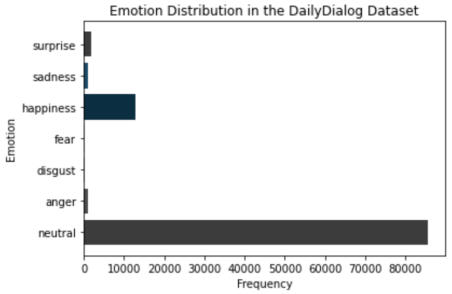

Figure 6: DailyDialog emotion label distribution

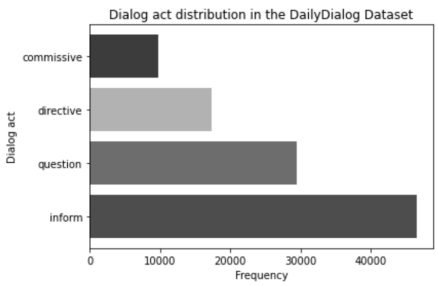

Figure 7: DailyDialog act label distribution

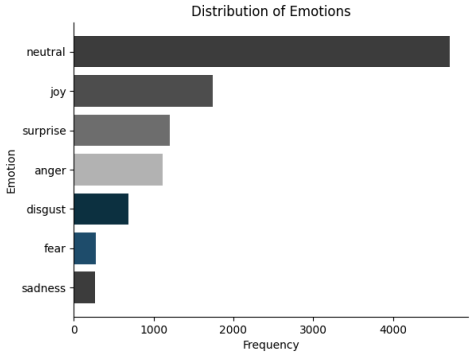

Figure 8: Emotion line label distribution

It is a quite long dataset with more than 13,000 dialogues each containing on average 8 speaker turns ($U_\tau = 8$) with around 15 tokens per turn ($T_x = 15$). As the dialogues in the dataset reflect our daily communications and topics, each conversation has a specific context and our proposed attention model is very suitable to be tested on this dataset.

3.2.2 EmotionLine dataset

After that, we will pursue our study on the EmotionLine dataset, the first dataset constituted of several utterances in dialogues with emotions labelled only based on their textual content. This dataset triggered our interest as it mainly comes from the famous TV show Friends that marked our childhood.

The labels of each utterance are the same as the DailyDialog dataset with the six Ekman's basic emotions plus the neutral emotion. Contrary on the previous dataset, it only contains about 2,000 dialogues. Each dialogue is a scene from an episode of Friends, with a number of utterances in a dialogues varying from 5 to 24 sentences.

In order to label each utterance from a dialogue, people were asked to think for at least 3 seconds and then annotate every utterance in a dialogue considering the whole context of the dialogue.

The label collecting the most votes was decided to be the final emotion of the utterance.

Similarly to the previous dataset, EmotionLine is very suitable to be tested on our model as in a conversation between two characters, the emotion of each sutterance is very dependent on the context of the conversations and on the past and future utterances.

EmotionLine Dataset was already splitted in three distinct datasets (train, validation and test), however it was not yet preprocessed. Therefore, we carried out a grouping of every utterance of a unique dialogue in order to create separate conversation.

We then performed a tokenization of the utterances, in order to separate words within a sentence, keeping 20 words per sentence ($T_x = 20$), and originally we decided to keep a maximum of 12 utterances per conversation ($U_\tau = 12$).

Let's note that this dataset is extremely imbalanced with about 47% of utterances being labelled as "neutral" emotion and only about 2% of utterances being defined as "disgust" or "fear" emotions, and we made a 300 dimension embedding of every word using FastText.

4 Results

Dataset	Weights	Max sentences	Accuracy
MELD	No	12	0.64*
		6	0.55*
	Yes	12	0.38*
		6	0.06*
DailyDialog Emotion	No	12	0.89
	Yes	12	0.55
DailyDialog Act	No	12	0.86
	Yes	12	0.83

Table 2: Results with HAN model

*Results with * means that the model only learned one label.*

Dataset	Weights	Max sentences	Accuracy
New MELD	No	12	0.61

Table 3: Results with HAN on transformed MELD dataset

Dataset	Weights	Max sentences	Accuracy
DailyDialog Act	No	12	0.78

Table 4: Results with Bi-LSTM model

On the DailyDialog dataset, without taking into account the label weights, the HAN model predict emotions with an accuracy of 0.89. Taking the weights into account does not seem to work, with an accuracy of only 0.55. In fact, the prediction of the "neutral" emotion is almost perfect, that of "surprise" is average, and the others are not learned by the model. With the weights, the model still tries to learn the other emotions but the overall results are very poor.

For the dialogue acts prediction, the HAN model with the weights obtains 0.83 of accuracy. Without the weights, the results increases to 0.86, but the prediction of the labels is slightly less harmonious because the "directive" and "commissive" acts are slightly more neglected. The results are still very good.

Unfortunately, and as expected, for the EmotionLine dataset, our model failed to detect other emotions than the "neutral" one, with an accuracy of 0.64, roughly corresponding to the share of neutral labels in the dataset. We then tried to use different class weights when defining our loss function in order to tackle this imbalanced learning but the results were not satisfying at all.

Afterwards, we decided to regroup the 4 least frequent emotions (anger, sadness, disgust and fear) accountable for less than 15% of all the utterances into a single one called "bad" emotion. In parallel, we randomly removed some "neutral" labelled utterances in dialogues in order to lower the proportion of this emotion. We were aware that this arbitrary selection could deteriorate the context and the sense of the dialogues. However, this deterioration is negligible because a neutral utterance adds very little context to a dialogue. In fact, in most cases, a neutral utterance will be a dialogue such as: "Yes. Yeah. Hi." etc.... The accuracy of 0.61 obtained with this dataset "New MELD" is quite low, nevertheless the prediction is balanced and better if we want to predict other emotions.

Finally, in a last attempt to improve our results, we implemented a Conditional Random Field layer (CRF) as the last layer of our network. This improvement, which allows to decode labels in a dependent way instead of taking them one by one, unfortunately did not bring any significant improvement on each of the tested parameter configurations (weight and max sentences) on each dataset.

5 Discussion/Conclusion

The various attempts to improve the HAN model have not worked, probably because of the small size of the dataset. Indeed, in addition to being very unbalanced, there are very few labels that allow the model to train properly (only a few hundred for "sadness" "fear" and "disgust" labels). This explains why the results are very poor when lowering the maximum sentence, but also why the weights do not give satisfactory results. On the other hand, even though the DailyDialog dataset is unbalanced, it offers a quantitatively sufficient number of occurrences for the minority labels, leaving more chance for the network to learn when the weights were taken into account.

Nevertheless, this paper proposes a clear, precise and didactic implementation of a hierarchical attention encoding respecting the hierarchical structure of a conversation. It provides a deep understanding of some of the most crucial concepts of NLP and its particularities with respect to the textual supervised learning.

References

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information, 2016.

[2] Emile Chapuis, Pierre Colombo, Matteo Manica, Matthieu Labeau, and Chloé Clavel. Hierarchical pre-training for sequence labelling in spoken dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2636–2648, Online, November 2020. Association for Computational Linguistics.

[3] Sheng-Yeh Chen, Chao-Chun Hsu, Chuan-Chun Kuo, Ting-Hao, Huang, and Lun-Wei Ku. Emotionlines: An emotion corpus of multi-party conversations, 2018.

[4] Pierre Colombo. *Learning to represent and generate text using information measures*. PhD thesis, (PhD thesis) Institut polytechnique de Paris, 2021.

[5] Pierre Colombo, Emile Chapuis, Matthieu Labeau, and Chloé Clavel. Code-switched inspired losses for spoken dialog representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8320–8337, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[6] Pierre Colombo, Emile Chapuis, Matthieu Labeau, and Chloe Clavel. Improving multimodal fusion via mutual dependency maximisation. *EMNLP 2021*, 2021.

[7] Pierre Colombo, Emile Chapuis, Matteo Manica, Emmanuel Vignon, Giovanna Varni, and Chloe Clavel. Guiding attention in sequence-to-sequence models for dialogue act prediction. 2020.

[8] Pierre Colombo, Chloe Clavel, and Pablo Piantanida. A novel estimator of mutual information for learning to disentangle textual representations. *ACL 2021*, 2021.

[9] Tanvi Dinkar, Pierre Colombo, Matthieu Labeau, and Chloé Clavel. The importance of fillers for text representations of speech transcripts, 2020.

[10] Alexandre Garcia*, Pierre Colombo*, Slim Essid, Florence d'Alché Buc, and Chloé Clavel. From the token to the review: A hierarchical multimodal approach to opinion mining. *EMNLP 2019*, 2019.

[11] Deng Jiawen and Fuji Ren. Hierarchical network with label embedding for contextual emotion recognition. *Research*, 2021, 2021.

[12] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. Dailydialog: A manually labelled multi-turn dialogue dataset, 2017.

[13] Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. Dialoguernn: An attentive rnn for emotion detection in conversations, 2018.

[14] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, Florence, Italy, July 2019. Association for Computational Linguistics.

[15] Vipul Raheja and Joel Tetreault. Dialogue act classification with context-aware self-attention, 2019.

[16] Wojciech Witon*, Pierre Colombo*, Ashutosh Modi, and Mubbasir Kapadia. Disney at iest 2018: Predicting emotions using an ensemble. In *Wassa @EMNP2018*, 2018.

[17] Kailai Yang, Tianlin Zhang, Hassan Alhuzali, and Sophia Ananiadou. Cluster-level contrastive learning for emotion recognition in conversations. *IEEE Transactions on Affective Computing*, pages 1–12, 2023.

[18] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 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*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics.

[19] Li Yanran, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. Dailydialog: A manually labelled multi-turn dialogue dataset. 10 2017.

A Appendix: results with HAN

A.1 Emotion lines dataset

Label	Precision	Recall	F1-score
neutral	0.64	1.00	0.78
joy	0.00	0.00	0.00
surprise	0.00	0.00	0.00
anger	0.00	0.00	0.00
sadness	0.00	0.00	0.00
disgust	0.00	0.00	0.00
fear	0.00	0.00	0.00
Accuracy			**0.64**

Table 5: Results without weights, max sentences = 12

Label	Precision	Recall	F1-score
neutral	1.00	0.50	0.67
joy	0.00	0.00	0.00
surprise	0.12	0.54	0.20
anger	0.00	0.00	0.00
sadness	0.00	0.00	0.00
disgust	0.03	0.46	0.06
fear	0.04	0.10	0.05
Accuracy			**0.38**

Table 6: Results with weights, max sentences = 12

Label	Precision	Recall	F1-score
neutral	0.56	1.00	0.72
joy	0.00	0.00	0.00
surprise	0.00	0.00	0.00
anger	0.00	0.00	0.00
sadness	0.00	0.00	0.00
disgust	0.00	0.00	0.00
fear	0.00	0.00	0.00
Accuracy			**0.56**

Table 7: Results without weights, max sentences = 6

Label	Precision	Recall	F1-score
neutral	0.00	0.00	0.00
joy	0.00	0.00	0.00
surprise	0.00	0.00	0.00
anger	0.00	0.00	0.00
sadness	0.06	1.00	0.11
disgust	0.00	0.00	0.00
fear	0.00	0.00	0.00
Accuracy			**0.06**

Table 8: Results with weights, max sentences = 12

Label	Precision	Recall	F1-score
neutral	0.69	0.95	0.80
bad	0.39	0.08	0.13
joy	0.36	0.51	0.42
surprise	0.00	0.00	0.00
Accuracy			**0.61**

Table 9: Results on New MELD without weights, max sentences = 12

A.2 DailyDialog act

Label	Precision	Recall	F1-score
inform	0.76	0.92	0.83
question	0.86	0.90	0.88
directive	0.64	0.41	0.50
commissive	0.64	0.28	0.39
Accuracy			**0.86**

Table 10: Results without weights, max sentences = 12

Label	Precision	Recall	F1-score
inform	0.88	0.66	0.75
question	0.90	0.87	0.89
directive	0.51	0.65	0.57
commissive	0.40	0.72	0.51
Accuracy			**0.83**

Table 11: Results with weights, max sentences = 12

A.2.1 DailyDialogue Emotion

Label	Precision	Recall	F1-score
Neutral	0.93	0.96	0.94
Joy	0.00	0.00	0.00
Surprise	0.00	0.00	0.00
Anger	0.00	0.00	0.00
Sadness	0.49	0.49	0.49
Disgust	0.00	0.00	0.00
Fear	0.00	0.00	0.00
Accuracy			**0.89**

Table 12: Results without weights, max sentences = 12

Label	Precision	Recall	F1-score
Neutral	0.99	0.53	0.69
Joy	0.07	0.29	0.12
Surprise	0.02	0.02	0.02
Anger	0.05	0.06	0.05
Sadness	0.19	0.80	0.30
Disgust	0.06	0.56	0.11
Fear	0.11	0.71	0.20
Accuracy			**0.55**

Table 13: Results with weights, max sentences = 12