Deep Learning-based Categorical and Dimensional Emotion Recognition for Written and Spoken Text

Bagus Tris Atmaja, Kiyoaki Shirai, Masato Akagi

Abstract—The demand for recognizing emotion in text has grown increasingly as human emotion can be expressed via text and many technologies, such as product reviews and speech transcription, can benefit from text emotion recognition. The study of text emotion recognition was established some decades ago using unsupervised learning and a small amount of data. Advancements in computation hardware and in the development of larger text corpus have enabled us to analyze emotion in the text by more sophisticated techniques. This paper presents a deep learning-based approach for the recognition of categorical and dimensional emotion from both written and spoken texts. The result shows that the system performs better on both categorical and dimensional task ($> 60\%$ accuracy and $< 20\%$ error) with a larger dataset compared to a smaller dataset. We also found the recognition rate is affected by both the size of the data and the number of emotion categories. On the dimensional task, a larger amount of data consistently provided a better result. Recognition of categorical emotion on a spoken text is easier than on a written text, while on dimensional task, the written text yielded better performance.

Index Terms—Deep learning, recurrent neural network, text emotion recognition, categorical emotion, dimensional emotion, spoken text.

1 Introduction

The studies of the emotion recognition are increasingly needed due to high demand for advance implementation e.g. emotional intelligence measurement and market analysis. Emotion, an affective state and brief response to the evaluation of events as being of major significance [1], shows potential benefits if we can detect it precisely. Detecting emotion is a key whether a student in class is confused, engaged, or certain when interacting with a tutorial system, whether a caller to a help line is frustrated, or whether a blog post or a tweet indicated depression. Additionally, detecting emotions like fear in novels, could help us trace what groups or situations are feared and how that changes over time [3]. In business market analysis, acquisition of customer emotion is very useful for both marketing strategy and evaluation. A simple example is by analyzing user review in such online shop application. Obtaining the precise emotion from user review can be useful for measurement of customer satisfaction, future marketing strategy and future product development.

Automatic emotion recognition can be performed in several ways; facial expression, speech, gesture or hand movement. It also can be detected based the language of interaction, both spoken language and written languages. As spoken and written languages are different in some aspects, for example formality, both also may result some differences in emotion recognition performance. Both spoken and written language can be acquired in the same format, i.e. text. A spoken text is a speech transcription usually contains speaker information including his/her emotion, while a written text is a plain format which can be difficult to infer writer information from this kind of text. However, modern text conversation like chat or social media status (Twitter or Facebook), now is closer to spoken text form where it is easier to infer writer information. Advancements in the study of Natural Language Processing (NLP) further the development of textual analysis and increase interest in computational linguistics as tool to analyzing emotion in text.

Emotion itself can be viewed from two major theories, categorical or dimensional emotion. The first view is based on the fixed atomic unit, a discrete category which separates an emotion class from other emotion classes. The best-known atomic theory of emotion is the one proposed by Ekman [2]. In his theory, emotion is divided into 6 categories: surprise, happiness, anger, fear, disgust, and sadness. Another view on emotion theory is dimensional emotion which, views emotion as a numerical score in a 2D or 3D space rather than as discrete categories. Most dimensional models include two dimensions i.e. valence (V) and arousal (A), and some add a third, dominance (D). These three dimensions respectively represent pleasantness, intensity and degree of emotion in response of the stimulus [3]. Having knowledge on both categorical and dimensional emotions will deepen our understanding of human emotion and offer more advantages. For example, a quick response can be taken by knowing target emotion category and the productivity of worker can be identified from its valence, arousal, and dominance (VAD) score [4].

This paper deals with categorical and dimensional emo-

- *Bagus Tris Atmaja is with the Department of Engineering Physics, Institut Teknologi Sepuluh Nopember Surabaya, 60132, Indonesia, and School of Information Science, JAST, Nomi, 9231292, Japan. E-mail: bagus@ep.its.ac.id*
- *Kiyoaki Shirai and Masato Akagi are with School of Information Science, JAIST, Nomi, 9231292, Japan Email: {kshirai, akagi}@jaist.ac.jp*

tion recognition from spoken and written text. A text emotion recognition system based on deep learning approach has been built for this purpose. Specifically, this system is a variant of recurrent neural network (RNN) named gated recurrent unit (GRU). This deep learning-based classifier has been used to evaluate a number of text datasets containing sentences with categorical and dimensional emotion label. Various metrics for objective measurement have been used for different categorical and dimensional tasks based on classification and regression approaches. The result shows the proposed system performs better on a larger dataset than on a small dataset. Some extensions of analysis have been performed including the use of weighting for word embedding as input feature of the system and the impact of data size and unbalance data.

The organization of this paper is described as follows. Section 2 introduces related work to this research. Section 3 explained dataset used on this research, which datasets used for categorical emotion and which datasets containing dimensional emotion, along with its each characteristic. Section 4 contains input features used to feed deep learning system and its extraction process from input text. Section 5 shows construction of the deep neural network system as classifier of the text emotion recognition. Section 6 shows obtained result and discussions. Finally, section 7 concludes works presented on this paper.

2 Related Work

Currently, most classification tasks in many areas are performed using deep learning using supervised learning technique. Given pairs of input and output label, a set of networks are trained to map those input features to target output. Supported by advancement of computer hardware like the use of GPU (graphical processing unit), the computation time to train the network become shorter than ten or twenty years ago which is the reason why this method gain more interest recently.

Significant research have been conducted on text emotion recognition, from text corpus construction [5], [6] to feature extraction [7], classifier development [14], [16] and its application [4]. Strapparava and Mihalcea has developed text corpus for text emotion task [5] and presented an explanation of their corpus annotation process in [8]. Various identification techniques has been used to evaluate the text corpus including the presence of the word in Wordnet affect [9], Latent Sentiment Analysis (LSA) of various words and Naive Bayes classifier. The result shows that no single method absolutely superior to others. The author also reported a similar result achieved by some systems participating in that task [5].

Kim et al. used unsupervised methods to evaluate emotion in text [10]. They used Majority Class Baseline (MCB) as a baseline method and compared it with LSA, Non-negative Matrix Factorization (NMF) and VAD model derived from affective norm of English words (ANEW) [11]. They found that VAD method achieve best result on ISEAR dataset while NMF performed better on SemEval and Fairy Tales dataset [10].

Razek and Fasson [14] used intelligent learning system to recognize emotion in text. They proposed a text-based emotion recognition approach that uses personal text data to recognize user current emotion. A dominant meaning technique was used to recognize user emotion by emotion dominant meaning tree method. They obtained the highest precision and recall on sadness category using ISEAR dataset to build the tree.

TABLE 1
Resume of dataset used in text emotion recognition (Dim: dimensional, Cat: Categorical).

Dataset	Emotion		Text	
	Dim	Cat	Written	Spoken
Affective Text	V[1]	✓	✓	-
ISEAR	A[2]	✓	✓	-
IEMOCAP	✓	✓	-	✓
EmoBank	✓	-	✓	-

[1]Only valence, not used in this research.
[2]Only arousal, not used in this research.

Although deep learning technique shows promising results in some areas, to our knowledge, only a few researchers applied this deep learning technique for text emotion recognition and some of them proposed RNN-based deep learning technique for text emotion recognition. Zhang et al. [26] surveyed some deep learning methods for sentiment analysis including the use of deep learning for emotion recognition. While they listed some research methods and its result on text emotion recognition, most methods are developed to recognize emotion in categorical task, in addition to development of capturing emotional words, building chatting machine, and learning emotional context. The use of GRU in this research is also listed in the papers. While the papers reported the use of GRU to build chat bot and to recognize categorical emotion, this paper proposed evaluation of GRU for both categorical and dimensional emotion with the same architecture. The system architecture is also evaluated for both written and spoken text. The detail of deep learning system is explained in Section 6.

3 Dataset

To understand the characteristics of datasets used on this research, the following short explanation of each dataset is presented.

3.1 SemEval Affective Text

Affective Text [5] is a dataset provided in International Workshop on Semantics Evaluations (SemEval) for a challenge (task number 14)[1]. The task focuses on the classification of emotions and valence. Hence, the dataset consists of news headlines drawn from major newspapers such as New York Times, CNN, and BBC News, as well as from the Google News search engine. Two data sets were made available: a development data set consisting of 250 annotated headlines, and a test data set with 1,000 annotated headlines. The used emotion category is Anger, Disgust, Fear, Joy, Sadness, Surprise. For each emotion category, an interval of [0, 100] was set and annotator are asked to score

1. This dataset can be downloaded at http://web.eecs.umich.edu/~mihalcea/affectivetext/

it on that range, where 0 means the emotion is missing from the given text, and 100 represents maximum emotional load. The categorical text emotion recognition presented in this paper is a conversion from the highest values for each category to 1. Hence, the label data is in binary format, 1 for target emotion, and 0 for other emotion categories.

3.2 ISEAR

ISEAR (International Survey on Emotion Antecedents and Reactions)[2] is a dataset collected by group of psychologists from over the world that contains emotional statements. Student respondents, both psychologists and non-psychologists, were asked to report situations in which they had experienced all of 7 major emotions (joy, fear, anger, sadness, disgust, shame, and guilt). In each case, the questions covered the way they had appraised the situation and how they reacted. This dataset consists of 7666 utterances, and all of them are used in this research.

3.3 IEMOCAP

IEMOCAP [19] is acronym of interactive emotional dyadic motion capture, a multimodal dataset to investigate verbal and non-verbal analysis for understanding expressive human communication. The modalities measured on this dataset included speech, visual, text and motion capture (face, head and hand movement). From those modalities, only text transcription is used. 11 emotion categories are used along with 3-dimensional VAD values of emotion on original dataset. The text corpus consists of 10039 of utterances with unbalance distribution for emotion category. From those 11 categories of emotion in the dataset, 6 emotions are used for categorical emotion recognition as in SemEval Affective Text dataset.

3.4 EmoBank

EmoBank [6][3] is a large-scale text corpus manually annotated with emotion according to the psychological Valence-Arousal-Dominance scheme. It was built at JULIE Lab, Jena University. As the dataset provide valence, arousal and dominance (VAD) value for given sentence/utterance, this dataset is suitable for dimensional text emotion recognition. The range of VAD score is from 1 to 5 same to IEMOCAP dataset with the same scoring method, i.e. using 5-scales Self-Assessment Manikin (SAM). EmoBank dataset consists of 10062 sentences with 17177 unique indexes (words).

Table 1 shows the resume all dataset used in this research. Table 2 shows excerpt of SemEval, IEMOCAP and ISEAR dataset and its categorical emotion label, and table 3 shows excerpt of IEMOCAP and EmoBank text with its dimension label.

TABLE 2
Excerpt of Affective Text, ISEAR and IEMOCAP dataset with its categorical label.

Text	Label
Affective Text	
'Mortar assault leaves at least 18 dead'	sadness
'Goal delight for Sheva'	joy
'Vegetables, not fruit, slow brain decline'	surprise
'Bombers kill shoppers'	fear
IEMOCAP	
'Excuse me.'	neutral
'That's out of control.'	angry
'Did you get the mail? So you saw my letter?'	sad
'Did you get the letter?'	excitement
ISEAR	
'When I was involved in a traffic accident.'	fear
'When I lost the person who meant the most to me. '	sadness
'When I did not speak the truth.'	shame
'When my uncle and my neighbour came home under police escort.'	guilty

TABLE 3
Excerpt of IEMOCAP and EmoBank dataset with its VAD dimension labels.

Text	v	a	d
IEMOCAP			
'Excuse me.'	2.5	2.5	2.5
'That's out of control.'	2.5	3.5	3.5
'Did you get the mail? So you saw my letter?'	2.5	2.0	1.5
'Did you get the letter?'	3.5	3.0	2.0
EmoBank			
'Remember what she said in my last letter?'	3	3	3.2
'If I wasn't working here.'	2.8	3.1	2.8
'Goodwill helps people get off of public assistance.'	3.44	3	3.22
'Sherry learned through our Future Works class thatshe could rise out of the mire of the welfare system and support her family.'	3.55	3.27	3.46

4 Text Feature: Word Embedding

To solve a classification or regression task using neural network approach, a set of input features are needed to feed the system. One of the common features used in text processing is word embedding. A word embedding is a vector representation of a word. A numerical value in the form of vector is used to make the computer to be able to process a text data as it only process numerical value. This value is the points (numeric data) in a space of dimension, which the size of dimension is equal to the vocabulary size. The word representations embed those points in a feature space of lower dimension [20]. In the original space, every word is represented by a one-hot vector, a value of 1 for the corresponding word and 0 for others. This element with value of 1 will be converted into a point in range of vocabulary size.

To obtain a vector of each word in an utterance, that utterance in the dataset must be tokenized. Tokenization is a process to divide an utterance to the number of constituent words. For example, the text "That's out of control" from IEMOCAP dataset will be tokenized as ['That's', 'out', 'of', 'control']. Suppose the number of vocabulary is 2182 (number of words in IEMOCAP dataset with six emotion categories), then the obtained word vector is something similar to

```
text_vector = [42, 44, 11, 471].
```

2. This dataset can be downloaded at http://www.affective-sciences.org/researchmaterial

3. This dataset is openly available at https://github.com/JULIELab/EmoBank

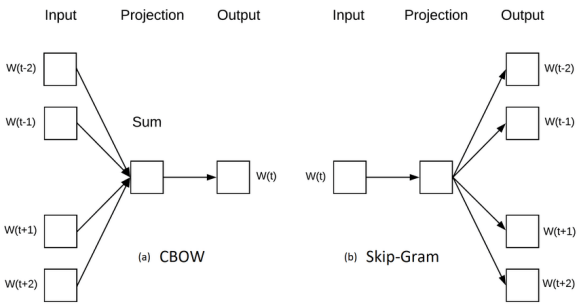

Fig. 1. Two architectures of word2vec: CBOW and Skip-gram [24].

To obtain the fixed length for each utterance, a set of zeros can be padded in front of or behind the obtained vector. The size of this zeros sequence can be obtained from the longest sequence, i.e. an utterance within the dataset which has the longest words, subtracted by the length of vector in the current utterance.

A study to vectorize certain words has been performed by several researchers [21], [22], [23]. The vector of those words can be used to weight the word vector obtained previously. The following word embedding techniques are used in this research.

4.1 word2vec

Classical word embedding paradigm used unsupervised learning algorithm such as LSA, N-gram and the similar methods. Due to advancements in neural network theory supported by the speedup of computer hardware, the search of word vector shifted to deep learning-based algorithm. Mikolov et al. [21] developed word representation using so-called word2vec (word to vector) using neural network language model trained in two steps. First, continuous word vectors are learned by using a simple model, and then the N-gram neural net language Model (NNLM) is trained on top of these distributed representations of words [24]. Two new model architectures are proposed to obtain word vector: the Continuous-Bag-of-Word (CBOW) architecture to predict the current word based on the context, and the Skip-gram to predict surrounding words given the current word. Figure 1 shows those two different architectures and how they process the input to the output.

From those two approaches, skip-gram was founded as an efficient method for learning high-quality distributed vector representations that capture precise syntactic and semantic word relationships [21]. The objective of the Skip-gram model is to maximize the average log probability

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c<j<c,c\neq 0}\log p(w_{t+j}|w_t) \tag{1}$$

where c is the size of the training context (which can be a function of the center word w_t). Larger c results in more training examples and thus can lead to a higher accuracy, at the expense of the training time. The basic Skip-gram formulation of $p(w_{t+j}|w_t)$ can be defined using the softmax function and computational efficiently can be approached by a hierarchical softmax [21].

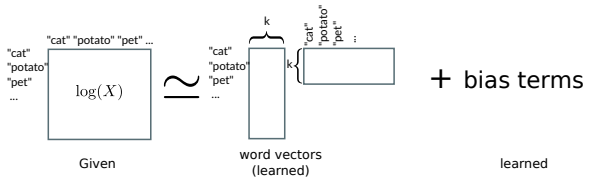

Fig. 2. Ilustration of GloVe representation.

4.2 GloVe embedding

Pennington et al. [22] combined global matrix factorization and local context window methods for learning the space representation of word. In GloVe (Global Vector) model, the statistics of word occurrences in a corpus is the primary source of information available to all unsupervised methods for learning the word representations. Although many such methods now exist, the question still remains as to how meaning is generated from these statistics, and how the resulting word vectors might represent that meaning. Glove captured the global corpus statistics from the corpus, for example, a Wikipedia document or a common crawl document.

In GloVe model, the cost function is given by

$$\sum_{i,j=1}^{V} f(X_{i,j})(u_{i,j}^T v_j + b_i + c_j - \log X_{i,j})^2 \tag{2}$$

where,

- V is the size of the vocabulary,
- X denotes the word co-occurrence matrix (so $X_{i,j}$ is the number of times that word j occurs in the context of word i)
- the weighting f is given by $f(x) = (x/x_{\max})^\alpha$ if $x < x_{\max}$ and 1 otherwise,
- $x_{\max} = 100$ and $\alpha = 0.75$ (determined empirically),
- u_i, v_j are the two layers of word vectors,
- b_i, c_j are bias terms.

In a simple way, GloVe is a weighted matrix factorization with the bias terms, as shown in Figure 2.

4.3 FastText

Mikolov et al. [23] improved word2vec CBOW model by using some strategies including subsample frequent words technique. This modification of word2vec is trained on large text corpus such as news collection, Wikipedia and Web Crawl. They named the pre-trained model with that modification as FastText. The following probability p_{disc} of discarding a word is used by FastText to subsample the frequent words:

$$P_{disc}(w) = 1 - \sqrt{t/f_w} \tag{3}$$

where f_w is the frequency of the word w and t is a parameter > 0.

FastText also counts the classical N-gram word representation by enriching word vector with bag of character n-gram vectors learned from a large corpus. In that computation, each word is decomposed into its character n-grams N and each n-gram n is represented by a vector x_n. The new word vector is then simply the sum of both representations

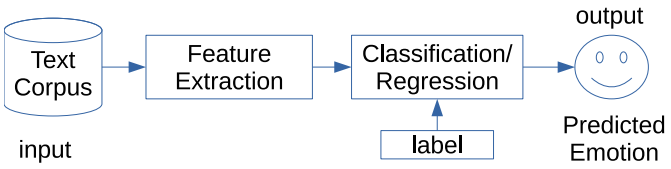

Fig. 3. Block diagram of text emotion recognition system.

$$v_w + \frac{1}{|N|}\sum_{n\in N} x_n. \tag{4}$$

where v_w is the old word vector. The set of n-grams N is limited to 3 to 6 characters in practical implementation.

5 Deep Learning-based Text Emotion Recognition System

5.1 System overview

The previous two sections explain the necessary blocks to construct text emotion recognition system. The third block is a classification process for categorical emotion recognition or a regression process for dimensional emotion recognition. Figure 3 shows the block diagram of this complete text emotion recognition system.

The text corpus is the datasets explained in Section 2. The data in the text corpus consists of two parts, the text data which a set of features will be extracted from, and the emotion/dimension label which is fed to classifier. The feature extraction process converts the sentences in the text corpus to vector form by word embedding technique. This word vector is fed into classifier to produce the output, an emotion category or three dimensions of valence, arousal, and dominance. Both classification task for categorical emotion recognition and regression task for dimensional emotion recognition can be approached by the same classifier model. The following subsection explains the classifier used in this research.

5.2 RNN and GRU Network

Machine learning paradigm has shifted from rule-based system to deep learning-based system. In rule-based system, the algorithm is written manually using set of rules to predict the output. In deep learning-based system, a stack of many functions learn to map input to output. The stack commonly is neural networks. Mathematically, artificial neural network (dense network/fully connected network) can be expressed as follows,

$$y = \sigma(b + w^\top x) \tag{5}$$

where,

- b is bias
- w is weighting vector
- x is input vector
- y is output (scalar or vector)
- σ activation function (sigmoid, tanh, ReLu, softmax or linear)

For deep neural network or deep learning, where the number of hidden layers is more than 1 (shallow network), for example with two and three hidden layers, that equation become

$$y_2 = \sigma(b_2 + w^\top(b_1 + w^\top x))$$
$$y_3 = \sigma(b_3 + w^\top(b_2 + w^\top(b_1 + w^\top x)))$$

The use of more than 1 hidden layer, i.e. deep networks, make easier to train the input-output pair and performs well than a shallow one. A variant of deep learning that works efficiently for sequence modeling (like text processing) is recurrent neural network (RNN). The RNN handles sequence by having a recurrent hidden state (h_t) whose activation at each time is dependent on that of the previous time ($t-1$).

Given a sequence of $x = (x_1, x_2, \ldots, x_T)$, the update of RNN hidden state is formulated,

$$h_t = \begin{cases} 0 & t = 0 \\ \phi(h_{t-1}, x_t), & otherwise \end{cases} \tag{6}$$

where ϕ is a nonlinear function. The update (output of layer) of the recurrent hidden state is implemented as

$$h_t = \sigma(W_{hh}h_{t-1} + W_{hx}x_t) \tag{7}$$

Note that bias term is not expressed in that Equation 7 for simplicity. The output of RNN then is defined as,

$$y_t = \sigma(W_{yh}h_t + b_y) \tag{8}$$

In implementation, Equation 7 can be simplified as multiplication of matrix W_h, which is horizontal stack of matrix W_{hh} and matrix W_{hx}, with vertical stack of matrices h_{t-1} and x_t, multiplied by activation function. Figure 4 shows graphical illustration of RNN and its unrolled version from $t-1$ to $t+1$ along with its comparison to (dense) fully connected network.

One of the variants of RNN is gated rate recurrent unit (GRU). GRU supports the gating of the hidden state, a mechanism which is enabled for when the hidden state should be updated and also when it should be reset. This mechanism is learned, and they address some limitation of RNN like whether early observation is highly significant for predicting all future observations. For example, if the first observation is of great importance we will learn not to update the hidden state after the first observation. Likewise, we will learn to skip irrelevant temporary observations. Lastly, we will learn to reset the latent state whenever needed [26].

The most two important terminologies for GRU is those gated R_t and reset gate Z_t. Beside those variables, a candidate of hidden state is computed as $\tilde{H}_t$ and final update gate as H_t. Here, we converted those variable symbols to vector. All of those four variables are computed below.

$$\mathbf{R}_t = \sigma(\mathbf{X}_t\mathbf{W}_{xr} + \mathbf{H}_{t-1}\mathbf{W}_{hr} + \mathbf{b}_r) \tag{9}$$
$$\mathbf{Z}_t = \sigma(\mathbf{X}_t\mathbf{W}_{xz} + \mathbf{H}_{t-1}\mathbf{W}_{hz} + \mathbf{b}_z) \tag{10}$$
$$\tilde{\mathbf{H}}_t = \tanh(\mathbf{X}_t\mathbf{W}_{xh} + (\mathbf{R}_t \odot \mathbf{H}_{t-1})\,\mathbf{W}_{hh} + \mathbf{b}_h) \tag{11}$$
$$\mathbf{H}_t = \mathbf{Z}_t \odot \mathbf{H}_{t-1} + (1 - \mathbf{Z}_t) \odot \tilde{\mathbf{H}}_t. \tag{12}$$

Figure 5 shows graphical illustration of GRU that accommodates Equation 9-12 above.

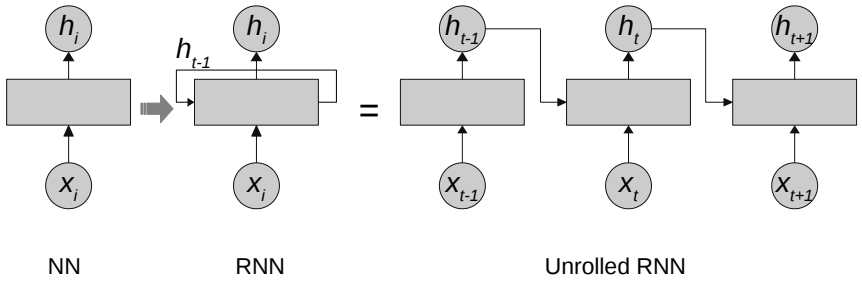

Fig. 4. Graphical illustration of simple neural network (NN), RNN and its unrolled version

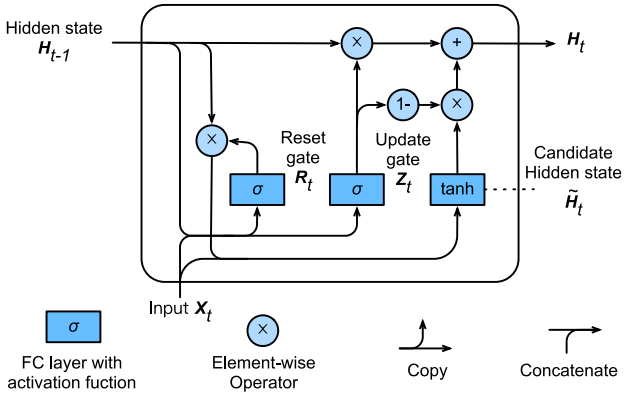

Fig. 5. Graphical illustration of GRU [26]

5.3 Network Model

The GRU neural networks model is implemented in Keras [27]. The networks model consist of five layers: embedding, two GRUs layers and two dense layers. The GRU layers itself are implemented using GPU accelerated layer using NVIDIA CUDA Deep Neural Network library (CuDNN). Figure 6 shows the neural networks model with its input and output dimensions.

The number of units (nodes) shown in the Figure 6 is for categorical model. While for dimensional model we used the same architecture, the number of units are decreased to 64, 64, 32 and 3 for GRU 1, GRU 2, Dense 1 and Dense 2 respectively. The embedding layer as the first layer obtained input from text features, while the number of units at the last layer show the number of output. A number of six or seven units with binary values are used at the last layer to represent the number of emotion categories while dimensional model used three units with floating-point values represent VAD score. A softmax activation function is used for categorical task and a linear activation function is used for regression task. The configuration of these DNN model is available in public repository for benchmarking and reproducibility.[4]

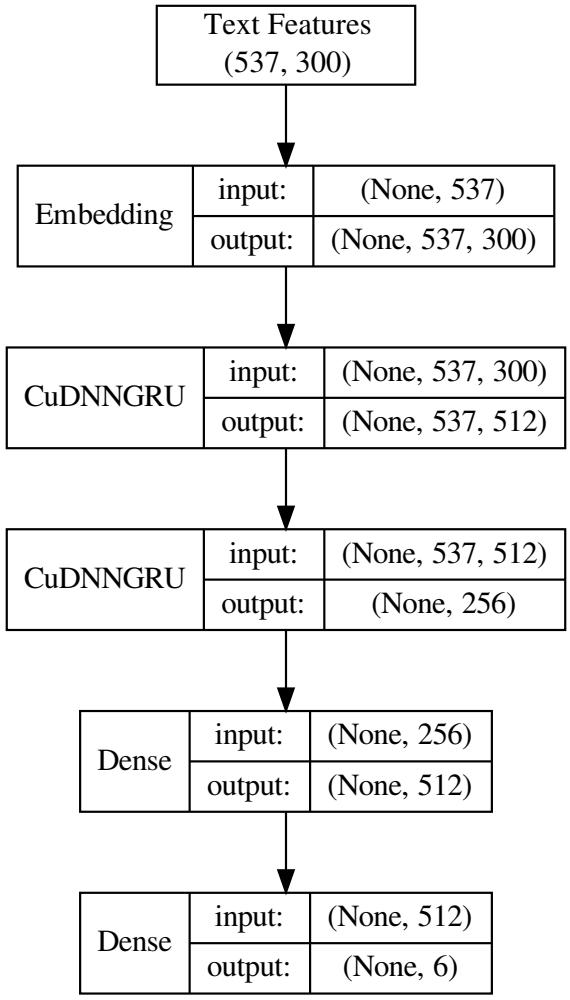

Fig. 6. GRU neural network model architecture for text emotion recognition.

6 Result and Discussions

6.1 Categorical emotion recognition

The first evaluation on this text emotion recognition research is to evaluate the system on categorical emotion. Three datasets are used for this purpose: SemEval, IEMOCAP and ISEAR. Both SemEval and IEMOCAP datasets have the same 6 categories while the ISEAR dataset has 7 categories. The data for each dataset is split into two sets: training and test set with 80:20 ratio. From the total data in training data, about 20% is used for validation.

Table 4 shows performance of test data using the proposed deep learning-based text emotion recognition. The evaluation shows that the developed networks perform poor on SemEval Affective Text, fair on ISEAR dataset, and

4. https://github.com/bagustris/isst_2019

TABLE 4
Results of Categorical Text Emotion Recognition.

Dataset	Precision	Recall	F-Score	Accuracy[1]
Affective Text	0.10	0.28	0.15	0.32
IEMOCAP	0.75	0.72	0.72	0.72
ISEAR	0.56	0.54	0.54	0.56

[1]The score is from test set, the train set accuracy is higher than this score.

well on IEMOCAP dataset. Four metrics are used for measuring performance: precision, recall, f-score and accuracy. All metrics are performed on test data. This means that all those metrics measure the difference between true value and predicted value by the system.

The result shown in table can be explained as follows. First, for the smaller dataset, our system can not learn as much as for larger dataset, like the common deep learning approach that require big dataset to learn the mechanisms inside the data. Secondly, the lower result on Affective Text and ISEAR dataset compared to IEMOCAP dataset could be caused by characteristics of written text compared to spoken text. Basically, written text does not have rich affective information before it is spoken. The speaker gives (additional) emotion information on the written text when he/she spoke it out. While SemEval is text collection from news headlines, ISEAR is written expression for given emotion category. Hence, beside the size of data, the results also shows improvement from SemEval data. Additionally, as we convert the multi categorical emotion labels to single emotion label in the SemEval dataset, this also may causes the obtained performance is lower than previous report [10]. However, our result on ISEAR dataset outperforms their highest result using VAD technique.

To obtain in which emotion category our system performs well and in which category our system performs worse, an analysis of the accuracy of each category is performed using the confusion matrix. For this evaluation, as the SemEval data gives worse result, we eliminate the result from that Affective Text dataset, and only presented results from IEMOCAP and ISEAR dataset. For IEMOCAP dataset, the confusion matrix shows that the system perform best on angry and sad emotion category, which has the bigger number of sentences (1103 and 1084 sentences) in the training dataset compared to other categories. For ISEAR dataset, although the biggest number of sentences in train data is anger, the system recognizes joy and fear better than any other emotion categories. Although we cannot explain why our system performs well on joy and fear emotion categories, other researchers also reported that they obtain the best performance in another category instead of anger category [14], [28]. Figure 7 and 8 shows obtained confusion matrix plot from those datasets. Note that the disgust emotion category is not shown in IEMOCAP confusion matrix due to its number of utterances is very small.

6.2 Dimensional emotion recognition

The biggest advantage of using dimensional data is no need to balance the dataset as the output label is numeric value. By using all sentences in the dataset, the capability of deep learning can be maximized as it learn from the data.

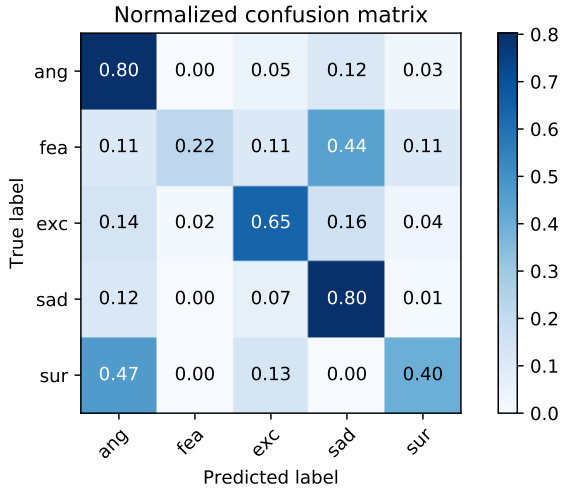

Fig. 7. Confusion matrix of text emotion recognition system for IEMOCAP dataset.

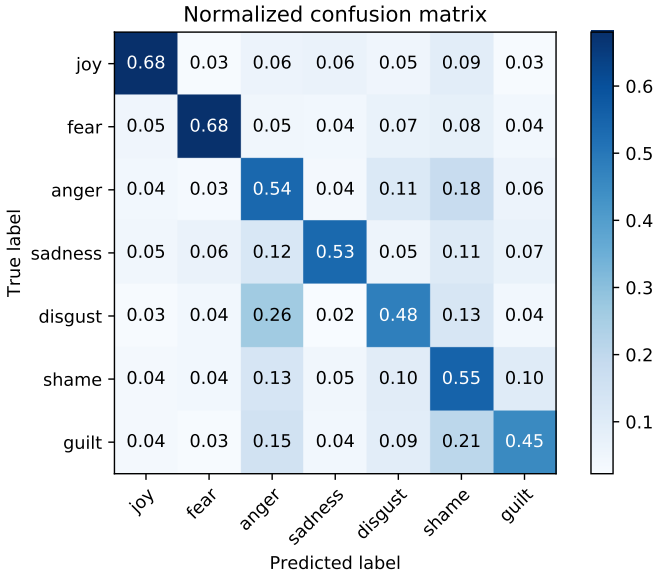

Fig. 8. Confusion matrix of text emotion recognition system for ISEAR dataset.

The more data means the longer learning process can be performed. Hence, the result might be better than learning on the smaller data. On this dimensional text emotion recognition task, the system is trained to learn the score of emotion dimensions (valence, arousal, dominance) from given word embeddings. This is a regression task to predict the dimension of test data given the training data.

Figure 5 shows the results of error score from test data from IEMOCAP and EmoBank dataset. The system used to model the dimensional emotion is the similar to the system used to train categorical emotion, but with smaller units on each layer. The main different is the last dense layer used 3 units without non-linear activation function with floating value while categorical task used 6 or 7 units with binary value. Each error on that table is the average of 5 experiments to eliminate the effect of random initialization.

On that table, it is clearly shown that the system per-

TABLE 5
Performance comparison of dimensional text emotion recognition on testing set.

Dataset	MSE	MAPE (%)
IEMOCAP	0.52	19.74
EmoBank	0.087	7.32

TABLE 6
Comparison on the use of pre-trained word embedding model on IEMOCAP categorical text emotion recognition.

Word Vector Method	Train Acc.[1]	Test Acc.
No weighting	79.23	64.25
Word2Vec	78.70	67.99
GloVe	78.51	72.52
FastText	77.59	70.90

[1]The score is maximum accuracy from validation set.

forms well on both dataset. However, the error of the system from EmoBank dataset is much smaller that IEMOCAP dataset which can be explained as follows. First, the EmoBank is genre-balanced corpus from headlines, blog, essays, fiction, letters, newspaper and travel guide that may contains more informative emotion than other text data. Second, the annotation process is performed by writer and reader with strong agreement [6] with any float score from 1 to 5 scale while IEMOCAP is unbalanced dataset which is annotated from multimodal measurement using scoring system in range of 1 to 5 with 0.5 steps. Those reasons might be the cause why the systems obtained better performance on EmoBank dataset.

6.3 Pre-trained word embedding weighting

To extend our analysis on the use of text features, we evaluate the use of weighting on word embedding extracted from text data. Three pre-trained models explained in Section 4 are evaluated with initial no weighting condition. This weighting factor is defined in embedding layer. The size of all pre-trained model is 300 dimensions.

Table 6 shows performance in term of train and test accuracy from IEMOCAP categorical emotion recognition task. It is clearly shown that weighting factor from GloVe embedding improve the accuracy of test data by 8 %. Without weighting model, the word embedding only shows good performance on training data, but lower performance on test data. It can be concluded that GloVe embedding is also suitable for text emotion recognition despite the original purposes for analogy, similarity and named entity recognition tasks [22].

6.4 The impact of data size & number of categories

Distribution of each emotion category in a dataset may affects the performance of emotion recognition. If the data is unbalance for each category, the system may learn better for a category that contains more data compared to other categories with less data as shown in [13]. This also may occurs to bigger data vs. smaller data. To observe the effect of this data size and number of categories, we evaluate the system on various data size number of categories composition.

Table 7 shows impact of data size on categorical emotion. In this task, we variate the size of data by selecting several

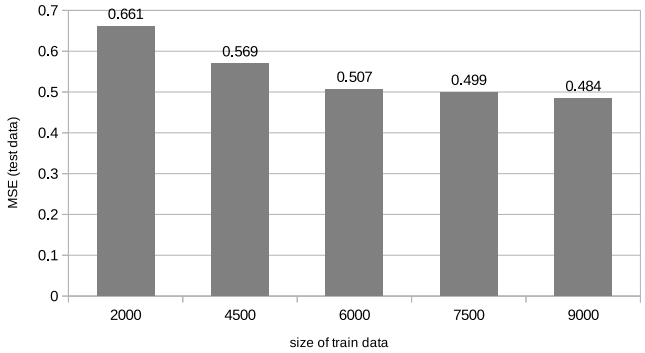

Fig. 9. MSE loss of various size of data in IEMOCAP dimensional text recognition.

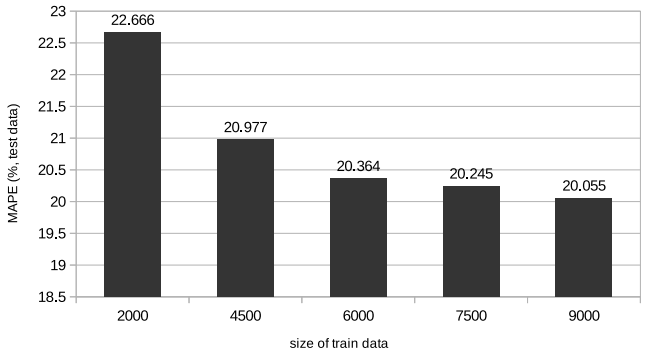

Fig. 10. MAPE of various size of data in IEMOCAP dimensional text recognition.

categories (3, 4, 6, and all 9 categories) from IEMOCAP dataset. All sentences in those selected categories are used. The composition which uses four categories (set 3) is commonly used in speech and multimodal emotion recognition and close to balance each other. For testing accuracy, as our main interest of evaluation metrics, the result shows that that set 3 data only performs better compared to other sets with more categories (six and nine categories). The more number of categories, with the more number of sentences for each category, the lower obtained testing accuracy. This can be explained that the system can learn better with small number of categories (three of four categories), even for closely imbalanced data. However, for bigger dataset, the deep learning system may need balanced dataset for each category. This finding may similar to how our brain works. We, human being, easier to recognize small number categories than larger number of categories on the same amount of data.

Beside the impact of data size on categorical task, we perform the impact of data size on dimensional task, where number of data is not limited by category. Figures 9-12 shows the impact of data size for dimensional text emotion recognition by measuring its errors, mean squared error (MSE) and mean absolute percentage error (MAPE), using IEMOCAP and EmoBank dataset. The MAPE metric is the main interest as it compares the result in the same scale (100-0%) across datasets, while the MSE loss of different dataset has different upper level boundary.

TABLE 7
The impact of data size and number of categories on categorical text emotion recognition using IEMOCAP dataset.

Set	# of sentences in each category									# total	accuracy		
	hap	sur	sad	ang	exc	neu	fea	fru	dis		train	val	test
1	595	107	1084	-	-	-	-	-	-	1786	0.98	0.81	0.73
2	595	107	1084	1103	1041	-	40	-	-	3970	0.95	0.68	0.55
3	-	-	1084	1103	1041	1708	-	-	-	4936	0.94	0.63	0.60
4	595	107	1084	1103	1041	1708	40	1849	2	7529	0.91	0.50	0.44

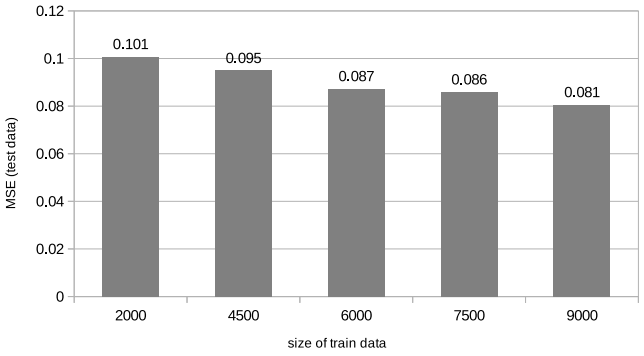

Fig. 11. MSE loss of various size of data in EmoBank dimensional text recognition.

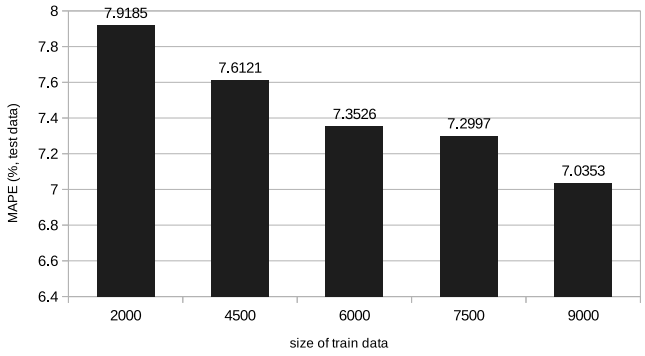

Fig. 12. MAPE of various size of data in EmoBank dimensional text recognition.

Figure 9 shows the loss of the deep learning system on test data while Figure 10 shows its performance in term of MAPE. Both figures show the consistency of the deflation of performance from both MSE loss and MAPE performance. The biggest deflation for both metrics is at the same step, i.e. when changing the number of train data from 2000 to 4500. Consequently, there are room for improvement available for larger amount of data as the result shows only small improvement. We found on this dimensional/regression task that the bigger size of the data, the better performance obtained (smaller error), as shown in Fig. 9 - 12 that quite different with the result from categorical task.

On the EmoBank evaluation using different size of data, the decrease of MSE loss during training session by increasing the size of data is also shown (Fig. 11). The similar pattern also occurs on the EmoBank dataset which has the big deflation on 4500 data compared to 2000 data. However, this patter occurs almost on all change of data except from 6000 data to 7500 data which has the smallest deflation.

The evaluation on data size on dimensional tasks shows consistently deflation of error and loss for both IEMOCAP and EmoBank dataset. EmoBank dataset shows more consistent result on evaluation by the system due to naturalness of its text dataset characteristics and well annotation while IEMOCAP is a multimodal dataset and annotation. This result also may superiority of dimensional views against atomic-unit (categorical) views on emotion theory.

7 Conclusion

In this paper, we presented our works on building and evaluating a text emotion recognition system by utilizing a deep learning approach that works for both categorical and dimensional task and for both written and spoken text. The proposed system consists of two GRU layers with a dense and final layer. The result shows that a text with indirect emotion information (spoken text, expressive text from experience) can be recognized easier than plain text from news or headlines. For categorical and dimensional tasks, we found that the system performed well on a smaller number of categories than on larger categories. Not only the size of data matters but also the number of categories in data affect the performance of the categorical task. A small dataset with the small number of categories produced better performance than a bigger number of categories with an imbalanced dataset. On the bigger dataset, the number of categories decreases the performance significantly on categorical task. For the dimensional task, we found that the bigger data shows a smaller error of performance and it consistently occurs on two datasets. Finally, the same system can be used for both categorical and dimensional task by adjusting the properties of the final dense layer. A dimensional task also can be performed with a smaller number of (layer) units than the number on categorical task without any significant performance degradation. For the future works, a combination of text modality with another, such as speech, can be performed to evaluate the performance improvement as human being also perceives emotion from multimodal perception.

References

[1] Scherer, Klaus R. "Psychological models of emotion." The neuropsychology of emotion 137, no. 3 (2000): 137-162.
[2] Ekman, Paul. "Are there basic emotions?." (1992): 550.
[3] Jurafsky, Dan, and James H. Martin. Speech and language processing. Vol. 3. London: Pearson, 2014.
[4] Mäntylä, Mika, Bram Adams, Giuseppe Destefanis, Daniel Graziotin, and Marco Ortu. "Mining valence, arousal, and dominance: possibilities for detecting burnout and productivity?." In Proceedings of the 13th International Conference on Mining Software Repositories, pp. 247-258. ACM, 2016.

[5] Strapparava, Carlo, and Rada Mihalcea. "Semeval-2007 task 14: Affective text." In Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), pp. 70-74. 2007.
[6] Buechel, Sven, and Udo Hahn. "EMOBANK: Studying the impact of annotation perspective and representation format on dimensional emotion analysis." In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 578-585. 2017.
[7] Quan, Changqin, and Fuji Ren. "An exploration of features for recognizing word emotion." In Proceedings of the 23rd International Conference on Computational Linguistics, pp. 922-930. Association for Computational Linguistics, 2010.
[8] Strapparava, Carlo, and Rada Mihalcea. "Learning to identify emotions in text." In Proceedings of the 2008 ACM symposium on Applied computing, pp. 1556-1560. ACM, 2008.
[9] Strapparava, Carlo, and Alessandro Valitutti. "Wordnet affect: an affective extension of wordnet." In Lrec, vol. 4, no. 1083-1086, p. 40. 2004.
[10] Kim, Sunghwan Mac, Alessandro Valitutti, and Rafael A. Calvo. "Evaluation of unsupervised emotion models to textual affect recognition." In Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, pp. 62-70. Association for Computational Linguistics, 2010.
[11] Bradley, Margaret M., and Peter J. Lang. Affective norms for English words (ANEW): Instruction manual and affective ratings. Vol. 30, no. 1. Technical report C-1, the center for research in psychophysiology, University of Florida, 1999.
[12] De Myttenaere, Arnaud, Boris Golden, Bndicte Le Grand, and Fabrice Rossi. "Mean absolute percentage error for regression models." Neurocomputing 192 (2016): 38-48.
[13] Johnson, Justin M., and Taghi M. Khoshgoftaar. "Survey on deep learning with class imbalance." Journal of Big Data 6, no. 1 (2019): 27.
[14] Razek, Mohammed Abdel, and Claude Frasson. "Text-Based Intelligent Learning Emotion System." Journal of Intelligent Learning Systems and Applications 9, no. 01 (2017): 17.
[15] Zhang, Lei, Shuai Wang, and Bing Liu. "Deep learning for sentiment analysis: A survey." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8, no. 4 (2018): e1253.
[16] Mulcrone, Kaitlyn. "Detecting emotion in text." University of Minnesota—Morris CS Senior Seminar Paper (2012).
[17] Calvo, Rafael A., and Sunghwan Mac Kim. "Emotions in text: dimensional and categorical models." Computational Intelligence 29, no. 3 (2013): 527-543.
[18] Bakker, Iris, Theo van der Voordt, Peter Vink, and Jan de Boon. "Pleasure, arousal, dominance: Mehrabian and Russell revisited." Current Psychology 33, no. 3 (2014): 405-421.
[19] Busso, Carlos, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. "IEMOCAP: Interactive emotional dyadic motion capture database." Language resources and evaluation 42, no. 4 (2008): 335.
[20] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.
[21] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In Advances in neural information processing systems, pp. 3111-3119. 2013.
[22] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014.
[23] Mikolov, Tomas, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. "Advances in pre-training distributed word representations." arXiv preprint arXiv:1712.09405 (2017).
[24] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).
[25] Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).
[26] Aston Zhang and Zachary C. Lipton and Mu Li and Alexander J. Smola. "Dive into Deep Learning. " (2019).
[27] Chollet, Francois. "Keras: The python deep learning library." Astrophysics Source Code Library (2018).
[28] Balahur, Alexandra, Jesus M. Hermida, and Andres Montoyo. "Building and exploiting emotinet, a knowledge base for emotion detection based on the appraisal theory model." IEEE transactions on affective computing 3, no. 1 (2011): 88-101.
[29] Gal, Yarin, and Zoubin Ghahramani. "A theoretically grounded application of dropout in recurrent neural networks." In Advances in neural information processing systems, pp. 1019-1027. 2016.