Natural Language Processing Technologies for Sentiment Analysis: the Words We Used to Describe Death

Author: Zhiwen GUO
Supervisor: Dr Trent Lewis

Wednesday 9th December, 2020

Submitted to the College of Science and Engineering in partial fulfilment of the requirements for the degree of Master of Science (Computer Science) at Flinders University - Adelaide Australia.

Abstract

With the population ages, global climate changes, pandemic spreads, other chronic diseases are increasingly resulting in people's death, the trend in deaths rate increasing may be unavoidable. However, death and dying are taboo in contemporary western societies. Under this circumstance, people may not have the ability to face death when they are exposed to death or dying. For this reason, a study on death and dying called Dying2Learn (Tieman et al. 2018) has been proposed. The principal purpose of the Dying2Learn research is to have a deeper look at the public's views on the topic of death and dying so that to improve the people's death competence (the capabilities and attitudes to deal with death). The Dying2Learn is a survey-based study, applied linguistic sentiment methodology to activities held on Massive Open Online Courses (MOOCs). These courses are designed for encouraging the public to discuss dying and death, so many participants of the courses made comments on dying or death. All the discussions have been collected as study data. However, as the Dying2Learn MOOCs are running year after year, a large amount of text-based data are available, which make it a heavy workload for researchers to analyse. Let alone maintain the analysis consistency. Therefore, efficient and accurate solutions have been called for analysing all these data.

Based on this background, the present research focused on investigating different natural language processing technologies and developing an efficient and accurate solution for Dying2Learn research. After reviewing different baseline algorithms for sentiment analysis such as Native Bayes classifier, SVMs classifiers and lexicon-based algorithms, we explored a widely used algorithm for sentiment analysis, which is the combination of Global Vectors for Word Representation (Pennington et al. 2014) with Long short-term memory (Hochreiter & Schmidhuber 1997) network. Also, we proposed two novel methods that make use of Global Vectors for Word Representation (Pennington et al. 2014), the Universal Sentence Encoder (Cer et al. 2018) and Warriner et al. (2013)'s affective lexicon.

The main contribution of this present research is that a novel solution has been proposed for Dying2Learn research. The proposed method can be used as a black box which only requires the data from Dying2Learn research as inputs; it automatically analyses all the data and outputs the results. So, the proposed method meet the objectives of this present study that develop an efficient and accurate solution.

Acknowledgments

I would like to express my appreciations and gratefulness to my thesis supervisor, Dr Trent Lewis, for allowing me to work on the project Dying2Learn, where I can apply my knowledge learnt from lectures to a realistic project. His continuous support and pieces of advice inspired me a lot.

I would like to give special thanks to Prof. David M W Powers for inspiring me and broadening my horizons so that I had many different ideas about the project.

Finally, I would like to thank my parents and my family for all of their support and understanding, and the many sacrifices they have made to let me continue pursuing my research.

Contents

List of Figures

List of Tables

1 Introduction

People's lifespan and awareness of health are continuously increasing (Zuo et al. 2018). Although this leads people to live longer and better than ever before, the trend in death rate increasing may be unavoidable (Commission 2013) as the population ages, the global climate changes, pandemic spreads, other chronic diseases are increasingly resulting in people's death. However, death and dying are taboo in contemporary western societies. People avoid talking about death and dying (McIlfatrick et al. 2013, Gellie et al. 2014), let alone prepare for death or manage this part of life. Under this circumstance, people may not have the ability to face death when they are exposed to death or dying (Balk et al. 2007, Fonseca & Testoni 2012, Gellie et al. 2014).

Massive Open Online Courses (MOOCs) provide a unique opportunity for people to discuss and learn more about death. Being able to talk about death and dying openly can help individuals and communities with the outcomes of their health and make decisions on care options. In addition, online learning has been employed in palliative care (Hughes et al. 2016) and used as a strategy to support awareness and use of the CareSearch evidence resources in the CareSearch project. In this background, a six-week Massive Open Online Course (MOOC), Dying2Learn, was designed as a community platform for people to discuss death, dying, and palliative care. It aims to enable participants to talk and discover about concerns around living, death and dying openly and supportively, and to exchange unheard ideas and views around death and dying (Tieman et al. 2018, Miller-Lewis et al. 2020), trying to increase the participants' abilities to deal with death.

Also, the Dying2Learn MOOC offers researchers an opportunity to have a closer look if online conversations about death and dying can influence death competence. According to Robbins (1994), death competence refers to the capabilities and attitudes people have to deal with death. In the six-weeks MOOC, there were different activities in different modules. By analysing the data from these activities, researchers tried to identify the changes in death competence during the MOOC, determining the contribution of the MOOC to community discussions around death and dying (Tieman et al. 2018).

However, one restriction with the research is the survey-based measurement maybe not suitable for text-based description. Initially, researchers used a five-point Likert scale which includes 'strongly disagree', 'disagree', 'not sure', 'agree' and 'strongly agree' to measure what degrees the participants agree with statements of death attitudes and MOOC satisfaction. After that, they used statistical approaches to analyse these data, determining the contributions of the MOOC to death competence (Tieman et al. 2018). Nevertheless, there are still large amounts of unstructured, text-based information about death and dying available from the MOOC which the study was done by Tieman et al. (2018) did not examine. As mentioned before, many different activities were in different modules of the MOOC,

and many of them are text-based-description activities. For example, one of these activities was the "3 Words to describe feelings about death" (3words activity), which asked participants to use three words to describe their feelings about death in the introduction module and the final reflections module. Although these activities were designed to encourage participants to use emotional words related to death, how to measure the sentiment of the words or comments was still a problem. Moreover, as the MOOC runs year after year, the volume of the data may increase rapidly, and to maintain consistency can be a heavy workload for researchers to analyse all the data. Therefore, efficient and accurate solutions have been called.

Nature language processing (NLP) technologies may help with processing and analysing those large amounts of text-based data from the MOOC. According to Collobert et al. (2011), NLP refers to the process which extracts simpler representations from complex text that can describe limited aspects of the textual information and motivated by specific applications. NLP aims to develop algorithms for computers to 'understand' natural languages and solving practical problems involving languages (Manning et al. 2017). There are different tasks in NLP, from speech recognition to semantic interpretation and discourse processing. For the Dying2Learn MOOC, since some activities asked participants to use words to describe their feelings, some others asked participants to make comments on the topic death and dying, so the data from those activities can be divided into two categories: words and sentences. Furthermore, one of the objectives of the MOOC is to 'understand' the words and the comments participants used in those activities, which is also one of the aims of NLP. So, some NLP technologies should be solutions for Dying2Learn research.

Many applications of sentiment analysis have proved that NLP technologies can be the solutions for Dying2Learn research. Sentiment analysis is also known as opinion mining. The word 'sentiment' was firstly used in the papers by Das & Chan (2001) and Tong (2001). It refers to the automatic analysis of evaluative text and tracking of predictive judgments. After that, a sizeable number of papers used the term 'sentiment analysis' to mean the specific application of studying affective states and subjective information using natural language processing. In general, sentiment analysis investigates at three levels which are document level, sentence level, entity and aspect level (Liu 2012). The opinions from others are always important for us during decisions making. This is the reason why sentiment analysis is one of the high demanded topics in Artificial Intelligence in recent years; and sentiment analysis has been employed in many applications like review-related websites, as a sub-component technology, or being in business and government intelligence (Pang & Lee 2008). To our knowledge, although no study applied sentiment analysis for the topic of death and dying, many studies used sentiment analysis on social media for products reviews (Cabral & Hortacsu 2010), predicting the stock market (Bollen et al. 2011) and political sentiment (Laver et al. 2003,

Mullen & Malouf 2006). Besides, one task of Dying2Learn MOOC is similar to the applications mentioned above, classifying the sentiment of words and comments. It is reasonable to believe that sentiment analysis can be applied to Dying2Learn research.

The present research aims at exploring different natural language processing technologies to propose a useful and accurate solution for the Dying2Learn MOOCs. Because a large amount of text-based data for the MOOCs have not been examined, an automatic, accurate and efficient solution would help Dying2Learn research, and the solution can also be applied to other sentiment analysis related project.

The present research will discuss and review the related works of sentiment analysis in the Literature Review section. After that, the datasets, the proposed methods and the evaluations for different proposed methods will be presented in the Experiments section. The analysis of the results from different proposed methods will be shown in the Results and Analysis section. The discussion and the future work of this research will be presented in the Discussion section. The conclusion section will summarise this research.

2 Literature Review

2.1 The difficulties of sentiment analysis

Tasks in NLP come in varying levels of difficulty; sentiment analysis is inherently at the hard level. The difficulties of NLP tasks can be divided into three levels. For example, the spell checking; keyword search and finding synonyms are at the easy level, parsing information from documents is at the medium level, while Machine Translation, semantic analysis, coreference, question answering are at the hard level (Manning et al. 2017). Sentiment analysis is at the hard level since it requires more 'understanding'. One challenge in sentiment analysis is that sentiment can be expressed in a latent way (Pang et al. 2002). For example, the sentence "What a day!" contains no word expressing any sentiment, but this sentence can be positive or negative by context. The other difficulty in sentiment analysis is order dependence. For example, the sentence

> "Removed due to copyright restriction. Original quote can be viewed in (Pang & Lee 2008)"

includes three positive words and one positive phrase, but the sentiment of this sentence is decided by the last part "However, it can't hold up" to be negative (Pang & Lee 2008). These difficulties require the algorithms for sentiment analysis can capture the semantic (connected with the meanings of words) and syntactic (relating to the grammatical arrangement of words in a sentence) information to 'understand' the sentiment of the sentences. However, many machine learning algorithms may not help with the semantic and syntactic information, which may be

the reason why Chomsky (1965, 1969) claimed that machine learning algorithms give no insight into the syntax. He derided researchers in machine learning use purely statistical methods to mimic the behaviour in the real world when he attended at the Brains, Minds, and Machines symposium held during MIT's 150th birthday party (Cass 2011).

2.2 Baseline algorithms for sentiment analysis

2.2.1 Naive Bayes

Naive Bayes is a probabilistic classifier, which is a baseline algorithm for sentiment analysis (Pang & Lee 2008). It is one of those supervised machine learning algorithms that use purely statistical methods to mimic the behaviour in realities. For a sentence S, out of all classes of sentiment $c \in C$, the classifier returns the class C_i which has the maximum posterior probability given the sentence (Jurafsky & Martin 2014). The Naive Bayes classifier so-called because it is constricted by the naive Bayes assumption, which assumes that given a positive (or negative) sentence, the probability of words occurs in that sentence is independent. It means the Naive Bayes classifiers does not consider the dependence in the sentence and may not able to capture the syntactic information. Despite this restriction, the Naive Bayes classifier is widely used as a baseline algorithm for sentiment analysis (Pang et al. 2002, Dave et al. 2003, Airoldi et al. 2004, Gamon 2004, Mullen & Collier 2004, Matsumoto et al. 2005); indeed, the Naive Bayes classifier performs well for certain datasets; Yu & Hatzivassiloglou (2003) achieved high accuracy by using Naive Bayes classifier to separate facts from opinions and identifying the polarity of opinion sentences. Because of two issues, this algorithm may not be suitable for the Dying2Learning. As mention before, it cannot capture the syntactic information, which means it may not be able to identify the sentiment of sentences. The biggest problem with this approach is that it cannot deal with words, let alone the unlabelled data from the Dying2Learn.

2.2.2 Support Vector Machines

Like Naive Bayes classifiers, support vector machines (SVMs) are in supervised machine learning algorithm category and widely used as baseline algorithm for sentiment analysis (Pang et al. 2002, Dave et al. 2003, Airoldi et al. 2004, Gamon 2004, Mullen & Collier 2004, Matsumoto et al. 2005). SVMs are usually used for classification. In traditional text classification tasks, SVMs usually outperform Naive Bayes classifiers (Lewis 1998). The basic mechanism behind SVMs is to minimise the loss function, finding a hyperplane which not only can separate data points in one class from those in the other, but also for which the separation, or margin, is as large as possible; this is why SVMs are also known as large margin classifiers. SVMs have been applied to sentiment analysis by many researchers. Tanesab

et al. (2017) use SVMs to analyse the sentiment fo comments from Youtube. Huq et al. (2017) use SVMs to conduct sentiment analysis on Twitter data. Usually, SVMs need to work with feature extraction procedure which is the process of transforming the input data into a set of features by term-weighting schemes (like term frequency-inverse document frequency) or affective lexicon (Zainuddin & Selamat 2014, Tanesab et al. 2017). The basic idea of term-weighting schemes is to extract the most important features as an input to the classifier by calculating frequency, while affective lexicon can be used to identify the sentiment of words in the sentence determining the sentiment of the sentence. Although the SVMs can work with words and sentences, it still cannot capture the semantic and syntactic information based on its feature extraction mechanism; in addition, because SVM is a supervised machine learning algorithm, but the data from Dying2Learning is unlabelled, so it may not be a suitable solution.

2.2.3 Lexicon-based algorithms

One restriction of these supervised algorithms mentioned previously is that they both trained with labelled data, so Bakshi et al. (2016) used an unsupervised algorithm to analyse tweets. It can be categorised as a lexicon-based algorithm. The basic idea of this algorithm is to create a dictionary by assigning polarities to words; 1 for positive, -1 for negative and 0 for neutral. According to the created dictionary, the algorithm will assign correspond score to a word only if that word is in the dictionary. The algorithm determines the sentiment of a sentence by assigning scores each word in a sentence and summing all the scores. If the result is bigger than 1, the sentence will be classified as positive, while the result is small than -1, the sentence will be labelled as negative; the sentence will be determined as neutral if the result equals to 0. However, there are two problems in this method. The first one is that the lexicon created manually can be unreliable. The study done by Pang et al. (2002) shows that human-produced lists of sentiment indicator words might be less trivial than one might initially think. In that study, Pang et al. (2002) asked two graduate students in computer science to chose keywords that they would consider to be good indicator words of sentiment in movie reviews. The result showed that statistics-based word list achieved higher accuracy than human-produced lists. In fact, researchers do not need to create a dictionary by themselves, since there are some large-scale affective lexicons have been developed. The second drawback of this algorithm is that it still cannot deal with the semantic and syntactic information contained in a sentence. It still cannot correctly classier the example sentence mentioned before. With this algorithm, that example sentence will be classified as positive as there are four positive words but only one negative word in the sentence.

Bakshi et al. (2016) are not the only researchers that have applied lexicon-based algorithms for sentiment analysis, and they are not necessary to create a dictionary manually since a few large-scale sentiment norms have been developed for the high

demand in emotional ratings of words. Leveau et al. (2012) developed a computer program to identify the sentiment of texts on the basis of lexicons. The lexicon they used is Affective Norms for English Words (ANEW) (Bradley & Lang 1999). These norms use three dimensions, which are in line with Osgood et al. (1957)'s theory of emotions, to measure the emotional ratings for 1, 034 words. The first dimension, the most important one, concerns the valence (the pleasantness of a stimulus) of the emotion invoked by a word, going from 'unhappy' to 'happy'. The second dimension address the degree of arousal (the intensity of emotion provoked by a stimulus) evoked by a word, rating from 'calm' to 'excited'. The third dimension refers to dominance (or power, the degree of control exerted by a stimulus) of a word, denoting something that is weak / submissive or strong / dominant (Warriner et al. 2013). However, the number of words covered by the ANEW may not be sufficient for large-scale factorial experiments. Although there is another tool frequently used for studying text called Linguistic Inquiry and Word Count (LIWC) (Pennebaker et al. 2001), it only works in document level and identifies the sentiment of text by using discrete labels like positive and negative. Because the emotional ratings of words are in high demand, Warriner et al. (2013) based on the ANEW developed norms to rate the majority of the well-known 13, 915 English lemmas (the base forms of words. i.e., the ones used as entries in dictionaries). Warriner's lexicon (Warriner et al. 2013) is also in line with Osgood et al. (1957)'s theory of emotions, using continuous ratings ranging from 1-9 with valence, arousal, and dominance three dimensions to measure the sentiment of a word, which provides researchers with an advanced lexicon to reference. Although lexicons can be directly used to measure the sentiment of words from Dying2Learn, they may not be the best choice. The first reason is, even the Warriner et al. (2013)'s lexicon extended the ANEW norms to nearly 14, 000 English lemmas, some words like 'Ok', 'Jesus' and 'when' are still not in the lexicon. The second reason is that lexicons cannot be used to identify the sentiment of sentences directly, and as mentioned before, the algorithm used by Bakshi et al. (2016) cannot capture the syntactic information. So only using affective lexicon for Dying2Learn research may not be the suitable approaches.

2.3 Word Embeddings

In the NLP field, word embeddings are used to capture semantic and syntactic information. In many previous NLP works, words were treated as atomic units, which act like indices in dictionaries and cannot indicate the similarity between words (Mikolov, Chen, Corrado & Dean 2013). In reality, words in a language may have some relations like synonym, antonym, and hypernym (a word whose meaning includes a group of other words). The objective of the word embeddings is to encode word tokens into some vectors which can represent points in some N-dimensional 'word' space that the relationships between words can be captured. It means these

dimensions can encode all semantics of human languages to indicate semantic information like tense (past or present or future), count (singular or plural), and gender (masculine or feminine). Using syntactic analogy questions, Mikolov, Yih & Zweig (2013) demonstrated that the word embeddings capture syntactic regularities. For example, the analogy "the king is to queen as man is to woman" can be encoded as simple algebraic operations performed on word vectors as vector ('King') - vector ('Man') + vector ('Woman'), results in the word vector of the word 'Queen'.

2.3.1 Word2vec algorithm

Word2vec (Mikolov, Chen, Corrado & Dean 2013) is one of word embedding algorithms. Word2vec algorithms are based on distribution semantic theory and utilise neural networks to obtain word embeddings. The idea behind Word2vec algorithms is distribution similarity, which is a fundamental hypothesis in linguistics that similar words have similar context. The context of a word refers to the set of words surrounding that word in a corpus. In 1957, Firth (1957) summarised this hypothesis as: "You shall know a word by the company it keeps." Word2vec algorithms contain two different algorithms: continuous bag-of-words (CBOW) and skip-gram. These two algorithms belong to the unsupervised learning algorithm category. The way these two algorithms make use of neural networks is to leverage neural networks whose parameters are the word vectors to perform some certain tasks. Therefore, the CBOW model aims to predict a centre word given the surrounding context in terms of word vectors, while skip-gram model predicts the distribution (probabilities) of context words given a centre word. These two models make use of the similarity of the word vectors for centre words and context words to calculate the probability of context words given a centre word (or vice versa). Because the word vectors decide the probability, the algorithms can get the word embeddings, which are the parameters of the model mentioned before, by using the backpropagation algorithm (Rumelhart et al. 1986) to penalise the model parameters that caused the error to minimise the loss function (the loss function is used to calculate the error between predicted probability and the real probability (Rong 2014).

2.3.2 GloVE algorithm

Global Vectors for Word Representation (GloVE) algorithm (Pennington et al. 2014) is another word embedding algorithm. Comparing to Word2vec, GloVe algorithm makes use of global co-occurrence statistics. Pennington et al. (2014) proposed that the statistics of word occurrences in a corpus is the primary source of information for learning word representations; with the ratio of words co-occurrence probabilities, certain aspects of meaning can be extracted. For example, suppose we are examining the relationship between the word 'ice' and 'steam' in a big corpus, and we have another four words, 'solid', 'gas', 'water' and 'fashion', in that corpus. Since 'solid' often appear in the context of the word 'ice' but rarely occurs in the

context of the word 'steam', so the probability of 'solid' appears in the context of 'ice', P (solid | ice), is big, while the probability of 'solid' appears in the context of 'steam', P (solid | steam), is small. Accordingly, the ratio of these two probabilities, P (solid | ice) / P (solid | steam), will be big, which indicates the 'solid' more related to the word 'ice' than the word 'steam'. Therefore, the loss function of the GloVe algorithm is not only decided by word vectors but also takes into account the words co-occurrence counts in its loss function. The method for the GloVe algorithm to learn word vectors is similar to that for Word2vec algorithm, which is to minimise the loss function using the Backpropagation algorithm. This is how the GloVe makes use of global co-occurrence statistics to produce a vector space with meaningful sub-structure. The GloVe algorithm also outperforms other word embeddings algorithms on different NLP tasks like analogy task, similarity tasks and named entity recognition (Pennington et al. 2014).

2.4 Sequence Models

2.4.1 Recurrent Neural Networks

Although word embeddings can capture the semantic information of single words, it still cannot capture the meaning of long phrases or sentences. Therefore, sequence models usually work with word embeddings to help with the syntactic and the meaning of sentences. The recurrent neural network, or RNN (Rumelhart et al. 1986), is one of the sequence models specialised for sequential data such as a sentence, a piece of music. RNNs are constructed by hidden layers which each represent a time-step T; at each time-step T (each hidden layer), there are two inputs to the hidden layer: the output of the previous layer and the input as that time-step. Besides, all these hidden layers share the same parameters, which make it possible to extend and apply the model to words in different positions and generalise the model across the input features. Such parameter sharing is crucial for RNN to capture a particular piece of information that can occur in different positions in the sentence. For example, let us consider two sentences:

1. "The Word2vec algorithm was published in 2013."
2. "In 2013, the Word2vec algorithm was published."

If we want to extract the information relevant to the year 2013, RNNs can capture that information. This is because RNNs make use of the sharing parameters for different hidden layers to share statistical strength across different positions in the sentence (Goodfellow et al. 2016). Hence, RNNs are capable of conditioning the model on all previous words in the sentence.

However, there are two widely known issues making it difficult to train RNNs. They are the vanishing and the exploding gradient problems (Bengio et al. 1994, Pascanu et al. 2013). The reason for these two problems is gradient values during

the backpropagation phase. Since RNNs like other neural network trained by the backpropagation algorithm, but the gradients tend to either very small (vanishing problem) or extremely big (exploding problem) when propagated over many hidden layers. This is mainly because of the multiplication of many Jacobians, so the weights given to long-term interactions are exponentially smaller than the weights for short-term, which also leads to the difficulty for long-term dependencies. For instance, suppose we have two sentences:

1. "We walked to the lake near our home. Maryann walked there too. We said hi to _"
2. "We walked to the lake near our home. Maryann walked there at the same time that day. We just walked there after a long day work for relaxing. We said hi to _".

As mentioned before, RNNs should have the ability to conditioning the model on all previous words in the sentence and predict the word "Maryann" for the blank in these two sentences correctly. However, because of the gradient vanishing problem, the RNNs are more likely to predict that word correctly for the first sentence than the second one. One solution for the vanishing and the exploding gradient problems proposed by Hochreiter & Schmidhuber (1997) is the Long short-term memory neural network.

2.4.2 Long short-term memory

Long short-term memory neural networks, or LSTM (Hochreiter & Schmidhuber 1997), is a variant of RNNs, which is capable of learning long-term dependencies. The main idea behind LSTM is the self-connected units (or self-loops, cell state in different literature (Goodfellow et al. 2016)) and the gate units. The self-connected units produce a path with only some minor linear interactions where the gradient can flow for long durations. Those gate units are a way to let information through this path optionally, and they are made up of a sigmoid function and pointwise multiplication operations. The sigmoid function outputs numbers between zero and one, describing how much information (in this case, information refers to the gradient values) should be let through. If the sigmoid function outputs a value of zero, it means "let nothing through", while a value of one means "let everything through". By employing the self-connected units and gate units, LSTM can solve the gradient vanishing and exploding problems effectively. Therefore, LSTM outperforms RNNs and has been employed in different NLP tasks like machine translation (Sutskever et al. 2014) and parsing (Vinyals et al. 2015).

2.5 Word Embeddings and Sequence Models

Because word embeddings and sequence model can capture the semantic and syntactic information, so it can be used for sentiment analysis. Socher et al. (2013)

used word embeddings and Recursive Neural Tensor Network (RNTN), a variant of RNNs, to conduct sentiment analysis for movie reviews. They achieved 87.6% and 80.7% accuracy for binary sentiment classification(positive and negative) and fine-grained sentiment classification (5-class, negative, somewhat negative, neutral, positive and somewhat positive) at phrases level respectively. They also achieved that 85.4% and 45.7% accuracy for binary sentiment classification and fine-grained sentiment classification at the sentence level, which were state of the art at that time. The basic logic in this approach is to regard sentiment analysis as text classification tasks, making use of word embeddings to capture semantic and leveraging the sequence neural networks to take word order into account. This is the main reason why the combination of word embeddings and sequence model outperforms SVMs and other approaches for sentiment analysis discussed before (Socher et al. 2013). Hence, this approach can be the solution for Dying2Learning research. The only problem with this approach is that it is also a supervised learning algorithm. It needs to be trained with labelled data, but data from Dying2Learning research are unlabeled.

2.6 Sequence to Sequence Models

Besides the words embeddings and sequence models, there is another approach called sequence to sequence models, or "Seq2Seq" may be the solution for Dying2Learn research, since it can also be used for capturing the meaning of sentences. Sequence to sequence models was proposed by Cho, Van Merriënboer, Gulcehre, Bahdanau, Bougares, Schwenk & Bengio (2014) and shortly later by Sutskever et al. (2014) for machine translation, which is the task of translating a sentence x from one language (the source language) to a sentence y in another language (the target language). A Seq2Seq model is an end-to-end model made up of two sequence models: an encoder and a decoder. The encoder's job is to encode the input sequence (input sentence) word by word into a final fixed-dimensional "context vector" which can capture the meaning of the input sentence. The decoder based on this "context vector" to generate an output sequence, which is a sentence in the target language. Because of this mechanism, the decoder relies heavily on the final context vector to translate, but not all the necessary information can be compressed into a fixed-dimensional context vector, making it difficult for the model to deal with long sentences (Cho, Van Merriënboer, Bahdanau & Bengio 2014). Therefore, Bahdanau et al. (2014) proposed neural machine translation by jointly learning to align and translate.

2.7 Attention Mechanism

The approach of jointly learning to align and translate, also known as "Seq2Seq with attention mechanism" (Bahdanau et al. 2014), copes better with long sen-

tences. Unlike the basic Seq2Seq models, where the decoder only based on the fixed context vector to translate, Seq2Seq with attention models allows the decoder to search different positions in the source sentence, finding out where the most related information is concentrated. The model then based on the context vector, these source positions and the previously generated word to predict a word in the target language. The idea behind the attention mechanism is, instead of encoding the whole input sentence into a fixed size context vector, the model encodes the input sentence into a set of vectors and chooses a subset of these vectors adaptively when decoding the translation. This is the way attention mechanism free the Seq2Seq models from compressing all the information into a fixed-size vector which ignores the length of the input sentence.

2.8 Transformer models

The Transformer models (Vaswani et al. 2017) outperform the Seq2Seq with attention models. Vaswani et al. (2017) proposed that the attention mechanism is all we need. It means, instead of working with any sequence models, the Transformer models relying on attention mechanism. Similar to the Seq2Seq models, the Transformer models also have encoders and decoders in their architectures, but the mechanism behind is significantly different. When the model encoding each word (each position) in an input sequence, a sub-layer in encoders called self-attention allows the encoders to look at other position in the input sequence for clues that can help a better encoding for that word. The output of the encoders is a set of attention vectors which are used by the decoders to focus on appropriate places in the input sequence. The decoders also have self-attention layers, but they are slightly different from that in the encoders: the self-attention layers in decoders only allow the decoders to look at the earlier positions in the output sequence. To illustrate the self-attention mechanism, suppose we have a sentence:

"A cat did not jump over a dog because it was too tired".

Because of the self-attention mechanism, when the encoders encode the word 'cat' in the sentence, it will associate the word 'it' with 'cat' to indicate the word 'it' refers to the word 'cat'. Due to the self-attention mechanism, the encoder of Transformer models can encode the meaning of sentences into vectors more precisely. To our knowledge, although no one has used the Transform models for sentiment analysis, the encoders in this model can be a solution for the sentences from Dying2Learn research.

2.9 Universal Sentence Encoder

Because of the limitation of training data available for many NLP tasks, Cer et al. (2018) proposed pre-trained sentence encoding models, Universal Sentence Encoder

(the USE). The biggest challenge for deep learning is nearly all of the neural networks, including the Transformer models, are data-hungry. Besides, sentence embeddings are highly demanded in many NLP tasks. Pre-trained models for encoding sentences into vectors can be helpful in this perspective. The USE was designed to be as general-purpose as possible, so the models were trained by using multiple-task learning, whereby an encoder is used for multiple down-stream tasks. It means the USE can be used for different kinds of NLP tasks such as text classification, semantic similarity, clustering. Different from word embedding algorithms, which generate word embeddings for fixed vocabularies, the USE can encode sentences into vectors without any limitation on the number of sentences.

2.10 The gap this research will fill

Back to the sentiment analysis, as mentioned before, the difficulties of sentiment analysis require algorithms that can capture semantic and syntactic information. However, the baseline algorithms we reviewed before have different disadvantages. Naive Bayes classifiers and SVM classifiers cannot capture semantic and syntactic information. Although the lexicon-based approach can measure the sentiment of words very precisely, it is still limited to the number of lemmas the affective lexicons contain. In addition, the lexicon-based approach is worthless for sentences. The word embeddings, the sequence models and the Transformer models can capture the semantic and syntactic information, but, to our knowledge, the Transformer models only used for neural machine translation and only the combination of word embeddings and sequence models approach has been used to classify discrete sentiment (binary or five classes). Therefore, the present research will make use of different natural language processing technologies such as the combination of word embeddings and sequence models; GloVe pre-trained word embeddings and the Universal Sentence Encoder to propose a useful and accurate solution for the Dying2Learn MOOCs and fill the gap mentioned above.

3 Experiments

3.1 Datasets

The dataset used in this research is from 3words activity in the Dying2Learn MOOC 2017, which contains 1,985 entries, including the words participants used to describe their feelings about death in the introduction module (mw: N792) and the final reflections module (mwr: N410) of the MOOCs as well as the words participants used to describe their opinions towards other people's feelings of death in the introduction module (ow: N783). Because some of the participants used three sentences (or phases) instead of three words to describe their attitude, the data entries can

be into three categories, words, pharses and sentences. The details of the dataset shown in Table 1

Name	Module	Size	Lables
N792	Introduction	792	mw
N783	Introduction	783	ow
N410	Final reflections	410	mwr

Table 1: Details of the datasets from Dying2Learn MOOC 2017. The names of the datasets are based on the number of entries they contain. The labels in the table: 'mw' refers to the words used for describing participants personal feelings in introduction module, when 'mwr' refers the words used in final reflection module. 'ow' refers to the words used to describe how other people think about death and dying.

Two researchers labelled the data which initially came in without scores. Participants used words and sentences to describe their feelings about death, but this makes it hard for researchers to estimate the sentiment changes during the MOOC. Two researchers in the lab, Dr Lauren Miller-Lewis and Dr Trent Lewis, labelled the data. For the words from the dataset, they used Warriner et al. (2013)'s lexicon to measure the sentiment of words. If a word were absent from the lexicon, that word would be lemmatised first, and if the lemma of the word were still absent, the word would be stemmed and then recheck the lexicon. Lemmatisation is to convert words from any form to its lemma. For example, the word 'gone' will be reduced to its base-form 'go'. Stemming means take off the suffix of words. For instance, 'saves', 'saved', 'saving', and 'saver' after stemming will be 'sav'. Both lemmatising and stemming were conducted by using Stanford CoreNLP (Manning et al. 2014). If the lemma and stem of a word are still absent from Warriner et al. (2013)'s lexicon, that word is labelled as 'MISS'. For phrases and sentences in the dataset, researchers analysed the sentiment of them and then found out words which can summarise the meaning of them in the Warriner et al. (2013)'s lexicon manually. Some examples are shown in Table 2.

Original	Labels	Replacement	Scores		
			Valence	Arousal	Dominance
go	ORIG	-	6.32	4.86	5.33
gone	LEMMA	go	6.32	4.86	5.33
limitation	STEM	limit	4.53	4.29	4.37
not sure	REPLACE	unsure	3.37	4.55	3.64
ok	MISS	-			

Table 2: Examples of the dataset labelled by researchers. The labels 'ORIG', 'LEMMA', 'STEM', 'REPLACE' and 'MISS' refer to 'original words', 'lemmatised', 'stemming', 'manually replace' and 'missing', respectively.

Although using lexicon may have some limitations, but the results from the

researchers provide a reference to evaluate the methods proposed in this present research. One drawback of using lexicon is that some frequently used words like 'OK', 'Jesus', 'nothing' are not in the lexicon, and Lemmatising or stemming cannot help. This is the reason why these words were labelled 'MISS' in the dataset. Let alone the lemma or stem from lemmatising and stemming may not have the same sentiment of the original words from participants. However, the labelled data can be a source to reference for evaluating and comparing the performances of the proposed methods.

3.2 Methods

Three different methods will be explored. All these approaches will be applied to the data from Dying2Learn research, and their performances will be evaluated. These three different methods are:

1. Making use of the combination of GloVe pre-trained word embeddings (Pennington et al. 2014) and LSTM (Hochreiter & Schmidhuber 1997) with Warriner et al. (2013)'s affective lexicon to predict seven-class classification task which uses 0 to 6 for denoting the ratings of sentiment;
2. Leveraging the Glove pre-trained word embeddings and Warriner et al. (2013)'s affective lexicon to classify the sentiment of words from Dying2Learn research;
3. Using the USE (Cer et al. 2018) and Warriner et al. (2013)'s affective lexicon classify the sentiment of sentences from Dying2Learn research;

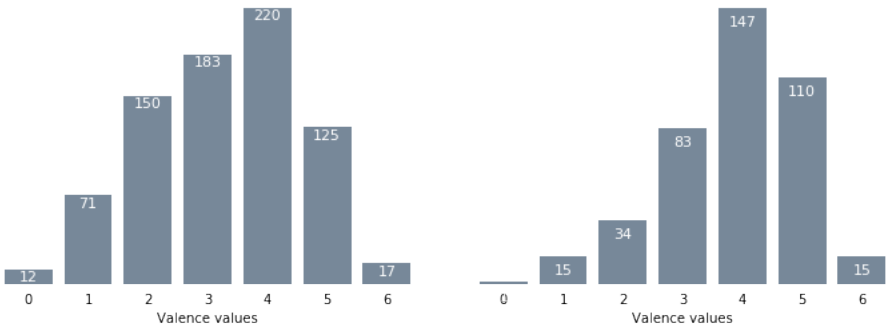
The details of these approaches will be represented in the following sections.

3.2.1 Method 1: GloVe embeddings with LSTM

As discussed in the literature review section, word embeddings encode the semantic information into word vectors, while sequence models can capture syntactic information. Besides, this approach can help with the difficulty that the limitation on words in the lexicon. Similar to the mechanism behind the word embeddings, by using part of the labelled dataset as training data to train the model, the model can obtain weight matrices associated with the hidden layers in the model which can map the inputs into another sub-space where can help sentiment classification. Another way to interpret this method is to regard the labelled data as seeds; the neural network learns the pattern of these seeds and then predict the sentiment of new inputs. Although sequence models are originally used for text classification on the sentence level, a word can be regarded as a concise sentence. Therefore, this approach can be applied for words and sentences from Dying2Learn.

Some preprocessing procedures for the data need to be carried out before training the model. The training data N792 used in this approach was collected from the introduction module of the MOOC 2017, which contains 792 entries of data.

Because the sentiment of each word in the lexicon was measured in three dimensions, valence, arousal, dominate. It means that each word in the lexicon has three scores for these three dimensions. However, these three dimensions scores for each word can be various. For example, the word 'able' has 6.64 in valence but 3.38 in arousal. Because Warriner et al. (2013) proposed valence is most important dimension to estimate people's sentiment. Therefore, in this project, we took the valence dimension as primary. Besides, as mentioned before, the 3words activity asked participants to use three words to describe their feelings. That is, each data entry for a participant has three words. Although this method can be used for sentence-level sentiment classification, the length of the sentences in the dataset are various. To examine the performance of this method, only the words from the dataset were used to train the models. For all these reasons, the first step of preprocessing was to concatenate the three words of each entry to be as input and average their scores in the valence dimension as the label for the input. Besides, Warriner et al. (2013)'s lexicon uses continuous values, but sequence models can only work on discrete values. Hence, it is necessary to round the scores into integers. The labels were then re-ranged from 1-9 to 0-6 (totally seven classes). The distribution of training set after preprocessing shown in Figure 1 (a)



(a) Training Set (N792) (b) Test Set (N410)

Figure 1: The distribution of (a) Training Set (N792), and (b) Test Set (N410) after the preprocessing procedure of method 1.

GloVe embeddings algorithm and LSTM outperform their counterparts, so both of them were employed in this research. The GloVe embeddings official website (Pennington et al. 2014) provides different pre-trained word embeddings. The biggest one contains 840 billion tokens, 2.2 million vocabularies, while the smallest one contains 6 billion tokens, 400 thousand words. Since the biggest one can be computationally expensive and the smallest still has good performance, so the smallest one used in this approach. It was trained on Wikipedia 2014 and Gigaword 5. The vectors for each word has 300 dimensions. The sequence model used in this approach was the LSTM network. The whole network was implemented in Keras, a deep learning API written in Python, with learning rate, epochs and batch size were

set to be 0.5, 100, and 32, respectively. For optimising the performance, different parameters settings were used for the models. Two LSTM layers and one LSTM layer were used to construct the network. For the same reason, different numbers of neurons were used in the layer of models. To prevent overfitting, the dropout was used and set to be 0.5, which means 50% of the neurons in a layer would be knocked out. Since the task is sentiment classification, the activation function used in this network was softmax. The details of the model shown in Table 3

Parameters	**Setting**
Layers	2, 1
Neurons	(128, 128), (128, 64), (64, 64) , (64, 10), 128, 64, 10, 5
Activation	softmax
Dropout	0.5
Optimizer	adam
Loss function	categorical crossentropy
Epochs	100
Batch size	32

Table 3: For optimising the performance of method 1, different parameters settings were used for the models.

3.2.2 Method 2: GloVe embeddings with Warriner's lexicon

The main idea of this approach is to use word embeddings to 'expand' the lexicon. Back to the method used by researchers, the two main disadvantages: some frequently used words are absent from Warriner et al. (2013)'s lexicon, the sentiment of lemmas and stems may be different from the original words. These two disadvantages can be overcome by computing word similarity.

As discussed in the literature review section before, word embeddings can be used to measure the similarity between words. The idea behind word embeddings is the meaning of a word given by the words that frequently appear close-by (Mikolov, Yih & Zweig 2013). Word embeddings can encode word tokens into some vectors which can represent points in some N-dimensional 'word' space that the relationships between words can be captured. In this word space, words have similar meanings their vectors will get together. Hence, we can obtain the word similarity by computing the cosine similarity (Equation 1), which is defined as a measure of similarity between two non-zero vectors of inner product space.

$$s = \frac{p \cdot q}{||p||||q||}, \text{ where } s \in [-1, 1] \tag{1}$$

The mechanism behind this approach is to convert all the words from the dataset and lexicon into vectors first. After that, retrieve the most similar words in the lexicon for the words in the dataset by computing cosine similarity between each

word vector of the dataset with all word vectors of the lexicon so that we can use the scores of the most similar words to measure the sentiment of the words absent from lexicon. The pipeline of method 2 is shown in Figure 2. For example, suppose we need to find the most similar word to substitute the word 'OK' which is not in the lexicon (the procedure is shown in Figure 3). We look up the corresponded GloVe pre-trained vectors for all the words in the lexicon. At the same time, we look up the corresponded GloVe pre-trained vector for the word 'OK'. Then, compute the cosine similarity between the vector of 'OK' and the vectors of the words from the lexicon. By sorting the cosine similarity values, we can obtain the most similar vector. After that, we convert that vector back to its corresponded word, which is the most similar word in the lexicon for the word 'OK'. In the end, we look up the most similar word scores in the lexicon. We can use these scores to measure the sentiment of the word 'ok'.

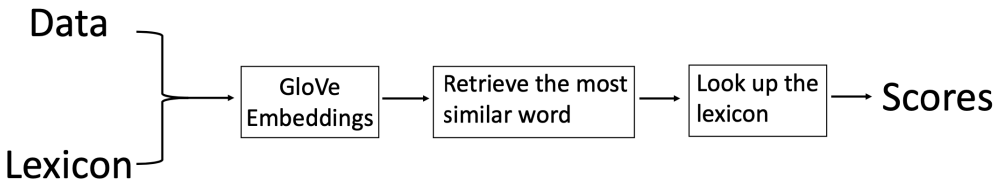


Figure 2: The pipeline of method 2

The most similar word in the lexicon for any word is the word itself if that word in the lexicon. So if the result of cosine similarity is a value of 1 means the word is in the lexicon, while it outputs a value of less 1, we sorted the similarity scores and retrieved the one with the highest score.

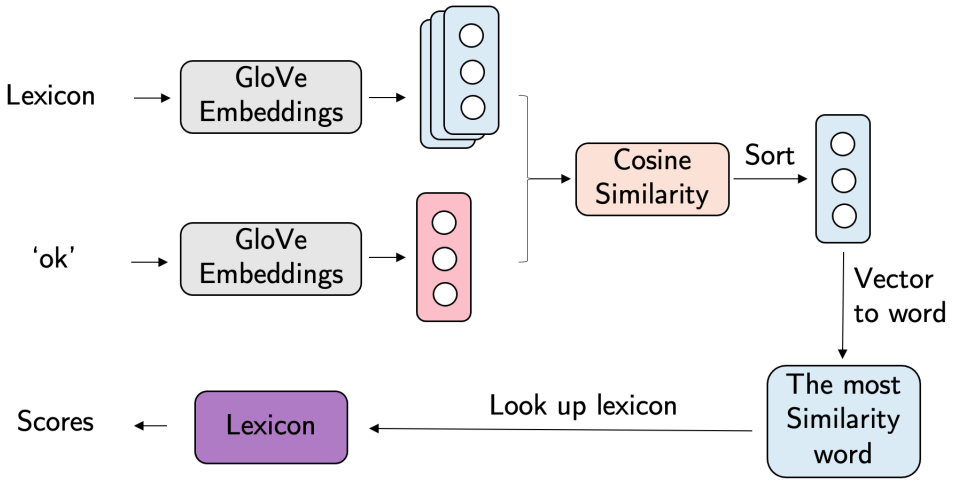


Figure 3: An example of applying method 2

There are two reasons why this approach can overcome the disadvantages of the lexicon-based method. The first reason is, by using the scores of the most similar words, we can still use Warriner et al. (2013)'s lexicon to measure the sentiment of the words that are not in the lexicon. The second reason is word embeddings encode

the semantic information into word vectors, so the cosine similarity of word vectors capture the semantic similarity between words, which guarantee the measurement of sentiment is relatively accurate.

Although the pre-trained GloVe embeddings are relatively robust, some pre-processing steps were still taken out. The word embeddings used in this approach was the same as that used in the word embeddings and sequence models. Because it only contains the most commonly used 400 thousand words, 31 lemmas in the Warriner et al. (2013)'s lexicon and 13 words from the dataset are not in these pre-trained word embeddings. After comparing the effect of computational expensive and 41 lemmas are missed, we adopted the smallest pre-trained embeddings in this approach. So the first preprocessing step took out was to eliminate the words absent from the GloVe pre-trained word embeddings. Another preprocessing step need to take out in this approach is to separate the words from the sentences in the dataset since this approach can only work on word level.

3.2.3 Method 3: The USE with Warriner's lexicon

The idea of this approach is the same as that of method 2, which is to make use of the semantic similarity and Warriner's affective lexicon, but the mechanism behind is slightly different. The biggest difference between method 2 and method 3 is the difference between the USE and the pre-trained word embeddings. As mention in the literature review section, the USE (Cer et al. 2018) is a pre-train model, while GloVe embeddings are pre-trained word vectors. It means that the USE can encode sentences into vectors without any limitation on the number of sentences, but the size of the pre-trained GloVe embeddings is fixed. Another difference is the USE is used for sentence-level, but GloVe embeddings are working on words level only. In fact, the USE is trained and optimised for greater-than-word length text, such as sentences, phrases or short paragraphs.

Because of the differences in the mechanism between the USE pre-trained models and the GloVe pre-trained embeddings, this approach overcomes the disadvantages of method 2. Since a word can be regarded as a short sentence, so the USE can be used for words as well while GloVe embeddings only work for words. As mentioned in literature review section before, by using transfer learning, which means a single model is used to feed various downstream tasks, the USE was trained on a diversity of data sources and a variety of tasks to accommodate a wide variety of natural language understanding tasks dynamically. The input is variable-length English text, and the output is a 512-dimensional vector (Cer et al. 2018), So the USE models can generate vectors for any new input, which means any uncommon word, any phrase, sentence, or text can be encoded into a high dimensional vector. Method 3 makes use of this mechanism, paraphrasing a phrase or a sentence into a word. The Transformer is used initially to translate sentences in one language to another, while in this method, we used the USE, which adopts the encoder of

the Transformer models, to "translate" sentences into words. The pipeline of this method was nearly the same as that of method 2, which is to convert all the words from the dataset and lexicon into vectors and make use of cosine similarity (Equation 1) so that we can use the scores of the most similar words measuring the words which are absent from the lexicon. The only difference in this method is that the USE was used to generate vectors instead of pre-trained GloVe embeddings. The pipeline of method 3 shown in Figure 4.

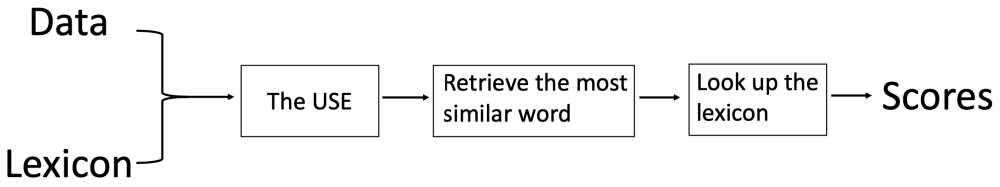


Figure 4: The pipeline of method 3

Suppose we use method 3 to find out the substitution for the word 'OK'. The procedure of method 3 is almost the same as that of method 2. The only difference in the procedure is that method 3 makes use of the USE to encode the words in lexicon into vectors and encode the word 'OK' into a vector. The procedure is shown in Figure 5.

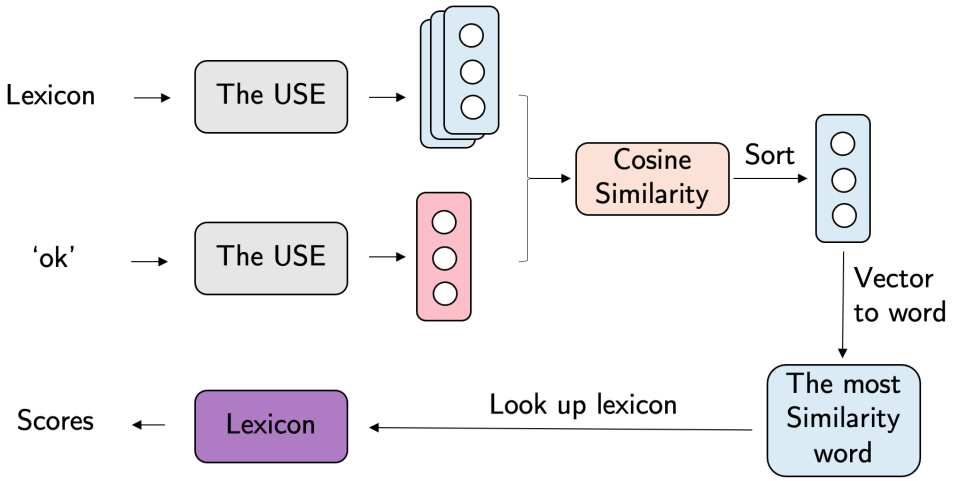


Figure 5: An example of applying method 3

The USE used in this approach was trained with a Transformer encoder. There are two different architectures in the USE. One makes use of the Transformer structure, while the other leverage deep averaging network, also known as DAN (Iyyer et al. 2015). Due to the one based on the transformer architecture outperforms the one based on DAN in terms of accuracy, so this approach makes use of the Transformer based USE. The USE pre-trained model is robust so that there is no need for preprocessing before applying the module.

3.3 Evaluation methods

The evaluation methods for these three methods were different, given the fact that they are different paradigms. Method 1, GloVe embeddings and LSTM, is based on word embeddings and sequence models, while method two and method three can be categorised as lexicon-based methods.

For method 1, as different parameters settings were used to find out the optimal model, Stratified 10-Fold Cross-Validation and T-Test were used to estimate the performance of different models and identify if their performances are significant differences. Since the training set (N792) is imbalanced, the performance comparison was based on the informedness of the models. To evaluate the optimal model, we used the dataset N410, which was collected in the final reflection module of the MOOC 2017, as a test set. Also, the confusion matrix was used to have a closer look at the performance of method 1. The confusion matrix is usually used in the evaluation of different classifiers. It can illustrate if the system is confusing different classes. In this research, the rows of the matrix denote the instances of actual classes while the columns represent the instances in predicted classes. The test set contained 410 entries of data and was preprocessed as the procedure as that of the training set used in method 1.

For the second and the third method, we referenced to the results from two researchers as mentioned in the data section before, and used cosine similarity and plots to evaluate the performances of these two methods. The results from these two approaches are the scores of the most similar words (the most similar words can be the word itself if that word in the lexicon), and we have the results from two researchers. Besides, each word in these two outcomes is measured by three scores in three dimensions, valence, arousal, dominance, which each dimension can be regarded as a high dimensional vector. By computing cosine similarity between the dimension vector of the results from these two methods with dimension vectors of the results from two researchers (valence to valence, arousal to arousal, dominance to dominance), we can obtain the similarity between them to measure the performances of the proposed methods. If the value of the cosine similarity between two outcomes is one, it means the results of the proposed methods are the same as the results obtained by researchers in the Dying2Learn research. Other than that, the words chosen by algorithms were compared to the words chosen by researchers to have a closer look at the performances of these two algorithms.

4 Results and Analysis

4.1 Method 1

From the results shown in Table 4, there was no significant difference between the models with different parameters settings. In terms of informedness, although

models with two layers had higher informedness mean, the results of T-Test showed there was no significant difference between the models. In this case, we chose the model with two LSTM layers each contains 128 neurons as a representative of Method 1.

Two layers	**(128, 128)**	**(128, 64)**	**(64, 64)**	**(64, 10)**
Mean	**0.6511**	0.6304	0.6477	0.606
SD	0.0994	0.1043	0.0696	0.0971
T-Test	There is no significant different			
One layer	**128**	**64**	**10**	**5**
Mean	0.5524	0.5861	0.5962	0.5752
SD	0.1036	0.1102	0.0937	0.0904
T-Test	There is no significant different			

Table 4: Informedness of different models

Although the capacities of different setting networks are different, LSTM maybe not suitable for this case. From the mathematical perspective, the weights associated with a hidden layer is a matrix; inputs of a neural network are vectorised as a vector. The matrix (weight) is used for mapping the vector (inputs) into another space in which the features of inputs can be separated. In terms of layers, the capacity of a network increases if the network with more layers. That is, that space grows since the neurons can collaborate to express many different functions. In terms of the size of hidden layers, it is the dimensions of the vector after the mapping. It means the bigger the size; the more differences of features can be captured. For the classification task, the performance should be better. However, back to this case, the use of LSTM may be a reason for the no difference between the performance of models with different parameter settings. The LSTM was used to capture the semantic and syntactic information for sentences. Although words can be regarded as short sentences, the three words data which were concatenated as inputs may not have any dependency, let alone if these three words express similar emotions. It means that the LSTM may not be able to the semantic and syntactic information. Therefore, LSTM may not be suitable for this case, which led to no significant difference between the models.

The model was overfitting. Overfitting refers to that the network performs well on the training set but cannot generalise to the test set. From the results shown in Table 5, the model got high informedness on the training set, but only 0.64 informedness on the test set, which means the obtained model was overfitting. A reason led to overfitting can be training set errors. As mentioned before, the training set N792 is imbalanced, which means even in the smallest net, overfitting can occur.

From the results of the confusion matrix (shown in Figure 6), the performance of the model was not bad. Except for class '1', class '2' and class '3', the deviation

Dataset	Informedness
Traning set	1.0
Test set	0.64

Table 5: The informedness of the model used in this research

of the predictions for each class is not more than one. Given the fact that the scores in Warriner et al. (2013)'s lexicon are continuous ratings and sentiments are not discrete, but all of the scores were rounded and averaged in the preprocessing stage of method 1, the deviation of the predictions for each class is reasonable.

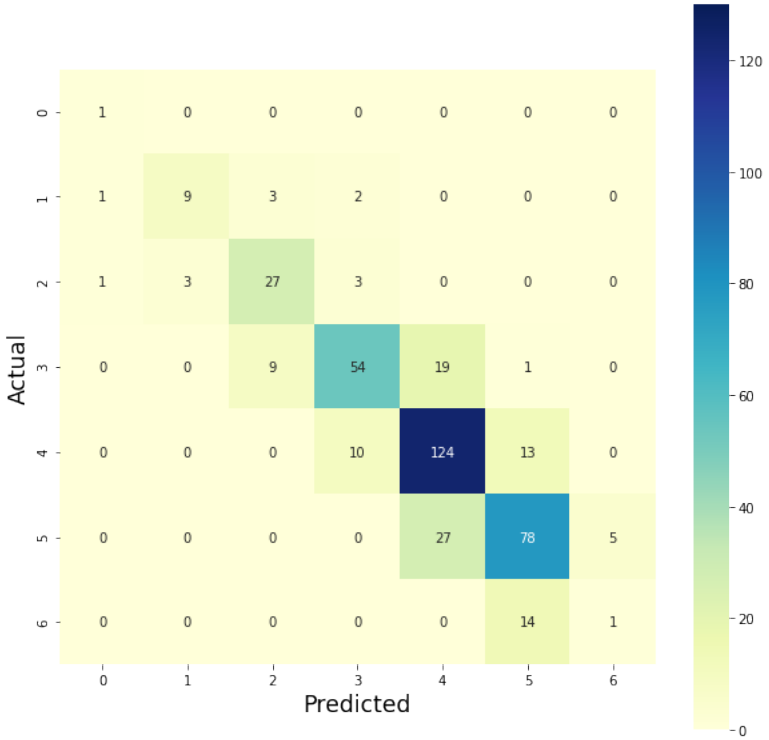


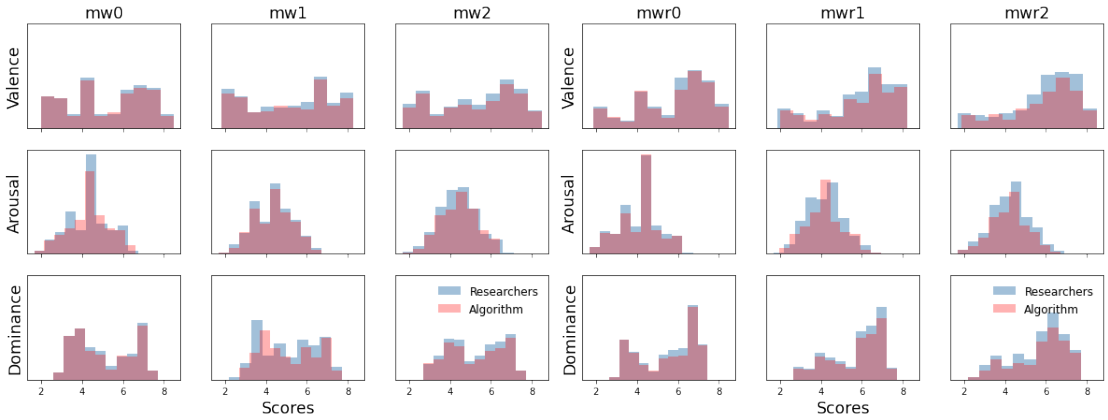
Figure 6: The confusion matrix of the model for method 1

One more finding from the experiment was the informedness of the same setting models might be various. The main reason for this is that the dropout layer randomly sets input units to 0 with a probability at each step during training time to helps prevent overfitting. It means some neurons would be randomly knocked out during the training time. Hence, even the settings of networks are the same, but the performance of the networks can be different.

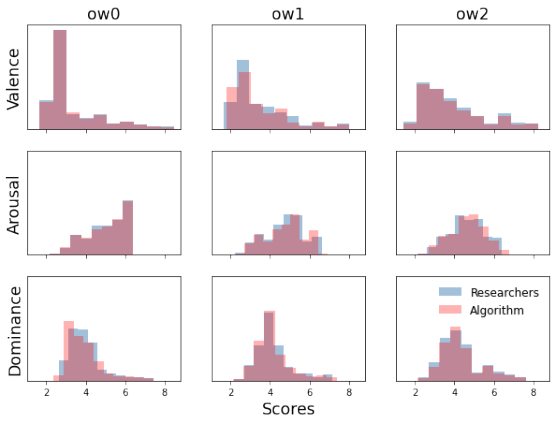
4.2 Method 2

From the Figure 7, although we can notice that the number of words chosen by the algorithm is smaller than that of the words chosen by researchers, the distributions of scores for the words chosen by researchers overlap with the distribution of scores

for the words chosen by method 2. Both of them have the same trend. The reason why the number of words chosen by the algorithm is fewer is that 13 words from the dataset are not in these pre-trained word embeddings and all of them were eliminated in the preprocessing stage. From the distribution perspective, the method 2 performs well for Dying2Learn research.



(a) Introduction module: N792 (b) Final reflections module: N410



(c) Introduction module: N783

Figure 7: The comparison of the distribution of results from method 2 and that from researchers

The cosine similarity of two scores sets shows that the results of the method 2 are similar to the results obtained by researchers in the Dying2Learn research. The values of the cosine similarity can indicate the performance of the proposed method if reference the results obtained by researchers. In this method, because the most similar word in the lexicon for any word is the word itself if that word in the lexicon, we only considered the scenario that the words which are not in Warriner et al. (2013)'s lexicon. This method achieved averaged cosine similarity 0.9856, which indicated the method performed well on the dataset. The cosine similarities for different datasets is shown in Table 6.

The results of the comparison showed that some words chosen by method 2 might not be suitable. Some examples of the comparison are shown in Table 7.

	N792	N410	N783
Cosine similarity	0.984	0.991	0.980

Table 6: The cosine similarities for dierent datasets from method 2.

In this comparison, only the words absent from lexicon were examined. Compared to the words obtained by lemmatisation and stemming, some of the words from method 2 may not make sense. The most interesting point is the method 2 used the same word 'go' as the researchers did to substitute the word 'gone', while the algorithm used 'do' to replace the word 'going' whose lemma should be 'go'. For the words 'caring' and 'touching', the method 2 used 'compassionate' and 'funny' as the most similar words, while researchers used 'care' and 'touch'.

Original Words	Researchers	Method 2
Lemmatisation & Stemming		
going	go	do
gone	go	go
fascinated	fascinate	fascination
accepting	accept	accept
caring	care	compassionate
cessation	cease	cease
touching	touch	funny
prepared	prepare	prepare
Manually		
trepidation	apprehension	nervousness
emotive	emotional	expressive
goodbyes	farewell	tearful
goodbye	farewell	farewell
autonomous	independent	autonomy
daunting	overwhelming	task
MISSING		
ok	-	do
when	-	time

Table 7: Examples from the results of method 2, compared to the original words and the words chosen by researchers.

For the words chosen by researchers manually, some of the words from method 2 still made no sense. Method 2 used 'nervousness', 'expressive', 'task' to be the most similar words in the lexicon for the words 'trepidation', 'emotive' and 'daunting' respectively. However, these words may have different sentiments from that of the original words. For the words 'goodbyes' and 'goodbye', method 2 used two different words 'farewell' and 'tearful' respectively.

For those words marked MISS by researchers, the method at least found out some words from the lexicon for them. For the word 'ok', the method used 'do' as the substitution while used 'time' for the word 'when'.

One main factor affects the performance of the method. Although the plots and the cosine similarity indicate the method performed well on the dataset, the mechanism of word embeddings affects the performance of this method. Word embeddings, including GloVe and Word2vec, are based on a fundamental hypothesis in linguistics that similar words have similar context. This mechanism may lead to the words that occur in the context frequently will be classified as semantic similar. For example, the word 'daunting' may usually have a similar context with the word 'task'. This may be the reason why the word 'goodbyes' and 'goodbye' have different substitutions 'tearful ' and 'farewell' since them have different frequencies occur in the contexts of 'goodbyes' and 'goodbye'. As a result, some of the substitutions from method 2 may not make sense.

4.3 Method 3

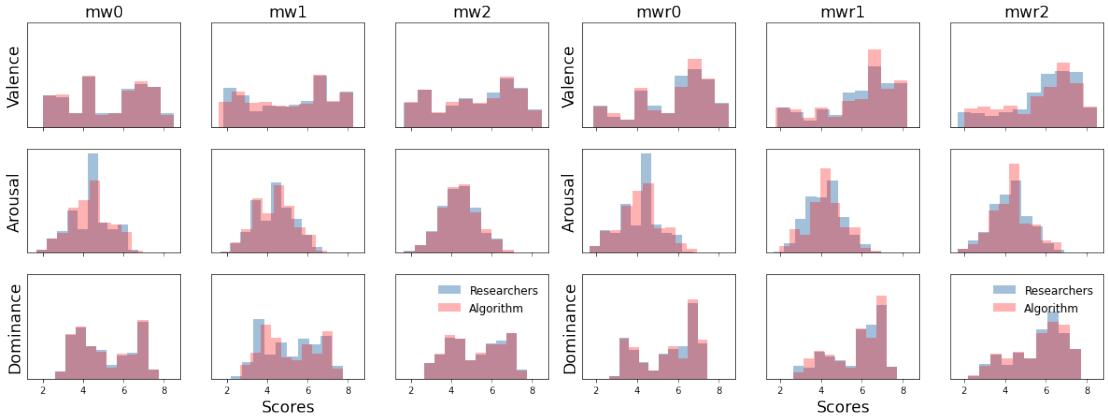
From Figure 8, the distribution of scores for the words chosen by method 3 overlap with the distributions of scores for the words chosen by researchers. Both of them have the same trend. For the dataset N783, the distributions of two outcomes entirely overlap with each other in dominance dimension. Since the USE is an encoder, any word or phrase and sentence can be converted into a vector. There was no need for method 3 to find out any words absent from the pre-trained embeddings like what method 2 did. From the distribution perspective, the performance of method 3 similar to that of researchers for Dying2Learn research.

The cosine similarity indicates that the results from method 3 are similar to the results from researchers. Different from the method 2, this method 3 can be used for words, phrases and sentences. Hence, we examined all types of data. To evaluate the performance of the method, we examined the cosine similarity (Entire Cosine Similarity in Table 8) for the entire dataset. As the same reason mentioned before, the most similar word in the lexicon for any word is the word itself if that word in the lexicon, we had a closer look at the performance of this method for the words absent from lexicon. Hence we computed another cosine similarity (Part Cosine Similarity in Table 8) which only focuses on the words absent from lexicon. The average of entire cosine similarity was 0.9947 while the average of part cosine similarity was 0.9769. Both of these two indicators show the method performed well on the dataset.

Cosine similarity	**N792**	**N410**	**N783**
Part	0.980	0.977	0.972
Entire	0.996	0.993	0.996

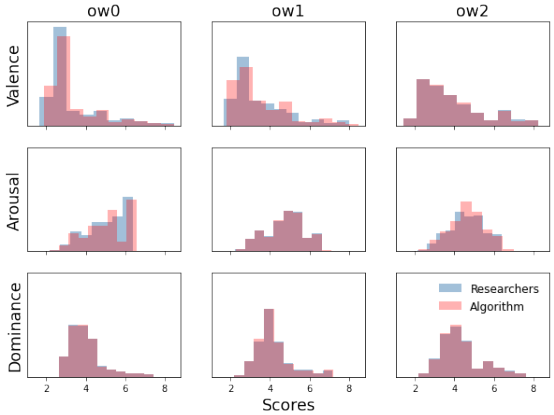
Table 8: The cosine similarities for dierent datasets from method 3.

Compared to the words from Lemmatisation and Stemming, some of the words from method 3 are better. The results are shown in Table 9. For the word 'gone', re-



(a) Introduction module: N792

(b) Final reflections module: N410



(c) Introduction module: N783

Figure 8: The comparison of the distribution of results from method 3 and that from researchers.

searchers used its lemma 'go' for replacement, while method 3 adopted 'disappear'. The sentiments of 'gone' and 'disappear' may be more similar than that between 'gone' and 'go'. The same scenario for the word 'hard-work', when researchers used 'hard' to substitute, the method adopted the word 'hardworking'.

Nearly all of the words chosen by the method are similar to those adopted by researchers. For example, the method used the words 'apprehensive' to substitute 'trepidation', while the researchers used 'apprehension'. For the word 'daunting', the method adopted 'troublesome', researchers used 'overwhelming' instead. With the USE, the same substitutions 'farewell' for 'goodbyes' and 'goodbye'.

Method 3 can find out substitutions for the words marked miss from the dataset. For the word 'ok', the method adopted the word 'fine' and used the word 'Moment' to substitute the word 'when'. The most exciting one was the method adopted the word 'god' to substitute the word 'Jesus'.

For phrases and sentences, method 3 performed better than researchers occasionally, but it failed to deal with negation. For instances, method 3 adopted 'life' for the phrase 'part of life', but researchers used 'part' instead. For the sentence "I

Original Words	Researchers	Method 3
Lemmatisation & Stemming		
going	go	go
gone	go	disappear
fascinated	fascinate	fascination
accepting	accept	accept
caring	care	care
hard-work	hard	hardworking
touching	touch	touch
prepared	prepare	prepare
Manually		
trepidation	apprehension	apprehensive
emotive	emotional	maudlin
goodbyes	farewell	farewell
goodbye	farewell	farewell
autonomous	independent	independent
daunting	overwhelming	troublesome
MISSING		
ok	-	fine
when	-	moment
Jesus	-	god
Phrases		
part of life	part	life
at peace	peaceful	peace
Sentences		
I prepare for death with joy	prepare	joy
Feelings of sadness about leaving loved ones behind	feeling	grieving
Not quite ready	unprepared	ready

Table 9: Examples from the results of method 3, compared to the original words and the words chosen by researchers.

prepare for death with joy", method 3 used 'joy' while researchers used 'prepare'. The word 'joy' can express more emotion than the word 'prepare'. Another example is the sentence "Feelings of sadness about leaving loved ones behind". When researchers used the word 'feeling' to summarise the sentence, the method used the word 'grieving', which is not in the sentence, to paraphrase the sentence. However, the method cannot deal negation. From the sentence "Not quite ready", it used 'ready' to substitute.

The Transformer structure is the main factor that makes method 3 outperformed other methods explored in this research. As discussed in the literature review section, the Transformer is initially used for neural machine translation. The encoder of Transformer models considers the relation between each word and the other words in the sentence and find out which word should be focused when it encodes that sentence. This mechanism makes the models be able to capture the semantic and syntactic information of the sentence. The USE makes use of the encoder of the Transformer models. Therefore, this method can summarise sentences into words. The main reason why method 3 cannot deal with negation may also be because of the Transformer mechanism. Back to the example "Not quite ready", although the vector encoded with the negation information, the encoder focused on the word 'ready'. Also, in the research, we try to "translate" a sentence into a word by using the USE. This may lead to a loss of information since the "translation" is originally for sentences to sentences.

5 Discussion

Many artificial intelligence algorithms are available today, but they may not fit for all the problems in reality. The method 1 was expected to be a solution for Dying2Learn research, but it has some limitations. For the dataset from Dying2Learn, method 1 cannot deal with words and sentences at the same time. If the model takes either a word, a phrase or a sentence as input and predict the sentiment of that input, there will be three results for each participant since they used three separated words to describe their feelings. There should be another algorithm to deal with these three results and identify a final emotion result for the participant. If we concatenate the phrase and sentences as what we did in method 1 for words, the length of inputs can be various, and this can affect the performance of the model. The results from Stanford sentiment treebank (Socher et al. 2013) shows that even the model with tree structure, it achieved 80.7% accuracy for five classes sentiment classification on the phrase level but only 45.7% accuracy on the sentence level. Besides, the model can only classify discrete values in one emotional dimension. However, according to Osgood et al. (1957), one dimension and discrete values may not be sufficient and adequate to describe the spectrum of people's sentiment. Therefore, method 1 is an excellent example to illustrate that although

many models and algorithms are available today, they may not be suitable for the dataset from reality.

Data is the most important element for neural networks. Some algorithms people used today can date back to 1990s or even earlier, such as backpropagation algorithm (Rumelhart et al. 1986), recurrent neural network (Rumelhart et al. 1986) and Long short-term memory neural network (Hochreiter & Schmidhuber 1997). Deep learning did not take off at that time, one reason was lacking data. Even now, we are in the big data era, lacking data may still be the biggest problem affecting the research. All the neural networks are data-hungry. Method 2 and method 3 are examples of this matter. People do not need to use the pre-trained word embeddings. They can obtain their own word embeddings by training word embedding algorithms on some specific corpora, or like Chawla et al. (2019) redefined the loss function of the algorithm to achieve higher performance on specific tasks. However, no matter in which way, a vast and high-quality corpus or many high-quality corpora are needed. Otherwise, the performance of the embeddings would be affected. Back to method 2, because it is not easy to access high-quality corpora, we adopted the smallest GloVe pre-trained embeddings, although it achieved high cosine similarity with the results from researchers, from the word by word comparison, we can conclude that the words from method 2 may not make sense. Due to the same reason, the limitation of training data available for many NLP tasks, researchers Cer et al. (2018) released the USE to help people with this difficulty. This was also the reason why we employed the USE in method 3.

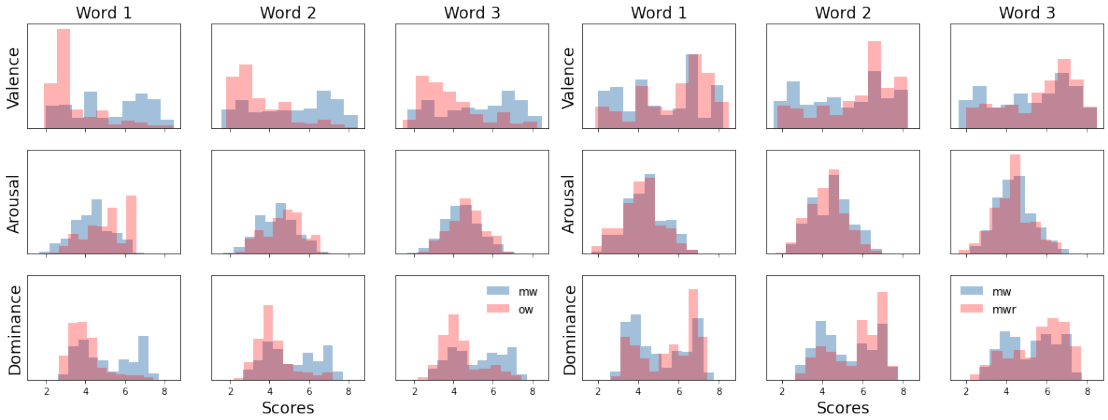
Methods	**Words**	**Sentences**	**Continues Ratings**
1. GloVe with LSTM	✓	✓	✗
2. GloVe with Lexicon	✓	✗	✓
3. The USE with Lexicon	✓	✓	✓

Table 10: The comparison of the three methods discussed in this study.

Compared to method 1 and method 2 (shown in Table 10), method 3 is the best of the approaches explored for Dy2ing2learn research. It can work on the sentences and words from the dataset at the same time. Also, it measures the sentiment of the data by Warriner et al. (2013)'s lexicon. Besides, method 3 can be used as a black box tool for researchers. Due to the input data do not need to be preprocessed. Researchers only need to input the data from Dying2Learn research; the algorithm will output the results. Therefore, method 3 meets the objectives of this research that propose an automatic, accurate and efficient solution. The only limitation with method 3 is that it cannot deal with the negation. Because of the Transformer mechanism, and we applied this mechanism for "translating" sentences into words, the algorithm focuses on the most important word but ignores the negation. Except for this limitation, method 3 is robust in any aspect.

The proposed solution (method 3) has been applied to the Dying2Learn MOOC

2017, and the results indicate the same suggestion as Tieman et al. (2018) concluded in their research that Death Attitudes increased significantly following the participation of the MOOCs. The results of applying method 3 on datasets from MOOC 2017 to analysis the changes in participants' sentiments are shown in Figure 9. We compared the words participants used in the introduction module of the MOOC to describe their personal feelings (N792) and how other people think about death and dying (N783). Also, we examined the changes in sentiments between the beginning (N792) and the end of the MOOC (N410). All the comparisons only considered the participants took part in corresponded sections. Figure 9 (a) illustrates that the words used by participants to describe their personal feelings of death and dying were happier, calmer and more dominated while they used more unpleasant, more arousing and more submissive words to describe the perceived perspective of others of death. The result of the comparison between N792 and N410 indicates that after taking part in the MOOC, people used more positive, more dominated but same level excitable words to describe their feelings about death (shown in 9 (b)).



(a) N792 (mw) VS N783 (ow) (b) N792 (mw) VS N410 (mwr)

Figure 9: The results of applying method 3 on datasets from MOOC 2017 to analysis the changes in participants' sentiments. 'mw' refers to the words used for describing participants personal feelings in introduction module, when 'mwr' refers the words used in final reflection module. 'ow' refers to the words used to describe how other people think about death and dying.

The limitations of this present research and future works for these limitations can be summarised as the following five aspects:

1. Improvement: Although method 3 is robust and outperformed other algorithms explored in this research, it cannot deal with the negation at the moment. An improvement will be taken out in the future work to solve this problem.
2. Applications: The proposed method, method 3, only have been applied to the data from MOOC 2017. However, there are three more years of data from Dying2Learn research. Besides, many comments in the Dying2Learn have not

been examined yet. Method 3 may help with all these data and identify the when and which activities exactly made the emotions of participants change. Hence, one of the future directions of this research can be applying the method 3 to all the MOOC activities.

3. Evaluation: the proposed solution was evaluated in terms of the distribution of the results and the cosine similarity. However, there should be a qualitative analysis of chosen words with domain experts to make sure the proposed solution (method 3) is suitable and reliable for Dying2Learn research.

4. Data: data and privacy are always problems with data-driven algorithms. For one side, data are always inadequate. For the other side, privacy information concern is another problem, especially for medical. Beaulieu-Jones et al. (2019) made use of generative adversarial nets, or GANs (Goodfellow et al. 2014), to generate simulated, synthetic participants that closely resemble participants. The results of their research suggest that synthetic data can be used to perform hypothesis-generating analyses and overcome the privacy information concern. Back to this research, lacking data was the biggest problem. If the GANs can be based on the data from Dying2learn to generate simulated data, then it would be helpful for the research. More than that, synthetic data also can be used to prevent the abuse of participants' privacy information. Therefore, the last direction of this research can be investigating the possibility of using GANs to generate data.

6 Conclusion

With deaths rate increasing, people are more likely exposed to death and dying. Under this circumstance, Dying2Learning research provides domain experts with an opportunity to have a closer look at the people's opinions about death and dying for improving the public's death competence. However, there are still many text-based comments from the activities of the research have not been examined yet. Therefore the present study developed an automatic and efficient solution for sentiment analysis by using natural language processing technologies to assist Dying2Learning research.

The difficulties of sentiment analysis algorithms, capturing semantic and syntactic information, were discussed in the literature review section. Based on these difficulties, we reviewed three types of baseline algorithms which are Naive Bayes, SVM and lexicon-based algorithms and concluded that these baseline algorithms might not be suitable for the Dying2Learn research. More than that, the algorithms used for capturing semantic and syntactic were also discussed in the literature review section.

To develop an effective and accurate algorithm for Dying2Learn research, we explored the neural network method (Method 1 in the Experiments section) and

proposed two novel methods which combines embeddings and Warriner's affective lexicon. The results from these three methods suggest that method 1 does not fit in the purpose of the research giving the fact that it cannot deal with words and sentences from the dataset at the same time. From the evaluation, method 2 performed well on the dataset, but when looking at the words chosen by method 2, it cannot be the solution. Besides, the method 2 cannot deal with sentences.

Method 3 is the best option explored for the dataset from MOOC. With the help of the Transformer architecture, the USE can encode words and sentences into vectors and capture the semantic and syntactic information. Although method 3 cannot deal with the negation at the moment, it outperformed the other two methods. For some phrases and sentences from Dying2Learn research, method 3 even outperformed human performance in some situations. It meets the objectives of this present study that develop an efficient and accurate solution. The proposed solution (method 3) has been applied to the Dying2Learn MOOC 2017, and the results indicate the same suggestion as Tieman et al. (2018) concluded in their research. Therefore, method 3 is the best of the approaches explored and can be used for Dying2Learn research. To further the study, we take out future work for the development to improve the method we proposed in this study.

Appendix

Content removed for privacy reasons

References

Airoldi, E., Bai, X. & Padman, R. (2004), Markov blankets and meta-heuristics search: Sentiment extraction from unstructured texts, *in* 'International Workshop on Knowledge Discovery on the Web', Springer, pp. 167–187.

Bahdanau, D., Cho, K. & Bengio, Y. (2014), 'Neural machine translation by jointly learning to align and translate', *arXiv preprint arXiv:1409.0473* .

Bakshi, R. K., Kaur, N., Kaur, R. & Kaur, G. (2016), Opinion mining and sentiment analysis, *in* '2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)', IEEE, pp. 452–455.

Balk, D. E., Wogrin, C. E., Thornton, G. E. & Meagher, D. E. (2007), *Handbook of thanatology: The essential body of knowledge for the study of death, dying, and bereavement.*, Association for Death Education and Counseling.

Beaulieu-Jones, B. K., Wu, Z. S., Williams, C., Lee, R., Bhavnani, S. P., Byrd, J. B. & Greene, C. S. (2019), 'Privacy-preserving generative deep neural networks

support clinical data sharing', *Circulation: Cardiovascular Quality and Outcomes* **12**(7), e005122.

Bengio, Y., Simard, P. & Frasconi, P. (1994), 'Learning long-term dependencies with gradient descent is difficult', *IEEE transactions on neural networks* **5**(2), 157–166.

Bollen, J., Mao, H. & Zeng, X. (2011), 'Twitter mood predicts the stock market', *Journal of computational science* **2**(1), 1–8.

Bradley, M. M. & Lang, P. J. (1999), Affective norms for english words (anew): Instruction manual and affective ratings, Technical report, Technical report C-1, the center for research in psychophysiology .

Cabral, L. & Hortacsu, A. (2010), 'The dynamics of seller reputation: Evidence from ebay', *The Journal of Industrial Economics* **58**(1), 54–78.

Cass, S. (2011), 'Unthinking machines', *Artificial intelligence needs a reboot, say experts* .

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C. et al. (2018), 'Universal sentence encoder', *arXiv preprint arXiv:1803.11175* .

Chawla, K., Khosla, S., Chhaya, N. & Jaidka, K. (2019), Pre-trained affective word representations, *in* '2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)', IEEE, pp. 1–7.

Cho, K., Van Merriënboer, B., Bahdanau, D. & Bengio, Y. (2014), 'On the properties of neural machine translation: Encoder-decoder approaches', *arXiv preprint arXiv:1409.1259* .

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. & Bengio, Y. (2014), 'Learning phrase representations using rnn encoder-decoder for statistical machine translation', *arXiv preprint arXiv:1406.1078* .

Chomsky, N. (1965), *Syntactic structures.*, Janua linguarum. Series minor; nr. 4, Mouton, The Hague.

Chomsky, N. (1969), 'Some empirical assumptions in modern philosophy of language'.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. & Kuksa, P. (2011), 'Natural language processing (almost) from scratch', *Journal of machine learning research* **12**(ARTICLE), 2493–2537.

Commission, A. P. (2013), *An Ageing Australia: Preparing for the Future*, Productivity Commission.

Das, S. & Chan, M. (2001), 'Extracting market sentiment from stock message boards', *Asia Pacific Finance Association* **2001**.

Dave, K., Lawrence, S. & Pennock, D. M. (2003), Mining the peanut gallery: Opinion extraction and semantic classification of product reviews, *in* 'Proceedings of the 12th international conference on World Wide Web', pp. 519–528.

Firth, J. R. (1957), 'A synopsis of linguistic theory, 1930-1955', *Studies in linguistic analysis* .

Fonseca, L. M. & Testoni, I. (2012), 'The emergence of thanatology and current practice in death education', *OMEGA-Journal of Death and Dying* **64**(2), 157–169.

Gamon, M. (2004), Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis, *in* 'COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics', pp. 841–847.

Gellie, A., Mills, A., Levinson, M., Stephenson, G. & Flynn, E. (2014), 'Death: a foe to be conquered? questioning the paradigm', *Age and ageing* **44**(1), 7–10.

Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. (2016), *Deep learning*, Vol. 1, MIT press Cambridge.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014), Generative adversarial nets, *in* 'Advances in neural information processing systems', pp. 2672–2680.

Hochreiter, S. & Schmidhuber, J. (1997), 'Long short-term memory', *Neural computation* **9**(8), 1735–1780.

Hughes, S., Preston, N. J. & Payne, S. A. (2016), 'Online learning in palliative care: does it improve practice?', *European Journal of Palliative Care* **23**(5), 236–239.

Huq, M. R., Ali, A. & Rahman, A. (2017), 'Sentiment analysis on twitter data using knn and svm', *IJACSA) International Journal of Advanced Computer Science and Applications* **8**(6), 19–25.

Iyyer, M., Manjunatha, V., Boyd-Graber, J. & Daumé III, H. (2015), Deep unordered composition rivals syntactic methods for text classification, *in* 'Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)', pp. 1681–1691.

Jurafsky, D. & Martin, J. H. (2014), 'Speech and language processing. vol. 3'.

Laver, M., Benoit, K. & Garry, J. (2003), 'Extracting policy positions from political texts using words as data', *American political science review* **97**(2), 311–331.

Leveau, N., Jhean-Larose, S., Denhière, G. & Nguyen, B.-L. (2012), 'Validating an interlingual metanorm for emotional analysis of texts', *Behavior Research Methods* **44**(4), 1007–1014.

Lewis, D. D. (1998), Naive (bayes) at forty: The independence assumption in information retrieval, *in* 'European conference on machine learning', Springer, pp. 4–15.

Liu, B. (2012), 'Sentiment analysis and opinion mining', *Synthesis lectures on human language technologies* **5**(1), 1–167.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S. & McClosky, D. (2014), The stanford corenlp natural language processing toolkit, *in* 'Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations', pp. 55–60.

Manning, C., Socher, R., Fang, G. G. & Mundra, R. (2017), 'Cs224n: Natural language processing with deep learning1'.

Matsumoto, S., Takamura, H. & Okumura, M. (2005), Sentiment classification using word sub-sequences and dependency sub-trees, *in* 'Pacific-Asia conference on knowledge discovery and data mining', Springer, pp. 301–311.

McIlfatrick, S., Hasson, F., McLaughlin, D., Johnston, G., Roulston, A., Rutherford, L., Noble, H., Kelly, S., Craig, A. & Kernohan, W. G. (2013), 'Public awareness and attitudes toward palliative care in northern ireland', *BMC palliative care* **12**(1), 34.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013), 'Efficient estimation of word representations in vector space', *arXiv preprint arXiv:1301.3781* .

Mikolov, T., Yih, W.-t. & Zweig, G. (2013), Linguistic regularities in continuous space word representations, *in* 'Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies', pp. 746–751.

Miller-Lewis, L., Tieman, J., Rawlings, D., Parker, D. & Sanderson, C. (2020), 'Can exposure to online conversations about death and dying influence death competence? an exploratory study within an australian massive open online course', *OMEGA-Journal of Death and Dying* **81**(2), 242–271.

Mullen, T. & Collier, N. (2004), Sentiment analysis using support vector machines with diverse information sources, *in* 'Proceedings of the 2004 conference on empirical methods in natural language processing', pp. 412–418.

Mullen, T. & Malouf, R. (2006), A preliminary investigation into sentiment analysis of informal political discourse., *in* 'AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs', pp. 159–162.

Osgood, C. E., Suci, G. J. & Tannenbaum, P. H. (1957), *The measurement of meaning*, number 47, University of Illinois press.

Pang, B. & Lee, L. (2008), 'Opinion mining and sentiment analysis', *Foundations and Trends in Information Retrieval* **2**(1-2), 1–135.

Pang, B., Lee, L. & Vaithyanathan, S. (2002), 'Thumbs up? sentiment classification using machine learning techniques', *arXiv preprint cs/0205070* .

Pascanu, R., Mikolov, T. & Bengio, Y. (2013), On the difficulty of training recurrent neural networks, *in* 'International conference on machine learning', pp. 1310–1318.

Pennebaker, J. W., Francis, M. E. & Booth, R. J. (2001), 'Linguistic inquiry and word count: Liwc 2001', *Mahway: Lawrence Erlbaum Associates* **71**(2001), 2001.

Pennington, J., Socher, R. & Manning, C. D. (2014), Glove: Global vectors for word representation, *in* 'Empirical Methods in Natural Language Processing (EMNLP)', pp. 1532–1543.
URL: *http://www.aclweb.org/anthology/D14-1162*

Robbins, R. A. (1994), 'Death competency: Bugens coping with death scale and death self-efficacy', *Death anxiety handbook: Research, instrumentation, and application* pp. 149–165.

Rong, X. (2014), 'word2vec parameter learning explained', *arXiv preprint arXiv:1411.2738* .

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986), 'Learning representations by back-propagating errors', *nature* **323**(6088), 533–536.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y. & Potts, C. (2013), Recursive deep models for semantic compositionality over a sentiment treebank, *in* 'Proceedings of the 2013 conference on empirical methods in natural language processing', pp. 1631–1642.

Sutskever, I., Vinyals, O. & Le, Q. V. (2014), Sequence to sequence learning with neural networks, *in* 'Advances in neural information processing systems', pp. 3104–3112.

Tanesab, F. I., Sembiring, I. & Purnomo, H. D. (2017), 'Sentiment analysis model based on youtube comment using support vector machine', *International Journal of Computer Science and Software Engineering* **6**(8), 180.

Tieman, J., Miller-Lewis, L., Rawlings, D., Parker, D. & Sanderson, C. (2018), 'The contribution of a mooc to community discussions around death and dying', *BMC palliative care* **17**(1), 31.

Tong, R. M. (2001), An operational system for detecting and tracking opinions in on-line discussion, *in* 'Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification', Vol. 1.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017), Attention is all you need, *in* 'Advances in neural information processing systems', pp. 5998–6008.

Vinyals, O., Kaiser, Ł., Koo, T., Petrov, S., Sutskever, I. & Hinton, G. (2015), Grammar as a foreign language, *in* 'Advances in neural information processing systems', pp. 2773–2781.

Warriner, A. B., Kuperman, V. & Brysbaert, M. (2013), 'Norms of valence, arousal, and dominance for 13,915 english lemmas', *Behavior research methods* **45**(4), 1191–1207.

Yu, H. & Hatzivassiloglou, V. (2003), Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences, *in* 'Proceedings of the 2003 conference on Empirical methods in natural language processing', pp. 129–136.

Zainuddin, N. & Selamat, A. (2014), Sentiment analysis using support vector machine, *in* '2014 international conference on computer, communications, and control technology (I4CT)', IEEE, pp. 333–337.

Zuo, W., Jiang, S., Guo, Z., Feldman, M. W. & Tuljapurkar, S. (2018), 'Advancing front of old-age human survival', *Proceedings of the National Academy of Sciences* **115**(44), 11209–11214.