# Infectious Disease Surveillance with GLEPI: A Natural Language Processing and Deep Learning System

MSc Research Project
Data Analytics

Emmanuel Adekola
18198627

School of Computing
National College of Ireland

Supervisor: Dr. Vladmir Milosavljevic

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | |
|---|---|
| **Student Name:** | Emmanuel Obaloluwa Adekola |
| **Student ID:** | 18198627 |
| **Program:** | MSc. Data Analytics **Year:** 2020 |
| **Module:** | Research Project |
| **Supervisor:** | Dr. Vladmir Milosavljevic |
| **Submission Due Date:** | August 17, 2020 |
| **Project Title:** | Infectious Disease Surveillance with GLEPI: A Real-time Natural Language Processing and Deep Learning System |
| **Word Count:** | 6848 **Page Count:** 22 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:

Date: August 17, 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies). | □ |
| You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| Office Use Only | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Table of Contents

# Abstract

Currently, the prevailing discussions on infectious disease outbreak surveillance are centred on mining unstructured data sources and reducing false notifications. Mining microblogs and other internet-based resources for infectious disease surveillance in an accurate and timely manner has become pertinent due to recent public health concerns.

In this paper, we implemented three deep learning-based frameworks to the natural language processing of microblog data to establish the best combination of techniques for infectious disease surveillance. We implemented LSTM, CNN and bi-directional LSTM frameworks. Our bi-directional LSTM model performed best with a 12.4% and 10.5% higher accuracy score than that of our LSTM and CNN models respectively.

In our bid to establish the best combination of techniques/parameters, we carried out an in-depth investigation on how number of epochs, dropout rate, and word embedding methods in our models affect performance.

Finally, we deploy GLEPI, a deep learning-based NLP framework that uses a bi-directional LSTM model to predict the validity of infectious disease related corpus.



**Keywords:** Natural Language Processing, Deep Learning, Neural Network, Sentiment Analysis, Infectious Disease

# Infectious Disease Surveillance with GLEPI: A Natural Language Processing and Deep Learning System

Adekola Emmanuel
18198627

## 1 Introduction

Over time, there has been several researches focused on mining big data for public health surveillance. A good number of these researches have been on infectious or communicable diseases, applying various combinations of techniques (Ertem et al., 2018). The current COVID19 pandemic has provided scientists and academic researchers with the opportunity to delve deeper into the space of analysing big data for infectious disease surveillance. In recent times, other infectious diseases like Ebola, Influenza, and Lassa Fever have been closely researched using multiple data sources and mixed techniques (Ahmed et al., 2019).

The humongous challenge of analysing big data, dealing with volume, velocity and variety amidst others, is daily being surmounted. There has been numerous research work focused on the improvement of natural language processing (NLP) techniques for different use cases (Chae et al., 2018). However, while there have been numerous studies on the use of supervised machine learning techniques for natural language processing of big data (Edo-Osagie et al., 2019), the application of deep learning techniques to the field is relatively recent.

Recent works on deep learning-based NLP systems and applications has achieved tremendous results. Various NLP tasks such as sentiment analysis, question answering (QA), and machine translation are being researched to achieve trailblazing results. Evidently, deep learning-based NLP systems and applications have achieved better accuracy when compared to leading machine learning techniques like Support Vector Machine (SVM) and Random Forest (RF) that have been widely recommended for NLP (Aiello et al., 2020).

For this research work, we applied deep learning-based NLP techniques to multiple data channels, to monitor the outbreak of infectious disease. We implemented and compared the performance of three deep learning models namely long short-term memory (LSTM), bi-directional LSTM and convolutional neural network (CNN).

In this paper, we described some of the current best practices for applying deep learning in NLP and present a deep learning-based NLP system named GLEPI. It performs the extraction, transformation, and analysis of infectious disease data from Twitter, and Google ranked web pages. The system offers an affordable solution that aids the job of disease detectives, providing health agencies and other institutions with timely surveillance data on infectious disease.

In a bid to scientifically document reproduceable steps for the setting up of a competitive deep learning-based NLP system, we came up with one research question and two objectives. We believe the following captures the intent for this research work and the development of GLEPI.

### 1.1 Research Question

The overall goal of this research project is to apply and evaluate leading deep learning-based NLP techniques for infectious disease surveillance. The set question for this research work is captured below.

1. To what extent can natural language processing for infectious disease surveillance be strengthened by exploring multiple deep learning models?

## 1.2 Research Objectives

In a bid to provide answers to the set question for this research, two main objectives were defined. We believe these defined objectives adequately represent the main goal and original intent of this research work.

1. Apply deep learning techniques to the natural language processing of multiple data sources.
2. Determine the best combination of techniques for infectious disease surveillance using unstructured data.

## 1.3 Data Sources

For this research work, we gathered data from official and non-official sources for infectious disease surveillance.

1. **Official Sources:** By official sources, we are referring to verified sources with repositories containing either semi-structured or unstructured data. For our research, we explored data from an institutional website. We trained and validated our model using the National Institute of Informatics Testbeds and Community for Information Access Research (NTCIR) classified tweets. The Medical Natural Language Processing for Web Document (MedWeb) classified tweets were used (Wakamiya et al., 2017).
2. **Non-official Sources:** By unofficial sources, we are referring to social media and other microblog data sources (Arsevska et al., 2018). For our research, we explored Twitter data and Google search top ranking verifiable news blogs.

In the next section, we provide a review of related research works on deep learning-based NLP. In section 3, we described and justified our adopted methodology and architecture. In Section 4, we present the details of our analysis and evaluation results. Section 5 and 6 provides further comparative analysis and outlines key points amidst suggestions for future works respectively.

# 2 Literature Review

NLP uses computational algorithms to analyse and represent human language in an automated fashion (Chae et al., 2018). NLP is behind several user-friendly applications like Google's widely used search engine, and Alexa, Amazon's voice assistant. NLP is also relevant in training machines to perform complex natural language tasks such as sentiment analysis and machine translation (Edo-Osagie et al., 2019).

Not until recent times, a good number of methods applied to NLP problems used inefficient machine learning models that are time-consuming due to hand-crafted labelling or annotation (Goel et al., 2020). This repeatedly led to issues like the curse of dimensionality due to the representation of linguistic information with high-dimensional features (Goel et al., 2020).

However, the recent advances in the use of neural based word and character embedding models have achieved state-of-the-art results on various applications to language-related tasks. The neural networking of low dimensional and distributed representations has been noted to outperform machine learning models like SVM or RF (Aiello et al., 2020).

## 2.1 Distributed Representations

As earlier noted, neural-based models provide a good alternative to challenges like the curse of dimensionality posed by traditional machine learning models.

1. Word Embeddings: Also known as distributional vectors, they are based on a distributional hypothesis (Feldman et al., 2019). This states that words that appears within similar contexts possess similar meanings. With the objective of predicting a word based on its context, word embeddings are pre-trained on a task with the use of a shallow neural network. The word vectors are embedded with syntactic and semantic information. This step is believed to be the game changer in many NLP tasks such as sentiment analysis. The introduction of continuous bag-of-words (CBOW) and skip-gram models has increased the popularity of distributed representation for NLP tasks. They became popular due to their efficiency in the construction of superior word embeddings and their usefulness for semantic compositionality.
2. Word2vec: As stated in the previous paragraph, CBOW and skip-gram models are game changers. CBOW uses a neural approach for the construction of word embeddings with the sole purpose of computing the conditional probability of a target word using its context within a set window. Skip-gram is another neural approach to construct word embeddings that predicts the surrounding context words based on a central target word (Gibbons et al., 2019). The two models determine their word embedding dimension through an unsupervised computational prediction of accuracy. A major challenge with word embedding is obtaining vector representations for some phrases like “cold tea” or “Texas Ranger”. These individual word vector representations can’t just be simply combined because the phrases don’t represent the individual word’s combination of meaning. When considering longer phrases or sentences, this gets even more complicated. Another limitation is the use of smaller window sizes. This is usually counterproductive for tasks that requires proper differentiation of words like sentiment analysis. It is also important to note that word embeddings are dependent on the application they are used for. While the re-training of task specific embeddings for every new task is a considerable practice, it is computationally expensive, and it is better addressed with the use of negative sampling (Goel et al., 2020). The model does not take polysemy into account and training data has a huge potential of introducing bias to the model.
3. Character Embeddings: Some tasks like named-entity recognition (NER) and parts-of-speech (POS) tagging perform better with the use of morphological information in words (Gibbons et al., 2019). Morphologically rich languages such as Chinese, Portuguese, Spanish are naturally processed better with the use of character embedding.

Finally, it’s important to note that word vectors are limited in how well they capture conceptual meaning of words (Feldman et al., 2019). Hence, it takes more than distributional semantics to understand the concepts behind words. There has been recent works and debate on meaning representation in the context of NLP systems (Feldman et al., 2019).

## 2.2 Convolutional Neural Network (CNN)

CNN is a neural-based approach that extracts higher-level features by applying a feature function to constituting words or n-grams. Amidst several tasks, its abstract features have been

successfully deployed for sentiment analysis, question answering, and machine translation. One of the early methods deployed, transformed words into a vector representation via a look-up table. This introduced a basic word embedding approach that is used to learn weights during the training of a network.

To perform sentence modelling with a primitive CNN, first, sentences are tokenized into words. Next, they are transformed into a word embedding matrix. Then, the application of convolutional filters on the input embedding layer to produce a feature map. This is followed up by a max-pooling operation that ensures the application of a max operation on each filter to reduce dimensionality and obtain a fixed length output. This procedure generates the final sentence representation.

An increase in the complexity of the basic CNN described above and its adaptation to NLP tasks like NER, word prediction, and POS is still currently understudied (Fast et al., 2018). It requires a window-based approach that considers a fixed size window of neighbouring words for each word. Also, a standalone CNN is applied to neighbouring words with the training objective of predicting the word at the centre of the window. This phenomenon is called word-level classification.

A dissuasive issue with basic CNN is its inability to model NLP tasks that requires long distance dependencies (Naoui et al., 2020). To work around this challenge, CNNs have been combined with time-delayed neural networks (TDNN) to enable larger contextual range during training. Another example of CNN variant that have been successful deployed for NLP tasks like sentiment prediction and question type classification is known as dynamic convolutional neural network (DCNN). Dynamically filtering span variable ranges, DCNN makes use of a dynamic k-max pooling technique to perform sentence modelling.

With the use of external knowledge, CNNs have also been deployed to handle complex tasks that requires varying lengths of texts like sentiment analysis on Twitter's microtexts. CNN has also been proven to be useful for other tasks like query-document matching, question-answer representations, and speech recognition. Also, DCNN has been deployed to hierarchically learn and compose lexical features for summarization of texts.

In conclusion, CNNs mine semantic clues in contextual windows efficiently but they perform poorly when it comes to long-distance modelling and sequential ordering of contextual information. Recurrent models perform better for such learning and are discussed next.

## 2.3 Recurrent Neural Network (RNN)

As highlighted above, RNNs are effective at processing sequential information. They recursively apply a computation to every input sequence instance based on the previously computed results. Sequences are sequentially fed to a recurrent unit represented by a fixed-size vector of tokens. The game changing ability RNN introduces is its capacity to memorize the results of previous computations and use same for current computation (Wang et al., 2015). This is why RNNs are suitable for the modelling of context dependencies. Amidst others, RNNs have been found useful for NLP tasks like language modelling, machine translation, and image captioning. RNN models are not necessarily superior to CNN models as they are both effective for different aspects of NLP tasks. RNN typically ingests one-hot encodings or word embedding inputs, and sometimes ingests abstract representations constructed by another model (Naoui et al., 2020). An established issue with simple RNNs is the vanishing gradient

problem. This makes learning and parameter tuning difficult in early layers. To overcome this limitation, other RNN based models like long short-term memory (LSTM) networks, gated-recurrent networks (GRU), and residual networks (ResNets) were developed. LSTM has input, forget, and output gates and calculates the hidden state by combining the three (Wang et al., 2015). A bit similar to LSTMs, GRUs consist of only two gates and are arguably more effective due to their less complexity (Joshi et al., 2019). One of our reviewed study explains that it is hard to say which of the gated RNNs are more effective, and they are selected based on the available computing power.

## 2.4 Attention Mechanism

The attention mechanism technique is a modification to the earlier described RNN-based technique. Along with the calculated information using the input hidden state sequence, the decoder part of the RNN-based framework uses the last hidden state (Joshi et al., 2019). This is important and crucial to tasks that rely on some form of alignment between output and input texts. Successfully, attention mechanism techniques have been deployed to tackle NLP tasks like in machine translation, dialogue generation, text summarization, sentiment analysis, and image captioning (Șerban et al., 2019). Several types and forms of attention mechanisms have been explored and this remains an important NLP research area.

## 2.5 Recursive Neural Network

Just like RNNs, recursive neural networks are used to model sequential data. Languages are seen as a recursive structure where words and sub-phrases have hierarchical composition use into higher-level phrases. Such structures use a representation of all its children nodes to capture a non-terminal node. A basic recursive neural network combines constituents using a bottom-up approach for the computation of higher-level phrases (Joshi et al., 2019). A variant, MV-RNN, uses matrix and vector to represent a word using parameters learnt by the network to represent each constituent. Recursive neural networks are quite flexible and can be combined with LSTM to deal with gradient vanishing problems. Recursive neural networks are widely used for sentence relatedness, parsing, sentiment analysis, and semantic relationship classification (Naoui et al., 2020).

## 2.6 Reinforcement Learning

Reinforcement learning trains agents to perform discrete actions followed by a reward (Joshi et al., 2019). It is widely used for natural language generation (NLG) like text summarization. For NLP, a reinforcement algorithm called REINFORCE was developed for image captioning and machine translation. This framework consists of an agent that interacts with input words and context vectors at every time step. The agent predicts the next word of a sequence at each time step and updates its internal state (Fast et al., 2018). This iterates until a reward is calculated at the end of the sequence.

## 2.7 Unsupervised Learning

This is the mapping of sentences to fixed-size vectors in an unsupervised manner. The distributed representations are trained using an auxiliary task and are used for capturing semantic and syntactic properties from languages (Șerban et al., 2019). A major example of

this is the ‘a skip-thought model’ (Naoui et al., 2020). Similar to word embedding technique, adjacent sentence is predicted using a centre sentence. It is trained using the seq2seq framework that allows decoder to generate target sequences while the encoder extracts generic features, learning word embeddings in the process.

### 2.8 Memory-Augmented Network

Memory-augmented network refers to the coupling of neural networks with some form of memory to solve NLP tasks like language modelling, part-of-speech tagging, visual QA, and sentiment analysis (Joshi et al., 2019). A good example is solving QA tasks. To do this, common sense knowledge is fed into the model to provide some form of memory. Probably the most advanced implementation of Memory-Augmented Network is the Dynamic Memory Networks, it employs neural networks models for input representation, attention, and QA.

From bodies of existing research works and implementations, capacity and effectiveness of neural-based models such as CNNs and RNNs are well in advanced stages (Joshi et al., 2019). Also, the awareness on the possibilities of applying reinforcement learning, deep generative models, and unsupervised learning to complex NLP tasks such as sentiment analysis, visual QA and machine translation has been raised. Attention mechanisms and memory-augmented networks have been hypothesized to be powerful but barely researched (Naoui et al., 2020).

The combination of all these powerful techniques provides a convincing basis to keep working on the demystification of language complexities. It also provides plethora of options that can be researched for infectious disease surveillance. In this paper, we present the result of our performance evaluation on RNN LSTM and CNN models for infectious disease sentiment analysis on MedWeb classified tweets, Twitter data and news articles.

## 3 Methodology

For this research, we adopted the CRISP-DM methodology. As earlier stated, we developed three models based on LSTM and CNN models to perform sentiment analysis. In this section, we provide details on our data, pre-processing steps and an overview of the implemented RNN LSTM and CNN models. The following summarizes the methods adopted.

1. Data Ingestion: In addition to MedWeb classified tweets, recent tweets and news articles were continuously streamed and downloaded using Centre for Disease Control (US CDC) list of infectious disease symptoms as keywords.
2. Natural Language Pre-processing: Texts were pre-processed; this included language filtering, removal of stop words, text cleansing, lemmatization, stemming and word tokenization.
3. Sentiment Analysis: Pre-trained and In-training word embedding techniques, LSTM and CNN text classification models were implemented with bi-directional LSTM model being deployed for further analysis of more recent dataset.

### 3.1 Data Pre-processing

Generally, the pre-processing of our datasets involves the removal of user mentions, spaces, non-alphabetic characters, apostrophes, web links, single characters, and stop words

(Carrillo-de-Albornoz et al., 2018). It also involved tokenization, lemmatization and stemming of words.

By tokenization, we mean the breaking down of text/sentences into words; it is the conversion of words into vectors for computer interpretation and understanding (Du et al., 2018). Stemming involves the chopping off of derivational affixes to get the base word. We implemented this and cautiously accepted the arguments from previous studies that sentiments are often associated to words (Carrillo-de-Albornoz et al., 2018). Finally, we remove inflectional endings. This morphological and vocabulary analysis is called lemmatization (Yoon et al., 2018).

## 3.2 Word Embedding

For the purpose of this research, two highly recommended embedding methods from literature were implemented. We applied a pre-trained embedding called GloVe embedding to the corpus to produce the best result for our CNN implementation. For our LSTM implementations, we implemented in-training embedding. The Keras embedding layer was trained on our dataset to achieve this.

## 3.3 Long Short-term Memory NLP Framework

As noted in the literature review section, LSTM is a popular RNN framework for modelling sequential data (Wang et al., 2015). It captures long term dependencies better than the vanilla RNN model. Like every other RNN, LSTM network gets the input at each time-step and the output from the previous timestep to produce an output to be used for the next time step (Joshi et al., 2020). The hidden layer(s) from all or the last time-step are then used for classification (Sosa, 2017, p.5).

To capture long term dependencies or solve the issue of gradient vanishing, LSTM network uses some internal gates. The LSTM framework uses a memory cell and three gates namely; input gate, output gate, forget gate. While the three gates regulate the flow of information in and out of the cell, the cell memorises values over arbitrary time intervals (Joshi et al., 2020). The equation for a simple LSTM is captured below:

$$f_t = \sigma(W^{(f)} x_t + U^{(f)} h_{t-1} + b^{(f)})$$
$$i_t = \sigma(W^{(i)} x_t + U^{(i)} h_{t-1} + b^{(i)})$$
$$o_t = \sigma\big(W^{(o)} x_t + U^{(o)} h_{t-1} + b^{(o)}\big)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W^{(c)} x_t + U^{(c)} h_{t-1} + b^{(c)})$$
$$h_t = o_t \odot \tanh(c_t)$$

Where:

$\sigma$ = sigmoid function for mapping the values within 0 and 1

$x_t \in R^d$ = input at timestep t

$d$ = feature dimension for each word

$\odot$ = element-wise product

$c_t$ = memory cell designed to lower the risk of vanishing gradient

$f_t$ = the forget gate to reset the memory cell

$i_t$ = input gate

$o_t$ = output gate

In this research, firstly, we used a four-layer sequential model with a LSTM layer of 256 units, which uses the embedding layer of Keras to prepare sequential input for the Dense layer to predict sentiment.

Secondly, we used a four-layer bi-LSTM model, which also uses the embedding layer of Keras to prepare sequential input for the Dense layer with relu and sigmoid activations to predict the sentiment of the tweet in review. The diagram below shows a simple LSTM network framework.

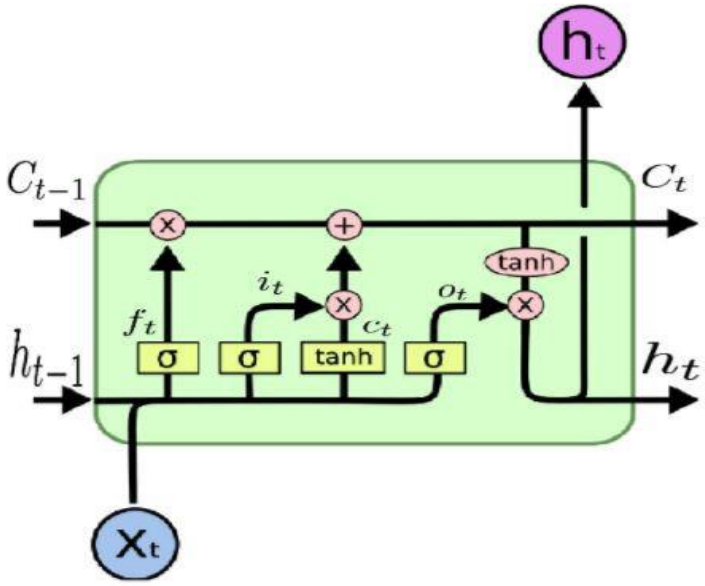


Figure 1: Simple LSTM network.

### 3.4 Convolutional Neural Network NLP Framework

The second framework implemented is based on convolutional neural network (CNN). In recent years, CNNs solutions have been successfully deployed for several NLP and computer vision tasks. From early works till date, CNN has performed excellently on text classification tasks (Joshi et al., 2019). Performing text classification with CNN requires the stacking together of the embedding from different words of a sentence to form a two-dimensional array, and the application of convolution filters to produce a new feature representation (Sosa, 2017, p.5). Some pooling techniques are applied on the new features, and then the hidden representations are formed from the concatenation of pooled features from different filters. To make final predictions, the hidden representations are followed by fully connected layer(s). The figure below shows CNN's general framework.

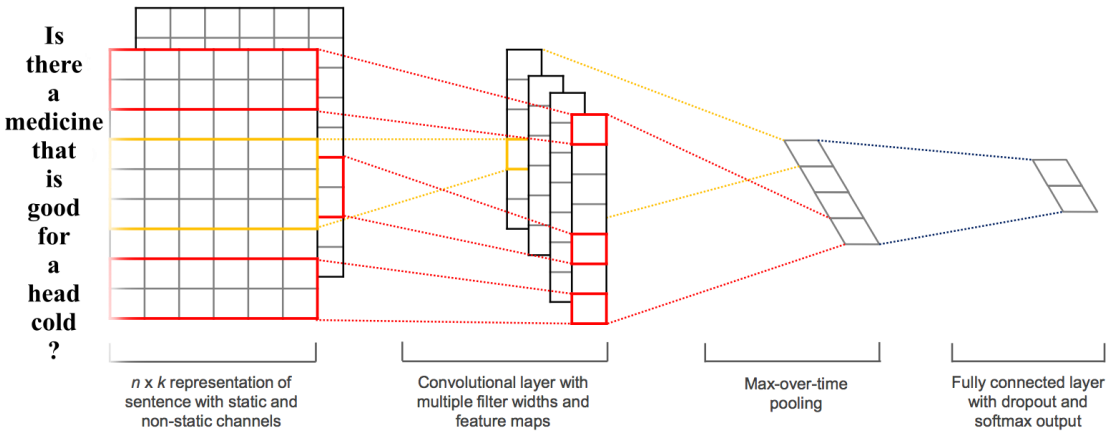


Figure 2: Simple CNN framework

In our implementation, we used a five-layer sequential model with a one-dimensional convolution layer which uses pre-trained GloVe vector embeddings to prepare sequential input for one-dimensional max pooling. Two Dense layers with rectified linear unit and sigmoid activation functions respectively were used for sentiment prediction.

### 3.5 Ethical Consideration

In the cause of our implementation, some ethical issues were discovered and addressed. We took all possible ethical issues into consideration; before, during and after research. We reviewed the European Union's General Data Protection Regulation (GDPR), Irish Data Protection Act, Irish e-Privacy Regulations, and adopted positions from existing research works on similar ethical issues (Garattini et al., 2019).

Learning from earlier works, the ethical strategies adopted for this research includes getting a general consent from Twitter to analyse its user data, privacy cautious data access/storage and de-identification of tweets containing personal identifiers (Garattini et al., 2019). We developed and adhered to a standardized operating guideline.

While we agree that there are divergent views on ethics from existing literatures (Ahmed et al., 2019), it is important to state that this research work is not in violation of any known statutory regulations.

## 4 Results

In this section, we present our findings on how natural language processing for infectious disease surveillance can be strengthened using deep learning models. To provide needed insight, we applied three deep learning classification techniques to a health tweet sentiment dataset. We compared the performance of LSTM, Bi-directional LSTM and CNN frameworks, noting that Bi-directional LSTM performed better than the other two with an accuracy score of 95.4%.

### 4.1 Data

As stated in section 1.3, the data used in this research was gotten from official and non-official sources. We present an overview of the dataset used for this research in the following outlines.

1. **NTCIR-13 MedWeb Dataset:** The Medical Natural Language Processing for Web Document (MedWeb) contains 2,560 classified pseudo tweets for eight symptoms or diseases (Wakamiya et al., 2017). The tweets were classified based on their sentiment; p representing positive sentiment and n representing negative sentiment. Positive sentiment classification means that the tweet is a valid report of a symptom or disease. The diseases/symptoms captured in the dataset are cold, cough, diarrhea, fever, hay fever, headache, influenza and runny nose (Wakamiya et al., 2017). For our analysis, we coded positive sentiments as 1 and all negative sentiments as 0. Also, we combined all symptoms leaving us with two classes; valid (positive) and invalid (negative) tweets. The next two figures below capture the top words for valid and invalid infectious disease tweets.

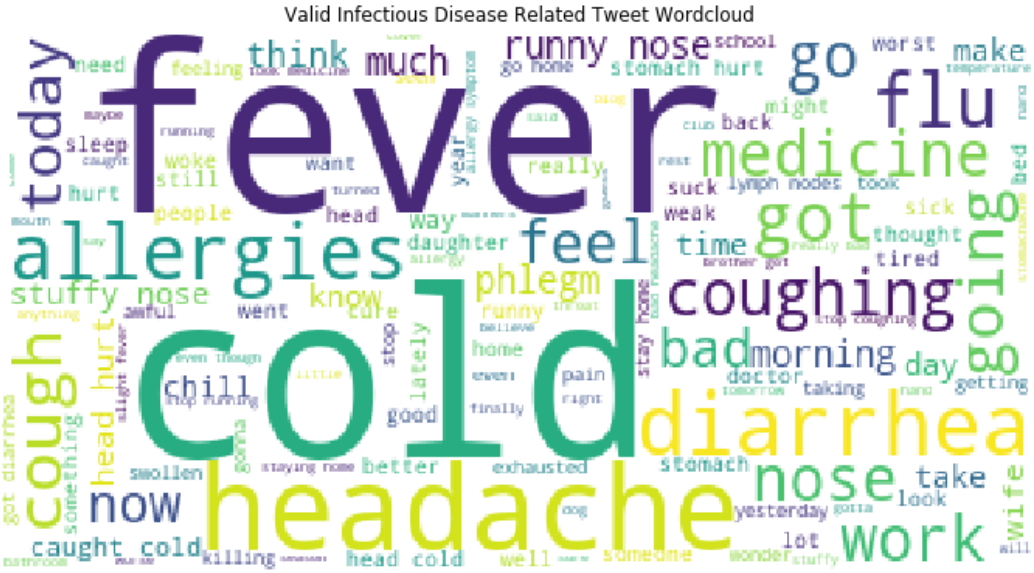


Figure 3: Word cloud for valid infectious disease top words.

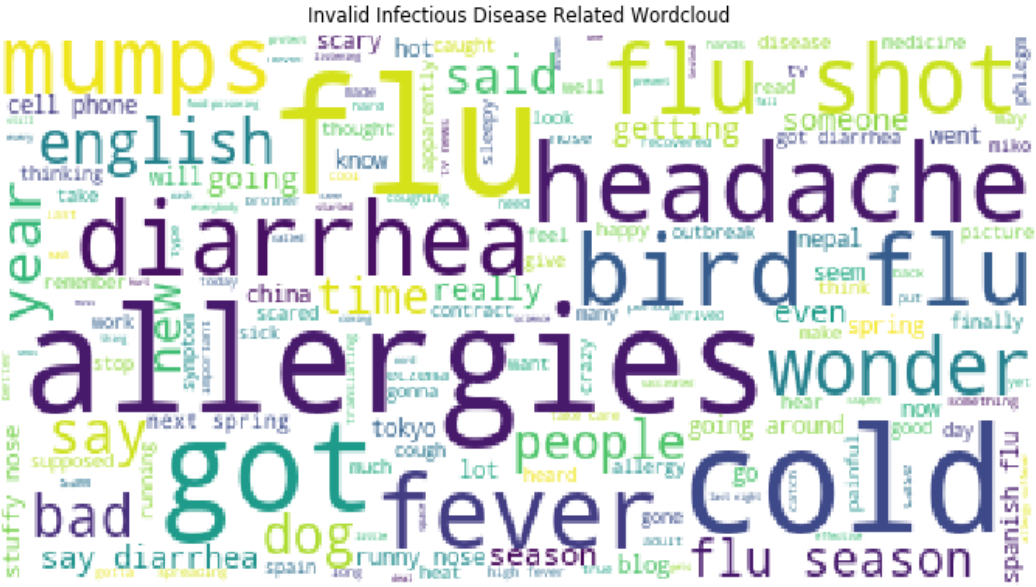


Figure 4: Word cloud for invalid infectious disease top words.

2. **News Articles:** We deployed a python-coded Google pagerank solution to identify top news websites based on the earlier listed symptoms. We then pick top five and scrape these websites for disease or symptoms, extracting numbers, location and time stamps (Yoon et al., 2018).
3. **Twitter Dataset:** Using python's tweepy library, we streamed twitter data using the earlier listed symptoms/diseases as keywords (Ahmed et al., 2019). We extracted tweet text, location and date.

In summary, our models were trained and validated using the MedWeb sentiment classification dataset. The Bi-directional LSTM model being the best performing model was then deployed to classify tweets (Twitter Dataset) and news articles into valid and invalid infectious disease corpus.

## 4.2 Parameters

Having gone through LSTM and CNN implementations of well cited papers, we adopted the parameters that produced best results and manually fine-tuned them where necessary. The parameter captured in the table below seemingly produced the best results to compare the model performances.

| | |
|---|---|
| Epoch | 10 |
| Batch Size | 20 |
| Training/Validation Dataset | 80/20 |
| Dropout | 0.001 |
| Embedding Dimension | 16 |
| Word Embedding | GloVe Embedding and Keras Trained Dataset-based Embedding |

Table 1: Parameters adopted for our LSTM and CNN implementations.

## 4.3 Model Performance and Comparison

In this section, we present the results of our sentiment analysis using the three earlier mentioned frameworks. Averagely, our CNN model achieved a 1% accuracy higher than our LSTM model but performed 11.8% worse than our bi-directional LSTM model. In other words, our bi-directional LSTM model performed best with a 12.8% and 11.8% higher accuracy score than that of our simple LSTM and CNN models respectively.

These results prove that our earlier intuition was not amiss, and that by introducing a bi-directional temporal information flow to our LSTM model, we were able to provide additional context to our network, and this resulted in a faster and better learning. It is safe to say that the 12.8% and 11.8% differences between our bi-directional LSTM model and the two others is a clear difference and not just a coincidence.

Comparing the performance of our CNN model to bi-directional LSTM model, we came up with a logical explanation that the convolutional layer of our CNN model is losing some of the text sequence/order information. We believe the bi-directional LSTM model performed better by quite a noticeable margin because it leverages both the input sequence and its reverse order.

Also, the fact stated above also explains why the bi-directional LSTM outperformed our simple LSTM model. The bi-directional LSTM model takes a step further by encoding the reverse order for every token in the input (Minaee et al., 2019). As with all LSTM based models, previous tokens are also linked.

Noting that the bi-directional model outperformed the other two models, it is pertinent to note that none of the models performed too poorly. The three models had an accuracy score of over 80% and are good enough models when compared to implementations from existing body of works. The table below presents the accuracy score for each of the models.

| **Model** | **Optimal Epochs** | **Optimal Accuracy** | **Average Accuracy** |
|---|---|---|---|
| LSTM | 6 | 85.4 | 84.6 |
| CNN | 2&4 | 87.3 | 85.6 |
| Bi-directional LSTM | 7 | 97.8 | 97.4 |

Table 2: Accuracy score of deep learning neural network models.

## 4.4 Learning Rate

Our research revealed that our models have varying learning rates. This means some models learn faster and start to over-fit quicker than others. In the order of learning rate, we have CNN, LSTM and Bi-directional LSTM.

| **Epoch** | **LSTM** | **CNN** | **Bi-directional LSTM** |
|---|---|---|---|
| 3 | 83.9 | 85.7 | 90.6 |
| 5 | 84.4 | 86.7 | 96.4 |
| 8 | 82.7 | 85.6 | 97.4 |

Table 3: Model accuracy at different epoch sizes

As presented in Table 3 and the plot below, training our CNN model with just a few epochs was enough to get accuracy above 80%. Our CNN model performed best at two and four epochs. We noted that too many epochs reduce our CNN model's accuracy

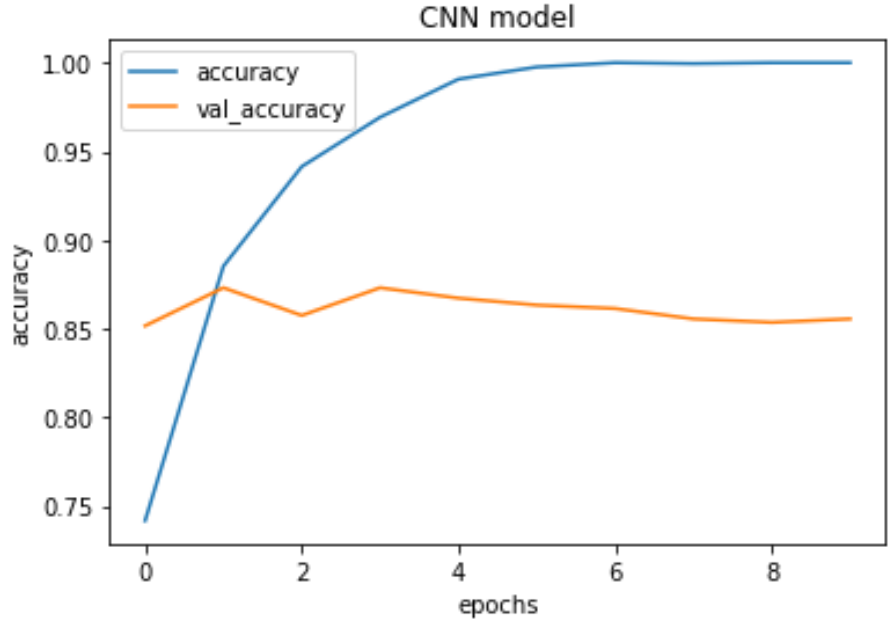


Figure 5: CNN model accuracy plot.

Our LSTM and bi-directional LSTM models' accuracy peaked at six and seven epochs respectively. They performed well with an accuracy of 70% with just a few epochs too. We noted that going above 10 epoch sizes starts reducing the models' accuracy. However, we also noted that an increase in epochs resulted in less overfitting.

The figures below show the described learning rates for our LSTM and bi-directional LSTM models.

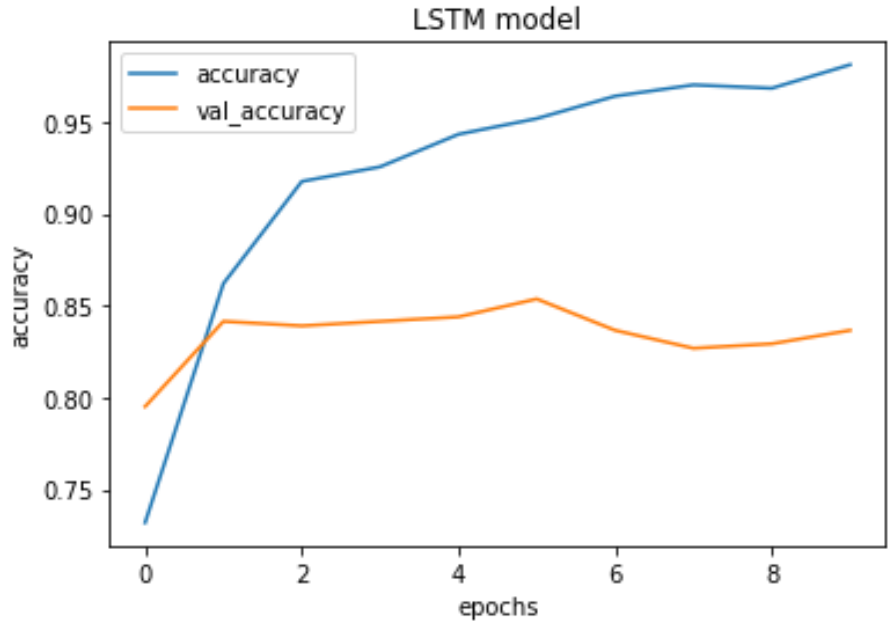


Figure 6: LSTM model accuracy plot.

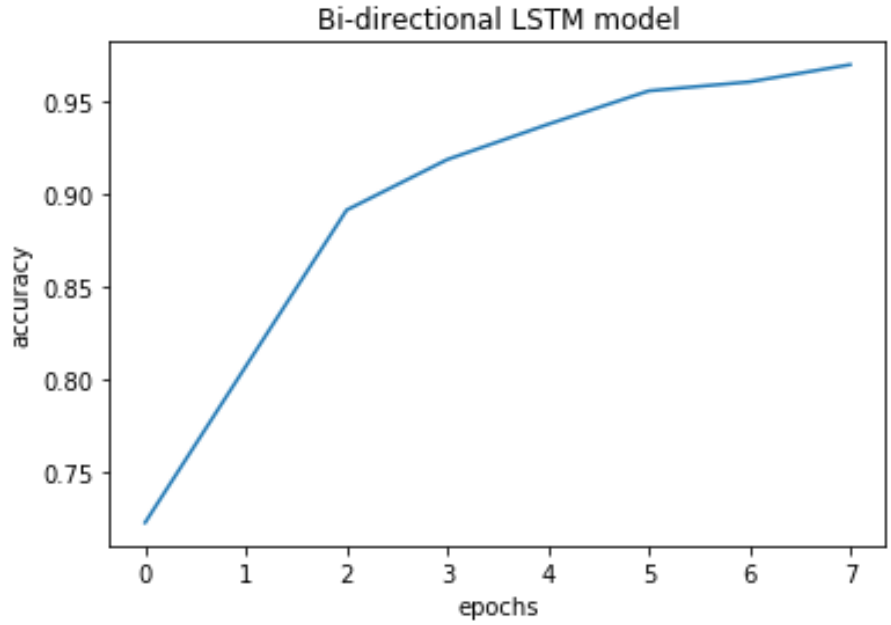


Figure 7: Bi-directional LSTM model accuracy plot.

## 4.5 Dropout Effect

We included dropout layers in our implementation to test its effect on our model performances. We included a dropout layer with 1% and 10% probability interchangeably, all other parameters remaining the same. Table 4 shows the result of applying dropout.

| **Dropout Rate** | **LSTM** | **CNN** | **Bi-directional LSTM** |
|---|---|---|---|
| 1% | 84.6 | 85.6 | 97.4 |
| 10% | 81.2 | 83.9 | 95.3 |

Table 4: Model accuracy at different dropout rate

### 4.6 Pre-trained versus In-Training Embeddings

In addition to using Keras embedding layer to learn the word embeddings during training, we also used the popular pre-trained GloVe word embeddings. While the GloVe word embeddings yielded better accuracy for our CNN model, it resulted in a lower but arguably good accuracy for our LSTM models. It seems it is due to the textual irregularities introduced into the MEdWeb dataset.

Tweets are limited to a maximum of 140 characters and it very likely to have slangs, misspellings, and word abbreviations that are not present in the pre-trained GloVe word embeddings. For obvious reasons, this complicates sentiment analysis.

We also noted that our Bi-directional LSTM model still outperformed CNN model even with GloVe embeddings. We believe this is so because of its reverse encoding learning (Minaee et al., 2019).

Table 5 presents the results of using GloVe pre-trained embeddings as opposed to using Keras in-training embedding layer.

| **Embeddings** | **LSTM** | **CNN** | **Bi-directional LSTM** |
|---|---|---|---|
| GloVe Embedding (Pre-trained) | 79.9 | 85.6 | 89.8 |
| Keras Embedding Layer (In- training) | 84.6 | 81.7 | 97.4 |

Table 5: Model accuracy using pre-trained and in-training embedding techniques

### 4.7 Model Deployment

In this section, we present a sample result of our deployment. Having implemented three models, we deployed the best performing model from the three. We applied the bi-directional model to our streamed Twitter dataset and news articles. The table below shows a sample snapshot of what our result output looks like.

| | |
|---|---|
| I have the worst runny nose today. | 1 |
| Apparently there are allergies in the fall, too. | 0 |
| Is there a medicine that's good for a head cold? | 1 |
| I have a super runny nose. There's no way I can go. | 1 |
| It seems like we're getting less pollen this year, but for those super sensitive to pollen, a small difference doesn't really matter. | 0 |

Table 6: Sample bi-directional LSTM model result

As captured in the table above, valid (positive sentiment) infectious disease tweets are coded as 1 while invalid infectious disease (negative sentiment) tweets are coded as 0.

## 5 Discussion

In this section, we present a brief comparison of our work to similar existing work from existing literature review. A close look at the implementation results of LSTM, CNN and Bi-directional LSTM models, revealed that our models performed better than many. We compare the performances of our model to that of two very closely related work in the field of deep learning-based NLP. The table below captures our comparison.

| Model | Accuracy | Dataset | Reference |
|---|---|---|---|
| LSTM | 80% | SST2 | (Minaee et al., 2019) |
| CNN | 80.2% | SST2 | |
| LSTM and CNN Ensemble | 90% | SST2 | |
| LSTM | 66.7% | Twitter | (Sosa, 2017, p.7) |
| CNN | 72.5% | Twitter | |
| LSTM and CNN Ensemble | 75.2% | Twitter | |

Table 7: Model comparison of related works

We noted that SST2 dataset being a less complicated dataset had a better result than that of its counterpart. However, with our carefully picked parameters, we were able to achieve an accuracy that is over 7% better than any result from the two compared works.

While the authors gave explanations on why they think their ensemble models performed better and should be a go to modelling options, we defy the odds by implementing our bi-directional LSTM model to put forward a new argument that a model's performance is as good as its implementation. By putting the temporal flow of information in both directions of network into use, we were able to achieve a near perfect accuracy of 97.4%.

It's safe to say that with the appropriate combination of parameters, model performances can be improved.

# 6 Conclusion and Future Works

In this paper, we have presented our implementation of three deep learning-based NLP models with the aim of determining the best combination of techniques for infectious disease surveillance using unstructured data. Optimally, our CNN model achieved a 1.9% accuracy higher than our LSTM model but performed 10.5% worse than our bi-directional LSTM model. In other words, our bi-directional LSTM model performed best with a 12.4% and 10.5% higher accuracy score than that of our LSTM and CNN models respectively.

In our bid to establish the best combination of techniques/parameters, we carried out an in-depth investigation on how number of epochs, dropout rate, and word embedding methods in our models affect performance. We were able to establish how these techniques/parameters go a long way in boosting model performance, providing future researchers with a better starting point.

As detailed in our literature review section, there are several other deep learning techniques and frameworks that can still be investigated for a possibility of better performance.

Our models seem to have a better performance than known existing CNN and LSTM models and might be worthwhile applying them to other tasks like text generation. Also, it might be worthwhile to try other types of RNNs aside LSTM and bi-directional LSTM. We think using a GRU or ResNets framework or architecture might yield a better result.

Lastly, it is important to state that while we capped our implementation at identifying valid and invalid infectious disease tweets in this paper, we hope to move forward by comparing structured data from official sources to formulate a working model that predicts infectious disease outbreak more accurately (Chae et al., 2018). We have no doubt that this research will help future implementation of accurate sentiment analysis.

## References

Ahmed, W., Bath, P.A., Sbaffi, L. and Demartini, G. (2019) 'Novel insights into views towards H1N1 during the 2009 Pandemic: a thematic analysis of Twitter data', Health Information & Libraries Journal, 36(1), pp.60-72.

Aiello, A.E., Renson, A. and Zivich, P.N. (2020) 'Social Media–and Internet-Based Disease Surveillance for Public Health', Annual Review of Public Health, 41(1), pp.101-118.

Arsevska, E., Valentin, S., Rabatel, J., de Hervé, J.D.G., Falala, S., Lancelot, R. and Roche, M. (2018) 'Web monitoring of emerging animal infectious diseases integrated in the French Animal Health Epidemic Intelligence System', PLOS One, 13(8), pp.1-25.

Carrillo-de-Albornoz, J., Vidal, J.R. and Plaza, L. (2018) 'Feature engineering for sentiment analysis in e-health forums', PLOS One, 13(11), pp.1-25.

Chae, S., Kwon, S. and Lee, D. (2018) 'Predicting infectious disease using deep learning and big data', International Journal of Environmental Research and Public Health, 15(8), pp.1-20.

Du, J., Zhang, Y., Luo, J., Jia, Y., Wei, Q., Tao, C. and Xu, H. (2018) 'Extracting psychiatric stressors for suicide from social media using deep learning', BMC medical informatics and decision making, 18(2), pp.78-87.

Edo-Osagie, O., Smith, G., Lake, I., Edeghere, O. and De La Iglesia, B. (2019) 'Twitter mining using semi-supervised classification for relevance filtering in syndromic surveillance', PLOS One, 14(7), pp.1-29.

Ertem, Z., Raymond, D. and Meyers, L.A. (2018) 'Optimal multi-source forecasting of seasonal influenza', PLOS Computational Biology, 14(9), pp.1-16.

Fast, S.M., Kim, L., Cohn, E.L., Mekaru, S.R., Brownstein, J.S. and Markuzon, N. (2018) 'Predicting social response to infectious disease outbreaks from internet-based news streams', Annals of Operations Research, 263(2), pp.551-564.

Feldman, J., Thomas-Bachli, A., Forsyth, J., Patel, Z.H. and Khan, K. (2019) 'Development of a global infectious disease activity database using natural language processing, machine learning, and human expertise', Journal of the American Medical Informatics Association, 26(11), pp.1355-1359.

Garattini, C., Raffle, J., Aisyah, D.N., Sartain, F. and Kozlakidis, Z. (2019) 'Big data analytics, infectious diseases and associated ethical impacts', Philosophy & Technology, 32(1), pp.69-85.

Gibbons, J., Malouf, R., Spitzberg, B., Martinez, L., Appleyard, B., Thompson, C., Nara, A. and Tsou, M.H. (2019) 'Twitter-based measures of neighborhood sentiment as predictors of residential population health', PLOS One, 14(7), pp.1-19.

Goel, R., Valentin, S., Delaforge, A., Fadloun, S., Sallaberry, A., Roche, M. and Poncelet, P. (2020) 'EpidNews: Extracting, exploring and annotating news for monitoring animal diseases', Journal of Computer Languages, 56, pp.1-12.

Joshi, A., Karimi, S., Sparks, R., Paris, C. and Macintyre, C.R. (2019) 'Survey of Text-based Epidemic Intelligence: A Computational Linguistics Perspective', ACM Computing Surveys (CSUR), 52(6), pp.1-19.

Joshi, A., Sparks, R., Karimi, S., Yan, S.L.J., Chughtai, A.A., Paris, C. and MacIntyre, C.R. (2020) 'Automated monitoring of tweets for early detection of the 2014 Ebola epidemic', PLOS One, 15(3), pp.1-10.

Minaee, S., Azimi, E. and Abdolrashidi, A. (2019) 'Deep-sentiment: Sentiment analysis using ensemble of CNN and bi-LSTM models.', New York University, 1(1), pp.1-6.

Naoui, M.A., Lejdel, B., Ayad, M. and Belkeiri, R. (2020) 'Integrating deep learning, social networks, and big data for healthcare system', Bio-Algorithms and Med-Systems, 16(1), pp. 21-30, De Gruyter. doi: 10.1515/bams-2019-0043.

Sosa, P.M. (2017) 'Twitter sentiment analysis using combined LSTM-CNN models.', Zugriff AM, 10(1), pp.1-9.

Șerban, O., Thapen, N., Maginnis, B., Hankin, C. and Foot, V. (2019) 'Real-time processing of social media with SENTINEL: A syndromic surveillance system incorporating deep learning for health classification', Information Processing & Management, 56(3), pp.1166-1184.

Wakamiya, S., Morita, M., Kano, Y., Ohkuma, T. and Aramaki, E. (2017), 'Overview of the NTCIR-13: Medweb task.', In Proceedings of the NTCIR-13 Conference, 5-8 December 2017, pp. 40-49.

Wang, X., Liu, Y., Sun, C.J., Wang, B. and Wang, X. (2015), 'Predicting polarities of tweets by composing word embeddings with long short-term memory', In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 26-31 July 2015, pp. 1343-1353, ACL Anthology. doi: 10.3115/v1/P15-2.

Yoon, J., Kim, J.W. and Jang, B. (2018) 'DiTeX: Disease-related topic extraction system through internet-based sources', PLOS one, 13(8), pp.1-16.