Attention-based Recurrent Neural Networks (RNNs) for Short Text Classification: An Application in Public Health Monitoring*

Oduwa Edo-Osagie[1], Iain Lake[1], Obaghe Edeghere[2], and Beatriz De La Iglesia[1]

[1] University of East Anglia, Norwich, United Kingdom
{o.edo-osagie, i.lake, b.iglesia}@uea.ac.uk
[2] Public Health England, Birmingham, United Kingdom
obaghe.edeghere@phe.gov.uk

Abstract. In this paper, we propose an attention-based approach to short text classification, which we have created for the practical application of Twitter mining for public health monitoring. Our goal is to automatically filter Tweets which are relevant to the syndrome of asthma/difficulty breathing. We describe a bi-directional Recurrent Neural Network architecture with an attention layer (termed ABRNN) which allows the network to weigh words in a Tweet differently based on their perceived importance. We further distinguish between two variants of the ABRNN based on the Long Short Term Memory and Gated Recurrent Unit architectures respectively, termed the ABLSTM and ABGRU. We apply the ABLSTM and ABGRU, along with popular deep learning text classification models, to a Tweet relevance classification problem and compare their performances. We find that the ABLSTM outperforms the other models, achieving an accuracy of **0.906** and an $F1$-score of **0.710**. The attention vectors computed as a by-product of our models were also found to be meaningful representations of the input Tweets. As such, the described models have the added utility of computing document embeddings which could be used for other tasks besides classification. To further validate the approach, we demonstrate the ABLSTM's performance in the real world application of public health surveillance and compare the results with real-world syndromic surveillance data provided by Public Health England (PHE). A strong positive correlation was observed between the ABLSTM surveillance signal and the real-world asthma/difficulty breathing syndromic surveillance data. The ABLSTM is a useful tool for the task of public health surveillance.

Keywords: Syndromic Surveillance · Sequence Modelling · Deep Learning · Natural Language Processing

1 Introduction

Text classification is a well established field related to Natural Language Processing (NLP) and data mining which has seen a lot of activity. Usually, literature published in this domain studies medium to large bodies of text such as film and internet reviews as well as news articles. However, with the proliferation of social media as a viable source of data for data mining, the issue of Tweet classification has become more prominent. Tweet classification is a natural yet specific extension of the text classification problem. Tweets are very short pieces of text, each limited to 280 characters only. Forms of expression vary when they are constrained in this way. This means that although we can apply existing text classification techniques, we have to pay special attention to the concise nature of Tweets so that it does not negatively impact the workings of these techniques.

We are motivated by a real world problem. This is the analysis of ***Tweets*** for the purpose of public health surveillance. Specifically, we have investigated the use of Tweets to obtain a signal for a given syndrome [6], that is asthma and/or difficulty breathing. For this, we collected Tweets related to our syndrome of interest - asthma and/or difficulty breathing - using keywords. Unfortunately, as explained previously [6], many Tweets contain terms like "*asthma*" or "*can't breathe*" but are not actually related to individuals expressing concern over asthma or difficulty breathing. Hence the classification of relevant/irrelevant Tweets for this particular syndrome is our problem. For some context, examples of Tweets that contain the keyword "*asthma*" include "*oh I used to have asthma but I managed to control it with will power*" or "*Does your asthma get worse when you*

* Supported by Health Protection Research Unit, Public Health England

exercise?". However, we do not consider these Tweets as relevant for our purposes. On the other hand, Tweets such as "*why is my asthma so bad today?*" express a person currently affected and will be considered as relevant.

Text classification using neural networks has been widely investigated and found to yield positive results [6,14,15]. These neural network models look at a document as a whole, examining the interrelations of words and word vectors in the document without giving any words special treatment. However, we believe that texts usually contain a number of *keywords* that inform the meaning and sentiment of the whole text. Such keywords should be used to inform the classification process. To this end, we propose to apply an attentive approach, which makes use of an encoder-decoder architecture, to short text classification, and we demonstrate its value specifically in the context of Tweet classification for public health monitoring.

Attentive neural networks pioneered for machine translation [9] have recently seen success in a range of tasks ranging from question answering, speech recognition to image captioning [1,3,32]. We propose adapting the attention mechanism for short text classification tasks such as Tweet analysis. We apply our attention mechanism using two popular RNN setups - Long Short Term Memory (LSTM) [10] and Gated Recurrent Unit (GRU) [2] networks in order to derive attention-based variants for comparisons. We call our attention-based LSTM, ***ABLSTM*** and our attention-based GRU, ***ABGRU***. After we employ our attention-based RNN classifiers to Tweet classification to generate an 'activity' signal over time, we compare our results to the activity recorded by syndromic surveillance systems maintained by Public Health England (PHE).

Our proposed approaches combine the characteristics of both deep learning and traditional classification algorithms. We combine the self-learning and intrinsic pattern recognition capabilities of deep learning with the use of keywords in classification employed by traditional classification methods. Through our experiments, we find that the *ABLSTM* and *ABGRU* are able to identify keywords in a Tweet relevant to its meaning and improve classification accuracy. As an example, Figure 1 shows a Tweet heatmap of perceived word importance generated with our *ABLSTM* network. The darker areas/words represent words which the model deems key to the message of the Tweet. We can see that the model does a good job of recognizing that *swelling*, *throat* and *difficulty*, *breathing* are important for determining whether the Tweet is relevant to our health context. We also found that using this attention-based approach to syndromic Tweet classification yields a signal that correlates well with the signal recorded by syndromic surveillance systems put in place by PHE for England.

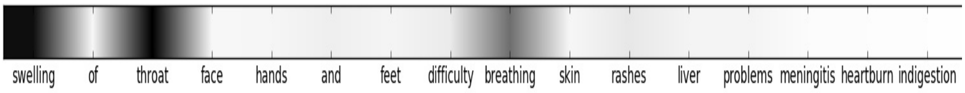

Fig. 1. Heatmap showing weights placed on words in a Tweet by our attentive RNN model

2 Related Work

The problem of text classification has a long history. In the 1960s, it was often referred to as *text categorization*, and was approached by employing a set of hand-crafted logical rules based on the specific language and its grammar and idiosyncracies [24]. In the 1990s, the study of automatic text categorization became more prominent. The approaches used in these studies involved the use of pre-marked data to automatically learn discriminatory rules for classification with which new samples could then be classified [12,34,33,30,18]. This was the precursor to the approaches used today. A number of learning algorithms have been applied to text which had been vectorised using a *tf*idf* weighting method, including Support Vector Machines [12], regression models [34], nearest neighbour classification [33], Bayesian models [30], and inductive learning [18]. These algorithms assume that independent key words or phrases are important to the text category and extract vector features representing those key words or phrases using statistical methods [29]. These methods generally yield successful results but the assumption is an oversimplification that brings some shortcomings. While independent keywords and phrases are important, there are other linking words which also give meaning to a text. The way words relate can provide context and disambiguation and without this, we potentially lose some information.

Recently, deep-learning-based methods have seen a lot of success for text classification. This is mostly due to the fact that such methods can automatically and effectively learn underlying features and interrelationships in data. Some authors [14,16] have adapted Convolutional Neural Networks (CNNs), which are normally used for images, to the task of text classification. They propose a semi-supervised approach by first learning word or region embeddings from a large unstructured corpus to be used as inputs to the CNN. The work was also expanded to use RNNs for the generation of region embeddings and text classification [15]. Lee and Dernoncourt [17] make use of an RNN for short-text classification. We previously employed CNNs and RNNs for the task of Tweet classification for syndromic surveillance and compared their performance [6]. Similar work was carried out by Hankin et al. [7] in the USA. Both our work and that of Hankin et al. attempt to use deep learning models for Tweet and news article classification in order to detect symptoms reported on these platforms. The identified reported cases can then be aggregated to create an estimate of the prevalence of the syndrome(s) under investigation.

While deep learning models have seen widespread success, they treat all the words as a block of input without giving any words or phrases special treatment. We would like to leverage the advantages of both the classical text categorization approaches, which employ keywords, and the modern deep learning approaches, which learn underlying relationships, for short text (Tweet) classification. Miyato et al. [23] make use of adversarial training to build semi-supervised text classification models. They make use of LSTMs and BLSTMs, making small changes or perturbations to the word embeddings during training. This approach is also similar to the semi-supervised transductive SVM approach [13] in that both families of methods push the decision boundary far from training examples. Zhou et al. [36] make use of attention BLSTMs for entity relation classification, which is the task of finding relations between pairs of nominal values. That is useful for applications such as information extraction and question answering. Zhang et al. [35] proposed an attention network with a hierarchical architecture for document classification. The hierarchical structure of the attention mechanisms is intended to mirror the hierarchical nature of documents. As such, it involves two levels of attention applied at the sentence level and the word level. It is better suited to large-scale text classification tasks that short text classification problems such as Tweet classification.

Our work is also related to Du and Huang [5] who used a BLSTM with attention for text classification. However, in their work, they compute the attention or weight of a word as the similarity between the embedding for that word and the hidden state of the BLSTM at the time step for that word. Rather than computing the attention from the hidden state, we opt for learning a function to approximate the values of the attention vector through back-propagation. Hence, the attention vector is a parameter to be learned directly. Furthermore, for the input to the classifier, Du and Huang [5] concatenate the attention vector and BLSTM output states. We propose the computation of a Tweet (or document) representation from the attention weights and hidden state. Such a representation could then also be used in a similar manner as a document embedding.

3 Model

In this section, we describe the proposed attention-based RNN. The attention RNN can be broken down into four parts:

1. ***Word Embedding***: This step vectorises the Tweet. It involves mapping each word in the Tweet to a fixed-dimension word embedding. In our work, we make use of GloVe embeddings which we build from a large unlabelled corpus of Tweets.
2. ***RNN***: Takes the output of the previous step as input. The RNN learns high level features from the given input.
3. ***Attention Layer***: Produces a weight vector which it uses in conjunction with the output states of the RNN to form a new Tweet representation.
4. ***Classification***: The attention-powered vector representation of the Tweet is fed into a classifier to obtain a prediction

Figure 2 shows a simple illustration of the workflow of the attentive RNN model. Each component of the process will subsequently be explored in more detail below.

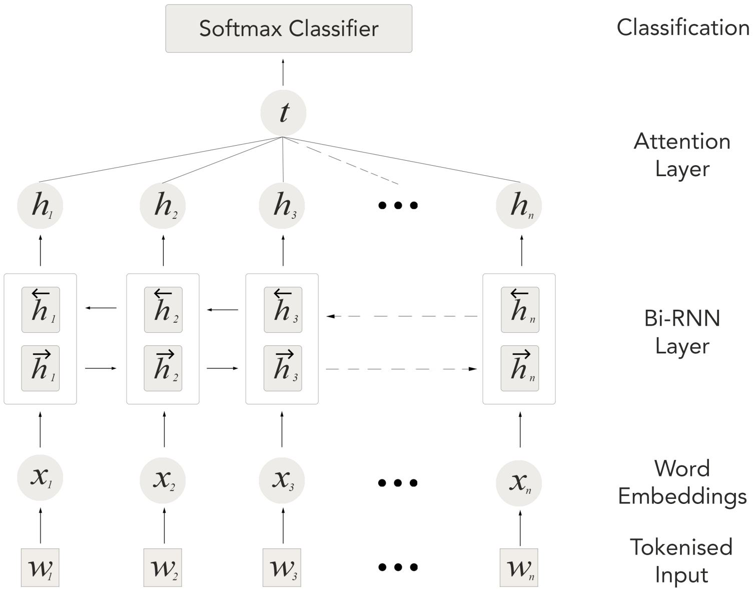

Fig. 2. Attention-based RNN model

3.1 Word Embeddings

Word embeddings (sometimes referred to as word vectors) are a powerful distributed representation of text learned using neural networks that have been shown to perform well in similarity tasks [11]. They encode semantic information of words in dense low-dimensional vectors. There are many different ways to learn word embeddings [26,22,19]. After learning, an embedding matrix X of size $|V| \times d$ is produced where V is the set of all the words in our vocabulary and d is the dimension of each word embedding. Given a Tweet T consisting of n words, $T = \{w_1, w_2, ..., w_n\}$, each word w_i is converted to a real-valued vector x_i by performing a lookup from the embedding matrix X. For this work, we built GloVe embeddings [26] from a set of 5 million unlabelled Tweets.

3.2 RNNs

RNNs are a category of neural networks that incorporate sequential information. That is to say, while in a traditional neural network, inputs are independent, in RNNs each node depends on the output of the previous node. This is particularly useful for sequential data such as text where each word depends on the previous one. While in theory, RNNs can make use of information in arbitrarily long lengths of text, in practice, they are limited to looking back only a few steps due to the vanishing gradient problem which occurs during the back-propagation algorithm. When tuning the parameters of the network due to long sequences of matrix multiplications, gradient values shrink fast and gradient contributions from earlier neurons become zero. As a result of this, information from earlier inputs (words in the text) do not contribute to the overall algorithm. Long Short Term Memory (LSTM) networks [10] and Gated Recurrent Unit (GRU) [2] networks are flavours of the RNN architecture which make use of a gating mechanism to combat the vanishing gradient problem. Succinctly, they are a solution for the short-term memory problem that simple RNNs possess in which they cannot properly update and learn weights for earlier inputs in a sequence. LSTMs and GRUs are very similar, the main difference is that GRUs have less parameters than LSTMs. As such, GRUs are faster and have been observed to exhibit better performance on some smaller datasets [2]. However, LSTMs have been shown to be better at learning in general [31].

Long Short Term Memory (LSTM) For simplicity, we make use of an LSTM with only one layer. The network has an input layer x, hidden layer h, LSTM cell state c and output layer y. Input to the network at timestep t is $x(t)$, output is denoted as $y(t)$, hidden layer state is $h(t)$ and LSTM cell state is $c(t)$. The LSTM cell state is controlled by a gating mechanism as highlighted above briefly. Each cell consists of the following gates which interact with each other to dictate the overall cell state:

- input gate (i)
- forget gate (f)
- write gate (g)
- output gate (o)

Each of these gates has its own weights and biases and is a function of the previous time step's hidden state $h(t-1)$. The hidden state of a layer can then be computed as a function of the cell state as shown below:

$$c(t) = f(t) \cdot c(t-1) + i(t) \cdot g(t) \tag{1}$$

$$h(t) = o(t) \cdot tanh(c(t)) \tag{2}$$

For the sake of brevity and simplicity of our equations, let us assume that there is only one hidden layer l so that we do not have to specify different equations for the different edge cases that would come with multiple layers, such as when execution is in the first layer and has no previous layer or when it is in a middle layer or the final layer. In the real world scenario, this is not the case as each hidden layer state is influenced by the hidden state in the previous time step as well as the state of the previous hidden layer. To adapt this, one may simply add the product of the weights and input of the previous layer to each activation function. The activation functions for the gates are computed as:

$$f(t) = sigmoid(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + b_f) \tag{3}$$

$$g(t) = tanh(W_{xg} \cdot x_t + W_{hg} \cdot h_{t-1} + b_g) \tag{4}$$

$$i(t) = sigmoid(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + b_i) \tag{5}$$

$$o(t) = sigmoid(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + b_o) \tag{6}$$

where W_{pq} are the weights that map p to q and b_p refers to the bias vector of p. For example, if we look at equation 3, W_{xf} refers to the weights going from input x to the forget gate f and so on while b_f refers to the bias of the forget gate f.

Gated Recurrent Unit (GRU) Again, for the sake of brevity and simplicity of our equations, let us assume that there is only one hidden layer l. The GRU cell state is controlled by a gating mechanism similar to the LSTM. Each cell consists of the following gates which interact with each other to dictate the overall cell state:

- update gate (z)
- reset gate (r)

The gates can be formalised as follows:

$$z(t) = sigmoid(W_{xz} \cdot x_t + W_z \cdot h_{t-1} + b_z) \tag{7}$$

$$r(t) = sigmoid(W_{xr} \cdot x_t + W_r \cdot h_{t-1} + b_r) \tag{8}$$

The hidden state of a layer is computed as a function of the input and gates as shown below:

$$h(t) = z(t) \cdot h(t-1) + (1 - z(t-1)) \cdot tanh(W_x + r(t) \cdot W_h \cdot h(t-1)) \tag{9}$$

where W_{pq} are the weights that map p to q and b_p refers to the bias vector of p. For example, if we look at equation 7, W_{xz} refers to the weights going from input x to the update gate z and so on, while b_z refers to the bias of the update gate z and W_z refers to the weights for the update gate itself.

Bi-directional Networks The above RNNs process sequences in time steps with subsequent time steps taking in information from the hidden state of the previous time steps. This means that they ignore future context. Bi-directional RNNs (Bi-RNNs) extend this by adding a second layer where execution flows in reverse order [28]. Hence, each layer in a Bi-RNN has two sub-layers: one moving forward in time steps and one moving backwards in time steps. To compute the hidden state $h(t)$ of a Bi-RNN layer, we perform an element-wise sum of the hidden states computed from both its sublayers:

$$h(t) = \overrightarrow{h(t)} \bigoplus \overleftarrow{h(t)} \tag{10}$$

where $\overrightarrow{h(t)}$ and $\overleftarrow{h(t)}$ are the hidden states of the forward and backward traversals of the bi-directional RNN.

3.3 Attention

In this section, we describe the attention mechanism used. The Bi-RNN layer takes in a sequence of vectors for each of the words in an n-worded Tweet $\{x_1, x_2, ..., x_n\}$, resulting in hidden states $\{h_1, h_2, ..., h_n\}$ where h_i is a vector derived from equation 10. That is, the hidden state of the Bi-RNN for the word w_i is h_i. Let H be a matrix containing these vectors such that $H \in \mathbb{R}^{k \times n}$ where k is the number of neurons in the hidden layer. The Tweet representation t is derived by taking a weighted sum of the hidden vectors with the attention weight for the relevant words. The attention weights, α such that α_i represents the attention weight for w_i, are obtained from trainable parameters and so are adjusted as the optimization algorithm trains the network:

$$M = tanh(H) \tag{11}$$

$$\alpha = softmax(w^T M) \tag{12}$$

$$t = M\alpha^T \tag{13}$$

where w is a trainable parameter in the network and w^T is its transpose. w, α and t have the dimensions k, n and k respectively. Finally, the hyperbolic tangent function (tanh) is applied to t, the Tweet attention vector, in order to squash it between the range [-1,1] and make it easier to train with the network:

$$t^* = tanh(t) \tag{14}$$

3.4 Softmax Classifier

Once, the new attention-based representation for the Tweet has been obtained, it is passed to a softmax classifier to make the class prediction. The softmax classifier predicts a class y from a discrete set of m classes Y by calculating the probability that the observed Tweet belongs to each class, $P(y|T)$, and assigning the Tweet the class with the highest probability:

$$P(y|T) = softmax(W_s t^* + b_s) \tag{15}$$

$$y = argmax_y P(y|T) \tag{16}$$

where W_s represents the softmax classifier network weight and b_s represents its bias term. The loss function used to train the entire network is the cross entropy loss function [4]:

$$L = -\frac{1}{m}\sum_{i}^{m} e_i log(o_i) \tag{17}$$

where L estimates loss between the observed and expected values. e is a one-hot encoded vector of the ground truth for t and o is the probability of each class being the target according to the softmax classifier.

4 Experiments and Results

We evaluate the performance of our attention-based RNN for Tweet classification. First, we evaluate our proposed approach's ability to automatically classify Tweets as **"relevant"** or **"irrelevant"** based on whether they associate with an individual expressing concern or discomfort over asthma/difficulty breathing or its symptoms. In these experiments, we compare the classification ability of our proposed approach to that of popular existing approaches. Next, we apply our attention-based RNN classifier to a continuous period of collected unlabelled Twitter data in order to generate a public health signal representing Twitter activity for asthma/difficulty breathing. We then compare this signal to data from real-world syndromic surveillance systems for evaluation.

Table 1. Performance of different classifiers on Tweet relevance classification tast.

Classifier	Metric	
ABGRU	*Accuracy*	0.900
	Precision	0.734
	Recall	0.656
	*F*1	0.682
	*F*2	0.666
ABLSTM	*Accuracy*	**0.906**
	Precision	0.752
	Recall	**0.672**
	*F*1	**0.710**
	*F*2	**0.687**
Convolutional Neural Network (CNN)	*Accuracy*	0.850
	Precision	0.507
	Recall	0.562
	*F*1	0.533
	*F*2	0.550
Recurrent Neural Network (LSTM)	*Accuracy*	0.889
	Precision	**0.762**
	Recall	0.557
	*F*1	0.644
	*F*2	0.589

4.1 Tweet Relevance Classification

Tweets were collected using the official Twitter streaming Application Programmer's Interface (API). The streaming API contains parameters which can be used to restrict the Tweets obtained (e.g. keyword search, where only Tweets containing the given keywords are returned). In conjunction with epidemiologists from Public Health England (PHE), we built a set of keywords likely to be connected to the symptoms for asthma/difficulty breathing syndrome. We then expanded on this initial set using synonyms from regular thesauri as well as from the urban dictionary in order to capture some of the more colloquial language used on Twitter. This set of keywords was used to restrict our Tweet collection. We also only collected Tweets we found to be geolocated to the UK, marked as originating from a place in the UK or marked as originating from a profile with its time zone set as the UK as our syndromic surveillance problem is in fact restricted to the the UK. The collected Tweets had to be cleaned with the removal of duplicates and retweets and replacing URLs and user mentions with the tokens "<URL>" and "<MENTION>" respectively. We considered implementing measures to prevent the false amplification of signals from users tweeting multiple times, potentially about the same thing. After further inspection however, we found that this was not necessary as it is discouraged by Twitter [8]. A similar concern existed for a single user posting Tweets across multiple accounts but this is also handled by Twitter's anti-spam efforts [27].

Five million Tweets were collected in total. 8000 of these Tweets were randomly selected and labelled to be used for development and experimentation. Tweets were labelled as relevant if they declared or hinted at an individual displaying symptoms pertaining to respiratory difficulties or asthma. The labelling was done by three volunteers. A first volunteer initially labelled the Tweets. A second volunteer checked the labels and flagged up any Tweets with labels that they did not agree with. These flagged Tweets were then sent to the third volunteer who then decided on which label to use. 23% of the labelled Tweets were labelled as relevant while 77% were labelled as irrelevant. This labelled dataset was then partitioned into a 70:30 training-test split. The 5 million Tweets were used to construct GloVe word embeddings while the labelled Tweets were used for experimentation. To assess the models under evaluation, accuracy can be a misleading metric as it may only be reflecting the prevalence of the majority class which is especially problematic in this application, as our dataset is quite unbalanced. Our aim is to detect Tweets which might suggest cases of a syndrome under surveillance (which for the purposes of this study was symptoms of asthma/difficulty breathing). As this is a health surveillance application, we need to prioritise that relevant Tweets are kept. We would like to reduce the number of irrelevant Tweets but not at the expense of losing the relevant Tweets in the signal. In essence, errors are not of equal cost for our application. Relevant Tweets that are classified as irrelevant (False Negative (FN) errors) should have a higher cost and hence be minimised; we can have more tolerance of irrelevant Tweets classified as relevant (False Positive (FP) errors). These subtleties are well captured by additional measures of model performance such as *Recall*, which can be interpreted as the probability that a relevant Tweet is identified by the model and *Precision*, which is the probability that a Tweet predicted as relevant is indeed relevant. The *F-measure* (sometimes referred to as *F-score*) combines precision and recall together in a meaningful way. The formula for positive real β is defined as:

$$F_\beta = (1 + \beta^2) \times \frac{Precision \times Recall}{(\beta^2 \times Precision) + Recall}. \tag{18}$$

The traditional F-measure or balanced $F1$-score [21] uses a value of $\beta = 1$. A variation of this, the F_2 measure, which uses $\beta = 2$, is more suited to our purpose as it weighs recall higher than precision. For this reason, in addition to accuracy, we also examine the $F1$-score for an insight into classification power and the $F2$-score for its utility in the context of syndrome detection. We implemented and applied our $ABLSTM$ and $ABGRU$ networks to the Tweet relevance classification task. The hyperparameters of the attention networks were selected using grid search. The dimension of our word vectors d was 200. The hidden layer size k was also 200. The learning rate of the optimization algorithm was 0.001. The dropout rate was set to 0.3 and the networks were trained for 50 epochs. The other parameters such as weights and biases were initialised randomly. We compare the results of applying both flavours of our proposed model and we also compare them to established deep learning text classification methods as a baseline. For this, we implemented the text classification CNN by Kim [16] and the short-text classification RNN by Nowak et al. [25]. The results of these comparisons are shown in table 1. Note that all our results were computed from the test partition.

We found that the attentive RNNs outperformed the other architectures, with the ABLSTM being the stronger attentive RNN. As shown in section 3.2, the gating mechanism used by the GRU is smaller and less complex than that of the LSTM. This means that ABGRU is faster but not quite as accurate as the ABLSTM. The LSTM RNN was seen to achieve a higher precision than the ABLSTM and ABGRU but it fell behind in terms of recall. Its recall was quite low and negatively impacted its overall performance. In effect, this translates to it being more likely to find negative class examples which were the majority class in the dataset and it suggests that it may be more suited to balanced datasets. However, our task of syndrome monitoring using social media deals with highly unbalanced data as most social media posts are not about health reporting. We also observed that the text CNN scored the worst in every metric so it performed quite badly at the Tweet relevance classification, even though it had perform well at other text classification tasks [16]. CNNs are good at extracting position-invariant features in space. They represent text as a 2D matrix made up of the word vectors of the constituent words from which the CNN learns which regions are important. However when applied to short Tweets, CNNs do not have a lot of salient spatial information to work with and so do not perform nearly as well as they would when applied to larger texts.

4.2 Document Embedding Capabilities

As was mentioned in section 3, the output of the attention layer is a Tweet attention vector, t. This vector summarizes the input word vectors while putting emphasis on important words. t is subsequently used as a vector representation for the Tweet in the classification part of the model. As such, the described model could also be applied to documents in other problems to create meaningful embeddings for them. We collected a random sample of Tweets, computed their attention vectors and performed t-distributed stochastic neighbour embedding (t-SNE) [20] dimensionality reduction to reduce their dimensions to 2. We then plotted these 2D attention vectors, shown in figure 3 in order to spatially visualize them. We found that Tweets with similar meanings and words appeared to be clustered together and away from irrelevant Tweets. In fact, it could be possible from 3 to draw a decision boundary line that roughly separates both classes. Below the red line, we see Tweets which are symptomatic of the asthma/dificulty breathing syndrome. Above the line, we see Tweets which may contain keywords related to asthma/difficulty breathing but are not expressing concern or suffering. It is also worth noting that "*wheezing*" is often used as slang to exaggerate laughter. Social media contains a lot of slang. The Tweet attention vectors capture the semantics of the different contexts of slang words, such as "*wheezing*", and this boosts its discriminatory ability. The attention vectors give us a semantic and discriminatory vector representation for our Tweets. In addition to its utility for short text classification, the attentive model we have described has the added ability to create useful document embeddings.

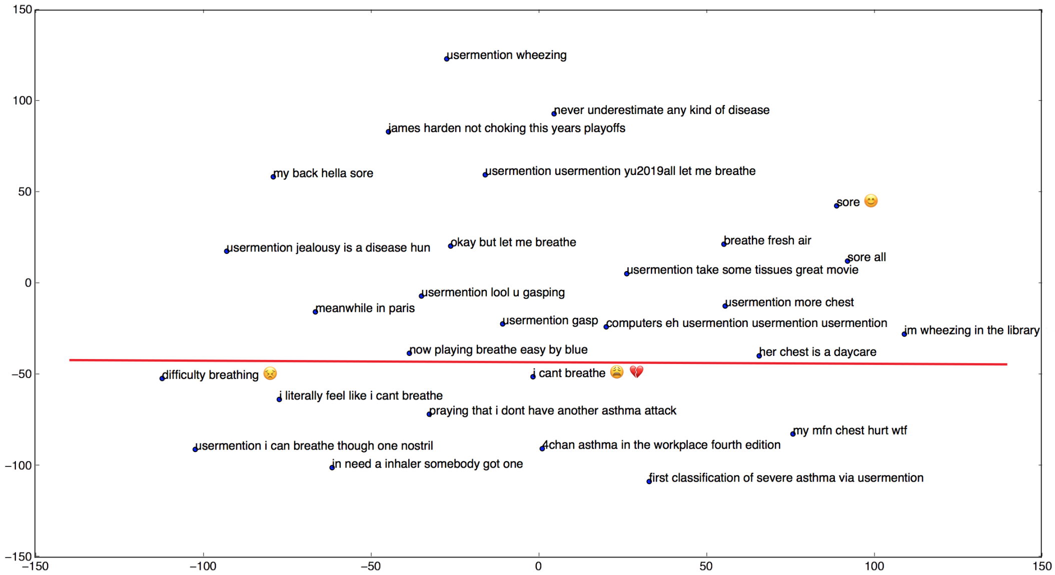

Fig. 3. Plot of Tweets representative of distances in embedding space. The axes represent t-SNE dimensional values.

Table 2. Pearson correlations and P-Values for extracted Twitter signals with syndromic surveillance signals.

	Twitter with ABLSTM filtering	Twitter with LSTM filtering	Twitter without filtering
GPOOH Asthma/ Wheeze/ Difficulty Breathing	0.792($p <$ 0.001)	0.637($p <$ 0.001)	0.555($p <$ 0.001)
NHS 111 Difficulty Breathing	0.830($p <$ 0.001)	0.586($p <$ 0.001)	0.361($p <$ 0.001)
NHS 111 Diarrhoea	0.207($p =$ 0.09)	0.125($p =$ 0.3)	0.027($p =$ 0.8)

4.3 Syndromic Surveillance evaluation

While we have shown in the previous section that the ABLSTM performs well at the task of Tweet relevance classification, we would like to demonstrate its utility to generate a signal for syndromic surveillance. To do this, we employ our ABLSTM to mine relevant Tweets for asthma/difficulty breathing in the UK. We then compare the results of our ABLSTM with recorded public health data. PHE runs a number of syndromic surveillance systems across England. For this experiment, we used Tweets outside of the labelled dataset used to build the classifier. We used unlabelled Tweets collected continuously between June 21, 2016 and August 30, 2016. We performed comparisons with relevant anonymised data from PHE's syndromic surveillance systems for this time period. PHE systems use primary care (general practitioner in hours and out of

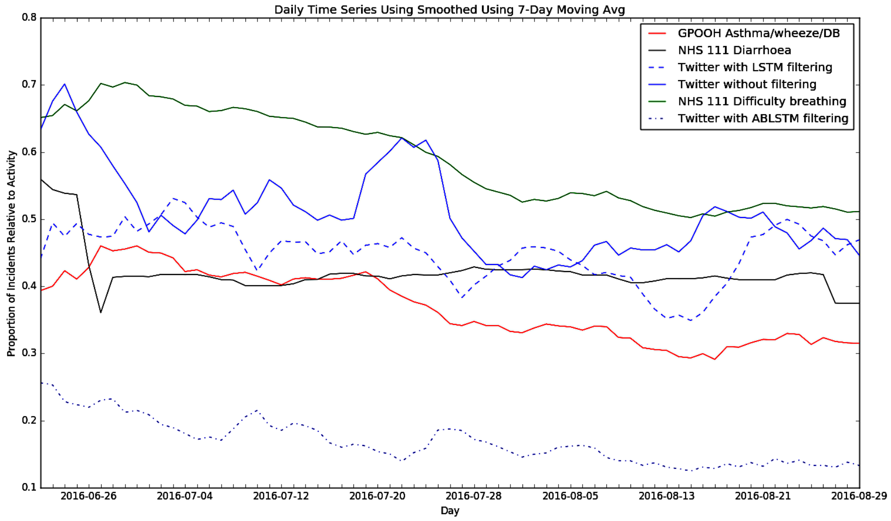

Fig. 4. Comparison of PHE syndromic surveillance indicators with Twitter signals.

hours) consultations, emergency department (ED) attendances and tele-health (NHS 111) calls We performed retrospective analyses comparing the signals generated by some of these systems to that generated by our ABLSTM. For this analysis, a number of 'syndromic indicators' monitored by PHE's syndromic surveillance systems were selected based upon their availability, quality and potential association to asthma/difficulty breathing. These indicators were ***"difficulty breathing"*** and ***"asthma/wheeze/difficulty breathing"***. We also made use of ***"diarrhoea"*** as a control indicator. *Difficulty breathing* and *diarrhoea* are generated from NHS 111 calls while *asthma/wheeze/difficulty breathing* are generated from GP Out-of-hours (GPOOH) consultations. For all indicators, daily counts of consultations for relevant syndromic indicators, together with daily counts of the consultations overall were used to compute daily proportions of consultations related to the indicators. Similarly, for ABLSTM we computed daily proportions of Tweets that were relevant to the syndrome of asthma/difficulty breathing relative to the number of Tweets collected each day. We used these daily proportions to plot comparative time series shown in figure 4. We also included the LSTM from our comparisons in section 4.1 in this experiment. We included it because it performed the best at the Tweet relevance classification task after our attentive RNNs and we wanted to observe how it measured against our attentive RNN in the real world and not just in the classification task with the limited test-partition data.

We smoothed the time series signals using a 7-day average to minimise the irregularities caused by the differences between weekend and weekday activities for GP out-of-hours services. Figure 4 shows that the signals for *asthma/wheeze/difficulty breathing*, *difficulty breathing* and Twitter with ABLSTM filtering follow very similar trends and have similar shapes. The signal for *diarrhoea* on the other hand, does not appear to be related to any others as we may expect. We also show a time series for the Twitter system without filtering. For this, we used the daily counts of collected Tweets and normalised each day's count by the average Tweet count for that week. We see in figure 4 that this raw Twitter signal does not match well with the *asthma/wheeze/difficulty breathing* signal. However, it still seems to match better than that of *diarrhoea*. To gain a clearer picture of how well the signals matched, we calculated the Pearson correlations between them. The results of this are shown in table 2. Table 2 confirms that the Attentive RNN (ABLSTM) does indeed perform well at Twitter mining for syndromic surveillance for this specific syndrome and displays a strong positive correlation ($r = 0.830$) with the recorded public health signal for asthma/ difficulty breathing. The Twitter signal with LSTM demonstrated a lower correlation with what may be considered the 'ground truth' ($r = 0.586$), and was not that far off from the correlation between the ground truth and Twitter without any filtering classifier applied to it.

5 Conclusion

We describe an attention-based RNN architecture for short text classification. We find from the literature that most Neural Network models used to classify Tweets treat all words as equal while focusing on making use of semantic relationships between words to get the overall meaning. Our proposed approach takes this a step further by not only trying to employ these semantic relationships, but also acknowledging the presence of key words and capitalizing on them. We demonstrate the utility of the described model for Tweet classification in a syndromic surveillance context. We monitor Twitter and employ our text classifiers to detect Tweets relevant to asthma/difficulty breathing. After learning and converting the words in Tweets to vectors, the Attentive bi-directional RNN derives a vector representation for the Tweet, which places emphasis on important words in the Tweet. We experimented with LSTM and GRU units for the cells in our attentive bi-directional RNN. The attentive bi-directional LSTM (ABLSTM) approach was found to outperform the popular text-CNN and LSTM at the task of Tweet relevance classification.

The also show that the attentive model has strong understanding capabilities that can not only be used for accurate short text classification, but could also be taken advantage of for building informative document embeddings.

We then evaluate the ABLSTM performance on the real-world task of syndromic surveillance by using it to generate a public health signal from Twitter and comparing it to the signal detected by PHE syndromic surveillance systems. We found that the signal generated using the ABLSTM had a strong correlation with the 'ground truth' signal generated by PHE.

While we found strong correlations between the ABLSTM and the syndromic surveillance data, we are yet to fully assess the syndromic surveilance utility of its application to Twitter as there were no real-world major incidents during our investigated periods and we only have Twitter data from these periods. We intend to repeat this analysis prospectively over a longer time period, where incidents may occur. Another limitation is that our syndromic surveillance data was collected with the geographical scope of England. However, as described in section 4.1,our location filtering was not accurate. Our Tweet filtering system focused on Tweets geolocated to the UK or marked loosely as originating from a place in the UK (e.g. by time zone). This makes our geographical filtering larger in scope (UK-level) than that of the syndromic surveillance data (England-level) and possibly inacurate. Better Twitter location filtering needs to be carried out in order to further fine-tune our syndromic surveillance framework. Despite that, we show that the ABLSTM performs better than popular neural network architectures for short text classification, i.e. Tweet classification. We also show that the described attention model can be used for creating meaningful document embeddings which not only summarize and encode the semantics of the document, but also automatically encodes emphasis on keywords in the document.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
3. Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-based models for speech recognition. In: Advances in neural information processing systems. pp. 577–585 (2015)
4. De Boer, P.T., Kroese, D.P., Mannor, S., Rubinstein, R.Y.: A tutorial on the cross-entropy method. Annals of operations research **134**(1), 19–67 (2005)
5. Du, C., Huang, L.: Text classification research with attention-based recurrent neural networks. International Journal of Computers Communications & Control **13**(1), 50–61 (2018)
6. Edo-Osagie, O., De La Iglesia, B., Lake, I., Edeghere, O.: Deep learning for relevance filtering in syndromic surveillance: A case study in asthma/difficulty breathing. In: International Conference on Pattern Recognition Applications and Methods 2019. No. 8 (2019)
7. ?erban, O., Thapen, N., Maginnis, B., Hankin, C., Foot, V.: Real-time processing of social media with SENTINEL: A syndromic surveillance system incorporating deep learning for health classification. Information Processing & Management **56**(3), 1166–1184 (may 2019). https://doi.org/10.1016/j.ipm.2018.04.011, `https://doi.org/10.1016%2Fj.ipm.2018.04.011`
8. Fennell, K.: Everything you need to know about repeating social media posts (March 2017), `\url{https://mavsocial.com/repeating-social-media-posts/}`, "[Online; posted 12-March-2017]"

9. Hermann, K.M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P.: Teaching machines to read and comprehend. In: Advances in Neural Information Processing Systems. pp. 1693–1701 (2015)
10. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
11. Jin, L., Schuler, W.: A comparison of word similarity performance using explanatory and non-explanatory texts. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 990–994 (2015)
12. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: European conference on machine learning. pp. 137–142. Springer (1998)
13. Joachims, T.: Transductive inference for text classification using support vector machines. In: Icml. vol. 99, pp. 200–209 (1999)
14. Johnson, R., Zhang, T.: Semi-supervised convolutional neural networks for text categorization via region embedding. In: Advances in neural information processing systems. pp. 919–927 (2015)
15. Johnson, R., Zhang, T.: Supervised and semi-supervised text categorization using lstm for region embeddings. arXiv preprint arXiv:1602.02373 (2016)
16. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
17. Lee, J.Y., Dernoncourt, F.: Sequential short-text classification with recurrent and convolutional neural networks. arXiv preprint arXiv:1603.03827 (2016)
18. Lewis, D.D., Ringuette, M.: A comparison of two learning algorithms for text categorization. In: Third annual symposium on document analysis and information retrieval. vol. 33, pp. 81–93 (1994)
19. Luong, T., Socher, R., Manning, C.: Better word representations with recursive neural networks for morphology. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning. pp. 104–113 (2013)
20. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
21. Maynard, D., Bontcheva, K., Rout, D.: Challenges in developing opinion mining tools for social media (2012)
22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
23. Miyato, T., Dai, A.M., Goodfellow, I.: Adversarial training methods for semi-supervised text classification. arXiv preprint arXiv:1605.07725 (2016)
24. Nosofsky, R.M., Gluck, M.A., Palmeri, T.J., McKinley, S.C., Glauthier, P.: Comparing modes of rule-based classification learning: A replication and extension of shepard, hovland, and jenkins (1961). Memory & cognition **22**(3), 352–369 (1994)
25. Nowak, J., Taspinar, A., Scherer, R.: Lstm recurrent neural networks for short text and sentiment classification. In: International Conference on Artificial Intelligence and Soft Computing. pp. 553–562. Springer (2017)
26. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
27. Roeder, L.: What twitter's new rules mean for social media scheduling (March 2018), `\url{https://meetedgar.com/blog/what-twitters-new-rules-mean-for-social-media-scheduling/}`, "[Online; posted 13-March-2018]"
28. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**(11), 2673–2681 (1997)
29. Sebastiani, F.: Machine learning in automated text categorization. ACM computing surveys (CSUR) **34**(1), 1–47 (2002)
30. Tzeras, K., Hartmann, S.: Automatic indexing based on bayesian inference networks. In: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 22–35. ACM (1993)
31. Weiss, G., Goldberg, Y., Yahav, E.: On the practical computational power of finite precision rnns for language recognition. arXiv preprint arXiv:1805.04908 (2018)
32. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057 (2015)
33. Yang, Y.: Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In: SIGIR'94. pp. 13–22. Springer (1994)
34. Yang, Y., Chute, C.G.: An example-based mapping method for text categorization and retrieval. ACM Transactions on Information Systems (TOIS) **12**(3), 252–277 (1994)
35. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1480–1489 (2016)

36. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). vol. 2, pp. 207–212 (2016)