An Evolutionary Approach to Automatic Keyword Selection for Twitter Data Analysis*

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

[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 approach to intelligent and automatic keyword selection for the purpose of Twitter data collection and analysis. The proposed approach makes use of a combination of deep learning and evolutionary computing. As some context for application, we outline the proposed algorithm using the case study of public health surveillance over Twitter, which is a field with a lot of interest. In doing so, we also describe an optimization objective function particular to the keyword selection problem, as well as metrics for evaluating Twitter keywords, namely: *reach* and *tweet retreival power*, on top of traditional metrics such as *precision*. In our experiments, our evolutionary computing approach achieved a tweet retreival power of **0.55**, compared to **0.35** achieved by the baseline human approach.

Keywords: Twitter · Evolutionary Computing · Syndromic Surveillance · Social Media Sensing

1 Introduction

Syndromic surveillance can be described as the real-time (or near real-time) collection, analysis, interpretation, and dissemination of health-related data, to enable the early identification of the impact (or absence of impact) of potential human or veterinary public health threats that require effective public health action [13]. As syndromic surveillance is concerned with the detection and understanding of public health threats, there is interest in rich, interesting and efficient data scources. De Quincey and Kostkova [1] introduced the potential of Twitter in detecting outbreaks. The amount of real-time information present on Twitter, either with regards to users reporting their own illness, the illness of others or reporting confirmed cases from the media, is both rich and highly accessible.

Twitter syndromic surveillance, like all other Twitter analysis applications, requires the collection of Tweets. This collection usually involves specifying some filtering criteria. The simplest and most commonly used criteria is filtering by

* Supported by Public Health England.

keyword. In any Twitter analysis application, choosing the right set of keywords can have a big impact on the system as it controls which Tweets are collected and observed. Choosing keywords that are too precise and strict will result in our system collecting mostly relevant Tweets, but also simultaneously only observing few Tweets, which will most likely only be a small sample of the relevant Tweets available. Conversely, choosing keywords that are too broad will result in our system observing a great deal of Tweets, most of which will not be relevant. Therefore, selecting the right keywords is an important and difficult task.

In this paper, we propose an intelligent and automatic approach to effective keyword selection. We leverage evolutionary algorithms and machine learning to quantify and distinguish semantic information in keywords, allowing us to select automatically select the best keywords to collect Tweets relevant to our application. For the sake of comparison, we also discuss a manual method of keyword selection, carried out by humans, and use this as a baseline in our experiments. We found that our automatic keyword selection algorithm was able to outperform the manual, human approach.

2 Twitter Keyword Selection and Related Work

Before we begin thinking about automatic keyword selection, we must first take a look at how keyword selection normally occurs. While this process might differ based on the purpose of the data collection, it will typically involve some domain knowledge relating to the purpose of the data collection. However, in addition to this, it is often useful to keep in mind that language on Twitter is usually very informal and colloquial. This must also be taken into account when selecting a set of keywords for any purpose. For syndromic surveillance, our goal when selecting keywords for collection was to choose keywords which may be relevant to our particular syndrome of interest. We worked in conjunction with experts from Public Health England (PHE), to create a set of formal terms that may be connected to the specific syndrome under scrutiny. This set of keywords was then further expanded using synonyms from thesauri and the urban dictionary[3]. Urban dictionary is a web resource which serves as an encyclopedia of sorts for slangs, so it was used to account for informal language that may occur in Twitter.

Much of the existing literature on keyword selection revolves around marketing and advertising efforts. Some researchers have looked at keyword selection as a feature selection problem [5]. Others have taking a classification point of view. For example, [7] applied a CNN to classify keywords as good or bad in order to maximaize advertising campaigns. More recently, with the influence of deep learning in natural language processing and the rise of deep word embeddings, such methods have seen a growing interest in keyword selection. [6] made use of Word2Vec embeddings to select keywords for websites using semantic similarity. In fact, similar to our work, [8] work towards the problem of social media

[3] `https://www.urbandictionary.com`

analysis, making use of GloVe embeddings to infer keywords which can be used to find online trolls on social media. The problem with the solutions provided by a lot of the existing research, is that they assume that all similar keywords under a concept (such as asthma, for example), will collect relevant documents. They also assume that the addition of keywords will not dilute the quality of the collected data, just because they are all similar. However, this is not always the case. For this reason, we go further and attempt to optimize the set of keywords selected in such a way that confirms the above assumptions.

3 Evolutionary Automatic Keyword Selection for Twitter Data Analysis

We propose a two-stage approach which makes use of deep learning and evolutionary optimization. The input to the system is a small initial set of reasonable keywords. Deep word embeddings are trained on a text corpus with a subject domain related to the task for which the keywords are intended. We make use of GloVe [12] word embeddings which capture sematic meaning of words from observed context. Using these embeddings, we expand the initial input keyword set with words/phrases which are semantically similar or related to the input keywords. This expanded set of keywords then forms our candidate set which makes up our problem space from which we attempt to select the optimal subset to be used for Twitter data collection and analysis.

The task of keyword selection is modelled as an optimization problem. Each possible set of keywords is seen as a candidate solution and the goal is to find the optimal solution. In this approach, we make use of Particle Swarm Optimization (PSO) which is an evolutionary algorithm based on swarm intelligence put forward by J. Kennedy in 1995 [4]. Loosely speaking, we model each potential set of keywords as a particle in the swarm. Each particle is moved around the search space with some velocity, which is influenced by its known best position, as well as the best positions found by other members of the swarm.

3.1 Stage I: Obtaining the Candidate Set of Keywords

The first stage involves producing a larger set of candidate keywords, C, from which our optimal set will be selected. This is done by expanding the input query set, I, using deep word ebeddings as mentioned above. 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 [3]. They encode semantic information of words in dense low-dimensional vectors. There are many different ways to learn word embeddings [12; 10; 9]. 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. For this work, we built GloVe embeddings [12] from a dataset of 5 million unlabelled Tweets collected without any keyword limitations. The trained GloVe embeddings allow our system to get an understanding of Twitter language and vernacular, as our application of choice is Twitter

data analysis. For each word in our input, I, we obtain the n most similar words/phrases. These are estimated as the n closest vectors to the query word vector in the embedding space. We make use of cosine distance as our distance measure for estimating similarity.

3.2 Stage II: Applying Particle Swarm Optimization to Keywords

We model the keyword selection task as an optimization problem. Here an individual or particle or solution is a set of keywords. In this problem, the goal is to find the set of keywords that provide the maximum (or minimum) of some objective function. While the obvious solution to such a problem would be simply to check each possible solution in the search space and select the best, this is not often feasible. This is die to the fact that as the problem space grows, the computational complexity increases exponsentially with it, making brute force search approaches a lot less feasible. PSO solves this problem by making use of a set population of particles, where each particle is a potential solution. Fig 1 shows an illustration of the particles in a problem space. These particles are then set loose to explore the search space in order to find an optimal solution. They tend to swarm and form clusters in optimal regions of the problem space. We now look towards how we can model the keyword selection problem using

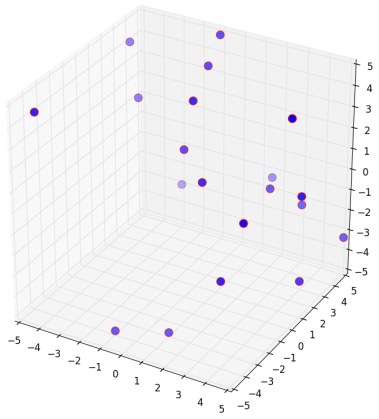

Fig. 1: Illustration of PSO particles in a search space

PSO. Following the construction of the candidate keyword set C as described above, we encode each keyword contained within it as a unique integer ranging from 1 to $|C|$. With this, we can now represent a set of keywords as a vector of integers, k, where each integer in the vector maps to a keyword in C. The size of k, denoted as D, must be determined before-hand and equates to the maximum size of the desired final optimal keyword set. Additionally, while values of 1 to $|C|$ represent keywords, a value of zero will be used to represent the absence of

a keyword. With this, when keyword vectors are mapped back to keyword sets, it will be possible to have sets of varying sizes (of up to $|k|$). Having developed a way to represent a set of keywords as a vector, we can also represent a set of keywords as a particle, as a particle is represented by a vector. With this, we can apply PSO to our candidate set, C, to intelligently and automatically select a set of keywords.

We start by randomly initializing a population of particles (i.e. keyword sets) from C. In essence, we create a set number of random vectors of size D, with values ranging from 0 to $|C|$. Each particle possesses a ***position***, x and a ***velocity***, v, and keeps track of the best position it has found, that is, its "personal best" or ***pbest***. The system keeps track of the "global best" or ***gbest***, which is simply the best position that has ever been found by any particle. The position of the i^{th} particle, $x_i = (x_i^1, x_i^2, x_i^3, ...x_i^D)$. The particles are all moved around the search space, with their positions updated based on their velocities, *pbest* values and *gbest*. More formally, after each iteration at time t, the position of the i^{th} particle is updated according to equation 1

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{1}$$

The velocity of the particle, v_i^{t+1}, (at time $t+1$) used to update its position can be computed as shown in equation 2 where ω is the inertia coefficient, c_1 and c_2 are acceleration coefficients and r_1 and r_2 are random floating point values between 0 and 1.

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (pbest - x_i^t) + c_2 r_2 (gbest - x_i^t) \tag{2}$$

There are three main components to the way the velocity of a particle is updated:

- ***Inertia Component:*** This component is intended to keep the particle moving (or not moving) in the direction it is headed, and is controlled by ω. Lower values of ω will speed up convergence while higher values encourage particle exploration of the search space [14].
- ***Learning Component:*** This component controls the size of the step a particle takes towards its next position in exploring the search space. It is controlled by the coefficient, c_1 [2].
- ***Social Component:*** This component implements swarm mentality, and causes a particle to move towards the best regions the swarm has discovered so far. It is controlled by c_2 [2].

Largely speaking, the particles in the swarm explore the search space based on the *pbests* and *gbest* within the swarm. These values are computed for a particle's position using the objective function, Z. Our goal is to minimize the value of the Z, which represents the underlying desire of the swarm system. It is the function to be optimized. We make use of an objective function that is particular to the task of selecting a keyword set. We wish to maximize the number of relevant documents (or Tweets in our application scenario) and minimize the number of irrelevant documents collected by a set of keywords. However, we don't wish

to achieve this by being too selective and only collecting very few documents. Our goal is a combination of the relevancy of the Tweets we collect and the volume of Tweets we collect. Both of these figures are important to us. As such, we developed an objective function that belies this. The objective function, Z, is computed as the mean of two terms, α and β. α is the ***irrelevance factor*** shown in equation 3, while β is the ***retrieval factor*** shown in equation 4.

$$\alpha = \sum_{i=1}^{D} \left(\frac{\sum_{j=1}^{|T|} k[i] \in T[j]}{\sum_{j=1}^{|T|} T[j] == irrelevant}\right) \tag{3}$$

$$\beta = \sum_{i=1}^{D} \left(1 - \left(\frac{\sum_{j=1}^{|T|} k[i] \in T[j]}{|T|}\right)\right) \tag{4}$$

Z is computed as:

$$Z = \frac{\alpha + \beta}{2} \tag{5}$$

Finally, putting this all together, PSO keyword selection can be carried out by iterating over the following steps:

1. The objective function is computed by each particle for their current position.
2. Each particle updates their *pbest* and the *gbest*.
3. Each particle is moved once their velocity and position are updated, using the *pbest* and *gbest* values computed from the objective function.

The steps are repeated either until the values converge, or a predetermined maximum number of iterations is reached.

4 Experiments and Results

We were interested in understanding if our proposed approach solved the task of intelligent and automatic keyword selection at all, and if so, also understanding how well it did so. We implemented and ran Tweet collections for the asthma/difficulty breathing for syndromic surveillance using our keyword selection approach and the typical manual approach, for a baseline. We undertook two sets of collection periods. The first collection period was a sort of "validation" collection period, inspired by the training/validation/test splits adopted when building machine learning models. This validation collection period was used by our proposed approaches to automatically generate keywords. These generated keywords were subsequently utilized in a second collection period, intended to allow us to measure how well the generated keywords perform. This can be seen as our "test" collection period. Our validation collection period ran for a seven day period from the 24^{th} of May, 2019 till the 1^{st} of July, 2019. Our test collection period ran for a further seven day period from the 1^{st} of July, 2019 till the 8^{th} of July, 2019. Only the PSO-based approach was involved in the validation

collection period, as the baseline approach does not need any data for setting it up. During the test collection period however, both approaches are involved.

One caveat to consider is that even though the evaluatory Tweet collections were performed simultaneously in parallel, due to the workings of the Twitter API, there is no guarantee that both systems will be exposed to the exact same Tweets at the exact same time. This is because of the fact that the Twitter streaming API only offers a sample of the entire real-time stream, the percentage of which will vary depending on the activity loads at the time. Despite this limitation of the free Twitter API, we should still be able to get some picture of how well our approaches perform. In this section, we describe the experimental setup for each approach, including the baseline standard keyword selection approach. After that, we present and discuss the results we obtained.

4.1 Experimental Setup: Baseline Approach

The baseline approach involved working with a group of domain experts to come up with useful keywords and augmenting these keywords with some terms from the Urban Dictionary. We came up with a list of keywords which are included in the appendices. Using these keywords, we ran a Tweet collection during the test collection period, from the 1^{st} of July, 2019, till the 8^{th} of July, 2019. The validation collection period was not used for this part of the experiments as there was no automatic keyword generation, rendering such a period unnecessary.

4.2 Experimental Setup: Particle Swarm Optimization-Based Keyword Selection Approach

The standard keyword set used in the baseline approach was used as the seed input I for creating the candidate set C. For each word in I, their five most similar words as inferred from our GloVe embeddings were added to the set. Using the candidate set of keywords, Tweets were collected during the validation collection period. At the end of this period, the PSO-based keyword generation algorithm was applied using the collected Tweets. We set our D, representing the maximum size of a keyword set to be 10. We made use of the PySwarm library of evolutionary algorithms to implement our PSO algorithm. Our setup had a swarm size of 100. After some experimentation, we set our ω to 0.8., and c_1 and c_2 to 1. This resulted in the following set of keywords being selected as the optimal arrangement: *wheezing*, *panting*, *gasping*, *puffing*, *coudn't breathe*, *wheeze*, *asthma*, *inhaler*, *sore eyes*. After obtaining the automatically selected keywords, we applied them during the test collection period, using them as query inputs.

5 Results

We utilized the set of keywords we obtained from our keyword selection algorithm as query inputs for Tweet collection. We also utilized the keywords obtained using the standard baseline approach. We applied the two distinct sets of keywords

in parallel during our test collection period - 1st of July, 2019, till the 8th of July, 2019. We then analyzed the Tweets collected by each set of keywords in order to understand how useful each keyword set was. We assessed the keyword sets based on their information retrieval ability. A lot of the traditional information retrieval metrics do not translate well, or cannot be calculated for our problem. For example, recall, which measures the fraction of relevant documents retrieved cannot be calculated because we have no way of knowing the total amount of relevant Tweets out there. Because of this, we made use of a combination of traditional metrics and developed problem-specific metrics. These metrics are ***precision*** and ***reach***.

Precision is a popular information retrieval metric which represents the proportion of retrieved documents which are relevant. In such an information retrieval context, precision is calculated as:

$$precision = \frac{|RelevantTweets| \cap |CollectedTweets|}{|CollectedTweets|} \tag{6}$$

In our scenario, precision measures the proportion of the collected Tweets which are relevant. When calculating the precision values for each keyword appraoch, we computed the precision over a random sample of the retrieved Tweets. We took random 2000-large samples of the Tweets colletected using each keyword selection approach and computed the precision from this sample. We did this because we wanted to manually label and count the number of relevant Tweets, instead of relying on one of our trained classifiers which are not perfect. Doing so allowed us to get an accurate and exact value for the number of relevant Tweets, and would not be feasible with the complete set of collected Tweets which are very large and would be incredibly time-consuming to manually label.

Reach is a metric we developed to help us capture the ability of a set of keywords to retrieve as many Tweets as possible, relevant or not. This is important because while it is useful to collect relevant Tweets, if we only observe a small amount of Tweets, we cannot create a useful signal which is appropriately representative of the activity related to the syndrome of interest. As such, reach measures the quantity of Tweets a set of keywords is able to collect. This could be computed simply as the proportion of the general Tweet stream that is collected using a set of keywords. However, the inner workings of the Twitter API is unknown to us. To overcome any bias introduced by the API and any rate limits it may impose, we calculate the *reach* of a set of keywords in relation to the simplest singular keyword possible. This can be formally represented as shown below:

$$reach = \frac{|CollectedTweets|_{\hat{k}} - |CollectedTweets|_K}{|CollectedTweets|_K} \tag{7}$$

$\hat{k}$ represents some arbitrary single unit keyword which is a simple and straightforward keyword. For example, in our scenario of *asthma/difficulty breathing* surveillance, we make use of the keyword "asthma" as $\hat{k}$.

Finally, we combined the precision and reach metrics into one metric by taking their harmonic mean, similar to the F-measure. We term this combined metric, the ***Tweet Retrieval Power (TRP)***.

$$TRP = 2\frac{precision \times reach}{precision + reach} \tag{8}$$

The TRP weights precision and reach evenly but similarly to the F-measure, it is possible to calculate variations of the TRP score which place different weights on precision and reach as below:

$$TRP_\beta = (1 + \beta^2)\frac{precision \times reach}{\beta^2(precision + reach)} \tag{9}$$

where TRP_β measures the Tweet retrieval ability when β times as much importance is placed on reach than precision.

Keyword Selection Approach	**Precision**	**Reach**	**TRP**
Baseline Human Approach	0.23	0.75	0.35
PSO Approach	0.48	0.65	0.55

Table 1: Performances of different approaches to keyword selection

Table 1 shows the results observed at the end of our analysis. We found the PSO approach to have the best Tweet Retrieval Power. The PSO approach produced a fair improvements in precision over the baseline human approach. It also resulted in a decrease in reach however. The precision achieved by the PSO keyword selection approach is more than double that seen by the baseline. In addition, while the reach of the baseline approach is better than that of the PSO approach, the margin between them is not very large. Finally, all things considered, the total Tweet Retreival Power of the PSO approach is greater than that of the manual baseline approach, and appears to possess a reasonable balance of precision and reach.

6 Discussion

In this paper, we investigated hybrid approaches to intelligently and automatically selecting keywords for use in collecting data, using Tweets for syndromic surveillance as a case study. We proposed an evolutionary algorithm inspired method which modelled the keyword selection task as an optimization problem. It made use of Particle Swarm Optimization (PSO) to determine the optimal set of keywords. The proposed algorithm was implemented and applied to the task of collecting Tweets for the surveillane of the *asthma/difficulty breathing* syndrome. For the sake of comparison, we also carried out a Tweet collection

with keywords selected using a manual, human approach. We then evaluated the results of both approaches, making comparisons between them.

We found that the PSO-based method performed better, outperforming the manual, human approach by a fair margin. While we observed a fair increase in relevance (precision) using our automatic keyword selection algorithms, we saw the opposite when looking at the reach metric. The baseline human approach to curating keywords seemed to have the most reach. Despite this, the boost in precision offered by the automatic keyword selection algorithm meant that it outperformed the baseline approach, yielding a higher TRP value. However, it is also important to remember that while we tried to keep things constant in our experiments, applying each keyword selection approach in parallel during the same periods, we cannot guarantee that they were exposed to the same environments and Tweets as that is an issue dependent on the Twitter API. Studies have estimated that using the Twitter streaming API, users can expect to receive anywhere from 1% of the tweets to 40% of tweets available in real-time, depending on the amount of activity at the time [11].

While we have introduced this technique for the intelligent and automatic selection of keywords and used them for surveilling the syndrome of *asthma/difficulty breathing*, this does not mean it does not generalize to other tasks. These techniques cannot only be applied for the purposes of surveilling other syndromes on Twitter, but also for any Tweet collection exercise, regardless of the purpose of said exercise. This is due to the fact that these techniques aim to maximize the relevance of collected Tweets to some query, together with the volume of Tweets collected. As long as there exists some defined query, the notion of "relevance" for its results must also exist. Because these are the main ideas behind our proposed approaches, they can be very easily adapted to any other problem and generalize very well.

While we have established that the PSO method collects better Tweets, it also runs in a very reasonable amount of time. This is because the time taken for each iteration is determined by how many particles the swarm contains and the dimension of each particle, as well as the computational resources available. Even with a mid-tier computer, a single iteration could never take longer than an hour in the absolute worst case. As such the PSO method is not only superior in terms of the quality of the keywords produced, but also in terms of the amount of time taken to produce said keywords.

Bibliography

[1] De Quincey, E., Kostkova, P.: Early warning and outbreak detection using social networking websites: The potential of twitter. In: International Conference on Electronic Healthcare. pp. 21–24. Springer (2009)

[2] Deb, K., Padhye, N.: Improving a particle swarm optimization algorithm using an evolutionary algorithm framework. KanGAL Report **2010**, 003 (2010)

[3] 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)

[4] Kennedy, J.: Particle swarm optimization. Encyclopedia of machine learning pp. 760–766 (2010)

[5] Kiritchenko, S., Jiline, M.: Keyword optimization in sponsored search via feature selection. In: New Challenges for Feature Selection in Data Mining and Knowledge Discovery. pp. 122–134 (2008)

[6] Lee, D., Kim, K.: Web site keyword selection method by considering semantic similarity based on word2vec. Journal of Society for e-Business Studies **23**(2) (2019)

[7] Liang, J., Yang, H., Gao, J., Yue, C., Ge, S., Qu, B.: Mopso-based cnn for keyword selection on google ads. IEEE Access **7**, 125387–125400 (2019)

[8] Liu, A., Srikanth, M., Adams-Cohen, N., Alvarez, R.M., Anandkumar, A.: Finding social media trolls: Dynamic keyword selection methods for rapidly-evolving online debates. arXiv preprint arXiv:1911.05332 (2019)

[9] 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)

[10] 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)

[11] Morstatter, F., Pfeffer, J., Liu, H., Carley, K.M.: Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose. In: Seventh international AAAI conference on weblogs and social media (2013)

[12] 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)

[13] Triple, S.: Assessment of syndromic surveillance in europe. Lancet (London, England) **378**(9806), 1833 (2011)

[14] Umapathy, P., Venkataseshaiah, C., Arumugam, M.S.: Particle swarm optimization with various inertia weight variants for optimal power flow solution. Discrete Dynamics in Nature and Society **2010**, 1–

15 (2010). https://doi.org/10.1155/2010/462145, https://doi.org/10.1155/2010/462145