STAT8002 Project

Text Mining and Classification of Chinese Suicide Newspaper Articles

Supervisor: Dr. Philip Yu

Name : LUI SIU MAN

UID: 3035358626

Submitted on 28 April 2017

Word Counts: 10293 words

Text Mining and Classification of Chinese Suicide Newspaper Articles

Abstract

Text classification has been used in many application domains for transforming unstructured text data into structural data and information. The study aims to examine the feasibility and performance of using text mining approaches to automatically classify suicide related Chinese newspaper articles from a dataset retrieved from a digital Hong Kong newspaper archive database. The corpus contians over 220K newspaper articles retrieved from the newspaper digital archive using the search terms "Suicide". This dataset covers news articles between Feb 1998 to Dec 2016. Using Word2Vec (Le and Mikolov 2014) representation and FastText (Bojanowski, Grave, Joulin, and Mikolov 2016) for document classification, the proposal system achieved a classification accuracy of 81-97% for various classification tasks.

1 Introduction

This study aims to investigate different text mining and machine learning approaches for classifying from suicide related Chinese newspaper articles. Suicide is one of the major global public health issues. Many studies have demonstrated a relationship between newspaper reporting, depiction of suicides incidence, and subsequent suicide behaviors (Yang et al. 2013). Media reporting of suicide may trigger imitative suicides and reporting of suicide of celebrity increase suicide risk in the population (Yip et al. 2006; Niederkrotenthaler et al. 2012).

Collaborations among researchers, social workers, and media are important for promoting suicide awareness and suicide prevention. Studying of the discursive characteristics of suicide newspaper reports is one way for researchers to better understanding and advise on the best practices for reporting suicides.

The wealth of information provided by digital newspaper archive and the accessibility machine learning and text mining tools for big data analysis motivated the implementation of an intelligent system that automatically extract information from suicide related newspaper articles to facilitate research in social sciences, media studies, and public health. While it is relatively easy to retrieve a big dataset, there are also many challenges to pre-process this big dataset so that to make it ready for further statistical modelling and analysis.

One of the main challenges is that, search results return by search engines or digital newspaper database are noisy. For instance, news articles contained the keyword "Suicide" in full text could be articles related to suicide car booming attacks, these articles are irrelevant for general suicide behavior and suicide prevention research. Hence, there is a need to classify the obtained articles into appropriate categories before it is useful for social science researchers. In the past, many of these related studies require extensive human efforts to classify, label, and annotate each newspaper article before further data analysis can be conducted. Hence, every often, the size of the dataset used and the length of the period studied are limited.

In this study, word embedding and machine learning are used to facilitate automated classification of Chinese suicide news into a pre-defined set of categories useful for suicide behavior and suicide prevention research. Different approaches and parameterization for are used and their classification performance are compared. This study involves a full corpus of over 220K Chinese newspaper articles related to Suicide, and the objective is to classify these articles into pre-defined categories suggested by research in the Center of Suicide Research and Prevention (CSRP). The four classification tasks are:

1. To determine if a newspaper article is related to suicide or not
2. To determine if a suicide news article is reporting a suicide new happened in Hong Kong
3. To determine if a suicide news article is reporting a suicide attempt, suicide incidence or suicide advice
4. To determine if a suicide news article is reporting student suicide

The overall framework of this study is shown in Figure 1.

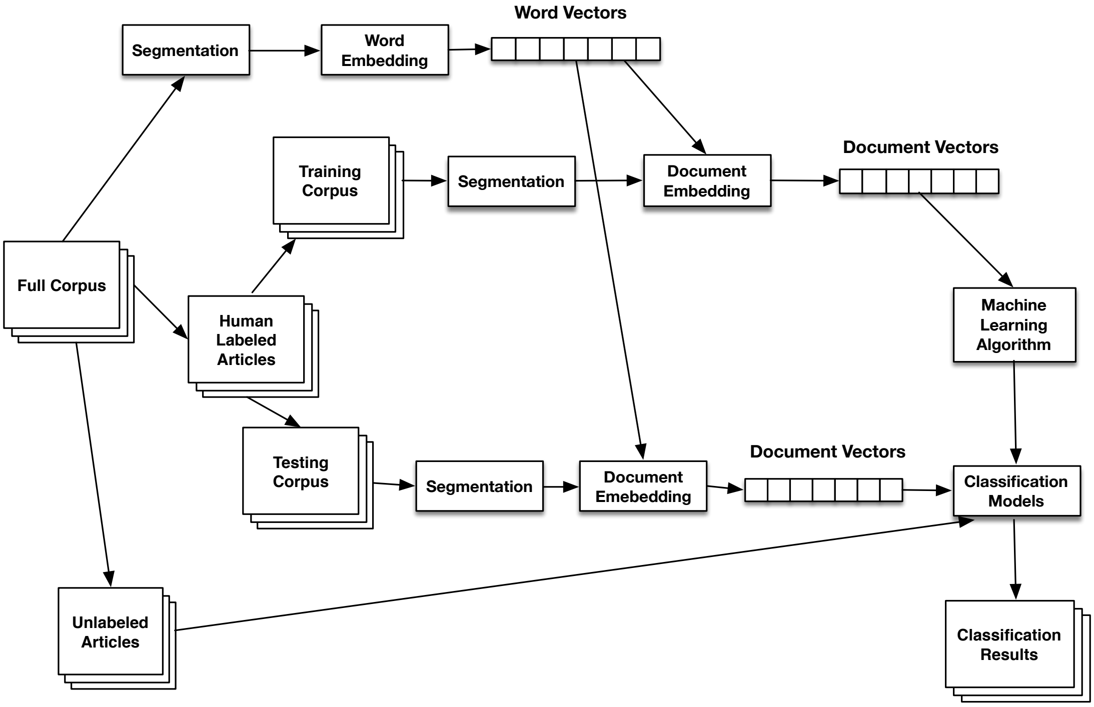


Figure 1 Overall framework of this study.

The remained of this paper is structured as follows. An overview of related works and technical background of the technique applied in this study for text segmentation, feature extraction, and document classification are provided in Section 2. Details of the text mining and machine learning classification processes used are then given in Section 3. Results and evolutions of the different approaches are reported in Section 4. Then Section 5 concludes this paper with a discussion of the limitations and future works.

2 Technical Backgrounds

Text mining is the process to deal with unstructured text data and turn them into structural and high quality information, minimizing human efforts to consume the text data. Text classification is an important component in text mining applications. Like many other classification applications, feature extraction and representation is a pre-requisite of text classification. For alphabet-based languages like English and other European languages, the use of word stems is recommended as the representation units of text corpus. However, extracting suitable features representation from Chinese text is challenging, because there is no natural delimiter (space) between characters of written Chinese text. Therefore, identifying suitable feature units by segmenting Chinese text is a major issue in Chinese text classification. Since there are no absolute precision and disambiguation for Chinese text segmentation, the inherent errors in this process will propagate to later steps in the text mining process. Segmented text are tokens of words, the process of turning these tokens of words into quantitative representations is called word embedding. This section presents the technical details of related to Chinese text segmentation, word embedding, and document classification techniques applied in this study.

2.1 Chinese Text Segmentation

For many languages like English, text document is formed by multiple sentences which are delimitated by punctuations, and each sentence is formed by one or multiple words and each word is formed by one or many characters. Text segmentation (tokenization) for English mainly involves splitting words by defined delimiters such as space, punctuation, and some specific affixes. However, for languages like Chinese, text contains of characters written without spaces in between, specific process is need to identify the boundaries of words, i.e. to determine where in a sequence of Chinese characters to put a delimiter such that the separated units (words) are meaningful units for downstream text mining tasks. This step is particularly important if further text mining tasks will involve semantic (meaning) and syntactic (grammar and structure) understanding of the text.

Existing Chinese text segmentation approaches can be classified into three different categories: (i) approaches based on and word matching, e.g., (Chen and Liu 1992), (ii) approaches based on linguistic grammatical rules, e.g., (Tsai et al. 2006), and (iii) approaches based on statistical models, e.g, (Sproat et al. 1996).

Dictionary-based approach simply assumes that a comprehensive dictionary exists and new text is segmented based on a simple word matching process. The dictionary usually come from manually segmenting and annotating very large corpus of news articles of People's Daily (a major newspaper in mainland China). However, this type of approaches cannot deal with unregistered words which are not included in the dictionary.

Linguistic grammatical rules use knowledge from linguistic to define rules and patterns for building a model which can be used for segmenting new text. While the statistical approaches examine statistic properties of very large training corpus to build models for segmentation. Therefore, the performance of segmentation with new target text significantly depends on the sufficiency and effectiveness of the training corpus for building the dictionary and rules. When the target texts are very different from the training corpus, performance of the mentioned approaches are in question.

For instance, target texts from Hong Kong Chinese newspapers, online social media may have many words and patterns that are outside those included in the training corpus (from People's Daily). In this case, unsupervised approach for segmentation will be needed.

In this study, four different Chinese segmentation approaches are considered. The first approach used methods available in a popular Chinese text segmentation software called JiebaR, which is software package available in R programming language [http://qinwenfeng.com/jiebaR/]. The second and third approaches are an unsupervised text segmentation, TopWORDS developed for domain specific Chinese text segmentation (Deng et al. 2016). Two different parameterization of TopWORDS are used. The last one is a naive method to just segment the text into characters which will serve as a baseline for comparison.

2.2 Segmentation using JiebaR

In JiebaR, four different text segmentation methods are available, namely MPSegment, HMMSegment, MixSegment, and QuerySegment. In this study, the MixSegment approach is a hybrid method which combined the MPSegment and the HMMSegment methods in JiebaR implementation. Both MPSegment and HMMSegment depend on the dictionary generated from the People's Daily training corpus for determining the values used in the calculation. The dictionary contains 584429 words. The details of MPSegment and HMMSegment are discussed below.

2.2.1 Maximum Probability Segmentation (MPSegment)

MPSegment is based on Maximum Probability Segmentation (Utiyama and Isahara 2001) considers all possible ways to segment the texts and pick the one with the highest probability. Consider the following example, given the Chinese text “有意見不同” (have divergent views), examples of possible ways to segment this are:

T_1 = 有 / 意見 / 不同
T_2 = 有意 / 見 / 不同

For MPSegment the objective is to

$$Max\ (P(T1), P(T2))$$

$$P(T) = P(w_1, w_2, \ldots\ldots, w_i) \approx P(w_1) \times P(w_1) \times\ \ldots \times P(w_i)$$

$$P(w_i) = \frac{Freq(w_i)}{N}$$

, where N is total number of terms in the training corpus, $Freq(w_i)$ is the frequent of word I in the corpus, and $w_1, w_2, \ldots\ldots, w_i$ are *words in T. As mentioned before, $P(w_i)$ is usually obtained from a very large training corpus, for any unregistered word, $P(w_i) = 1$. The probability of the related words in dictionary used by JiebaR is shown in the table below.*

Table 1 Examples of Maximum Probability Segmentation

Word	$P(w_i)$ *in dictionary used by JiebaR*
有	0.0180
有意	0.0005
意見	0.0010
見	0.0002
不同	0.0001

Therefore, in this example given above, $P(T_1) > P(T_2)$, thus the text “有意見不同” *will be segmented to* 有 / 意見 / 不同.

2.2.2 Hidden Markov Model Segmentation (HMMSegment)

HMMSegment in JiebaR is based on a Hidden Markov Model build from the training corpus which mainly from the newspaper articles in People's Daily. The main idea of this segmentation method based on some hidden states of each character to segment the text. The implementation in Jieba used a 4-states model (*B*, *E*, *M*, *S*) corresponding to B: beginning of word, E: end of word, M: middle of word, S: a single character word. The HMMSegment method can be summarized as below:

$$\boldsymbol{\lambda = (S, C, A, B, \pi)}$$

in the model, S = {s_1,s_2,…,s_N}, with N is the number of the states. e.g. B, M, E, S. Then, C = {c_1,c_2,…,c_L}, with C are the sequences of characters in the text, L is the number of characters. A is the state transition matrix with A = a_{ij} which provide the probabilities of transitions from one state (i) to another state (j), e.g. from state B to state E. B is the probability distribution such that B = b_j (k) represents the probability that state j output character v_k with v represents the one of the character in the dictionary which has k characters in it. $\boldsymbol{\pi}$ is the probability of each state in the initial of the a sentence. Hence,

$$\mathbf{a_{ij} = P\left(q_{t+1} = j \mid q_t = i\right), \quad 1 \le i, j, \le N \,,\ a_{ij} \ge 0 \,,\ \textstyle\sum_{j=1}^{N} a_{ij} = 1.}$$

$$\boldsymbol{b_j(k) = P\left(c_t = \ v_k \middle|\ q_t = \ s_j\right) ,}$$

$$\boldsymbol{1 \ \le j \ \le N \,, 1 \ \le k \ \le L,\ b_j(k) \ge 0 \,,\ \sum_{k=1}^{P} b_j(k) = 1}$$

Table 2 shows the tranisition matrix **A** in Jieba R implementation determined from the People's Daily newspaper training corpus.

Table 2 State transition matrix **A** from JiebaR implementation

		To State			
		B : beginning	**M: middle**	**E: end**	**S: single**
From State	**B**	0	0.1482	0.8518	0
	M	0	0.2836	0.7164	0
	E	0.7164	0	0	0.4455
	S	0.4862	0	0	0.5138

Chinese text are words mainly composed of 2 characters. The corresponding tranistion probablity from a character which is the begining of a word to the following character which is the end of a word, denoted as P (E | B) = 0.851. Comparing this probability to P (M | B) = 0.149, which represent the probability that a character is the beginning of a word, and followed by a character which is the middle of a word, you can see that P (E | B) = 0.851 is much higher than P (M | B) = 0.149. There are also some zeros in the transition matrix which represent the impossible transitions. For example, P (B | M) = 0 means that it is impossible for a character be the beginning of a word if the character before it is a character belongs to the middle of a word. Same idea applies to the transition probability of P (B | B) and P (E | E). If a character is the beginning of a word and the next character is also beginning of a word, the character should be considered as having a state of S, that is a single character word. The Chinese character "我" (I) is an example of a single character in Chinese.

Table 3 Values of $\boldsymbol{\pi}$ from JiebaR implementation

Initial State	Probability
B	0.7690
M	0
E	0
S	0.2310

Table 4 Values of $\boldsymbol{b_j(k)}$ of example characters from JiebaR implementation

Character	**State = B**	**State = M**	**State = E**	**State = S**
有	0.0045	0.0026	0.0067	0.0141
意	0.0015	0.0011	0.0018	0.0004
見	0.0006	0.0006	0.0019	0.0020
不	0.0139	0.0118	0.0006	0.0120
同	0.0027	0.0014	0.0020	0.0013

The objective of HMMSegment is $\mathrm{T}' = \mathrm{argmax}_{\mathrm{T}}\, \mathrm{P}\,(\,\mathrm{T} \mid \mathrm{C}, \lambda)$. Consider the text in the previous example "有意見不同" and the corresponding possible segmentation combinations:

T_1 = 有 / 意見 / 不同
T_2 = 有意 / 見 / 不同

To determine which segmentation combination is better, HMMSegment considers the two probabilities P ($\mathbf{T_1}$ | C, $\boldsymbol{\lambda}$) and P ($\mathbf{T_2}$ | C, $\boldsymbol{\lambda}$):

P ($\mathbf{T_1}$ | C, $\boldsymbol{\lambda}$) = P (S)*P (有 | S)* P (B | S) *P (意 | B) *P (E | B) *P (見 | E)
* P (B | E) * P (不 | B) *P (E | B) *P (同 | E)

P ($\mathbf{T_2}$ | C, $\boldsymbol{\lambda}$) = P (B)*P (有 | B)* P (E | B) *P (意 | E) *P (S | E) *P (見 | S)
* P (B | S) * P (不 | B) *P (E | B) *P (同 | E)

Therefore, in this example given above, $P(T_1) > P(T_2)$, thus the text "有意見不同" *will be segmented to* 有 / 意見 / 不同.

The MixSegment method on JiebaR implementation combined the MPSegment and HMMSegment methods mentioned above first handling the segmenting by MPSegment to segment words that can be found in the dictionary, then process that unregistered words with HMMSegment. Table 5 shows an example of applying the MixSegment method to segment text extracted from Suicide related Chinese newspaper article.

Table 5 Example results from JiebaR segmentation using MixSegment Method.

Chinese Text Extracted from Suicide Related Newspaper Article ***(Sing Pao 17-03-2016)***
... 教育局長吳克儉昨天在立法會表示，對近期學生輕生事件感到痛心和難過，局方昨日已去信學校，要求加強生命教育，並已採取措施減少學生學業壓力，認為學生輕生與新高中課程改革無直接關係。…
English Translation of the Text
... Secretary for Education Eddie Ng Hak-Kim was deeply concerned and distressed for the recent student suicides, the board has contacted schools, called for the strengthening of life education and has taken measures to reduce academic pressure of students. He thinks that there is no direct relationship between students suicide and the new high school curriculum reform.
Example of JiebaR Segmentation Results
教育局長 / 吳 / 克儉 / 昨天 / 在 / 立法會 / 表示 / 對 / 近期 / 學生 / 輕生 / 事件 / 感到痛心 / 和 / 難過 / 局方 / 昨日 / 已去 / 信 / 學校 / 要求 / 加強 / 生命 / 教育 / 並已 / 採取措施 / 減少 / 學生 / 學業 / 壓力 / 認為 / 學生 / 輕生 / 與 / 新 / 高中課程 / 改革 / 無 / 直接 / 關係

2.3 Unsupervised Segmentation with TopWORDS

TopWORDS (Deng et al. 2016) is an unsupervised approach based on the "word dictionary model" (WDM) by starting with an over-complete initial dictionary and then trim it down to an appropriate size based on statistical estimation.

With a set of unsegmented Chinese corpus $\boldsymbol{C}$ where S_C is the set of all possible segmentation combinations corresponding to C using dictionary $D = \{w_1, w_2, \ldots\ldots, w_N\}$ which contains N words (w_i), $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots\ldots, \theta_N\}$ as the probability vector of the word use in the corpus C, where $\sum_{i=1}^{N} \theta_i = 1$. Therefore, the probability model for a K-word segmented sentence S is as follows:

$$P(S \mid D, \theta) = \prod_{l=1}^{K} \theta_{ik}$$

$$P(C \mid D, \theta) = \sum_{s \in Sc} P(S \mid D, \theta)$$

and the conditional probability is the following:

$$P(S \mid C, D, \theta) \propto P(S \mid D, \theta)\ 1_{s \in Sc}$$

Therefore, the maximum-likelihood of segmenting C can be defined as follows:

$$S^* = \arg\max_{s \in Sc} P(S \mid C, D, \theta)$$

TopWORDS use an approach which average over all possible segmentation combinations of C such that to obtain a score γ_k defined as:

$$\gamma_k(C) = \sum_{s \in Sc} P(S \mid C, D, \theta) * I_k(S),$$

where $I_k(S) = 1$ if a word delimiter is put behind the *k*th character of C in segmentation of S, and otherwise $I_k(S) = 0$. Hence, $\gamma_k(C)$ is the total probability of having a word delimiter behind position k of C in all possible combinations of segmentation for C. Segmentation combinations can then be created by putting a word delimiter behind the position k if $\gamma_k\ (C)$ is greater than a given threshold τ_γ. Together with two other user-specified thresholds τ_L *and* τ_F for the length of word and frequency of the word in C, a dictionary D is generated which contains all possible words of lengths smaller or equals to τ_L and with frequency equal or larger then τ_F. Notice that this often an over-completed dictionary. Using a normalized counts vectors based on the observed counts of words in corpus C as a starting value for estimating *the MLE of* $\boldsymbol{\theta}$ = {$\theta_1, \theta_2, \ldots.., \theta_N$ }, non-words which have very low frequency in C will have *the estimated* θ close to zero.

Using a threshold τ_{trim} (default value is 10 $^{-8}$), words with $\theta < \tau_{trim}$ with will be removed from the over-completed dictionary. This step will be iterated to estimate the MLE of $\boldsymbol{\theta}$ again until no more words will be removed from the dictionary. Table 6 show the output of the words with top 10 value of $\boldsymbol{\theta}$ in the dictionary generated by TopWORDS using a corpus of suicide related newspaper articles. In this study, $\tau_L = 4$ *and* $\tau_L = 6$ are used. Table 7 shoes some example of words which are found in the dictionary generated with $\tau_L = 6$ but not in the dictionary generated with $\tau_L = 4$. Table 8 presents the segmentation results based on dictionary generated by $\tau_L = 4$. Comparing to the segmentation method using JiebaR, TopWORDS usually will generate fewer tokens with longer words.

Table 6 Top 10 words and the corresponding $\boldsymbol{\theta}$ in the dictionary generated by TopWORDS based on Suicide Related Corpus

```
自殺 , 0.0011260811867494901
表示 , 0.0011219041894804582
一名 , 9.480699443402915E-4
香港 , 9.134883901063539E-4
警方 , 8.875192877894716E-4
他們 , 7.426189391958928E-4
本報訊 , 7.42167241560748E-4
發現 , 7.292428391915934E-4
一個 , 7.286950947958128E-4
不過  , 6.62613358820602E-4
```

Table 7 Examples of words and the corresponding $\boldsymbol{\theta}$ in the dictionary generated with $\tau_L = 6$ but not in $\tau_L = 4$

```
撒瑪利亞會  , 7.559234703567094E-6
精神科醫生 , 3.8443331795166094E-5
患有情緒病 , 1.1716653002746032E-5
警員接報到場 , 2.984754486323608E-5
突然情緒失控  , 1.173788846726809E-5
```

Table 8 Example of TopWORDs segmentation results

```
教育局長 / 吳克儉 / 昨天 /  在立法會 /  表示 / 對近期  / 學生輕生 /  事件 / 感到痛
心 /  和 / 難過 / 局方 / 昨日 / 已 / 去信 / 學校 / 要求加強 / 生命教育 / 並已 / 採取
措施 / 減少 / 學生 / 學業壓力 / 認為 / 學生輕生 / 與 / 新高中 / 課程改革 / 無 / 直
接關係
```

2.4 Word Embedding

With the segmented text, each feature unit is a word. Next step is to determine how to represent these features quantitatively. Word embedding (Bengio et al. 2003; Collobert and Weston 2008) also referred as distributed word representation is the collective name for a set of language modelling and feature learning techniques in text mining. The simplest and naive approach is to use a one hot vector to represent each word, such that all words in a vocabulary of size K will be represented by a K-dimension vector space. In the one hot vector, the kth word will have 1 in the *k*th index of the vector and 0s in all other indexes. Let's consider a simple corpus with only 3 segmented sentences:

1. 我 / 愛 / 香港 (I love Hong Kong)
2. 我 / 愛 / 中國 (I love China)
3. 我 / 享受 / 跑步 (I enjoy running)

The size of the vocabulary of this corpus is 6 with words 我, 愛, 香港, 中國, 享受, 跑步. Hence each word in the corpus can be presented by a one-hot column vector with size 6. Table 9 presents the one-hot vectors of the example corpus.

Table 9 Example of one-hot vector representation

我 (I) = { 1, 0, 0, 0, 0, 0 }'
愛 (love) = { 0, 1, 0, 0, 0, 0 }'
香港 (Hong Kong) = { 0, 0, 1, 0, 0, 0 }'
中國 (China) = { 0, 0, 0, 1, 0, 0 }'
享受 (enjoy) = { 0, 0, 0, 0, 1, 0 }'
跑步 (running) = { 0, 0, 0, 0, 0, 1 }'

With this simple word presentation, words are considered as completely independent entities and this presentation cannot provide any information about the similarity or syntactic relationships between different words. Another major issue of this simple word presentation is that the dimension of these vectors growth as the size of vocabulary increase hence will be very inefficient for computation.

Another approach used in main traditional text mining application is to construct a window based word co-occurrence matrix to capture relationships between words. The values of the matrix can be based on the frequency of co-occurrences of words or existing or not existing co-occurrence within a specific window size. This approach is based on the assumption that related words are more likely to appear in the same documents than unrelated words. For example, "防止" , "自殺" , "燒炭", etc. are more likely to appeared in the same document than "香蕉" , "足球" , "回報率". Let consider the same simple corpus used before, to construct the co-occurrence matrix with window size 1, that is to consider 1 word left and one word right of that word in interest, the resulting matrix is shown in Table 10.

Table 10 Example of one-hot vector representation

	我	愛	香港	中國	享受	跑步
我	0	2	0	0	1	0
愛	2	0	1	1	0	0
香港	0	1	0	0	0	0
中國	0	1	0	0	0	0
享受	1	0	0	0	0	1
跑步	0	0	0	0	1	0

Single value decomposition (SVD) can then be applied to obtained a submatrix that with lower dimension by selecting the *k* singular vectors based on the required percentage of variance to be captured. This submatrix can then be used as the word embedding matrix. However, this approach still suffers from the problems resulting from the fact that most words do not co-occur, hence the matrix is very sparse and new words will appear very frequently, hence the dimensions of the co-occurrence matrix will change very often.

In this study, an iterative neural network approach called FastText (Bojanowski, Grave, Joulin, and Mikolov 2016) is used. FastText is an enhanced version of Word2Vec (Mikolov et al. 2013; Le and Mikolov 2014; Bojanowski, Grave, Joulin, and Mikolov 2016) approach with consideration of sub-word information. The technique details of Word2Vec and FastText are discussion below.

2.4.1 Word2Vec

Word2Vec is a neural network approach created by Mikolov et al. (Le and Mikolov 2014; Mikolov et al. 2013). This approach has drawn a lot of attention by demonstrating the use of word vectors to capture semantic and syntactic meanings of words. The iconic example is vector of (king) – vector of (man) + vector of (woman) = vector of (queen). This example suggests that Word2Vec presentation can capture the sematic meaning of the analogy of king to queen as to man to woman. Another example which demonstrates the capability of Word2Vec to capture syntactic structure of words is vector of (amazing) – vector of (amazingly) = vector of (calm) – vector of (calmly). Word2Vec is available with two different model namely Continues Bag of Word Model (CBOW) and Skip-gram Model.

2.4.1.1 Continuous Bag of Words Model (CBOW)

Both CBOW and Skip-gram are based on predictive approaches to determine the word vector representations. In CBOW, it predicts the target word based on the surrounding context words within a specific window size. Figure 2 depicts the neural network of CBOW model.

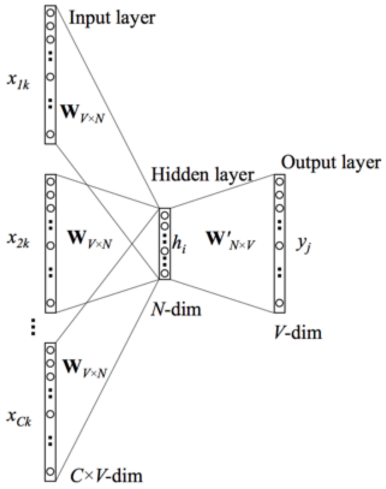


Figure 2 Continuous Bag of Words Model (CBOW) (Rong 2014)

Consider a sentence: "It is time to finish", with a widow size of C = 4 and target word $w_t =$ time, the objective of CBOW is to determine $P\left(w_t | w_{t-C/2} .. w_{t+C/2}\right)$, with $w_t = time, w_{t-1} = is, w_{t-2} = It, w_{t+1} = to, and\ w_{t+2} =$ finish. For a corpus with vocabulary of size V, the input layer contains the V-dimension one-hot vectors of the context words, and the weights W between the input layer and the hidden layer, is V X N dimension matrix. From the hidden layer to the output layer, the matrix W' is a N X V matrix. The hidden unit h is defined as follows:

$$\mathrm{h} = \frac{1}{\mathrm{C}}\ W'\ (x_1\ + x_2\ + \cdots + x_C\ )$$

$$\mathrm{h} = \frac{1}{\mathrm{C}}\ (v_{w1}\ + v_{w2}\ + \cdots + v_{wc}\ )'$$

where $(x_1\ + x_2\ + \cdots + x_C\ )$ are the one-hot vectors of the context words, v_w is the input vector of the context words comes from the rows of W matrix. Then a score $\mathrm{u_j}$ can be computed for each word in the vocabulary using the $\mathrm{j^{th}}$ column of the matrix W' and h:

$$u_j\ = v'_{wj}{}'h$$

Finally, with a softmax log-linear classification mode, the posterior distribution:

$$P\left(w_t | w_{t-\frac{C}{2}} .. w_{t+\frac{C}{2}}\right) = \mathrm{y_i} = \frac{\exp(u_j)}{\sum_{j'=1}^{V} \exp\ (u_{j'})}$$

2.4.1.2 Skip-Gram Model

The skip-gram model is like a mirror-image of the CBOW model, with the target words as the input and the context words as the output. Figure 3 depicts the neural network of the skip-gram model. The objective of the skip-gram model is to determine $P\left(w_{t-C/2} .. w_{t+C/2} | w_t\right)$. Notice that the hidden layer is now just a simple copy and transposing of the input to the hidden weight matrix W, and the posterior distribution:

$$P\left(w_{t-\frac{c}{2}} .. w_{t+\frac{c}{2}} \mid w_t\right) = \mathrm{y_{c,j}} = \frac{\exp(u_{c,j})}{\sum_{j'=1}^{V} \exp\,(u_{j'})}$$

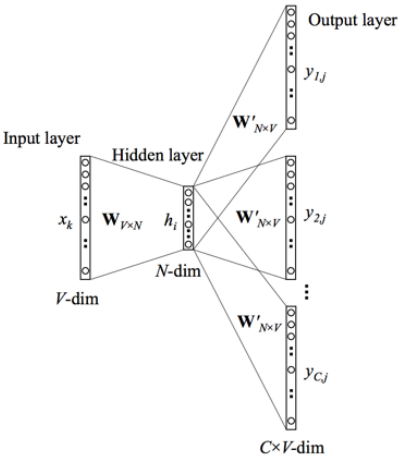


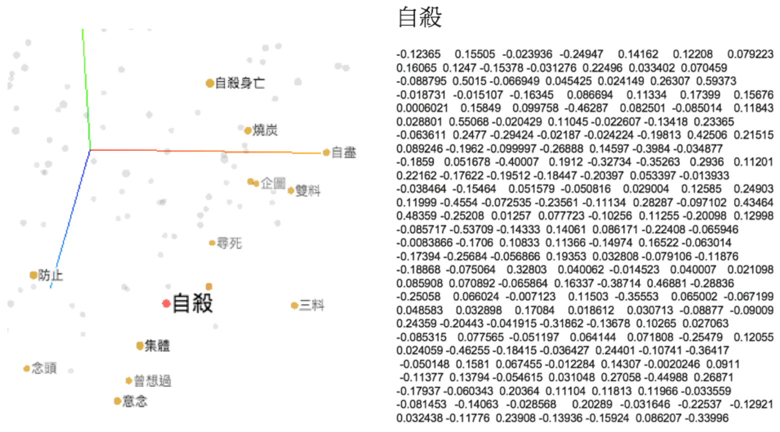
Figure 3 Skip-gram Model (Rong 2014)

2.4.1.3 FastText

FastText (Bojanowski, Grave, Joulin, and Mikolov 2016) is an enhanced version of Word2Vec with consideration of sub-word information. Given a word *w*, and addition set of vectors representing $\mathrm{G_w}$ for the n-grams of characters appearing in the word *w* is learned. The new word presentation then become the sum of the vector representation of the n-grams of the word, hence the new scoring function in the skip-gram is defined as follows:

$$\mathrm{S\,(w,c)} = \sum_{g\in Gw} z_g{}' v_c$$

Figure 4 shows the 200-diamension word vector representation and the corresponding vector projection of the word 自殺 trained using the suicide newspaper article corpus. The visual representation of the vectors projection is constructed using Embedding Projector provide by Google (http://projector.tensorflow.org/).

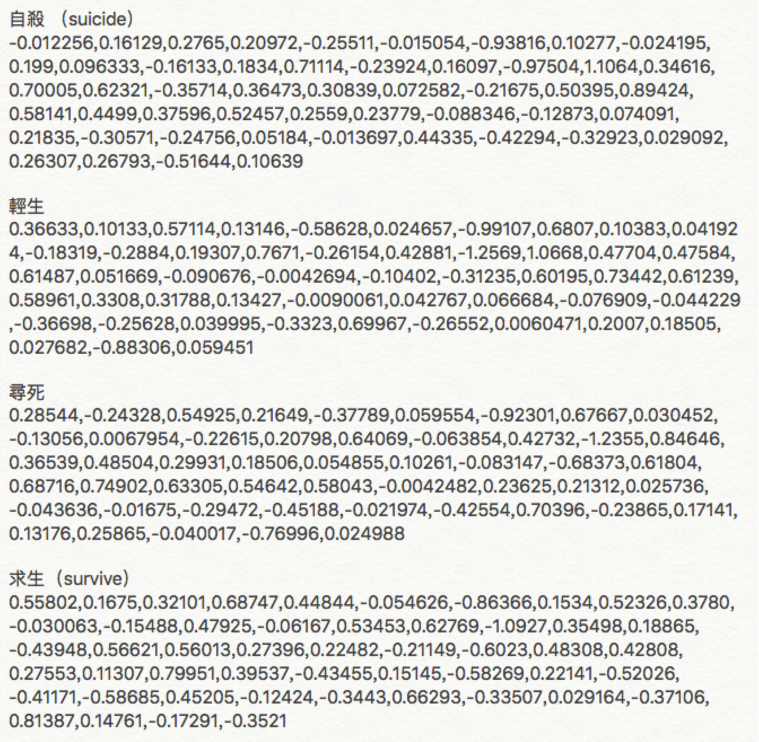


自殺

```
-0.12365   0.15505  -0.023936  -0.24947   0.14162   0.12208   0.079223
0.16065  0.1247 -0.15378 -0.031276  0.22496  0.033402  0.070459
-0.088795  0.5015 -0.066949  0.045425  0.024149  0.26307  0.59373
-0.018731  -0.015107  -0.16345   0.086694   0.11334   0.17399   0.15676
0.0006021   0.15849   0.099758 -0.46287   0.082501  -0.085014   0.11843
0.028801  0.55068 -0.020429  0.11045 -0.022607 -0.13418  0.23365
-0.063611  0.2477 -0.29424 -0.02187 -0.024224 -0.19813  0.42506  0.21515
0.089246 -0.1962 -0.099997 -0.26888  0.14597 -0.3984 -0.034877
-0.1859   0.051678  -0.40007   0.1912  -0.32734  -0.35263   0.2936   0.11201
0.22162 -0.17622 -0.19512 -0.18447 -0.20397  0.053397 -0.013933
-0.038464  -0.15464   0.051579  -0.050816   0.029004   0.12585   0.24903
0.11999 -0.4554 -0.072535 -0.23561 -0.11134  0.28287 -0.097102  0.43464
0.48359 -0.25208  0.01257  0.077723 -0.10256  0.11255 -0.20098  0.12998
-0.085717 -0.53709 -0.14333  0.14061  0.086171 -0.22408 -0.065946
-0.0083866 -0.1706  0.10833  0.11366 -0.14974  0.16522 -0.063014
-0.17394 -0.25684 -0.056866  0.19353  0.032808 -0.079106 -0.11876
-0.18868  -0.075064   0.32803   0.040062  -0.014523   0.040007   0.021098
0.085908  0.070892 -0.065864  0.16337 -0.38714  0.46881 -0.28836
-0.25058   0.066024  -0.007123   0.11503  -0.35553   0.065002  -0.067199
0.048583   0.032898   0.17084   0.018612   0.030713  -0.08877  -0.09009
0.24359 -0.20443 -0.041915 -0.31862 -0.13678  0.10265  0.027063
-0.085315   0.077565  -0.051197   0.064144   0.071808  -0.25479   0.12055
0.024059 -0.46255 -0.18415 -0.036427  0.24401 -0.10741 -0.36417
 -0.050148  0.1581  0.067455 -0.012284  0.14307 -0.0020246  0.0911
 -0.11377  0.13794 -0.054615  0.031048  0.27058 -0.44988  0.26871
-0.17937 -0.060343  0.20364  0.11104  0.11813  0.11966 -0.033559
-0.081453  -0.14063  -0.028568   0.20289  -0.031646  -0.22537  -0.12921
0.032438 -0.11776  0.23908 -0.13936 -0.15924  0.086207 -0.33996
```

Figure 4 a 200-dimension word vector and the visual representation of the word 自殺 "suicide" learned from the suicide newspaper article corpus with Jeiba Segmentation, Skip-gram model, 2-chars n-grams and widow size 7.

One way to validate the trained word vectors are to examine some regularity in words. Figure 5 shows the word vectors of 4 terms to be used as examples to check the validity of the trained word embedding.



```
自殺 （suicide）
-0.012256,0.16129,0.2765,0.20972,-0.25511,-0.015054,-0.93816,0.10277,-0.024195,
0.199,0.096333,-0.16133,0.1834,0.71114,-0.23924,0.16097,-0.97504,1.1064,0.34616,
0.70005,0.62321,-0.35714,0.36473,0.30839,0.072582,-0.21675,0.50395,0.89424,
0.58141,0.4499,0.37596,0.52457,0.2559,0.23779,-0.088346,-0.12873,0.074091,
0.21835,-0.30571,-0.24756,0.05184,-0.013697,0.44335,-0.42294,-0.32923,0.029092,
0.26307,0.26793,-0.51644,0.10639

輕生
0.36633,0.10133,0.57114,0.13146,-0.58628,0.024657,-0.99107,0.6807,0.10383,0.04192
4,-0.18319,-0.2884,0.19307,0.7671,-0.26154,0.42881,-1.2569,1.0668,0.47704,0.47584,
0.61487,0.051669,-0.090676,-0.0042694,-0.10402,-0.31235,0.60195,0.73442,0.61239,
0.58961,0.3308,0.31788,0.13427,-0.0090061,0.042767,0.066684,-0.076909,-0.044229
,-0.36698,-0.25628,0.039995,-0.3323,0.69967,-0.26552,0.0060471,0.2007,0.18505,
0.027682,-0.88306,0.059451

尋死
0.28544,-0.24328,0.54925,0.21649,-0.37789,0.059554,-0.92301,0.67667,0.030452,
-0.13056,0.0067954,-0.22615,0.20798,0.64069,-0.063854,0.42732,-1.2355,0.84646,
0.36539,0.48504,0.29931,0.18506,0.054855,0.10261,-0.083147,-0.68373,0.61804,
0.68716,0.74902,0.63305,0.54642,0.58043,-0.0042482,0.23625,0.21312,0.025736,
-0.043636,-0.01675,-0.29472,-0.45188,-0.021974,-0.42554,0.70396,-0.23865,0.17141,
0.13176,0.25865,-0.040017,-0.76996,0.024988

求生（survive）
0.55802,0.1675,0.32101,0.68747,0.44844,-0.054626,-0.86366,0.1534,0.52326,0.3780,
-0.030063,-0.15488,0.47925,-0.06167,0.53453,0.62769,-1.0927,0.35498,0.18865,
-0.43948,0.56621,0.56013,0.27396,0.22482,-0.21149,-0.6023,0.48308,0.42808,
0.27553,0.11307,0.79951,0.39537,-0.43455,0.15145,-0.58269,0.22141,-0.52026,
-0.41171,-0.58685,0.45205,-0.12424,-0.3443,0.66293,-0.33507,0.029164,-0.37106,
0.81387,0.14761,-0.17291,-0.3521
```

Figure 5 Four example 50-dimension word vectors the word 自殺, 輕生, 尋死, 求生 learned from the suicide newspaper article corpus with JeibaR Segmentation, Skip-gram model, 2-chars n-grams and widow size 7.

Using the cosine similarity to check the validity of the trained word vectors.

$$\text{Cosine similarity (A , B)} = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

Cosine similarity (自殺, 輕生) = 0.878
Cosine similarity (自殺, 尋死) = 0.845
Cosine similarity (自殺,求生) = 0.491

2.5 FastText LBoW Document Classifier

For the document classification, a Linear Bag of Words (LBoW) classifier is used in this study. This classifier combines the technique of Word2Vec to the average of vectors of all the words in a document as the document vector. For a document with N words with vectors $x_1, x_2 \ldots x_{N,}$ the document vector is:

$$y = \frac{1}{n}\sum_{i=1}^{N} x_i$$

Y is supplied as the input to the hidden layer for getting the classification results. Figure 5 depicts the LBoW.

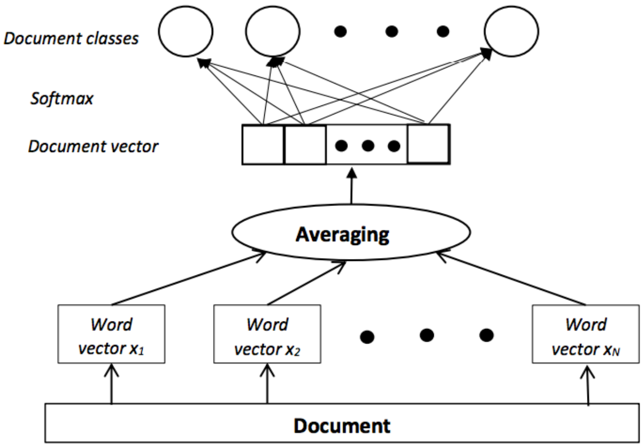


Figure 6 FastText LBoW Document Classifier
(Bojanowski, Grave, Joulin, Mikolov, et al. 2016)

3 Dataset and Experiment

The full corpus containing over 220k news articles representing the suicide related news from 1998 to 2016 published in major Hong Kong newspaper were used to train the word vectors which will be used later for encoding each news document into a document vector. Among these newspaper articles, 17175 were manually labeled into the required categories. The labeled cases were randomly split into a training and testing corpus. The training corpus contains 70% of the labeled cases, while the testing corpus contains 30% of the labeled corpus. Table 11 summarized the number of labeled cases in each class for the 4 classification tasks.

Table 11 Number of labeled cases in each classification task.

Task	**Label and no. of labeled cases**	**Total no. of labeled cases**
To determine if a newspaper article is related to suicide or not	Yes (8801) No (8374)	17175
To determine if a suicide news article is reporting a suicide new happened in Hong Kong	Yes (6511) No (1201)	7712
To determine if a suicide news article is reporting a suicide attempt, suicide incidence or suicide advice	Incident (3175) attempts (2613) advice (1244)	7032
To determine if a suicide news article is reporting student suicide	Yes (825) No (701)	1526

To investigate and fine-tune different parameters for text segmentation and word embedding in different approaches, a set of experiment with 64 possible configurations for each classification task have been conducted.

The configurations of the experiments is shown in Table 12. In total, there are 64 different setups for each classification task. Four different segmentation methods were tested, including the naive character segmentation, JiebaR segmentation with MixSegment, and 2 settings of the TopWORDS unsupervised segmentations. Given that there are 4 different classification task, a total of 64 * 4 = 256 models were trained. Word vectors trained in all setup is 50 dimensions. The total time required to segment the text, generate the word embedding, build the prediction model and perform predictions for all these 256 models is about 60 hours in a virtual Linux server with 6 Xeon E5-2630 CPU, 64 G RAM.

Table 12 Number of labeled cases in each classification task.

Configuration	Levels	No. of Levels
Segmentation approaches	Character Segmentation Jieba Segmentation Unsupervised Segmentation 1 with τ_L=4 Unsupervised Segmentation 2 with $\tau_L = 6$	4
Using news headline or new content for the classification tasks	Headline Content	2
Word embedding models	CBOW Skip-gram	2
Windows size for word embedding	5 7	2
Character n-grams in FastText	1 2	2

In terms for evaluating different classification models, the metric listed in Table 13 were used:

Table 13 Classification models evaluation metrics

Metrics	**Description**
Accuracy	This is the proportion of instance that are classified correctly by the classifier.
Marco Precision	Marco metrics use to overall confusion metric then for each class in the classification task, precision, recall, and F1 are calculated, then averaged the per class precision, recall, and F1 of all classes to obtain the Marco Precision, Marco Recall, and Marco F1. For a classification task with C number of classes:
Marco Recall	
Marco F1	
Marco-average Accuracy	Marco-average accuracy is calculated based the one-vs-all confusion matrix for each class, precision and recall for each class is first calculated and then averaged for calculating the accuracy.
Micro-averaged Accuracy	Micro-average accuracy is calculate based on the global one-vs-all confusion matrix which is the sum of the one-vs-all confusion metric of each class.
Majority Accuracy	Majority class accuracy simply use the majority class as the predicted value for all instance in dataset. Then accuracy is calculated with these predicted values.
Random Guess Accuracy	Random guess is the metrics calculated based on the results of randomly assign predicted value to each instance, with equal probability for each class.
Random Weighted Accuracy	Random weight guess is like random guess but assumed prior knowledge of the distribution of the data.
Expected Accuracy	Expected Accuracy is calculated based on the true value of the instance.
Kappa	Kappa is the overall accuracy minus the expected accuracy then divided by one minus expected accuracy.

4 Results

Table 14 to Table 21 presents the 10 top classifiers with the highest accuracy of each classification task for the training and the testing set. Overall, the accuracy of the top 10 classifiers range from 81% to 97%. Notice that except for the first classification task which is to determine if a newspaper article is a suicide news or not, all top 10 classifiers belong to those using the article content to build the word vectors. For determining if a newspaper article is suicide news or not, very often, the new headline provide good enough information, as words like 自殺，死，亡 will be included in the headline. Skip-gram models in general also has better performance than CBOW models.

Table 22 presents some example classification results and highlight some issue with the current classifiers. The first example is related to suicide bombing attack, the word suicide 自殺 is include in the content of the article, hence, it is returned by search results from the digital newspaper archive. The classifier has successfully predicted that it is not a suicide related news. Similarly, in example 2, the classifier also successfully predicted that the article is a Hong Kong suicide related news article, not related to student suicide, with a fatal suicide incidence.

However, example 3 shows a misclassified case, where a suicide new describing the suicide incidence of a university graduate is misclassified as a student related suicide news. Examine the content of the article we can see that many words related to study, university, classes, assignments., etc are included in the content. These words and the use patterns are very similar to those in student related suicide news.

5 Conclusion

This study investigated the use of various Chinese text segmentation and word embedding approaches to solve a real-world problem related to classifying Chinese newspaper articles related to suicide news. Using word vectors based on the enhanced Word2Vec word embedding generated from the segmented text, the classifiers have achieved rather good results. Researchers from CSRP has commented that even human classification may not achieve the same results and very often have inconsistency in classification by different people.

However, as shown in the examples in Table 22 there are still room for improvement the current approaches. The classifier may be improved by training a higher dimension word vectors to capture more regularity in the word uses and include additional information such as part of speech for training the word embed.

Table 14 Top 10 classifiers for classifying suicide or non – suicide news (Training Set).

Segment method	Headline or Content	Model	Window Size	N-gram	Accuracy	Macro Precision	Macro Recall	Macro F1	Marco Accuracy	Micro_ Accuracy	Major class Accuracy	Random Guess Accuracy	Random Weighted Guess Accuracy	Expected Accuracy	kappa
jbSeg	H	cbow	ws5	ng2	0.8599	0.8632	0.8632	0.8599	0.8599	0.8599	0.5311	0.5000	0.5019	0.4981	0.7208
unSeg1	C	skipgram	ws5	ng2	0.8556	0.8550	0.8554	0.8552	0.8556	0.8556	0.5311	0.5000	0.5019	0.5016	0.7103
unSeg1	C	skipgram	ws7	ng2	0.8535	0.8529	0.8530	0.8530	0.8535	0.8535	0.5311	0.5000	0.5019	0.5019	0.7060
unSeg1	C	skipgram	ws5	ng1	0.8504	0.8498	0.8510	0.8501	0.8504	0.8504	0.5311	0.5000	0.5019	0.5008	0.7003
charSeg	H	skipgram	ws5	ng2	0.8493	0.8544	0.8533	0.8493	0.8493	0.8493	0.5311	0.5000	0.5019	0.4974	0.7002
unSeg1	C	skipgram	ws7	ng1	0.8483	0.8476	0.8481	0.8478	0.8483	0.8483	0.5311	0.5000	0.5019	0.5015	0.6956
unSeg2	C	skipgram	ws5	ng2	0.8483	0.8476	0.8478	0.8477	0.8483	0.8483	0.5311	0.5000	0.5019	0.5018	0.6954
unSeg2	C	skipgram	ws7	ng2	0.8483	0.8476	0.8481	0.8478	0.8483	0.8483	0.5311	0.5000	0.5019	0.5015	0.6956
jbSeg	C	skipgram	ws7	ng2	0.8462	0.8457	0.8469	0.8459	0.8462	0.8462	0.5311	0.5000	0.5019	0.5006	0.6919
charSeg	H	skipgram	ws7	ng2	0.8440	0.8509	0.8486	0.8440	0.8440	0.8440	0.5311	0.5000	0.5019	0.4968	0.6901

Table 15 Top 10 classifiers for classifying suicide or non – suicide news (Test Set).

Segment method	Headline or Content	Model	Window Size	N-gram	Accuracy	Macro Precision	Macro Recall	Macro F1	Marco Accuracy	Micro_ Accuracy	Major class Accuracy	Random Guess Accuracy	Random Weighted Guess Accuracy	Expected Accuracy	kappa
jbSeg	H	cbow	ws5	ng2	0.8589	0.8615	0.8641	0.8588	0.8589	0.8589	0.5468	0.5000	0.5044	0.4980	0.7190
unSeg2	C	skipgram	ws5	ng2	0.8538	0.8522	0.8533	0.8527	0.8538	0.8538	0.5468	0.5000	0.5044	0.5036	0.7054
unSeg2	C	skipgram	ws7	ng2	0.8538	0.8522	0.8532	0.8527	0.8538	0.8538	0.5468	0.5000	0.5044	0.5036	0.7054
charSeg	H	cbow	ws7	ng1	0.8534	0.8536	0.8501	0.8514	0.8534	0.8534	0.5468	0.5000	0.5044	0.5064	0.7030
unSeg1	C	skipgram	ws5	ng2	0.8527	0.8511	0.8526	0.8517	0.8527	0.8527	0.5468	0.5000	0.5044	0.5032	0.7035
charSeg	H	skipgram	ws5	ng2	0.8520	0.8566	0.8582	0.8519	0.8520	0.8520	0.5468	0.5000	0.5044	0.4969	0.7057
jbSeg	C	skipgram	ws7	ng2	0.8520	0.8504	0.8524	0.8511	0.8520	0.8520	0.5468	0.5000	0.5044	0.5027	0.7023
unSeg1	C	skipgram	ws7	ng2	0.8520	0.8504	0.8514	0.8509	0.8520	0.8520	0.5468	0.5000	0.5044	0.5037	0.7017
jbSeg	C	skipgram	ws5	ng2	0.8518	0.8502	0.8518	0.8508	0.8518	0.8518	0.5468	0.5000	0.5044	0.5030	0.7018
unSeg1	C	skipgram	ws5	ng1	0.8500	0.8485	0.8504	0.8491	0.8500	0.8500	0.5468	0.5000	0.5044	0.5027	0.6984

Table 16 Top 10 classifiers for classifying Hong Kong or non – Hong Kong suicide news (Training Set).

Segment method	Headline or Content	Model	Window Size	N-gram	Accuracy	Macro Precision	Macro Recall	Macro F1	Marco Accuracy	Micro_ Accuracy	Major class Accuracy	Random Guess Accuracy	Random Weighted Guess Accuracy	Expected Accuracy	kappa
unSeg2	C	skipgram	ws7	ng1	0.9710	0.9532	0.9309	0.9416	0.9710	0.9710	0.8502	0.5000	0.7452	0.7514	0.8832
jbSeg	C	skipgram	ws7	ng2	0.9705	0.9507	0.9318	0.9409	0.9705	0.9705	0.8502	0.5000	0.7452	0.7505	0.8819
unSeg1	C	skipgram	ws5	ng1	0.9701	0.9525	0.9280	0.9397	0.9701	0.9701	0.8502	0.5000	0.7452	0.7520	0.8795
unSeg1	C	skipgram	ws5	ng2	0.9701	0.9535	0.9269	0.9396	0.9701	0.9701	0.8502	0.5000	0.7452	0.7526	0.8792
jbSeg	C	skipgram	ws5	ng2	0.9697	0.9490	0.9301	0.9393	0.9697	0.9697	0.8502	0.5000	0.7452	0.7505	0.8785
unSeg1	C	skipgram	ws7	ng2	0.9697	0.9532	0.9255	0.9387	0.9697	0.9697	0.8502	0.5000	0.7452	0.7529	0.8774
unSeg1	C	skipgram	ws7	ng1	0.9689	0.9514	0.9238	0.9370	0.9689	0.9689	0.8502	0.5000	0.7452	0.7529	0.8740
unSeg2	C	skipgram	ws5	ng1	0.9684	0.9511	0.9224	0.9360	0.9684	0.9684	0.8502	0.5000	0.7452	0.7532	0.8721
jbSeg	C	skipgram	ws7	ng1	0.9680	0.9456	0.9268	0.9359	0.9680	0.9680	0.8502	0.5000	0.7452	0.7505	0.8718
unSeg2	C	skipgram	ws5	ng2	0.9676	0.9493	0.9208	0.9343	0.9676	0.9676	0.8502	0.5000	0.7452	0.7532	0.8687

Table 17 Top 10 classifiers for classifying Hong Kong or non – Hong Kong suicide news (Test Set).

Segment method	Headline or Content	Model	Window Size	N-gram	Accuracy	Macro Precision	Macro Recall	Macro F1	Marco Accuracy	Micro_ Accuracy	Major class Accuracy	Random Guess Accuracy	Random Weighted Guess Accuracy	Expected Accuracy	kappa
unSeg2	C	skipgram	ws7	ng1	0.9672	0.9469	0.9217	0.9337	0.9672	0.9672	0.8502	0.5000	0.7452	0.7523	0.8675
jbSeg	C	skipgram	ws7	ng2	0.9672	0.9448	0.9240	0.9340	0.9672	0.9672	0.8502	0.5000	0.7452	0.7511	0.8681
unSeg1	C	skipgram	ws5	ng1	0.9663	0.9461	0.9189	0.9319	0.9663	0.9663	0.8502	0.5000	0.7452	0.7529	0.8637
unSeg1	C	skipgram	ws5	ng2	0.9659	0.9479	0.9152	0.9306	0.9659	0.9659	0.8502	0.5000	0.7452	0.7544	0.8612
jbSeg	C	skipgram	ws5	ng2	0.9651	0.9420	0.9181	0.9296	0.9651	0.9651	0.8502	0.5000	0.7452	0.7520	0.8591
unSeg1	C	skipgram	ws7	ng2	0.9642	0.9319	0.9269	0.9294	0.9642	0.9642	0.8502	0.5000	0.7452	0.7467	0.8588
unSeg1	C	skipgram	ws7	ng1	0.9630	0.9298	0.9238	0.9268	0.9630	0.9630	0.8502	0.5000	0.7452	0.7470	0.8536
unSeg2	C	skipgram	ws5	ng1	0.9625	0.9358	0.9143	0.9246	0.9625	0.9625	0.8502	0.5000	0.7452	0.7514	0.8493
jbSeg	C	skipgram	ws7	ng1	0.9613	0.9265	0.9205	0.9235	0.9613	0.9613	0.8502	0.5000	0.7452	0.7470	0.8470
unSeg2	C	skipgram	ws5	ng2	0.9609	0.9352	0.9076	0.9207	0.9609	0.9609	0.8502	0.5000	0.7452	0.7532	0.8414

Table 18 Top 10 classifiers for classifying 3 types of suicide news:
suicide attempt, suicide incidence, and suicide prevention advices (Training Set).

Segment method	Headline or Content	Model	Window Size	N-gram	Accuracy	Macro Precision	Macro Recall	Macro F1	Marco Accuracy	Micro_ Accuracy	Major class Accuracy	Random Guess Accuracy	Random Weighted Guess Accuracy	Expected Accuracy	kappa
jbSeg	C	type	skipgram	ws5	ng2	0.8759	0.8699	0.8614	0.8641	0.9380	0.8759	0.3803	0.2500	0.2882	0.2901
jbSeg	C	type	skipgram	ws7	ng2	0.8759	0.8704	0.8622	0.8647	0.9380	0.8759	0.3803	0.2500	0.2882	0.2900
unSeg1	C	type	skipgram	ws5	ng2	0.8735	0.8714	0.8664	0.8678	0.9367	0.8735	0.3803	0.2500	0.2882	0.2894
unSeg1	C	type	skipgram	ws7	ng2	0.8723	0.8696	0.8651	0.8664	0.9361	0.8723	0.3803	0.2500	0.2882	0.2893
unSeg1	C	type	skipgram	ws7	ng1	0.8719	0.8698	0.8647	0.8667	0.9359	0.8719	0.3803	0.2500	0.2882	0.2895
unSeg1	C	type	skipgram	ws5	ng1	0.8707	0.8656	0.8654	0.8642	0.9353	0.8707	0.3803	0.2500	0.2882	0.2876
jbSeg	C	type	skipgram	ws7	ng1	0.8674	0.8651	0.8566	0.8599	0.9337	0.8674	0.3803	0.2500	0.2882	0.2903
unSeg2	C	type	skipgram	ws7	ng1	0.8666	0.8667	0.8600	0.8625	0.9333	0.8666	0.3803	0.2500	0.2882	0.2900
unSeg2	C	type	skipgram	ws5	ng1	0.8658	0.8656	0.8591	0.8612	0.9329	0.8658	0.3803	0.2500	0.2882	0.2898
charSeg	C	type	skipgram	ws5	ng2	0.8654	0.8629	0.8514	0.8560	0.9327	0.8654	0.3803	0.2500	0.2882	0.2911

Table 19 Top 10 classifiers for classifying 3 types of suicide news:
suicide attempt, suicide incidence, and suicide prevention advices (Test Set).

Segment method	Headline or Content	Model	Window Size	N-gram	Accuracy	Macro Precision	Macro Recall	Macro F1	Marco Accuracy	Micro_ Accuracy	Major class Accuracy	Random Guess Accuracy	Random Weighted Guess Accuracy	Expected Accuracy	kappa
jbSeg	C	type	skipgram	ws5	ng2	0.8658	0.8656	0.8591	0.8612	0.9329	0.8658	0.3803	0.2500	0.2882	0.2898
jbSeg	C	type	skipgram	ws7	ng2	0.8654	0.8629	0.8514	0.8560	0.9327	0.8654	0.3803	0.2500	0.2882	0.2911
unSeg1	C	type	skipgram	ws5	ng2	0.8650	0.8641	0.8593	0.8607	0.9325	0.8650	0.3803	0.2500	0.2882	0.2894
unSeg1	C	type	skipgram	ws7	ng2	0.8638	0.8621	0.8499	0.8547	0.9319	0.8638	0.3803	0.2500	0.2882	0.2912
unSeg1	C	type	skipgram	ws7	ng1	0.8638	0.8623	0.8532	0.8569	0.9319	0.8638	0.3803	0.2500	0.2882	0.2905
unSeg1	C	type	skipgram	ws5	ng1	0.8614	0.8609	0.8556	0.8572	0.9307	0.8614	0.3803	0.2500	0.2882	0.2896
unSeg2	C	type	skipgram	ws7	ng1	0.8606	0.8558	0.8518	0.8536	0.9303	0.8606	0.3803	0.2500	0.2882	0.2889
jbSeg	C	type	skipgram	ws7	ng1	0.8541	0.8486	0.8423	0.8451	0.9271	0.8541	0.3803	0.2500	0.2882	0.2894
unSeg2	C	type	skipgram	ws5	ng1	0.8513	0.8482	0.8409	0.8434	0.9257	0.8513	0.3803	0.2500	0.2882	0.2899
charSeg	C	type	skipgram	ws5	ng2	0.8509	0.8492	0.8387	0.8432	0.9255	0.8509	0.3803	0.2500	0.2882	0.2902

Table 20 Top 10 classifiers for classifying student suicide or non – student suicide news (Training Set).

Segment method	Headline or Content	Model	Window Size	N-gram	Accuracy	Macro Precision	Macro Recall	Macro F1	Marco Accuracy	Micro_ Accuracy	Major class Accuracy	Random Guess Accuracy	Random Weighted Guess Accuracy	Expected Accuracy	kappa
jbSeg	C	skipgram	ws7	ng2	0.8704	0.8707	0.8698	0.8701	0.8704	0.8704	0.5159	0.5000	0.5005	0.5009	0.7403
jbSeg	C	skipgram	ws5	ng2	0.8651	0.8654	0.8645	0.8648	0.8651	0.8651	0.5159	0.5000	0.5005	0.5009	0.7297
unSeg1	C	cbow	ws5	ng1	0.8598	0.8596	0.8597	0.8597	0.8598	0.8598	0.5159	0.5000	0.5005	0.5004	0.7193
jbSeg	C	cbow	ws7	ng1	0.8598	0.8596	0.8597	0.8597	0.8598	0.8598	0.5159	0.5000	0.5005	0.5004	0.7193
jbSeg	C	cbow	ws5	ng1	0.8571	0.8570	0.8572	0.8570	0.8571	0.8571	0.5159	0.5000	0.5005	0.5003	0.7141
unSeg1	C	skipgram	ws7	ng2	0.8571	0.8570	0.8572	0.8570	0.8571	0.8571	0.5159	0.5000	0.5005	0.5003	0.7141
unSeg2	C	skipgram	ws5	ng2	0.8545	0.8544	0.8543	0.8543	0.8545	0.8545	0.5159	0.5000	0.5005	0.5006	0.7087
jbSeg	C	cbow	ws7	ng1	0.8466	0.8466	0.8469	0.8465	0.8466	0.8466	0.5159	0.5000	0.5005	0.5000	0.6931
jbSeg	C	skipgram	ws7	ng1	0.8466	0.8465	0.8462	0.8464	0.8466	0.8466	0.5159	0.5000	0.5005	0.5007	0.6927
jbSeg	C	skipgram	ws5	ng1	0.8466	0.8464	0.8464	0.8464	0.8466	0.8466	0.5159	0.5000	0.5005	0.5005	0.6928

Table 21 Top 10 classifiers for classifying student suicide or non – student suicide news (Test Set).

Segment method	Headline or Content	Model	Window Size	N-gram	Accuracy	Macro Precision	Macro Recall	Macro F1	Marco Accuracy	Micro_ Accuracy	Major class Accuracy	Random Guess Accuracy	Random Weighted Guess Accuracy	Expected Accuracy	kappa
jbSeg	C	skipgram	ws7	ng2	0.8148	0.8123	0.8139	0.8130	0.8148	0.8148	0.5556	0.5000	0.5062	0.5048	0.6260
jbSeg	C	skipgram	ws5	ng2	0.8148	0.8123	0.8139	0.8130	0.8148	0.8148	0.5556	0.5000	0.5062	0.5048	0.6260
unSeg1	C	cbow	ws5	ng1	0.8148	0.8132	0.8167	0.8138	0.8148	0.8148	0.5556	0.5000	0.5062	0.5021	0.6281
unSeg2	C	skipgram	ws5	ng2	0.8148	0.8132	0.8167	0.8138	0.8148	0.8148	0.5556	0.5000	0.5062	0.5021	0.6281
jbSeg	C	cbow	ws5	ng1	0.8025	0.8000	0.8000	0.8000	0.8025	0.8025	0.5556	0.5000	0.5062	0.5062	0.6000
unSeg1	C	skipgram	ws7	ng2	0.8025	0.8000	0.8000	0.8000	0.8025	0.8025	0.5556	0.5000	0.5062	0.5062	0.6000
jbSeg	C	cbow	ws7	ng1	0.8025	0.8000	0.8000	0.8000	0.8025	0.8025	0.5556	0.5000	0.5062	0.5062	0.6000
jbSeg	C	cbow	ws7	ng1	0.7901	0.7875	0.7889	0.7881	0.7901	0.7901	0.5556	0.5000	0.5062	0.5048	0.5762
jbSeg	C	skipgram	ws7	ng1	0.7901	0.7879	0.7861	0.7869	0.7901	0.7901	0.5556	0.5000	0.5062	0.5075	0.5738
jbSeg	C	skipgram	ws5	ng1	0.7901	0.7909	0.7944	0.7896	0.7901	0.7901	0.5556	0.5000	0.5062	0.4993	0.5808

Table 22 Example of predictions from unlabeled cases, model built from word vectors with Jeiba Segmentation, Skip-gram model, 2-chars n-grams and widow size 7.

	Examples of Segmented Newspaper Articles	Classification Results
1	塔利班 /的 /武裝 /分子 /昨天 /襲擊 /阿富汗 /首都 /喀布爾 /至少 /兩處 /地點 /情報 /機關 /警方 /及 /軍隊 /成為 /針對 /目標 /至少 /16 /人 /喪生 /50 /人 /受傷 /其中 /喀布爾 /西部 /的 /警察 /總部 /附近 /遭 /自殺式 /汽車 /炸彈 /襲擊 /爆炸 /威力 /強大 /事後 /保安人員 /與 /武裝 /分子 /在 /現場 /持續 /交火	P(Suicide = No) = 0.998 P(Student = No) = 0.916 P(HK = NO) = 0.998
2	本報 /特訊 / 一名 /曾 / 輕微 / 中風 /身懷 /未 /中彩 /賽馬 /彩票 /的 /男子 /昨日 /遭 /妻子 /發現 /在 /黃大仙 /翠鳳街 /住所 /吊頸 /喪生 /警方 /調查 /時 /在場 /檢獲 /一張 /預防 /中風 /的 /宣傳 /單張 /相信 /事主 /看過 /後 /感觸 /而 /萌死念 /與 /賭 /敗 /無關 /死者 /葉 /人瑞 /五十二歲 /與 /妻子 /及 /兩子 /一女 /同住 /翠鳳街 /五十四 /號五樓 /葉原任 /廚師 /在 /兩年 /前 /輕微 /中風 /後 /被 /解蝨 /治癒 /後 /失業 /了 /一段時間 /才 /尋 /獲 /一份 /工作 /葉 /因 /血壓高 /仍要 /定時 /服藥 /昨日 /中午 /他 /與 /妻女 /一起 /到 /住所 /附近 /酒樓 /品茗 /其 /後葉 /因到 /馬 /會場 /外 /投注 /處 /投注 /遂 /與 /家人 /分道揚鑣 /葉妻 /在 /下午 /三時 /四十五分 /返抵 /家中 /赫 /見 /丈夫 /在 /露臺 /上用 /鐵線 /吊頸 /立即 /報警 /由 /馳 /至 /消防員 /解下 /但 /送 /抵 /醫院 /時不治 /警方 /調查 /時 /在 /葉 /的 /身上 /尋獲 /三張 /賽馬 /投注 /記錄 /除 /第一場 /中腳 /失膽外 /第二 /三場 /均告 /落敗 /初時 /懷疑 /因 /輸 /馬 /而 /自殺 /但 /其 /投注額 /不大 /故 /認 /為 /與 /輸錢 /無關 /警員 /在 /了解 /案情 /時 /於 /現場 /檢獲 /一張 /關於 /中風病 /的 /宣傳 /單張 /相信 /葉因 /看 /了 /單張 /一時 /感觸 /而 /吊 /頸 /的 /成分 /居多	P(Suicide = Yes) = 0.902 P(Student = No) = 0.998 P(HK = Yes) = 0.998 P(Type = Incidence) = 0.992
3	香港大學 /教育 /學院 /一名 /於 /前年 /畢業 /的 /博士 /懷疑 /長期 /受 /失業問題 /困擾 /昨日 /被 /發現 /在 /西環 /高街 /住所 /墮樓 /死亡 /死者 /手部 /有 /割脈 /傷痕 /警方 /相信 /他 /割脈 /及 /跳樓 /雙料 /自殺 /死因 /並無 /可疑 /不 /排除 /他 /因為 /失業 /加上 /近日 /受 /感情 /打擊 /而 /尋死 /記者 /簡明 /恩 /鄭 /大康 /死者 /辛尚泰 /三十三歲 /於 /二年 /在 /港 /大 /教育 /學院 /課程 /及 /教育 /學系 /修畢 /博士 /課程 /其 /畢業論文 /題目 /為 /從 /形態學 /角度 /研究 /學習 /第二語言 /英語 /的 /效能 /並 /獲頒 /哲學 /博士學位 /辛 /畢業 /後 /長時間 /失業 /無法 /找到 /工作 /終日 /賦閒在家 /最近 /與 /女友 /分手 /現場 /消息 /稱 /辛尚泰 /於 /九六年 /搬入 /西環 /高街 /五十九 /號 /富裕 /大廈 /十三 /樓一 /單位 /居住 /平日 /住所 /經常 /有 /一名 /女性 /友人 /出入 /但近 /兩個 /月 /街坊 /已未 /見過 /她 /出現 /據 /知該 /女子 /為 /死者 /女友 /最近 /因 /感情 /問題 /分手 /事發 /昨午 /三時許 /一名 /姓葉 /男途 /人 /行經 /富裕 /大廈 /對開 /聽到 /隆然 /聲響 /並 /發現 /一名 /男子 /從 /高處 /墮下 /肝腦塗地 /倒 /斃 /馬路 /中 /救護 /員接 /到場 /將其 /送院 /搶救 /終 /證實 /死亡 /警方 /在 /死者 /身上 /發現 /一串 /鎖匙 /大廈 /看 /更 /亦 /證實 /墮樓 /男子 /的 /身份 /警方 /遂 /用 /鎖匙 /開啟 /單位 /大門 /發現 /一個 /房間 /的 /窗門 /打開 /廚房 /內有 /一把 /染血的 /刀 /地上 /仍 /留有 /大灘 /血漬 /事後 /調查 /發現 /死者 /的 /手腕 /有 /傷痕 /估計 /他 /在 /廚房 /割脈 /之後 /再 /在 /房間 /跳樓自殺 /警方 /在 /現場 /並無 /撿 /獲 /遺書 /其後 /接觸 /到 /死者 /的 /三十八歲 /胞兄 /他 /表示 /胞弟 /並無 /患病 /紀錄 /但 /透露 /其弟 /長時間 /失業 /呼籲 /要 /面對現實 /對於 /一名 /大學 /博士 /畢業生 /自殺 /死亡 /嶺南大學 /市場 /及 /國際 /企業 /學系 /副教授 /呂 /漢光 /表示 /惋惜 /他稱 /現時 /經濟 /不景 /但 /高學歷 /人士 /失業率 /其實 /不高 /惟 /香港 /是 /個 /高度 /商業化 /社會 /一切 /講究 /供求關係 /教育 /學系 /相比 /於 /工商管理 /較 /冷門 /競爭力 /較弱 /呂 /漢光 /說 /現時 /政府 /削減 /教育資源 /過去 /博士生 /畢業 /後 /返回 /大學 /象牙塔 /任 /研究 /助理 /的 /工作 /亦 /遭 /裁減 /令到 /堂堂 /博士 /畢業生 /亦 /要 /面臨 /失業 /壓力 /這 /情況 /甚是 /可悲 /他 /希望 /高學歷 /失業 /人士 /能 /面對現實 /為 /生活 /放下 /工作 /要求 /標準 /教協 /副會長 /區伯權 /對 /事件 /感到 /震驚 /他 /說 /讀 /教育 /學院 /行頭 /確比 /其他 /專科 /更窄 /加上 /出生率 /下降 /政府 /削資 /等 /要謀得 /教育 /一職 /將比前 /困難 /教協有 /一條 /減壓 /熱線 /可 /提供 /輔導 /電話 /27807337	P(Suicide = Yes) = 0.8086 P(Student = Yes) = 0.5390 P(HK = Yes) = 0.998 P(Type = Incidence) = 0.996

Acknowledgements. This work was supported by the General Research Fund, HKSAR [Grant numbers: GRF17628916, PI: Dr. Qijin Cheng].

Reference

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. "A Neural Probabilistic Language Model." *The Journal of Machine Learning Research* 3: 1137–55. doi:10.1162/153244303322533223.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. "Enriching Word Vectors with Subword Information." *arXiv:1607.04606v1 [cs.CL]*. http://arxiv.org/abs/1607.04606.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Tomas Mikolov, et al. 2016. "Bag of Tricks for Efficient Text Classification." *arXiv:1604.00289v1[cs.AI]*, 1–55. doi:1511.09249v1.

Chen, KJ, and SH Liu. 1992. "Word Identification for Mandarin Chinese Sentences." *Proceedings of the 14th Conference on Computational Linguistics*, 23–28. doi:10.3115/992066.992085.

Collobert, Ronan, and Jason Weston. 2008. "A Unified Architecture for Natural Language Processing." *Proceedings of the 25th International Conference on Machine Learning - ICML '08* 20 (1): 160–67. doi:10.1145/1390156.1390177.

Deng, Ke, Peter K Bol, Kate J Li, and Jun S Liu. 2016. "On the Unsupervised Analysis of Domain-Specific Chinese Texts." *Proceedings of the National Academy of Sciences* 113 (22): 6154–59. doi:10.1073/pnas.1516510113.

Le, Qv, and Tomas Mikolov. 2014. "Distributed Representations of Sentences and Documents." *International Conference on Machine Learning - ICML 2014* 32: 1188–1196. doi:10.1145/2740908.2742760.

Mikolov, Tomas, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013. "Efficient Estimation of Word Representations in Vector Space." *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, 1–12. doi:10.1162/153244303322533223.

Niederkrotenthaler, T., K.-w. Fu, P. S. F. Yip, D. Y. T. Fong, S. Stack, Q. Cheng, and J. Pirkis. 2012. "Changes in Suicide Rates Following Media Reports on Celebrity Suicide: A Meta-Analysis." *Journal of Epidemiology & Community Health* 66 (APRIL): 1037–42. doi:10.1136/jech-2011-200707.

Rong, Xin. 2014. "word2vec Parameter Learning Explained." *arXiv:1411.2738*, 1–19. http://arxiv.org/abs/1411.2738.

Sproat, Richard, William Gale, C Shih, and Nancy Chang. 1996. "A Stochastic Finite-State Word-Segmentation Algorithm for Chinese." *Computational Linguistics* 22 (3): 377–404. doi:10.3115/981732.981742.

Tsai, Richard Tzong Han, Hong Jie Dai, Hsieh Chuan Hung, Cheng Lung Sung, Min Yuh Day, and Wen Lian Hsu. 2006. "Chinese Word Segmentation with Minimal Linguistic Knowledge: An Improved Conditional Random Fields Coupled with Character Clustering and Automatically Discovered Template Matching." In *Proceedings of the 2006 IEEE International Conference on Information Reuse and Integration, IRI-2006*, 274–79. doi:10.1109/IRI.2006.252425.

Utiyama, Masao, and Hitoshi Isahara. 2001. "A Statistical Model for Domain-Independent Text Segmentation." *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics - ACL '01*, 499–506. doi:10.3115/1073012.1073076.

Yang, Albert C., Shih Jen Tsai, Cheng Hung Yang, Ben Chang Shia, Jong Ling Fuh, Shuu Jiun Wang, Chung Kang Peng, and Norden E. Huang. 2013. "Suicide and Media Reporting: A Longitudinal and Spatial Analysis." *Social Psychiatry and Psychiatric Epidemiology* 48 (3): 427–35. doi:10.1007/s00127-012-0562-1.

Yip, Paul S F, K. W. Fu, Kris C T Yang, Brian Y T Ip, Cecilia L W Chan, Eric Y H Chen, Dominic T S Lee, Frances Y W Law, and Keith Hawton. 2006. "The Effects of a Celebrity Suicide on Suicide Rates in Hong Kong." *Journal of Affective Disorders* 93 (1–3): 245–52. doi:10.1016/j.jad.2006.03.015.