applied sciences

MDPI

Article

Learning Word Embeddings with Chi-Square Weights for Healthcare Tweet Classification

Sicong Kuang * and Brian D. Davison

Department of Computer Science and Engineering, Lehigh University, 19 Memorial Dr. West, Bethlehem, PA 18015, USA; davison@cse.lehigh.edu

* Correspondence: sik211@lehigh.edu

Received: 16 July 2017; Accepted: 11 August 2017; Published: 17 August 2017

Abstract: Twitter is a popular source for the monitoring of healthcare information and public disease. However, there exists much noise in the tweets. Even though appropriate keywords appear in the tweets, they do not guarantee the identification of a truly health-related tweet. Thus, the traditional keyword-based classification task is largely ineffective. Algorithms for word embeddings have proved to be useful in many natural language processing (NLP) tasks. We introduce two algorithms based on an existing word embedding learning algorithm: the continuous bag-of-words model (CBOW). We apply the proposed algorithms to the task of recognizing healthcare-related tweets. In the CBOW model, the vector representation of words is learned from their contexts. To simplify the computation, the context is represented by an average of all words inside the context window. However, not all words in the context window contribute equally to the prediction of the target word. Greedily incorporating all the words in the context window will largely limit the contribution of the useful semantic words and bring noisy or irrelevant words into the learning process, while existing word embedding algorithms also try to learn a weighted CBOW model. Their weights are based on existing pre-defined syntactic rules while ignoring the task of the learned embedding. We propose learning weights based on the words' relative importance in the classification task. Our intuition is that such learned weights place more emphasis on words that have comparatively more to contribute to the later task. We evaluate the embeddings learned from our algorithms on two healthcare-related datasets. The experimental results demonstrate that embeddings learned from the proposed algorithms outperform existing techniques by a relative accuracy improvement of over 9%.

Keywords: word embedding; healthcare; classification

1. Introduction

More and more researchers have realized that Internet data could be a valuable and reliable source for tracking and extracting healthcare-related information. For example, in 2008, Google researchers found that they can "forecast" flu prevalence in real time based on search records [1]. Google later turned this research into one of their projects called Google Flu Trends (GFT) (https://en.wikipedia.org/wiki/Google_Flu_Trends). However, GFT later failed by missing the peak of the 2013 flu season by 140 percent [2]. One reason is the presence of too much noisy data [2]: people who search using the keyword "flu" might know very little about the symptoms of the flu. And some disease, whose symptoms are similar to the symptoms of the flu, is not actually the flu. The failure of GFT does not negate the value of the data but highlights the importance of classification of truly healthcare-related data from irrelevant and noisy data. Healthcare researchers desire to extract more healthcare information from information that people have shared online. Thus, we are more interested in tweets that talk about real disease symptoms as shown in Examples 4–6 which we name healthcare-related tweets, rather than those tweets that simply highlight healthcare information as seen

in Examples 1–3 which we name healthcare-noise tweets. The classification task in the healthcare field is a challenging one since both healthcare-related tweets and healthcare-noise tweets might contain some keywords such as "flu" and "health" which makes basic filtering approaches unworkable. Below, we show examples of healthcare-noise tweets (Examples 1–3) versus the truly healthcare-related tweets (Examples 4–6). (The example tweets are all drawn from a published dataset by Lamb et al. [3].) Compared to the healthcare-related tweets, we found that although the healthcare-noise tweets all have keywords such as "swine flu" and "flu shots", they are not really talking about the symptoms of the flu of individuals. With this motivation, the task of this work is to classify truly healthcare-related data from healthcare-noise data that is typically collected through the keyword filtering approach provided by the Twitter API (https://dev.twitter.com/streaming/overview).

1. Worried about swine flu? Here are 10 things you need to know: Since it first emerged in April, the global swine ..
2. Swine Flu - How worried are you? - Take our poll now and check out how others feel!
3. Missed getting a FREE FLU SHOT at Central last night? You've got three more "shots" at it.

4. feels icky. I think I'm getting the flu...not necessarily THE flu, but a flu.
5. Resting 2day ad my mthly blood test last 1 ok got apoint 4 flu jab being lky so far not getting swine flu thats something
6. 38 degrees is possible swine flu watching the thermometer go up. at 36.9 right now im scared :/

Social media, such as Facebook and Twitter, have been widely used by individuals to share real-life data about a person's health. It made the popular social media platforms, such as Twitter, a major source of healthcare-related data. Twitter provides support for accessing tweets via the Twitter API. Healthcare researchers have long been utilizing social media data to conduct their research [3–7]. Because of the popularity of social media platforms such as Twitter, the number of healthcare-related posts is growing fast. To extract further healthcare information, the most basic and crucial task is to discriminate and extract healthcare-related tweets from the massive pool of tweets. Researchers have made efforts to collect healthcare-related tweets [3,8]. Using modern machine learning algorithms and hand-crafted features, such as keyword-based binary features and support vector machines (SVM) with linear kernels [8], researchers are able to collect tweets that are potentially related to healthcare. However, many words are polysemous. For example, "cold" has a potential to talk about the disease but it might refer to the weather; besides the health-related concept, "virus" might also mean computer virus. Thus, tweets that are collected through ambiguous keyword filtering could be irrelevant. Another reason for the limitation of the keywords-based approach is that the set of important words can change over time, e.g., from H1N1, H5N1 to H7N9.

Recent years have seen the success of word embedding algorithms applied to many downstream natural language processing (NLP) tasks such as part-of-speech tagging and sentiment analysis [9–11]. Word embeddings usually learned from neural language models are well-known for representing the fine-granularity of words' semantic meaning. Word2Vec, developed by Mikolov et al. [9], has been shown to establish a new state-of-the-art performance in NLP tasks. Many other researchers have also contributed to the area of neural language model-based word embedding [12–15]. Word embeddings serve as machine-learned features for downstream classification tasks. Compared to the handcrafted keywords approach, the unsupervised word embedding algorithms can be adapted to different tasks and different corpora.

Mikolov et al. presented both the continuous bag of words (CBOW) and skip-gram models in their work. Our work extends the CBOW model. The CBOW model learns to predict the target word from the words in the context window surrounding the target word. The vector representations of the words in the context window are averaged in the process to predict the target word. Thus, the CBOW model treats every word in the context window equally in terms of their contributions to the prediction of the target word.

The CBOW model effectively learns a representation of the semantic meaning of each word as measured by a word-similarity evaluation, as long as the corpus is large enough. Mikolov et al. tested the CBOW model on the 6B-word Google news dataset. News articles are often written by professional reporters. Thus, the sentences are expected to be compact and the sentences should have meaningful semantic words. The intuitive and straightforward idea of equal contribution of every word in the context in the CBOW model is effective enough. However, when the corpus has plenty of slang expressions, abbreviations, emojis and unusual syntactics, such as in a microblog, the default combination that treats every word equally in the CBOW model might not be the optimum solution. We note that not all words in the context window contribute equally to the prediction of the target word. Incorporating all the words in the context window will largely limit the contribution of useful semantic words and bring more noisy or irrelevant words into the learning process.

Some existing word embedding work also learn weighted word embeddings [16–18]. Their weights, however, are based on existing pre-defined syntactic rules while ignoring the ultimate goal of the learned embedding.

Thus, motivated, we propose an alternative, to learn weights based on their relative importance in the classification task. Our intuition is that such learned weights place more emphasis on words that have comparatively more to contribute to the later classification task.

The chi-square (χ^2) statistical test is often used in feature selection for data mining [19]. It calculates the dependency between the individual feature and the class. By utilizing the χ^2 statistics for each word in the corpus as weights, we emphasize words that would later benefit the classification task and de-emphasize words that are usually independent of the class label.

We propose two algorithms based on the CBOW model. Inspired by the max-pooling layer of the convolutional neural network model (CNN), in a small context window setting, the first algorithm selects the word with the maximum χ^2 value to represent the context to predict the target word. The second algorithm keeps every word in the context window but weights them proportionally according to χ^2 values. The main contributions of this work can be summarized as follows.

- We are the first to propose to use the χ^2 statistic to weight the context in the CBOW model to enhance the contribution of the useful semantic words for the classification task and limit the noise brought by comparatively unimportant words.
- We propose two algorithms to train word embeddings using χ^2 on the task of healthcare tweet classification for the purpose of identification of truly health-related tweets from healthcare-noise data collected from a keyword-based approach.
- We evaluate our learned word embeddings for each of the proposed algorithms on two healthcare-related twitter corpora.

2. Related Work

Microblogging sites and online healthcare forums distribute many posts that share aspects of an individual's life and experience each day. The potential for working with great amounts of real healthcare-related clinical records, disease and symptom descriptions and even clinical transcripts attracted many researchers with interesting projects. To further extract and track healthcare information especially from users' social media profiles, the most basic and crucial task is to discriminate the healthcare-related tweets or target users from the massive pool of tweets and users that are irrelevant to the topic. Wang et al. note that prior research on eating disorders only focused on datasets collected from particular forums and communities [4]. Their goal was to identify behavioral patterns and psychometric properties of real users that suffered from eating disorders and not the patterns of people who simply discussed it on Twitter. They proposed a snowball sampling method to collect data based on the labeled eating disorder users' social media connections. Lamb et al. also used tweets to track influenza by distinguishing tweets about truly flu-affected people from the ones that express only concerns and awareness [3]. Because of the subtlety in distinguishing the two types of tweets, a keyword-based approach is insufficient since both sets contain typical keywords. Lamb et al.

proposed handcrafted feature types such as a word lexicon, stylometric features and part-of-speech template features. However, handcrafted features have the problem of scalability. Paul and Dredze [20] build an unsupervised topic model based on latent Dirichlet allocation (LDA) [21] to extract healthcare topics discussed in tweets. Ali et al. designed a platform to detect the trend and breakout of disease at an early stage [8]. They identify healthcare-related tweets from a pre-defined keywords list. Signorini et al. tracked H1N1 activity levels and public concerns on Twitter in real time [5]. They used SVM and handcrafted features such as age, recent clinic visits, etc., to track public sentiment with respect to H1N1, the swine flu. Interestingly, they found hygiene keywords such as "wash hands" positively correlated with the outbreak of the disease.

The popularity of the vector space model lies in its ability to quantify semantic similarities by the distributional structure of the language [22,23]. The assumption here is that words with similar distributional statistics tend to have similar semantic meaning. The distributional structure of the language can be captured by multi-dimensional vectors learned from the words' co-occurrence statistics. The research based on this assumption to quantify words' meaning and similarity is called distributional semantics [24]. There are multiple vector space models implementing distributional semantics, including Latent Semantic Analysis (LSA) [25] and Latent Dirichlet Allocation (LDA) [21]. Landauer and Dumais endowed LSA with a psychological interpretation and used LSA as a computational theory to solve the fundamental problem of knowledge acquisition and knowledge representation [26]. Enlightened by LSA's capability to capture similarity between words and its usage of Singular Value Decomposition (SVD) [27] to smooth the vector and handle the sparseness, Turney proposed capturing the relations between pairs of words and developed a new algorithm called Latent Relational Analysis (LRA) which also used SVD to smooth the data [28]. The context of a target word is defined as a small unordered number of words surrounding the target word in semantic space models. Pado and Lapata incorporated the syntactic information (dependency relations) to represent the context of the target word and formed a general framework for the construction of semantic space models [29]. The development of distributional vector representation of words greatly solves the scalability issue by releasing engineers from tedious handcrafted feature creation work. Neural network language models (NNLM) [30] produce a distributed vector representation of a word, known as a word embedding. The neural language model utilizes the neural network model to predict the word from the words appearing ahead of it [30], thus words with similar context will be mapped to close vector locations. In 2013, Mikolov et al. [9] used a three-layer neural network model to build word embeddings, to capture the semantic and syntactic regularities through the words in the context window of the target word. They proposed two models: the skip-gram and CBOW models. Both learn the vector representation of the word from the context in which the word resides. The skip-gram model trains the weights in the hidden layer and uses a softmax function to produce a probability of appearance in the context for every vocabulary word. Since it is very expensive to compute every word's probability in the corpus for every sample, Mikolov et al. adopt two mechanisms to further reduce the computation: hierarchical softmax and negative sampling. While hierarchical softmax uses a fixed Huffman tree structure with leaves as words in the vocabulary, negative sampling only samples n negative examples instead of the full vocabulary. Tian et al. extended the skip-gram model from Mikolov's work and generated multiple vector representations for each word in a probabilistic manner [31]. Researchers also incorporated syntactic information into neural language models. Levy and Goldberg [16] extended Mikolov et al. [9]'s skip-gram model by replacing the linear context with an NLP dependency-based syntactic context. Their model reported further improvement than the original model in the word similarity task (WordSim353 [32]).

In this work, we focus on an extension of the CBOW model. There are several existing works that also develop this line of research. Trask et al. [33] develop a very simple and effective method, incorporating additional information, the part-of-speech tag attached to each word during training. However, they did not invent a brand new model; instead, they used the CBOW model from Word2Vec. For example, for polysemy disambiguation to train the embedding of (banks, verb) in the sentence

"He banks at the bank", the input of CBOW is ("He", pronoun), ("at", adposition), ("the", determiner), ("bank", noun). For sentiment disambiguation, words are labeled with both the part-of-speech tag and sentiment for adjectives. Similarly, Liu et al. used part-of-speech information to weight the context window in the CBOW model [11]. They argue that in their learning algorithm the part-of-speech tags capture syntactic roles of words and encode inherently the syntactic relationships inside the word vector representation. However, the authors overlook the ultimate goal of the learned embeddings. The usage of the pre-defined syntactic rules to weight the context does not guarantee later success in the classification task in which the trained embeddings will be used.

Statistical measures have long been used in natural language processing. In terminology extraction, a fundamental processing step to extract technical terms from domain-specific textual corpora before complex NLP tasks, statistical measures such as mutual information, log likelihood and *t*-test are used to rank and identify the candidate terms from the texts. Zhang et al. developed a weighted voting algorithm that incorporated five existing term recognition algorithms to recognize both single- and multi-word terms in the text [34]. Most of the five term recognition algorithms adopt both statistical measures and frequency-based measures to rank the terms.

In this work, we propose using χ^2 to weight the context words according to the words' contribution to the classification task. There is existing work which also uses statistical measures in the vector space model. Gamallo introduced a count-based vector space model. Different from most of the co-occurrence context-based word vector space models, the context of the target word in Gamallo's model is the syntactic context (dependencies) of the target word [18]. To store the word–context sparse matrix, Gamallo used a global hash table. One inevitable weakness of count-based model is that the word–context matrix could be huge. Each word can have multiple contexts in the word–context matrix. To reduce dimensionality and only keep the most relevant and informative contexts of the target words, Gamallo used the log likelihood score to select the top R contexts for each word in the corpus. In our proposed algorithms, we also use an informativeness measure, the chi-square statistical test. Different from Gamallo's model, we use the chi-square statistical test to calculate the dependency between each word and the target class. The chi-square value for each word in the context is used as weights based on their relative importance in the later classification task. Another difference is that our work focuses on the neural language model while Gamallo's work extends from the count-based vector space model.

Word embedding algorithms have been applied to the healthcare field. Several studies have shown the performance of word embedding in extracting useful clinical concepts and information from either clinical notes or clinical free text [35,36].

3. Algorithms

In this section, we introduce two algorithms to learn word embeddings for healthcare tweet classification. We first introduce the background knowledge of the Chi-square statistical test and the CBOW model.

3.1. Chi-Square Statistical Test

The Chi-square (χ^2) statistical test has been widely accepted as a statistical hypothesis test to evaluate the dependency among two variables [37]. In natural language processing, the chi-square test is often applied to test the independence between the occurrence of the term and the occurrence of the class. It is often used as a feature selection method in NLP. Formula (1) is used to rank the terms that appear in the corpus [38].

$$\chi^2(\mathbb{D}, t, c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \quad (1)$$

where e_t and e_c are binary variables defined in a contingency table; $e_t = 1$ means the document contains term t and $e_t = 0$ means the document does not contain term t; $e_c = 1$ means the the document is in class c and $e_c = 0$ means the document is not in class c; N is the observed frequency in $\mathbb{D}$ and E is the expected frequency. For example, N_{11} is the observed frequency of documents appearing in class c containing term t; E_{11} is the expected frequency of t and c occurring together in a document assuming the term and class are independent. A higher value of χ^2 indicates that term t and class c are dependent, thus making term t a useful feature since the occurrence of t means the document is more likely to be seen in class c.

Utilizing the property of χ^2 that higher χ^2 values of term t indicate higher likelihood of occurrence in the class c, we use χ^2 to weight the context words in the CBOW model. The key aspect of our discovery is that words with higher χ^2 statistics tend to be keywords for class identification. Thus, we are using the chi-square statistical test to select the lexicon that particularly caters to the specific class identification task of short sentences such as tweets. Our rationale is that in our modified CBOW model, words are weighted according to their χ^2 statistics; words that are likely to be valuable for the classification task are more heavily weighted thus reducing the disturbance of the noise words which are not helpful comparatively to the later task.

3.2. Continuous Bag-of-Words Model (CBOW)

The CBOW model is a neural language model which consists of three layers: an input layer, a projection layer (also known as a hidden layer) and an output layer. In the input layer, the CBOW model uses context words both b before and after the target word to predict it; the vocabulary is represented as an input vocabulary matrix $\mathcal{V} \in \mathbb{R}^{n \times |V|}$; each column in $\mathcal{V}$ is represented as the vector representation of the words in the vocabulary; $\mathcal{V}$ is randomly initialized from the uniform distribution in the range $[-1, 1]$. The matrix $\mathcal{U} \in \mathbb{R}^{|V| \times n}$ is also initialized, which contains parameters learned by the neural language model during training. In the projection layer, the vector representation of the context, $\mathcal{C}$, is calculated as the arithmetic mean of the vector representation of all words w_i in the context window with b words before and after the target word, as shown in Formula (2).

$$\mathcal{C} = \frac{1}{2b} \sum_{i \in [-b,-1] \cup [1,b]} w_i \tag{2}$$

$\mathcal{C}$ is used to calculate the probability of the target word as shown in Formula (3), which is represented as a softmax function over the dot product of the vector representation of the context $\mathcal{C}$ and target word w_t.

$$p(w_t|\mathcal{C}) = \frac{e^{w_t \cdot \mathcal{C}}}{\sum_{w_i \in Vocab} e^{w_i \cdot \mathcal{C}}} \tag{3}$$

Finally, we can depict the loss function of the CBOW model in Formula (4).

$$\mathcal{L} = \sum_{w_t \in \mathbb{C}} \log p(w_t|\mathcal{C}) \tag{4}$$

where over all training tuples in the corpus $\mathbb{C}$, we are maximizing the probability of finding the target word w_t given $\mathcal{C}$, its context. However, to go over all the words in the vocabulary in Formula (3) is expensive. Instead of computing all the words in the vocabulary, distinguishing only the target word from several noise words largely reduces the computation load. This is called negative sampling. In this work, we adopt negative sampling when training the CBOW model. The window size $2b + 1$ and the word embedding dimension n are all hyperparameters.

By averaging the context words, the CBOW model overlooks the fact that the contribution of the words for the prediction should not be equal. We develop two algorithms to re-weight the context words.

3.3. Algorithm I

Inspired by the good performance of the max-pooling layer in the convolutional neural network model (CNN) in which only the maximum value within a window of the feature map is returned, instead of incorporating all the context words, we only select the word with the maximum χ^2 value to represent the context. Thus, Formula (2) of calculating the vector representation of the context is substituted by Formula (5)

$$\mathcal{C} = \underset{\chi^2(w_i),\ i\in[-b,-1]\cup[1,b]}{\arg\max} w_i \tag{5}$$

where $\chi^2(\cdot)$ represents the chi-square statistical value of w_i for the target class. Although the trained word embedding complies to the property of linear compositionality, in a small context window size, and a corpus containing as much noise such as Twitter, we choose the word from the context window that is likely to contribute the most to the later classification task. The expectation is that this will be more beneficial than the original strategy of averaging all of the context words. We emphasize a small context window size in this algorithm because when the context window is large, there is a greater chance that more than one word with a substantial contribution to the prediction will be included in the context window, thus selecting only the word with the maximum χ^2 statistic might not be beneficial. We test our algorithm in a context window size of 3($b = 1$), and in Section 4.4 show that our approach can improve performance on data from Twitter.

3.4. Algorithm II

In Algorithm I described above in Section 3.3, we remove all the other words that have smaller χ^2 values and only keep the word with the maximum value to represent the context. In contrast, Algorithm II weights every word in the context window proportionally according to its χ^2 test statistic. Thus, Formula (2) calculating the vector representation of the context is substituted by Formula (6).

$$\mathcal{C} = \frac{1}{\sum_{j\in[-b,-1]\cup[1,b]} \chi^2(w_j)} \sum_{i\in[-b,-1]\cup[1,b]} \chi^2(x_i) w_i \tag{6}$$

In the original CBOW model, the words in the context window are treated equally assuming equal contribution to the prediction task. However, the assumption is generally not held based on language characteristics. Previous work also tries to improve this by the pre-defined syntactic rules such as using part-of-speech to weight the words. For example, nouns and verbs are usually more important than prepositions, pronouns and conjunctions; thus they are often weighted heavier comparatively. However, they overlook the purpose and the usage of the learned embedding. The weighting mechanism based on pre-defined rules is not necessarily in line with the classification task. We propose using the χ^2 test statistics as the weighting strategy, which directly links the weights to the term's correlation to the classification task.

4. Experimental Method

In this section, we describe experiments on the two proposed algorithms on two carefully selected datasets.

4.1. Datasets

Our goal is to extract healthcare information from people's profile and Twitter posts. We are more interested in tweets that talk about real disease symptoms as shown in Examples 4–6 which we named healthcare-related tweets, rather than those tweets that popularize the healthcare information as seen in Examples 1–3 in the Introduction which we named healthcare-noise tweets. Our current classification task is a challenging one since both heathcare-related tweets and healthcare-noise tweets might contain keywords such as "flu" and "health" which makes basic filtering approaches unworkable.

We use two datasets in the experiments. The first dataset, called the healthcare dataset, is from Paul and Dredze [20]. It was collected and labeled using Amazon's mechanical turk (AMT) and has two labels: health-related and health-unrelated. All tweets were collected using healthcare keywords filtering as a first step; thus, even health-unrelated tweets contain healthcare keywords. Tweets that were not about a particular person's health (e.g., advertisements of flu shots and news information about the flu) were labeled as unrelated. The statistics of the healthcare dataset are shown in Table 1. The second, called the influenza dataset, is from Lamb et al.'s work [3]. It was also collected from Twitter. It contains tweets posted during the 2009 and 2012 outbreaks of swine and bird influenza. The data is also labeled as influenza-related and influenza-unrelated by AMT workers. The statistics of the influenza dataset are shown in Table 2.

Table 1. Tweet counts for the healthcare dataset (from [20]).

Data	Healthcare-Related	Healthcare-Unrelated	Total
Train	868	1301	2169
Test	217	325	532

Table 2. Tweet counts for the influenza dataset (from [3]).

Data	Influenza-Related	Influenza-Unrelated	Total
Train	2148	1609	3757
Test	537	402	939

4.2. Baselines

We compare the proposed two algorithms with the following baseline methods for healthcare tweet classification.

1. tf-idf + SVM: we calculate the tf-idf scores [39] for the words of each tweet as the features and train a support vector machine (SVM) classifier [40] using the Liblinear library [41].
2. skip-gram + CNN: we train Mikolov et al.'s skip-gram model on the training set for both datasets. We learn the word embedding for each word in the corpus to use as features and train a convolutional neural network model (CNN) for classification [42].
3. CBOW + CNN: we train the original CBOW model and learn the word embeddings for the word in the corpus as a feature and train a convolutional neural network model (CNN) for classification [42].

4.3. Experimental Setup

Preprocessing is necessary when working with the text of tweets. We strip the punctuation, the html tags and hypertext links, and downcase all letters. We use Tensorflow [43] to implement the two proposed algorithms. In both cases, we keep the default model setting as in the Tensorflow skip-gram model codec in Github (https://github.com/tensorflow/models/blob/master/tutorials/embedding/word2vec.py). Word embeddings are learned using a window size of $b = 1$, embedding dimension $n = 128$ and a negative sampling rate of 64. Words with frequency smaller than 3 are eliminated from the vocabulary. We use the stochastic gradient decent optimizer (SGD) to train the two algorithms with a learning rate of 1.0. To perform the χ^2 statistical test, we use sklearn [44] on the training sets of the two corpora. We train CNN models for the two datasets for the classification task. We use filter sizes of 3, 4 and 5 and 128 filters for each filter size in the training process.

4.4. Evaluation and Results

We completed the χ^2 statistical test on the two Twitter datasets. Boxplots for the χ^2 values in both datasets are shown in Figure 1. As we can see, most of the words in the two corpora have a very low χ^2 value. We list the words in Table 3 that ranked highest by χ^2 value. Words with higher χ^2 value are

recognizable as plausible keywords for the identification of the health-related tweets. Using the χ^2 statistics in Algorithms I and II, the result of the experiment is shown in Table 4.

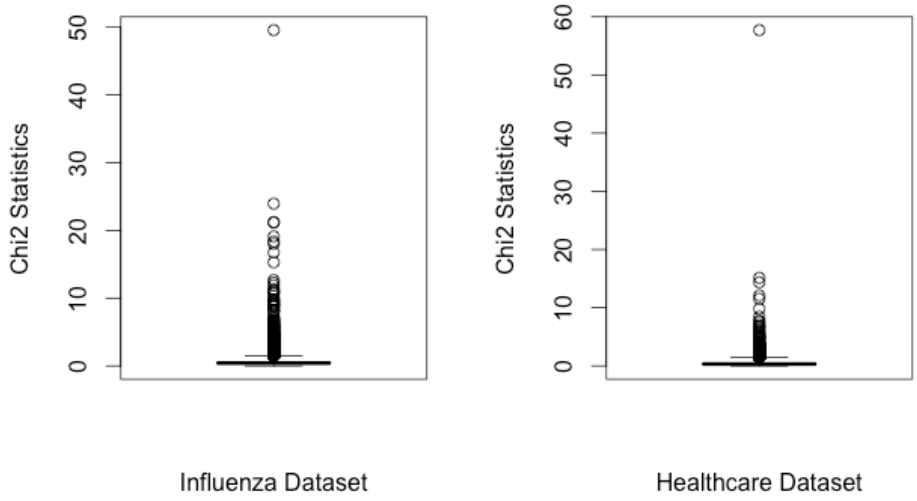

Figure 1. Boxplot of the values of the Chi-square (χ^2) statistical test for the healthcare dataset and the influenza dataset.

Table 3. Words with highest χ^2 value for both datasets.

	Healthcare Dataset	Influenza Dataset
1	headache	sick
2	sick	vaccine
3	allergies	throat
4	feeling	fear
5	flu	news
6	surgery	swineflu
7	cramps	bird
8	throat	shot

Table 4. Comparison of testset classification accuracy across the two datasets using word embeddings from various models. SVM: support vector machine; CNN: convolutional neural network; CBOW: continuous bag of words.

Method	Healthcare Dataset	Influenza Dataset
tf-idf + SVM baseline	59.96	57.18
Skip-gram + CNN baseline	66.61	66.99
CBOW + CNN baseline	69.00	66.67
Algorithm I + CNN	69.19	72.31
Algorithm II + CNN	69.93	72.84

Since we assume equal importance for the identification of both of the two classes, related versus unrelated, we choose accuracy, the commonly used evaluation criteria, as the metric to measure the classification performance as shown in Formula (7).

$$accuracy(y_{lab}, y_{pred}) = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} 1(y_{pred} = y_{lab}) \tag{7}$$

Since tweets in both classes (related versus unrelated) contain the keywords of the topic, it is not surprising that the keywords-based approach in the tf-idf + SVM baseline behaves poorly for both

datasets. For the two Word2Vec baselines, the CBOW model performs better than the skip-gram model for the smaller (healthcare) dataset; they have very similar results in the influenza dataset (which is a larger dataset). Overall, Algorithms I and II improve over the CBOW baseline model by 1.35% and 9.23% respectively. We can see that Algorithm I which chooses the word with the maximum χ^2 value also performs well in terms of accuracy. As we noted earlier, we have a small context window size of 3($b = 1$). When the context window is larger, more context words are included. It might not be optimal to choose only the word with the maximum statistical measurement score to form the context representation. Our experimental results indicate that the χ^2 weighting scheme of Algorithm II generally outperforms the others.

5. Conclusions

To improve tweet classification accuracy, we use the chi-square (χ^2) statistical test statistic to directly link the weight of each term to its correlation to the tweet classification tasks. We proposed two algorithms: in Algorithm I, assuming a small context window setting, we select the word with the maximum χ^2 value; in Algorithm II, we use the χ^2 statistics to proportionally weight the words in the context window. Our evaluation result shows improvement over the original CBOW Word2Vec model by as much as 9.2%.

Some natural directions for future work include hyperparameter optimization (e.g., selecting the best window size), and testing of other term weighting functions.

Author Contributions: Sicong Kuang and Brian Davison conceived the original idea, designed the experiments, and wrote the article; Sicong Kuang performed the experiments and analyzed the data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ginsberg, J.; Mohebbi, M.H.; Patel, R.S.; Brammer, L.; Smolinski, M.S.; Brilliant, L. Detecting influenza epidemics using search engine query data. *Nature* **2009**, *457*, 1012–1014.
2. Butler, D. When Google got flu wrong. *Nature* **2013**, *494*, 155.
3. Lamb, A.; Paul, M.J.; Dredze, M. Separating Fact from Fear: Tracking Flu Infections on Twitter. In Proceedings of the HLT-NAACL, Atlanta, GA, USA, 9–15 June 2013; pp. 789–795.
4. Wang, T.; Brede, M.; Ianni, A.; Mentzakis, E. Detecting and Characterizing Eating-Disorder Communities on Social Media. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM), Cambridge, UK, 6–10 February 2017; pp. 91–100.
5. Signorini, A.; Segre, A.M.; Polgreen, P.M. The use of Twitter to track levels of disease activity and public concern in the US during the influenza A H1N1 pandemic. *PLoS ONE* **2011**, *6*, e19467.
6. Paul, M.J.; Dredze, M. You are what you Tweet: Analyzing Twitter for public health. In Proceedings of the AAAI International Conference on Weblogs and Social Media (ICWSM), Barcelona, Spain, 17–21 July 2011; Volume 20, pp. 265–272.
7. Huang, X.; Smith, M.C.; Paul, M.J.; Ryzhkov, D.; Quinn, S.C.; Broniatowski, D.A.; Dredze, M. Examining Patterns of Influenza Vaccination in Social Media. In Proceedings of the AAAI Joint Workshop on Health Intelligence (W3PHIAI), San Francisco, CA, USA, 4–5 February 2017.
8. Ali, A.; Magdy, W.; Vogel, S. A tool for monitoring and analyzing healthcare tweets. In Proceedings of the ACM SIGIR Workshop on Health Search & Discovery, Dublin, Ireland, 28 July–1 August 2013; p. 23.
9. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
10. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global Vectors for Word Representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
11. Liu, Q.; Ling, Z.H.; Jiang, H.; Hu, Y. Part-of-Speech Relevance Weights for Learning Word Embeddings. *arXiv* **2016**, arXiv:1603.07695.

12. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
13. Tang, D.; Wei, F.; Yang, N.; Zhou, M.; Liu, T.; Qin, B. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 23–25 June 2014.
14. Ling, W.; Dyer, C.; Black, A.; Trancoso, I. Two/too simple adaptations of word2vec for syntax problems. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 31 May–5 June 2015; pp. 1299–1304.
15. Chen, Y.; Perozzi, B.; Al-Rfou, R.; Skiena, S. The Expressive Power of Word Embeddings. In Proceedings of the ICML 2013 Workshop on Deep Learning for Audio, Speech, and Language Processing, Atlanta, GA, USA, 16 June 2013.
16. Levy, O.; Goldberg, Y. Dependency-Based Word Embeddings. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics; Baltimore, MD, USA, 23–25 June 2014; Volume 2, pp. 302–308.
17. Qiu, L.; Cao, Y.; Nie, Z.; Yu, Y.; Rui, Y. Learning Word Representation Considering Proximity and Ambiguity. In Proceedings of the AAAI, Québec City, QC, Canada, 27–31 July 2014; pp. 1572–1578.
18. Gamallo, P. Comparing explicit and predictive distributional semantic models endowed with syntactic contexts. *Lang. Resour. Eval.* **2016**, *51*, 727–743.
19. Howell, D.C. Chi-square test: Analysis of contingency tables. In *International Encyclopedia of Statistical Science*; Springer: Berlin, Germany, 2011; pp. 250–252.
20. Paul, M.J.; Dredze, M. A model for mining public health topics from Twitter. *Health* **2012**, *11*, 16–26.
21. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
22. Harris, Z.S. Distributional structure. *Word* **1954**, *10*, 146–162.
23. Firth, J.R. A synopsis of linguistic theory 1930–1955. In *Studies in Linguistic Analysis (Special Volume of the Philological Society)*; Blackwell: Oxford, UK, 1957; pp. 1–32.
24. Clark, S. Vector space models of lexical meaning. In *Handbook of Contemporary Semantics*; University of Cambridge Computer Laboratory: Cambridge, UK, 2014; pp. 1–43.
25. Deerwester, S.; Dumais, S.T.; Furnas, G.W.; Landauer, T.K.; Harshman, R. Indexing by Latent Semantic Analysis. *J. Am. Soc. Inf. Sci.* **1990**, *41*, 391–407.
26. Landauer, T.K.; Dumais, S.T. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychol. Rev.* **1997**, *104*, 211.
27. Golub, G.H.; Reinsch, C. Singular value decomposition and least squares solutions. *Numer. Math.* **1970**, *14*, 403–420.
28. Turney, P.D. Similarity of semantic relations. *Comput. Linguist.* **2006**, *32*, 379–416.
29. Padó, S.; Lapata, M. Dependency-based construction of semantic space models. *Comput. Linguist.* **2007**, *33*, 161–199.
30. Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
31. Tian, F.; Dai, H.; Bian, J.; Gao, B.; Zhang, R.; Chen, E.; Liu, T.Y. A Probabilistic Model for Learning Multi-Prototype Word Embeddings. In Proceedings of the 25th Annual Conference on Computational Linguistics (COLING), Dublin, Ireland, 23–29 August 2014; pp. 151–160.
32. Finkelstein, L.; Gabrilovich, E.; Matias, Y.; Rivlin, E.; Solan, Z.; Wolfman, G.; Ruppin, E. Placing search in context: The concept revisited. In Proceedings of the 10th International Conference on World Wide Web, Hong Kong, China, 1–5 May 2001; ACM: New York, NY, USA, 2011; pp. 406–414.
33. Trask, A.; Michalak, P.; Liu, J. Sense2vec—A Fast and Accurate Method for Word Sense Disambiguation in Neural Word Embeddings. *arXiv* **2015**, arXiv:1511.06388.
34. Zhang, Z.; Brewster, C.; Ciravegna, F. A Comparative Evaluation of Term Recognition Algorithms. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008), Marrakech, Morocco, 28–30 May 2008; pp. 28–31.
35. Kholghi, M.; Vine, L.D.; Sitbon, L.; Zuccon, G.; Nguyen, A.N. The Benefits of Word Embeddings Features for Active Learning in Clinical Information Extraction. In Proceedings of the Australasian Language Technology Association Workshop 2016, Caulfield, Australia, 5–7 December 2016; pp. 25–34.

36. Choi, Y.; Chiu, C.Y.I.; Sontag, D. Learning low-dimensional representations of medical concepts. In Proceedings of the AMIA Summits on Translational Science, San Francisco, CA, USA, 21–24 March 2016; American Medical Informatics Association: Bethesda, MD, USA, 2016; p. 41.
37. Pearson, K. On the Criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. In *Breakthroughs in Statistics: Methodology and Distribution*; Kotz, S., Johnson, N.L., Eds.; Springer: New York, NY, USA, 1992; pp. 11–28.
38. Manning, C.D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.J.; McClosky, D. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 22–27 June 2014; pp. 55–60.
39. Robertson, S. Understanding inverse document frequency: On theoretical arguments for IDF. *J. Doc.* **2004**, *60*, 503–520.
40. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297.
41. Fan, R.E.; Chang, K.W.; Hsieh, C.J.; Wang, X.R.; Lin, C.J. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.* **2008**, *9*, 1871–1874.
42. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
43. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: tensorflow.org (accessed on 16 June 2017).
44. Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Mueller, A.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; et al. API design for machine learning software: Experiences from the scikit-learn project. In Proceedings of the ECML/PKDD Workshop: Languages for Data Mining and Machine Learning, Prague, Czech Republic, 23–27 September 2013; pp. 108–122.