EYE WITNESS MESSAGE IDENTIFICATION ON FOREST FIRES DISASTER USING CONVOLUTIONAL NEURAL NETWORK

Rinaldi[1], M. Reza Faisal[2], M. Itqan Mazdadi[3], Radityo Adi Nugroho[4], Friska Abadi[5]
[12345]Ilmu Komputer FMIPA ULM
A. Yani St. KM 36 Banjarbaru, South Kalimantan
Email: 1611016110013@mhs.ulm.ac.id

Abstract

Social media, one of which is Twitter, is a medium for disseminating information that is growing rapidly at this time. The advantage of Twitter which has such a huge impact is its speed in spreading news and information that is happening. One of the information that is often reported through social media is information about natural disasters. Therefore, a lot of research on sensor social networks has been carried out by researchers using data from social media with the aim of obtaining valid data for the disaster emergency response process. In this study, the classification of eye witness messages for forest fires was carried out using Convolutional Neural Network and feature extraction Word2Vec with dimensions of 100. Twitter data used amounted to 3000 data and divided into 3 classes, namely eyewitnesses, non-eyewitnesses, and unknowns. The research was conducted to determine the accuracy performance obtained from testing using several types of configurations hyperparameter. Based on the results of the tests carried out, the best accuracy value was 81.97%.

Keywords: *Convolutional Neural Network, Natural Disasters, Twitter, Word2Vec*

1. INTRODUCTION

The rapid development of social media nowadays has a big influence on society, especially in the field of information dissemination. Social media has proven to have a big impact in accelerating the spread of news and news because social media users will immediately report what they see after seeing or experiencing an event or event. Information about natural disasters is one of the events that is often reported through social media [3]. Therefore, a lot of research has been carried out using social media data as a social network sensor [4, 9] which can be used as a medium for disseminating the main source of information. In addition, data from social media can be used to identify eye witness messages during natural disasters [2] with the aim of obtaining credible and valid important data for the disaster emergency response process.

Previously, there were two important things done in research on social network sensors, namely feature extraction to convert social media data into structured data. The feature extraction process can be done in various ways such as TF-IDF, Unigram and Bigram, Word Embedding and so on. Then the structured data is processed to create a classification model. Previous research was conducted using n-gram and bag of word and TF-IDF weighting as a feature extraction technique and random forest method in making a classification model for eye witness identification into 3, namely eyewitness, non-eyewitness, and don't know based on messages from social media twitter [2]. Based on research using n-gram and bag-of-word, it still produces out-of-vocabolary and semantic ambiguity problems that do not present word relationships between sentences in social media data, resulting in suboptimal classification performance.

A social sensor network research has been conducted to monitor epidemic outbreaks [6]. In addition, other studies also use convolutional neural network and word2vec to classify news articles and twitter tweets [1]. Based on this research, it is concluded that the word embedding technique and also the classification model using a convolutional neural network can improve the accuracy performance of the data classification.

Based on the above problems, this study will focus on the method of making a convolutional neural network classification model and Word2Vec feature extraction. This research will use twitter data related to forest fires that occurred in Indonesia during the period 2014-2019. The data to be used is data that already has an eye witness grouping label. The data is then processed using the word embedding technique so that the resulting features are low in dimensions and building a classification model using a convolutional neural network to create a model that is expected to perform classification optimally.

2. RESEARCH METHODOLOGY

2.1 Dataset

The data that will be used for the classification process is twitter data obtained in the period 2014-1019 totaling 3000 data. The data will be divided into 3 main classes. The amount of data sharing used is shown in table 1.

Table 1 Amount of data share for classification

Eyewitness	Non-Eyewitness	Don't Know	Total
1000	1000	1000	3000

2.2 Preprocessing

2.2.1 Remove Duplicate

Remove duplicate step is done to remove duplicate data. Remove duplicate is used for tweet data that has the same tweet_id and text content. The purpose of remove duplicate is so that the data has a unique value because no other data is the same.

2.2.2 Labelling

Data that has gone through the remove duplicate process will then be manually labeled according to the type of eyewitness, the label type category is shown in table 2.

Table 2 Eyewitness label category

Label type	Characteristics
Eyewitness	Use words as first person or telling experience. Sentences that express emotions. Sentence describing the location of the incident.
Non-Eyewitness	Sentences that use a third-person perspective or tell events that have experienced other people. Sentences expressing concern or hope An article or news with a link
Don't Know	is not a sentence that informs about forest fires

2.2.3 Removing

Removing is the process of removing unnecessary fields. Fields other than the text column will be deleted because the data to be used is tweet data in the text column. In the tweet data that has been collected, several fields have been removed, including screen_name, timestamp, timestamp_epochs, tweet_id, tweet_url, user_id, and username.

2.2.4 Cleansing

Cleansing is done to remove characters other than text, including special characters in comments such as punctuation marks (such as: comma (,), period (.), Question mark (?), Exclamation point (!) And so on), numeric numbers (0 - 9), and other characters (such as: $,%, *, and so on). Cleansing is also carried out to remove existing links / URLs in tweet data.

2.2.5 Case Folding

Case folding is the stage of converting the entire text in a document into a standard form. In this study case folding is used to change all words into lowercase letters. The purpose of case folding is to make it easier to correct words in the text. Examples of the results of applying cleansing and case folding to the data are presented in table 3.

Table 3 Cleansing and Case folding data

Data before Cleansing & Case Folding	Data after Cleansing & Case Folding
Håнåнå kebakaran jenggot tuh antek" pks dr org"nya Hijbutahrir terutama #Jonru cs...cm org koplax yg gampang... http://fb.me/6sFXlyEcj	kebakaran jenggot tuh antek pks dr org nya hijbutahrir terutama jonru cs cm org koplax yg gampang
Zola Bilang Hanya 3 Bulan, Kerugian Akibat Kabut Asap Mencapai Rp12 Triliun: Kebakaran hutan dan lahan di Jambi... http://bit.ly/1Rk69kN	zola bilang hanya bulan kerugian akibat kabut asap mencapai rp triliun kebakaran hutan dan lahan di jambi
Zaman bpk presiden, Malaysia dan Singapur terima ekspor ASAP. Zaman JKW pemerintahan Malaysia thn ini mengucapkan Terimakasih ke pak Jkw sebab tdk ada kebakaran hutan lagi yg asapnya NOL.	zaman bpk presiden malaysia dan singapur terima ekspor asap zaman jkw pemerintahan malaysia thn ini mengucapkan terimakasih ke pak jkw sebab tdk ada kebakaran hutan lagi yg asapnya nol

2.3 Word Embedding

Creating a word embedding model is useful for making every word in the data have a vector value. The method used to create word embedding is Word2Vec and uses 95000 twitter data, including data to be used in the classification stage. Then the pretrained Word2vec [7] model was made with a total of 100 dimensions.

2.4 Classification

The classification process is carried out by the k-fold cross validation method to divide the training data partitions and test data according to the k value. In this study, k with a value of 10 is used so that the data will be divided into 10 partitions.

CNN is used to classify disaster messages on twitter into several classes of eye witnesses. This CNN did well in performing text classification tasks (such as sentiment analysis) and has since become the basic standard for a new architecture for text classification [8]. CNN is also very effective at deriving features from fixed length segments of the entire dataset and works well for Natural Language Processing (NLP) problems. There is no addition of handcrafted features because all features are studied directly by the algorithm from the dataset. The CNN model architecture used in the classification process is as follows:

2.4.1. *Input Layer*

In the input layer, Twitter text data that has been transformed into an index will be forwarded to the embedding layer to add vector values according to the predetermined dimensions, namely 100 dimensions. The length of the data that will be used as input consists of two, namely the length of the data with a value of 17, which is the mode value of the word length of the entire data and the maximum length of words from the entire data, which is 57.

2.4.2. *Embedding Layer*

The embedding layer functions to add vector values obtained by using Word2Vec [7] to 100 dimensions of data so that the input data dimension is n x 100 where n is the length of the input data or the length of the twitter data to be used. The vector value will be initiated according to the number of words in the data used for testing according to the word index in the pretrained word embedding. For words that do not have a vector value, the pretrained word embedding will be initiated with a random vector value. In the embedding layer, a trainable or non-static vector parameter attribute is added with the value true so that the model can update the vector value during the training process.

2.4.3. *Convolutional Layer*

This CNN model was formed using 3 layers of a convolutional layer. The way the convolutional layer works starts with a filter that will move according to the size of the filter kernel and the number of filters then vertically along the entire input matrix. After the convolution process is carried out based on filter size and filter kernel, as well as non-linear operations using the ReLU activation function, a feature map is obtained that contains important features with lower dimensions in each convolutional layer. Then this feature map will be input to the max pooling layer in the next layer.

2.4.4. *Max Pooling Layer*

The Max Pooling Layer will take the highest value of the elements that are in the scope of the filter window, so that the most important information is obtained from the convolutional feature map of each convolutional layer. The matrix results from each max pooling layer will then be combined and followed by a flatten process to convert the matrix into a vector which will then be forwarded to the fully connected layer for the classification process.

2.4.5. *Fully Connected Layer*

The output from the previous hidden layer, in the form of a feature map that has been reshaped into a vector, will go through the fully connected layer before being connected to the output layer for the classification process of data into predetermined classes.

2.4.6. *Output Layer*

In this layer, the softmax activation function and the loss function sparse_categorical_crossentropy are used, because there are 3 multiclass output variables represented in labels consisting of the number 0 for the eyewitness class, 1 for the non-eyewitness class, and 2 for the unknown class.

The CNN classification model was trained with batch size = 50 and epoch = 30. As well as using the Adam optimization function as an optimizer to reduce loss functions or errors contained in neurons and filters. To avoid overfitting during the training process, a dropout regulation technique was used [6] with a rate of 0.5 for the default hyperparameter and 0.2 obtained from hyperparameter tuning which randomly deactivates the number of neurons during the training phase. The design of the convolutional neural network model to be built is shown in Figure 1.

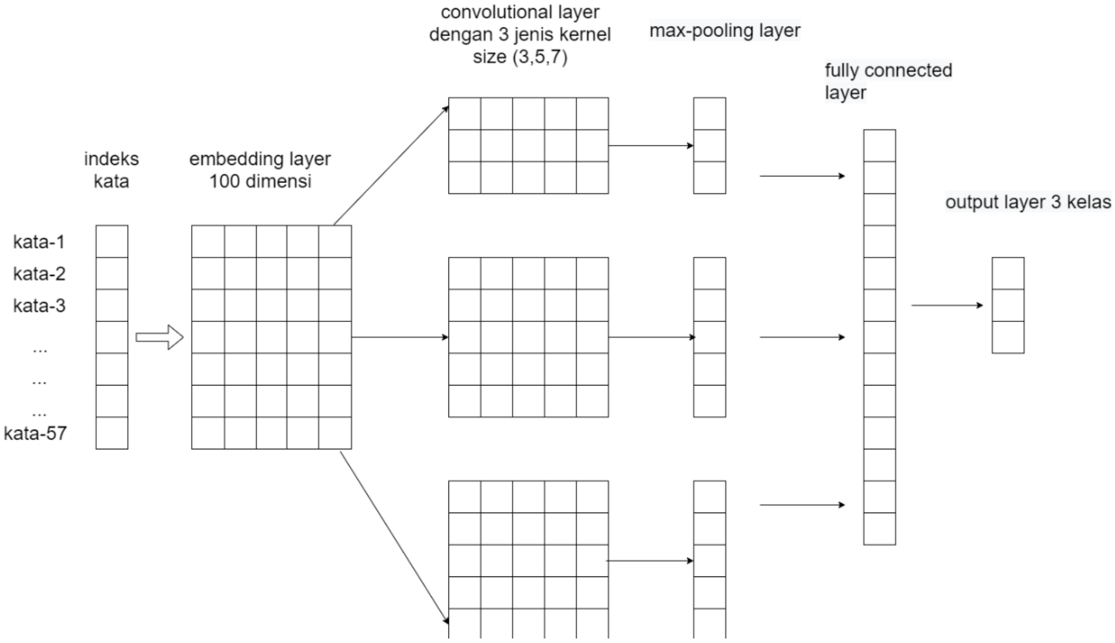


Figure 1 CNN Classification Model Design

2.5 Evaluation

After the classification process, a performance evaluation is carried out based on the confusion matrix of the test results to obtain accuracy, sensitivity / recall, and specificity. The formula for accuracy is in the following equation:

$$\text{Accutation} = \frac{\text{SS+BB+TT}}{\text{total data}} \text{X } 100\% \qquad \text{... (1)}$$

Rumus untuk perhitungan sensitivity dan spesificity terdapat pada persamaan berikut :

$$\text{Sensitivity eyewitness} = \frac{\text{TP(S)}}{\text{TP(S)+FN(S)}} \text{x}100\% \qquad \text{... (2)}$$

$$\text{Sensitivity non-eyewitness} = \frac{\text{TP(B)}}{\text{TP(B)+FN(B)}} \text{x}100\% \qquad \text{... (3)}$$

$$\text{Sensitivity don't know} = \frac{\text{TP(T)}}{\text{TP(T)+FN(T)}} \text{x}100\% \qquad \text{... (4)}$$

$$\text{Specificity eyewitness} = \frac{\text{TN(S)}}{\text{TN(S)+FP(S)}} \text{x}100\% \qquad \text{... (5)}$$

$$\text{Specificity non-eyewitness} = \frac{\text{TN(B)}}{\text{TN(B)+FP(B)}} \text{x}100\% \qquad \text{... (6)}$$

$$\text{Specificity don't know} = \frac{\text{TN(T)}}{\text{TN(T)+FP(T)}} \text{x}100\% \qquad \text{... (7)}$$

3. RESULTS AND DISCUSSION

3.1 Results

The data used for testing were 3000 twitter data regarding forest fire disasters which were divided into 3 classes, namely eye witnesses, non-eye witnesses, and unknowns. The data that has been obtained are first cleaned at the preprocessing stage to equalize the shape of the data. After the data is clean, the data can be used for the classification process. Before classification using the CNN model, first the text data will be assigned a vector value with a pretrained word embbedding that is trained from the 95000 fabric twitter data. The process of converting words into vectors is carried out using Word2Vec with dimensions of 100.The words that have been assigned a vector value are shown in table 5.

Table 5 Data with vector values

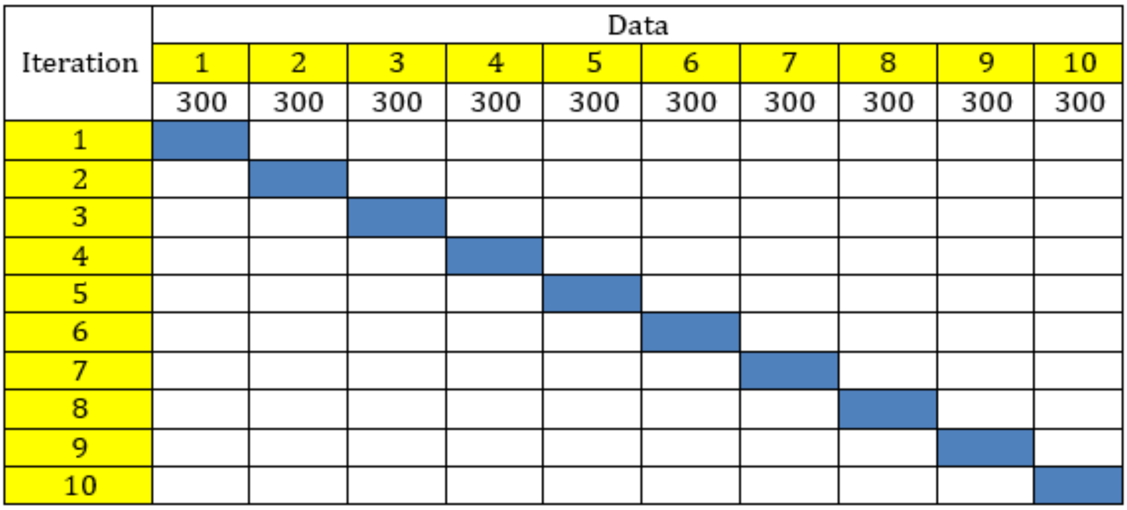
word	v1	v2	v99	V100
gempa	0.22752914	-0.026566252	-0.22735706	-0.855792
di	0.247078	0.028301394	-0.6187041	-0.45651615
ada	-0.5308554	0.15626042	-0.48434815	-0.44323444
bencana	0.49454045	0.27460954	-0.4654763	-0.72823805
dan	0.2143934	0.2183165	0.09396511	-0.25565255
dengan	0.5574413	-0.2484517	0.13603958	-0.6985348
kebakaran	-0.17860343	0.63471633	0.04312394	-0.2287788
untuk	0.4147127	-0.0794483	-0.21024393	-0.8214908
saya	-0.20656265	0.20617461	-0.20224054	-0.7100925
hutan	-0.16703685	0.6430618	0.30531082	-0.64005995
bot	-0.20343119	0.42710945	-0.32928905	-0.9651787
asap	-0.2360047	0.809586	0.32357427	-0.36645967
halo	0.11015763	0.015900763	0.008810158	-0.9534915
melaporkan	0.16459662	0.112611026	0.025729423	-0.8675922
silakan	0.21155356	0.19847843	-0.24407634	-0.9958149
yang	0.38930404	0.0943736	0.17304125	-0.2651987
ini	-0.01939561	-0.27471688	-0.3516146	-0.64498776
yg	0.08138049	0.17102708	-0.13241854	-0.26250315
aku	-0.45107993	0.2409588	0.011166853	-0.67897326
rt	0.3275481	0.50895375	-0.43042934	0.15715387
ya	0.110101916	-0.2794417	-0.25006995	-0.0980093
ga	-0.59050184	0.15488352	-0.14995517	-0.3044784
kabut	-0.035050724	0.9773146	0.40793917	-0.44069836
lagi	-0.64936864	-0.0037389724	-0.15791205	-0.49976176
banjir	0.2081302	0.17649962	0.29254964	-0.5504369

After the vector value for each word is obtained through word2vec, then testing the classification model is carried out. Testing was carried out with 4 types of hyperparameter combinations. The variable of each hyperparameter is shown in table 5. The value of the input length is divided into two, namely 17 is the value of the word length mode and 57 is the maximum value of the word length of all data used for classification.

Table 5 hyperparameter variables for testing

Variable	Testing 1	Testing 2	Testing 3	Testing 4
Input length	17	17	57	57
Filter	50	100	50	100
Kernel filter	3,4,5	3,5,7	3,4,5	3,5,7

Testing the CNN classification model uses the 10-fold cross validation data validation method to divide the data partition into training data and test data where there will be 9 training data partitions and 1 partition as test data. The data partition division is shown in Figure 2.



Iteration	Data									
	1	2	3	4	5	6	7	8	9	10
	300	300	300	300	300	300	300	300	300	300
1	■									
2		■								
3			■							
4				■						
5					■					
6						■				
7							■			
8								■		
9									■	
10										■

Figure 2 Data divide using 10-fold CV

After that, the classification process is carried out using a Convolutional Neural Network. The result of the classification is in the form of confusion matrix which is calculated to find the performance of accuracy, sensitivity / recall, and specificity. The results of the calculation of the performance of the CNN model are shown in Figure 3.

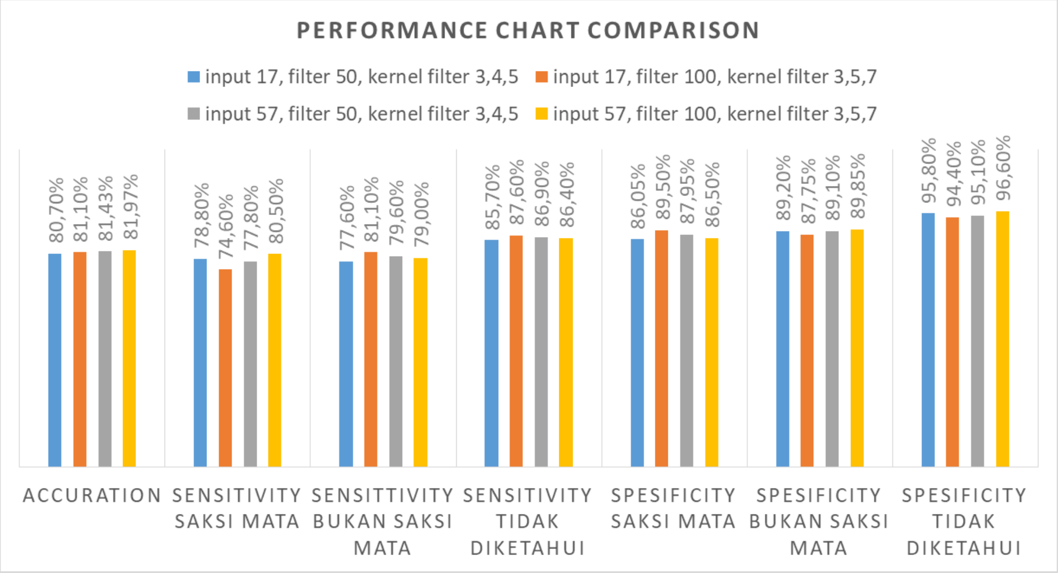


Figure 3 Performance chart

3.2 Discussion

Tests are carried out using 4 types of hyperparameter configurations. From the 4 types of testing, the results obtained were in the form of accuracy, sensitivity and specificity for each class. Overall, the highest accuracy is 81.97% which is obtained from testing using a hyperparameter combination using an input length of 57 which is the maximum word length of all data, 100 filters, and a filter kernel of 3.5,7. The next result that is calculated is the sensitivity and specificity for each class based on the test parameters. The sensitivity value shows the accuracy of the model in predicting the positive class, while the specificity value is used to measure the accuracy of the model in predicting the negative class. The sensitivity and specificity values that are considered are the sensitivity and specificity values in the eye witness class. The overall test shows that the sensitivity value using a combination of the hyperparameter input 57, filter 100, and filter kernel 3,5,7 has a higher value of 80.50% compared to other hyperparameter tests. Meanwhile, the highest specificity value is obtained from testing using the input hyperparameter 17 which is the mode length of the entire data, filter 100 and filter kernel 3,5,7.

Based on the results of several tests, it was found that the CNN classification model with hyperparameter maximum word input length, filter 100, and filter kernel 3,5,7 had the best performance based on the accuracy and sensitivity values of the eyewitness class. Overall, CNN performed quite well in classifying eyewitness messages by class using several types of hyperparameters. However, the test hyperparameter with the best value for classification is to use the maximum data length, filter 100, and filter kernel 3,5,7.

4. CONCLUSION

Based on the results of testing the Convolutional Neural Network classification model by calculating the value of accuracy, sensitivity, and specificity, it can be concluded that the CNN model can be used to classify eye witness messages for forest fires and has a fairly good performance. The results of the accuracy performance of each test have a value between 70% -80% where the highest accuracy value is obtained by using a hyperparameter with a maximum input length, filter 100, and filter kernel 3,5,7. However, in general, each test performed quite well in classifying eye-witness messages from forest fires. For further research it is recommended to use other word embedding methods such as fastText and GloVe to find out the results of CNN classification performance when using word embedding other than Word2Vec.

REFERENCES

[1] Beakchoel Jang, Inhwan Kim, and Jong Wook Kim. 2019. "Word2Vec Convolutional Neural Networks for Classification of News Articles and Tweets". *PLos ONE* 14(8): e0220976. https://doi.org/10.1371/journal.pone.0220976

[2] Kiran Zahra, Muhammad Imran, and Frank O Ostermann. 2020. "Automatic Identification of Eyewitness Messages on Twitter during Disasters". *Information and Processing* 57(1):102-107. https://doi.org/10.1016/j.ipm.2019.102107

[3] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2015. "Processing Social Media Messages in Mass Emergency: A Survey". *ACM Computing Surveys* 47(4):67 pp 1-38. https://dl.acm.org/doi/10.1145/2771588

[4] Nicholas A Christakis and James H Fowler. 2010. "Social Network Sensors for Early Detection of Contagious Outbreaks*".PLoS ONE* 5(9): e12948–e12948. https://doi.org/10.1371/journal.pone.0012948

[5] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks From Overfitting". *The Journal of Machine Learning Research* 15(1):1929-1958. https://dl.acm.org/doi/epdf/10.5555/2627435.2670313

[6] Ovidiu Serban, Nicholas Thapen, Brendan Maginnis, Chris Hankin, and Virginia Foot. 2019. "Real-time Preprocessing of Social Media with SENTINEL: A Syndromic Surveilance System Incorporating Deep Learning for Health Classification". *Information Processing and Management* 56(1): 1166-1184. https://doi.org/10.1016/j.ipm.2018.04.011

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Efficient Estimation of Word Representations in Vector Space". *arXiv:1301.3781 [cs]*. https://arxiv.org/abs/1301.3781

[8] Yoon Kim. 2014. "Convolutional Neural Networks for Sentence Classification". *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* 1746-1751. https://www.aclweb.org/anthology/D14-1181

[9] Yuri Kryvasheyeu, Hachui Chen, Esteban Moro, Pascal Van Hentenryck, and Manuel Cabrian. 2015. "Performance of Social Network Sensors during Hurricane Sandy". PLos ONE 10(2): 1-19. https://doi.org/10.5061/dryad.15fv2