Natural Language Processing With Disaster Tweets

Yoonjung Choi, Abhiteja Mandava, Ishaan Bhalla, Sakruthi Avirineni
CMPE 255 Data Mining
San Jose State University

***Abstract* – Social Network Service have become not only an important sources of emergency information during disaster but also a medium for expressing immediate responses of warning, evacuation or rescue. As a result, predicting context of SNS is a crucial concern in our society. By analyzing context, we can utilize this study to track, monitor and predict disasters from the real time data and this study would help making prediction model. In this paper, data was retrieved from the company Figure-Eight, and key problem is dealing with the natural language processing by using ensemble model handling different feature vectors. We proposed ensemble model using combinations with different feature vectors and classifiers. The results are compared with each classifier with different feature vectors and existing ensemble classifiers that apply the same data set to several classifiers. After analyzing data, factors normalizing data set and transforming to feature vectors were identified, and measures to improve accuracy were proposed.**

***Keywords* – NLP, PCA, Ensemble**

I. Introduction

Social Network Service has been playing a crucial role in communicating in our society and Natural Language Processing has been widely used to analyze SNS and extract potential patterns. Twitter is one of the popular SNS platforms and many tweets has been delivered in emergency situation. Since there are demands for companies to utilize this tweets, we investigated natural language processes and developed prediction model having the best performance.

In this paper, for pre-processing, we cleaned data set from unnecessary information such as URL, Emojis or HTML tags and normalized data set by using useful algorithms; tokenizer, stopwords and lemmatization. We transformed given text into feature vectors by using Count Vectorizer, Inverse Document Frequency, Word2Vec and Word2Vec with PCA applied, and trained each numerical feature vector on different models; Decision Tree, Support Vector Machine, Logistic Regression, and Ensemble Model. We found each combination how each model has the higher performance on which feature vectors. Then, after fine-tuning, we made ensemble model training each classifier with feature vectors that resulted in higher accuracy, unlike an existing ensemble model using the same data set.

As a result, the ensemble model of SVM with Count Vectorizer, Logistic Regression with Count Vectorizer and Decision Tree with Tf-Idf gave the better results. The Results were compared based on different performance matrics such as Accuracy, Recall, Precision, F1 Score.

II. Data Exploration

A. Data set

The data set has been collected from the company figure-eight and originally shared on their 'Data For Everyone' website [1]. We found the data set from Kaggle Competition [2]. It contains 7613 tweets data with the following features:

Feature	Dtype	Descrition
id	int64	
keyword	object	39 null-values
location.	object	2533 null-valus
text	object	tweetter content
target	int64	0:non-disaster, 1:disaster

Fig. 1. The table shows feature, type, and description.

The below figures are bar charts of the count for top 15 'keywords' feature of each target.

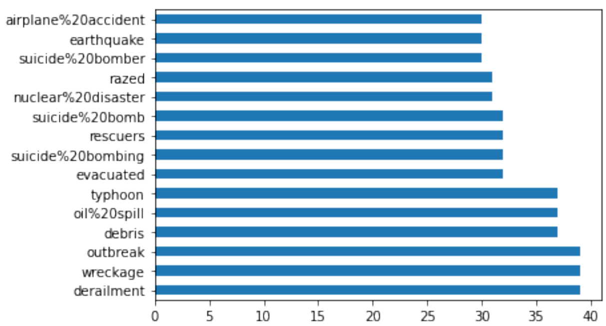

Fig. 2. Top 15 of disaster tweets' keywords

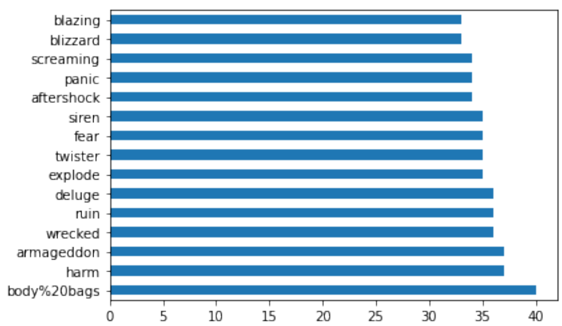

Fig. 3. Top 15 of non-disaster tweets' keywords

The figure 4 shows the percentage of feature 'target's distribution.

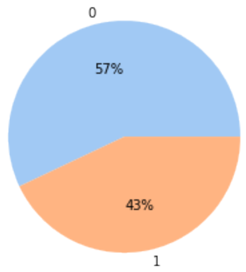

Fig. 4. there are 4342 samples for target(0:non-disaster) and 3271 samples for target(1:disaster)

The figure shows the missing values of 'location' and 'keywords' features. The 'location' feature does not have format and it is not generated automatically. That's why it has dirty values, such as 'have car; will travel', 'peekskill. new york, 10566', or 'milky way'. We do not use 'location' as a feature.

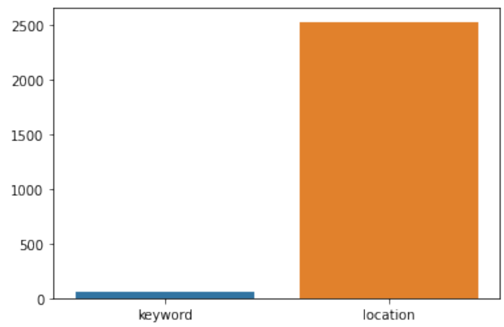

Fig. 5. The missing values of data set

In the Natural Language Processing, there are several common processing steps and many practical algorithms. We need to clean data, normalize data, and transform data into numerical feature vectors.

B. Data Cleaning

We should modify data to filter meaningless data. For cleaning text, we changed all words to lowercase, removed URL, HTML tags, Emojis, punctuation and ASCII codes.

text
Crying out for more! Set me ablaze
On plus side LOOK AT THE SKY LAST NIGHT IT WAS ABLAZE http://t.co/qqsmshaJ3N
@PhDSquares #mufc they've built so much hype around new acquisitions but I doubt they will set the EPL ablaze this season.
INEC Office in Abia Set Ablaze - http://t.co/3ImaomknnA
Barbados #Bridgetown JAMAICA ‰ÛÒ Two cars set ablaze: SANTA CRUZ ‰ÛÓ Head of the St Elizabeth Police Superintende... http://t.co/wDUEaj8Q4J

Fig. 6. The table shows partial of original 'text' feature

CleanText
crying out for more set me ablaze
on plus side look at the sky last night it was ablaze
phdsquares mufc theyve built so much hype around new acquisitions but i doubt they will set the epl ablaze this season
inec office in abia set ablaze
barbados bridgetown jamaica two cars set ablaze santa cruz head of the st elizabeth police superintende

Fig. 7. The table shows changes after cleaned meaningless data

C. Data Preprocessing

Now, we have a cleaned text set and we should apply some methods to normalize words. NLTK [3] provides easy-to use interfaces for natural language processing

1) Tokenization: Tokenization divides strings into lists of substrings. We can use library to find the words and punctuation in a sentences.

TokenizedText
['crying', 'out', 'for', 'more', 'set', 'me', 'ablaze']
['on', 'plus', 'side', 'look', 'at', 'the', 'sky', 'last', 'night', 'it', 'was', 'ablaze']
['phdsquares', 'mufc', 'theyve', 'built', 'so', 'much', 'hype', 'around', 'new', 'acquisitions', 'but', 'i', 'doubt', 'they', 'will', 'set', 'the', 'epl', 'ablaze', 'this', 'season']
['inec', 'office', 'in', 'abia', 'set', 'ablaze']
['barbados', 'bridgetown', 'jamaica', 'two', 'cars', 'set', 'ablaze', 'santa', 'cruz', 'head', 'of', 'the', 'st', 'elizabeth', 'police', 'superintende']

Fig. 8. The table shows changes after tokenization

2) Stopwords: we should remove commonly used words (such as "the", "a", "is", "in").

RemoveStopWords
['crying', 'set', 'ablaze']
['plus', 'side', 'look', 'sky', 'last', 'night', 'ablaze']
['phdsquares', 'mufc', 'theyve', 'built', 'much', 'hype', 'around', 'new', 'acquisitions', 'doubt', 'set', 'epl', 'ablaze', 'season']
['inec', 'office', 'abia', 'set', 'ablaze']
['barbados', 'bridgetown', 'jamaica', 'two', 'cars', 'set', 'ablaze', 'santa', 'cruz', 'head', 'st', 'elizabeth', 'police', 'superintende']

Fig. 9. The table shows changes after stopwords

3) Stemming: Stemming is the process of producing morphological variants of a root/base word. For example, words such as "Likes", "liked", "likely" and "liking" will be reduced to "like" after stemming. There are different algorithms for stemming. Porter Stemmer, one of them, is a basic stemmer and it is straightforward and fast to run.

PorterStemmer
['cri', 'set', 'ablaz']
['plu', 'side', 'look', 'sky', 'last', 'night', 'ablaz']
['phdsquar', 'mufc', 'theyv', 'built', 'much', 'hype', 'around', 'new', 'acquisit', 'doubt', 'set', 'epl', 'ablaz', 'season']
['inec', 'offic', 'abia', 'set', 'ablaz']
['barbado', 'bridgetown', 'jamaica', 'two', 'car', 'set', 'ablaz', 'santa', 'cruz', 'head', 'st', 'elizabeth', 'polic', 'superintend']

Fig. 10. The table shows changes after stemming

4) Lemmatization: Lemmatization is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form. Both stemming and lemmatization are word normalization techniques, but we can find the word in dictionary in case of lemmatization. For example, original words 'populated' changed 'popul' in Stemming, but it is not changed in lemmatization. Lemmatization is more better performed than Stemming [4]. We decided to apply lemmatization.

LemmatizedText
['cry', 'set', 'ablaze']
['plus', 'side', 'look', 'sky', 'last', 'night', 'ablaze']
['phdsquares', 'mufc', 'theyve', 'built', 'much', 'hype', 'around', 'new', 'acquisition', 'doubt', 'set', 'epl', 'ablaze', 'season']
['inec', 'office', 'abia', 'set', 'ablaze']
['barbados', 'bridgetown', 'jamaica', 'two', 'car', 'set', 'ablaze', 'santa', 'cruz', 'head', 'st', 'elizabeth', 'police', 'superintende']

Fig. 11. The table shows changes after lemmatization

5) Data Visualization: After normalized text, we made data visualization by using word cloud. In disaster tweet's words, we can discover disaster related words; suicide, police, news, kill, attack, death, california, storm, flood. In the other hand, the non disaster tweets shows that time, want, great, feel, read, but also injury or emergency are found.

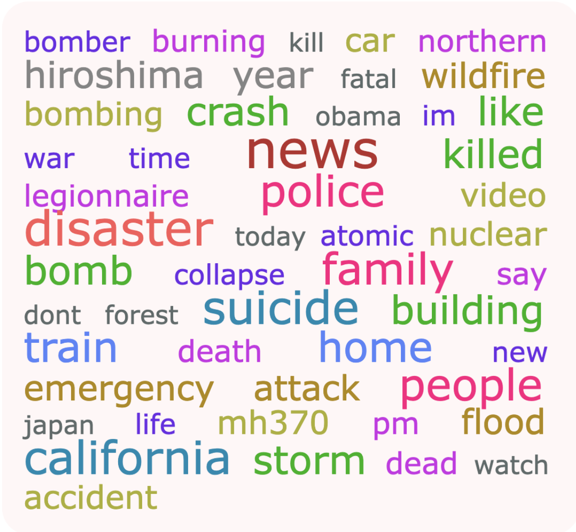

Fig. 12. target(1) disaster tweets' words

Fig. 13. target(0) non disaster tweets' words

The Figure 13 represent histogram of the number of words at each sample.

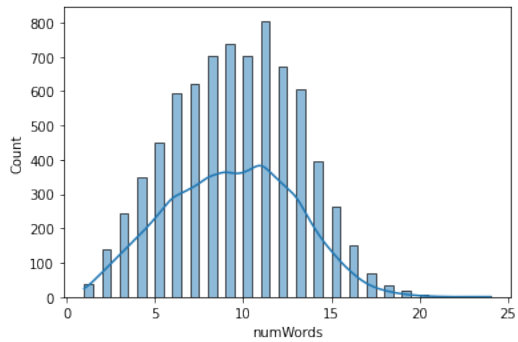

Fig. 14. histogram of lengths of tweets' words

D. Transforming numerical feature vectors

Bag of Words model is a simplified representation used in natural language processing. A text is represented as the bag of its words, disregarding grammar and describes the occurrences of words with in a document.

1) CountVectorizer: CountVectorizer can be used for bag of words model. This convert a collection of text documents to a matrics of token counts.

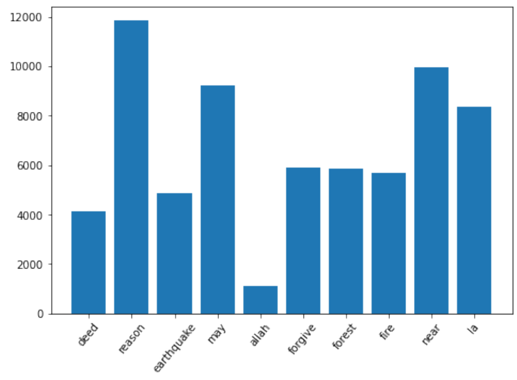

Fig. 15. Bar chart shows the number of ten words from dictionary.

2) *TF-IDF:* The Term Frequency Inverse Document Frequency is a measure of whether a term is common or rare. It gives weight more to a term that occurs in only a few documents.

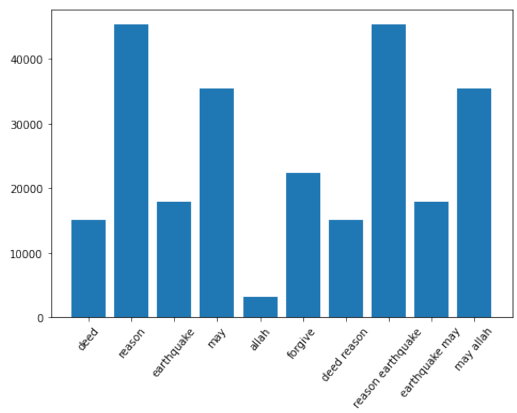

Fig. 16. Bar chart shows the number of ten words from dictionary.

3) *Word2Vec:* Word2Vec uses a neural network model to learn word associations from a large corpus of text [5]. Word2Vec represents words in vector space in a way of similar meaning words are positioned in close locations but dissimilar words are placed far away.

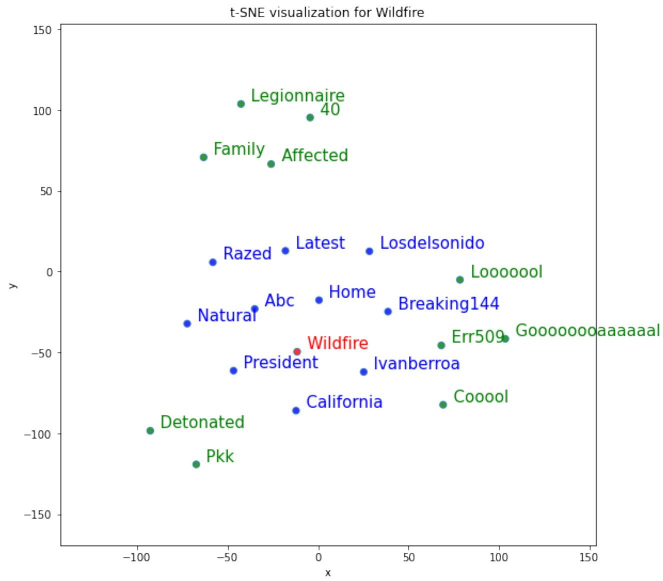

Fig. 17. For word 'wildfire', blue words represent most similar words and green words represent most dissimilar words.

4) *Word2Vec with PCA applied:* As principal component analysis is a strategy to reduce dimension, we applied PCA with 100 components on feature vectors from Word2Vec. The below figure is shown when applying PCA with 2 components.

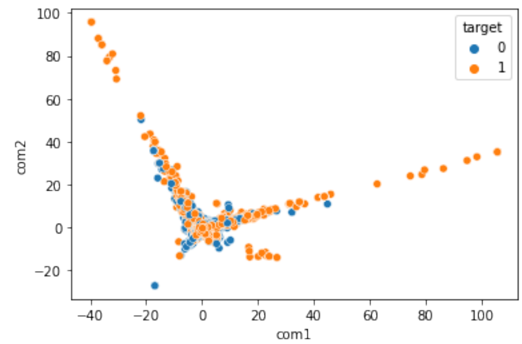

Fig. 18. Scatter plot of feature vector applied Word2Vec and PCA.

III. METHODS

We trained each numerical feature vectors on the basic models to find which feature vectors can yield better performance. For the fine-tuning, we adjusted parameters on the model with the selected feature vector. We repeated the same steps on other models. We also trained each feature vectors with ensemble method.

1) *SVM:* Support Vector Machine is a supervised learning model used for classification and regression problems. We trained each numerical feature vectors on basic SVM, which means no changes of parameters. In case of SVM, CountVectorizer feature vector has higher accuracy and f1 score than other feature vectors.

SVM	Count Vector	Tf-Idf	W2V	W2V + PCA
accuracy	0.799	0.761	0.624	0.709
Recall	0.639	0.493	0.163	1.434
Precision	0.864	0.923	0.857	0.809
F1 Score	0.735	0.643	0.274	0.565

Fig. 19. Performance of basic SVM with different feature vectors

We adjusted parameters to yield best accuracy. In the final SVM model, it has default C value as 1, gamma value as 'auto', kernel value as 'sigmoid'. We obtained the result and confusion matrics of the model.

SVM	Accuracy	Recall	Precision	F1
score	0.800	0.668	0.839	0.744

Fig. 20. Performance of fine-tuned SVM

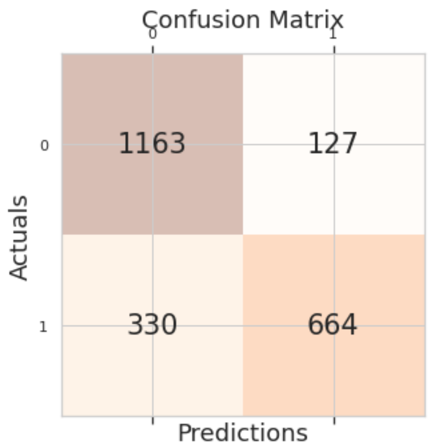

Fig. 21. Confusion matrics of fine-tuned SVM

2) Logistic Regression: Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. It is also a supervised learning model used for classification problems. We trained each feature vectors on basic Logistic Regression without fine-tuning, and Count Vector has higher accuracy as well.

LR	Count Vector	Tf-Idf	W2V	W2V + PCA
accuracy	0.797	0.776	0.669	0.751
Recall	0.693	0.539	0.314	0.603
Precision	0.813	0.903	0.806	0.776
F1 Score	0.749	0.677	0.452	0.678

Fig. 22. Performance of basic Logistic Regression with different feature vectors

From the fine-tuning, we finalized parameters as C=0.15, penalty='l2', tol=0.001, solver='saga', random state=42, max iter=1000. We obtained the following result and confusion matrics:

LR	Accuracy	Recall	Precision	F1
score	0.800	0.672	0.837	0.746

Fig. 23. Performance of fine-tuned Logistic Regression

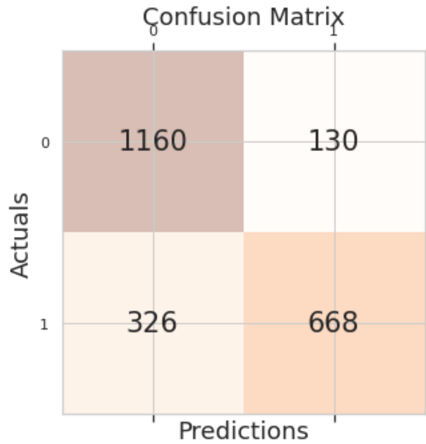

Fig. 24. Confusion matrics of fine-tuned Logistic Regression

3) Decision Tree: Decision Tree is a non-parametric supervised learning method used for classification and regression problems. We trained each numerical feature vectors on basic decision tree, and Tf-Idf feature vector has a little bit higher accuracy rather than others.

DT	Count Vector	Tf-Idf	W2V	W2V + PCA
accuracy	0.749	0.752	0.655	0.683
Recall	0.671	0.684	0.616	0.621
Precision	0.731	0.730	0.614	0.640
F1 Score	0.700	0.706	0.615	0.630

Fig. 25. Performance of basic Decision Tree with different feature vectors

From the fine-tuning, we finalized parameters as min samples split=8. We obtained the result and confusion matrics.

DT	Accuracy	Recall	Precision	F1
score	0.756	0.675	0.741	0.706

Fig. 26. Performance of fine-tuned Decision Tree

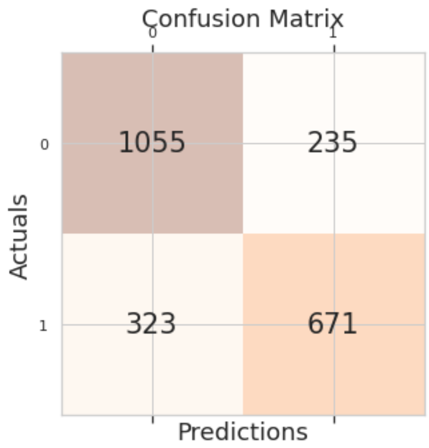

Fig. 27. Confusion matrics of of fine-tuned Decision Tree

4) Ensemble: Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would. We used hard voting classifier and trained each feature vectors on ensemble model consisted of SVM, Logistic Regression and Decision Tree. The figure 27 shows each ensemble's accuracy and ensemble model with CountVectorizer feature vector yields better accuracy

Hard Voting	Count Vector	Tf-Idf	Word2Vec
Accuracy	0.805	0.785	0.626

Fig. 28. Accuracy of each ensemble model with each feature vector

Based on the hard voting, we made custom ensemble model combined of SVM with CountVectorizer, Logistic Regression with CountVectorizer, and Decision Tree with Tf-Idf. As a result, we got 0.806 accuracy. The following figure is about confusion matrics of custom ensemble model.

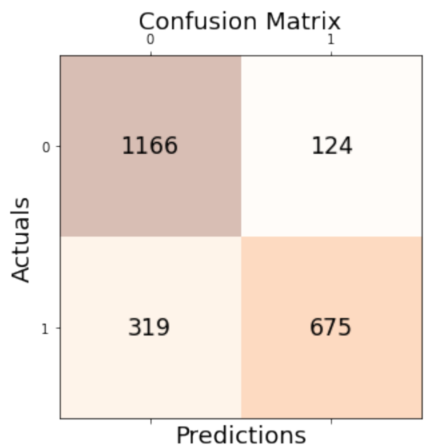

Fig. 29. Confusion matrics of ensemble model

IV. Comparison

A. Performance Matrics

1. Accuracy

Accuracy is a metric that generally describes how the model performs across all classes. It is calculated as the ratio between the number of correct predictions to the total number of predictions.

2. Precision

The precision is calculated as the ratio between the number of Positive samples correctly classified to the total number of samples classified as Positive (either correctly or incorrectly).

3. Recall

The recall is calculated as the ratio between the number of Positive samples correctly classified as Positive to the total number of Positive samples.

4. F1 Score

The F1-score combines the precision and recall of a classifier into a single metric.

5. ROC Curve

ROC curve is a graphical plot that illustrates recall(x-axis) and precision(y-axis) for different thresholds.

B. Comparison

(insert ROC Curve graph)

Also, other submissions of Kaggle competition used similar steps using algorithms to transform to numerical feature vectors and classifiers including ensemble models as well. However, there is no comparison to find each combination of feature vectors and classifiers, to make custom ensemble models. Our model considered finding suitable combination of a feature vector and a classifier and then, applying ensemble model.

V. CONCLUSION

VI. Limitations And Future research

We obtained the qualified data set from company, so we assumed that content of data are true. However, the thing that content could be fake is the main limitation of this study. Overcoming these limitations can be done in future research. By dealing with distinguishing the content is fake or not first, we can predict emergency situations and properly respond them.

REFERENCES

[1] OPPEN, URL: `https://appen.com/datasets-resource-center`.

[2] Kaggle, "Natural Language Processing with Disaster Tweets", URL: `https://www.kaggle.com/competitions/ nlp-getting-started/data`.

[3] NLTK, "Natural Language Toolkit", URL: `https://www.nltk.org/index.html`.

[4] Baeldung, “Naturalstemming-vs-lemmatization”, URL: https://www.baeldung.com/cs/stemming-vs-lemmatization.

[5] Wikipedia, “word2vec”, URL: https://en.wikipedia.org/wiki/Word2vec.