Computer Modeling in Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2021.014355

ARTICLE

An Automated System to Predict Popular Cybersecurity News Using Document Embeddings

Ramsha Saeed[1], Saddaf Rubab[1], Sara Asif[1], Malik M. Khan[1], Saeed Murtaza[1], Seifedine Kadry[2], Yunyoung Nam[3,*] and Muhammad Attique Khan[4,*]

[1]National University of Sciences and Technology, Islamabad, Pakistan

[2]Beriut Arab University, Beirut, Lebanon

[3]Department of Computer Science and Engineering, Soonchunhyang University, Asan, Korea

[4]Hitec University, Taxila, Pakistan

[*]Corresponding Authors: Yunyoung Nam. Email: ynam@sch.ac.kr; Muhammad Attique Khan. Email: attique@ciitwah.edu.pk

Received: 19 September 2020 Accepted: 03 February 2021

ABSTRACT

The substantial competition among the news industries puts editors under the pressure of posting news articles which are likely to gain more user attention. Anticipating the popularity of news articles can help the editorial teams in making decisions about posting a news article. Article similarity extracted from the articles posted within a small period of time is found to be a useful feature in existing popularity prediction approaches. This work proposes a new approach to estimate the popularity of news articles by adding semantics in the article similarity based approach of popularity estimation. A semantically enriched model is proposed which estimates news popularity by measuring cosine similarity between document embeddings of the news articles. Word2vec model has been used to generate distributed representations of the news content. In this work, we define popularity as the number of times a news article is posted on different websites. We collect data from different websites that post news concerning the domain of cybersecurity and estimate the popularity of cybersecurity news. The proposed approach is compared with different models and it is shown that it outperforms the other models.

KEYWORDS

Embeddings; semantics; cosine similarity; popularity; word2vec

1 Introduction

With the expansion of the web and the continuously growing news industry, online news consumption has flourished substantially. Hence, the competition for gaining readers' attention on news articles is increasing day by day. The task of monitoring the popularity of news articles has become quite challenging for the editorial teams in any news organization these days. The editors are under the mounting pressure of maximizing the impact of the content they post. The ability to forecast the popularity of news content can improve the competitiveness of news platforms. It can help the editorial teams manage their web content by effectively deciding which news items

to post, promote, or demote. Given a large volume of online news content, the editors need to anticipate the popularity of news articles quite early after publication or before publication.

The task of popularity prediction can broadly be classified into two major approaches, the approaches that predict popularity after publication of the news content and the approaches that predict popularity before its publication [1]. The first approach utilizes data such as user interaction with the news articles in terms of the number of likes, shares, and comments. Using this approach, popularity can be estimated after the article gets published online as the features used by this approach become available after article's publication. Hence, a waiting period is introduced for popularity estimation as the articles do not get social interaction instantly after being published. Moreover, this approach is not effective for cases where negligible social feedback is available. The second approach uses the content of the articles to make predictions. This approach can help the content providers decide beforehand whether to post certain content or not. The machine learning models for content based approach use features, a few of which are: Bag of Words (BoW) [2], n-grams word embeddings [3], grammatical construction [4], and keywords count, closeness to topics, data channel, sentiment polarity [5].

The content based approach has also been used to predict popularity by using article similarity features [6,7]. However, in these techniques similarity among the news articles has been computed between term frequencies of the articles. This does not take into account the overall meaning of the text ignoring word order and usage of synonymous words to communicate the same meaning. In order to incorporate semantics in the content based approach of measuring popularity using article similarity, this work proposes to combine document embeddings of news articles with similarity matching techniques. The proposed approach estimates the popularity of a news item by matching the similarity of its embeddings with embeddings of other news articles published on different websites. We define popularity as the number of times a news article is shared on different websites. In this work, we intend to predict the popularity of news articles related to the domain of cybersecurity. We extract data from different websites that post cybersecurity news articles. To the best of our knowledge, no such dataset already exists. Hence, the primary contributions of this research are:

- Creating first of its kind dataset comprising news articles from multiple cybersecurity news websites
- Developing a semantically enriched article similarity approach for content based popularity prediction systems

The rest of the paper is organized as follows: Section 2 presents the literature review, Section 3 details the process of corpus collection, Section 4 describes the proposed methodology, Section 5 discusses the results while Section 6 concludes the paper.

2 Literature Review

Rizos et al. [8] modeled the popularity prediction of online content as a binary classification problem. They used Random Forest (RF) model fed with features based on user graphs and comment trees to predict news popularity after its publication. It was found by Keneshloo et al. [9] that content based features indicating the freshness of an article such as topic intersection, click count, number of similar articles, and content similarity of the most similar articles are the most important features for predicting popularity. Multi-Linear Regression model (MLR) fed with all these features was employed to determine news popularity. Fernandes et al. [10] used RF as a binary classification model to predict the popularity of a given article prior to its publication using a range of features including but not limited to digital media content, number of keywords, day

of the week publishing the news, sentiment, and closeness to topics. Khan et al. [5] observed that type of data channel, closeness to the topics, article publication on weekends, and the number of shares of keywords were the useful features and Gradient Boosting (GB) was found to be the best classifier. Zhou et al. [11] divided popularity into three levels. They used Decision Tree (DT) fed with user-based features, time features, and textual features to predict the popularity. Choudhary et al. [12] selected a set of optimum features by applying correlation coefficient techniques such as Kendall–Rank, Pearson, and Spearman. They used Naive Bayes (NB) and RF for classification where NB performed better than RF. They further reduced the feature set by using Genetic algorithm. Neural Netwrok (NN) [13] yielded the best prediction score for this feature set. Obiedat et al. [14] used five different classification algorithms C4.5, RF, Logistic Function, Bayes Net, and Simple Cart to predict popularity of news articles where RF showed the best results. Xiao et al. [15] built time sensitivity based model to predict popularity values of Twitter posts when they are published at different times. The posts were decomposed into syntactic units to reduce data sparsity, and features such as temporal information, neighborhood influence, user activity level are exploited. Yang et al. [16] proposed a named entity based model for extracting textual factors. A popularity gain matrix is learned for each named entity overall semantic topics to predict news popularity. Multivariate Adaptive Regression Splines (MARS) model was proposed by Moniz et al. [2] which used BoW and sentiment scores to predict the article's popularity upon its publication. Hensinger et al. [17] used Support Vector Machine (SVM) fed with BoW and topic labels to determine popularity. Long et al. [18] used ensemble classifiers fed with topics and Doc2Vec vectors of news content to predict popularity. Guan et al. [3] coupled Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM) to classify popular news. Liu et al. [4] proposed a linear regression model using grammatical construction of news titles as features to predict popularity. Besides grammatical construction, the grade of the author, publication time, content length, and category score were also used as features. Popularity prediction has been modeled as clustering problem by Aswani et al. [19] where the chaotic cuckoo search algorithm was combined with K-means clustering to cluster popular and unpopular news articles. Abbar et al. [6] converted the articles into vectors using Term Frequency Inverse Document Frequency (TFIDF) and predicted popularity by measuring similarity between these vectors. Natarajan et al. [7] estimated popularity using cosine similarity between the number of common words of news articles and tweets. However, similarity between textual information does not always depend on a particular set of keywords rather it depends more on the meaning of the text.

3 Dataset Creation

We collected about 30,000 news articles from 20 websites that publish cybersecurity news. We scraped data for a period of 4 years, i.e., from 2016 to 2019. To evaluate the results, we selected a test set of 1000 samples annotated with four popularity levels, i.e., unpopular (UP), weakly popular (WP), moderately popular (MP), and very popular (VP).

A news from a given source is UP if it is not published by any other site in the dataset within the period of a week, a news is WP if it is published on just one other website within a week, a news is MP if it is published on two other websites within a week, and a news is VP if it is published on more than two websites within a week.

Tab. 1 shows the number of samples for all popularity levels.

Table 1: Number of samples for all popularity levels in the dataset

Categories	Number of samples
Unpopular	812
Weakly popular	110
Moderately popular	57
Very popular	21

4 Proposed Framework: Word2vec Based Popularity Prediction System (W2V-PPS)

In this section, we detail the description of the proposed system to predict popular cybersecurity news. Fig. 1 shows the overall architecture of the system.

4.1 Data Preprocessing

Data is preprocessed by removing punctuations and stopwords from the text. We remove punctuations and stopwords from the data as they do not play any significant role in analyzing the content for popularity estimation.

4.2 Embedding Vector Generation using Word2vec

Word embeddings are the vector representations of words where semantically similar words are closer to each other in the vector space. We train skipgram model of word2vec [20] on 30,000 cybersecurity news articles. The model is trained using the following parameters: $size = 300$, $min\_count = 5$, $window = 10$. Here *size* is the dimensionality of the vector, *min_count* is the minimum frequency of words in the corpus to be considered, and *window* is the maximum number of words considered in the context. Skipgram model finds word representations useful for predicting the contextual words in the given sentence. Given a set of words in a sentence $w_1, w_2, w_3, \ldots, w_T$, skipgram model maximizes the average log probability as:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j} \mid wt) \tag{1}$$

where T is the size of the corpus, and c is the training window size over the context.

Let v_w and v'_w represent input and output vectors of a word w, and W represents the total number of words in the vocabulary, the basic skipgram model defines the probability $p(w_t + j \mid w_t)$ using the softmax function as:

$$p(w_o \mid w_I) = \frac{exp(v'^{T}_{w_o} v_{w_I})}{\sum_{w=1}^{W} exp(v'^{T}_{w} v_{w_I})} \tag{2}$$

4.3 Embedding Vector Averaging

After generating representation for individual words, we need to produce document vectors to represent the entire news article. Since averaging word vectors has been found to be a suitable technique to represent text such as phrases [20], we use this technique to create document vectors of news articles. Document vectors are created by averaging the vectors of all the words in a document. Let $w_1, w_2, \ldots, w_t$ be a set of words in document d and $v_1, v_2, \ldots, v_t$ be the set of

embeddings for each word w in the document, then the document embedding D can be represented as the mean of the vectors $v_1, v_2, \ldots, v_t$ as given in the following equation:

$$D = \frac{1}{t}\sum_{i=1}^{t} v_i \tag{3}$$

where t is the number of words in the document.

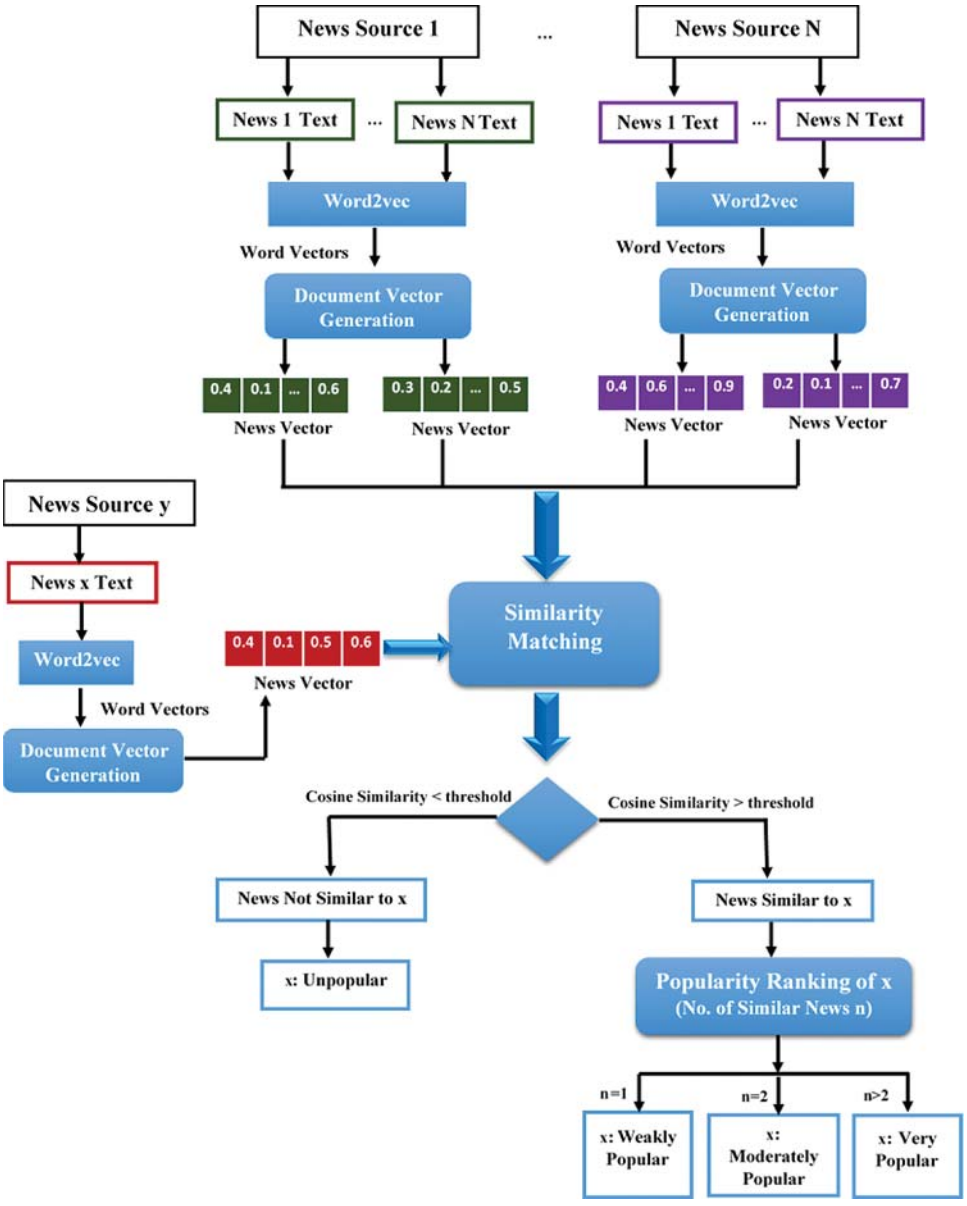

Figure 1: System architecture for news popularity estimation

The process of generating document vectors for news articles is illustrated in Fig. 2.

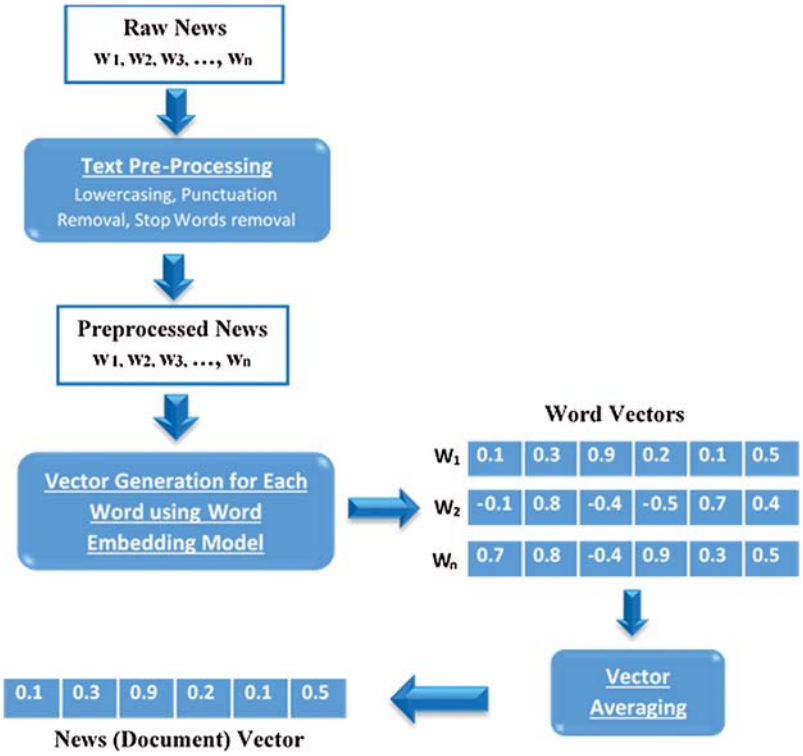

Figure 2: Document vector generation for news articles

4.4 Popularity Estimation

To identify the popularity of a news published by a site, the document vector of the news is matched with the vectors of every other news published within a week of its publication using cosine similarity. The value of cosine similarity between two input vectors ranges from 0 to 1. If the cosine is 0, it means that the vectors do not match with each other. The closer the value is to 1, the greater is the match between the vectors. The cosine of 1 means that vectors are identical to each other. Hence, to label the news items similar or not, we set a threshold of 0.70. If the similarity score between two news vectors is greater than the threshold of 0.70, the news articles are considered similar to each other. Let $S = s_1, s_2, \ldots, s_n$ be a set of all news sources in the database. Let $N = n_1, n_2, \ldots, n_t$ be a set of all news articles for a source $s \in S$. Let x be a news article from another source y whose popularity is to be estimated against the news articles from all the sources in the database. The similarity of x with a news article n denoted as $sim(x, n)$ is given by cosine similarity between their document embeddings D.

$$sim(x, n) = Cosine\ \ Similarity(D(x), D(n)) \tag{4}$$

where $D(x)$ represents document embedding of the news x while $D(n)$ represents document embeddings of the news articles n.

For a source $s \in S$, let *Scores* be a set of similarity scores of news articles belonging to s with the news x belonging to source y

$$Scores = \{sim(x, n_1), sim(x, n_2), \ldots, sim(x, n_t)\} \tag{5}$$

The news articles similar to the news x can be represented as the set of news having similarity scores with x greater than the threshold of 0.70. From each source s, only the most similar article is selected.

$$sn(x) = \{n \mid sim(x, n) \in R \wedge sim(x, n) > 0.70 \wedge sim(x, n) = max(Scores)\} \tag{6}$$

where $sn(x)$ denotes the singleton set of news from a source s similar to x.

The similarity of the given news x is calculated with the news articles from all the sources.

$$SN(x) = \{sn(x)_1, sn(x)_2, \ldots, sn(x)n\} \tag{7}$$

where $SN(x)$ is the set of news articles from all the sources similar to x.

After finding the news similar to the given news, popularity of the given news is calculated based on the occurrences of similar news on all the sources in the database.

A news x from a source y is considered unpopular if it is not found on any other source, which means the set $SN(x)$ is empty.

$$Unpopular\ \ News(UP) = \{x \mid n(SN(x)) = 0\} \tag{8}$$

where $n(SN(x))$ denotes the number of elements in the set $SN(x)$.

If the given news x from source y is published on one other website/source, we consider it as WP. This means the set $SN(x)$ contains only one element.

$$Weakly\ \ Popular(WP) = \{x \mid n(SN(x)) = 1\} \tag{9}$$

If the given news x from source y is published on two other websites/sources, we consider it as MP. This means the set $SN(x)$ contains two elements.

$$Moderately\ \ Popular(MP) = \{x \mid n(SN(x)) = 2\} \tag{10}$$

If the given news x from source y is published on more than two other websites/sources, we consider it as VP. This means the set $SN(x)$ contains more than two elements.

$$Very\ \ Popular(VP) = \{x \mid n(SN(x)) > 2\} \tag{11}$$

5 Experiments and Results

In this section, we perform different experiments on a small dataset of 1000 samples and discuss the results. We evaluate our model for each popularity level individually using Precision (P), Recall (R), and F-score (F). The scores in the tables and figures are given in %ages.

5.1 Comparison of the Proposed W2V-PPS Model with the Baseline Models

In this section, we compare the proposed framework with the following baseline models:

- **BoW based popularity prediction system (BoW-PPS):** In this approach BoW [21] is used to convert the news headlines into vector representations. Cosine similarity is computed between BoW vectors of new items. Popularity is estimated based on the number of most similar news items.

- **TFIDF based popularity prediction system (TFIDF-PPS):** In this approach, cosine similarity is computed between TFIDF vectors of new headlines. Popularity is estimated based on the number of most similar news items.

Experiments are performed with the cosine similarity threshold of 0.70. Tab. 2 shows the results of the proposed model and the baseline models. The F-scores are visually represented in Fig. 3.

Table 2: Results yielded by the proposed and the baseline models

Models	Popularity levels											
	Unpopular			Weakly popular			Moderately popular			Very popular		
	P	R	F	P	R	F	P	R	F	P	R	F
BoW-PPS	83	100	91	55	11	18	100	5	10	0	0	0
TFIDF-PPS	83	100	91	48	11	18	0	5	10	0	0	0
W2V-PPS	97	99	**98**	79	74	**76**	80	63	**71**	62	62	**62**

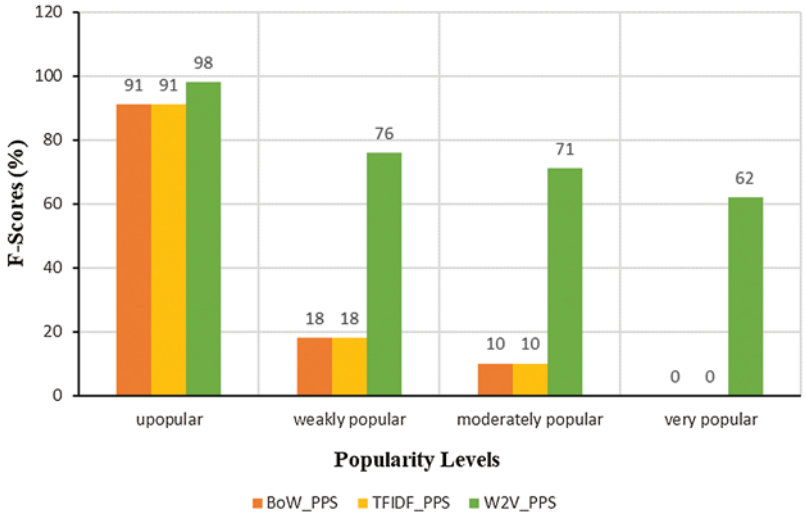

Figure 3: Comparison of F-scores yielded by the proposed and the baseline models

It is evident from the Tab. 2 that the proposed model performs better than the baseline models producing the F-scores of 98%, 76%, 71% and 72% for unpopular, weakly popular, moderately popular, and very popular level, respectively. BoW and TFIDF based systems yield very low results. The reason for the low performance of these approaches is that they are based on the number of occurrences of the particular words in the document while ignoring word order and text semantics. In natural language, there are different ways of writing a sentence which is not bound by the use of particular identical words. Semantics of the news content can effectively be represented by creating content/document vectors based on embedding models. Hence, W2V-PPS based on its ability to estimate popularity by exploiting semantic representations of the news articles performs remarkably better than both the baseline models.

5.2 Effect of Embeddings

In this section, we explore the effect of other embedding models. Besides training wordvec on cybersecurity news dataset in the proposed framework, we train fasttext [22] word embedding model on the same dataset and compare the proposed model with fasttext based popularity prediction system. We further investigate the effect of state-of-the-art pretrained word2vec and fasttext embeddings. We use the pre-trained word2vec model by Google which has been trained on a Google news dataset comprising about 100 billion words. The model generates 300-dimensional embeddings for 3 million words and phrases. A data driven approach explained in [20] has been used to generate embeddings for phrases. We further experiment with pretrained fasttext. The pretrained fasttext model [23] by Facebook contains 2 million word vectors and has been trained on common crawl data comprising about 630 billion tokens. We make these comparisons with the pretrained word embedding models to test if the word2vec model trained on a smaller but domain specific data of cybersecurity news performs better than the state of the art models pretrained on generic and relatively larger datasets providing coverage for various domains. Tab. 3 shows the results achieved by using different embeddings. F-scores achieved by the system with different embeddings are also shown in Fig. 4.

Table 3: Results yielded by the popularity prediction system with different embedding models

Embedding models	Popularity levels											
	Unpopular			Weakly popular			Moderately popular			Very popular		
	P	R	F	P	R	F	P	R	F	P	R	F
Cybersecurity news fasttext	99	14	25	10	8	9	2	4	3	3	90	5
Pretrained google news word2vec	92	97	95	58	53	55	67	53	59	40	10	15
Pretrained common crawl fasttext	97	84	90	40	62	49	40	53	45	24	62	35
Cybersecurity news word2vec	97	99	**98**	79	74	**76**	80	63	**71**	62	62	**62**

It is evident from the results shown in Tab. 3 that cybersecurity news word2vec used in the proposed W2V-PPS model performs better than all the other embedding models. The reason for better performance of cybersecurity news word2vec than cybersecurity news fasttext is that the fasttext model represents word embeddings by the sum of their char n-grams which is useful for capturing morphological nuances of a sentence. However, most of the words which are semantically similar to each other are standalone words, and hence are not related to their morphemes. Therefore, at semantic tasks, including char n-grams or syntactic information into word representations could make the embeddings worse for small training corpora. A similar observation has also been made in [22] where the inclusion of morphological information degraded the performance on semantic tasks.

Cybersecurity news word2vec based system also performs better than both the state-of-the-art pretrained models of word2vec and fasttext. Both the pretrained models have been trained on very large datasets making them very useful for generic tasks. Since cybersecurity news Wor2vec has been trained specifically on the news related to cybersecurity, therefore it yields better results on the undertaken task. This indicates the importance of using models trained on the domain specific data for a particular domain. This also justifies the significance of training our own model for the given problem rather than using pretrained embeddings.

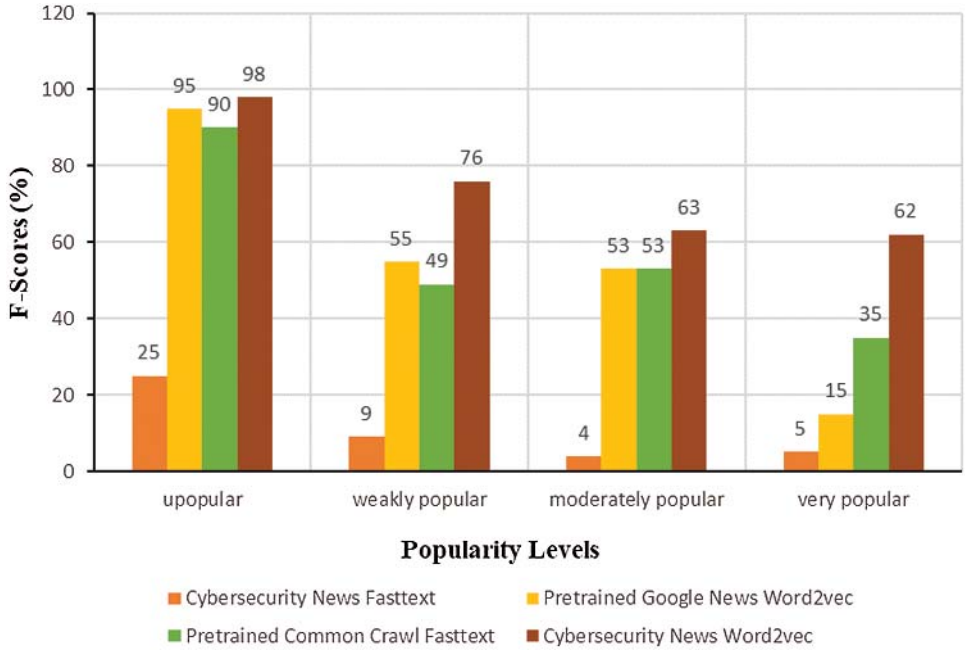

Figure 4: Comparison of F-scores yielded by the popularity prediction system with different embedding models

5.3 Effect of Varying Similarity Threshold

We perform experiments with the proposed W2V-PPS model at different cosine similarity thresholds to identify the threshold that yields the best results. We varied the threshold from 0.63–0.84 in increments of 0.07, producing 4 values in total, i.e., 0.63, 0.70, 0.77, 0.84. We chose 0.63 and 0.83 as the starting and ending thresholds respectively as we observed that the model produced lower results below the threshold of 0.63 than those observed at 0.63. Similarly, the model produced lower results above the threshold of 0.84 than those observed at 0.84. We chose the step size of 0.07 as smaller step sizes did not show any remarkable difference in performance. Tab. 4 and Fig. 5 show the variations in F-scores yielded by the proposed model for all the popularity levels at varying thresholds. It can be observed that the model achieves the best results at the threshold of 0.70. The performance of the proposed model increases at the first increment and then diminishes with further increments of the threshold.

Table 4: Effect of different thresholds on F-scores yielded by proposed W2V_PPS

Similarity	Popularity levels			
Thresholds	Unpopular	Weakly popular	Moderately popular	Very popular
0.63	81	29	23	29
0.70	**98**	**76**	**71**	**62**
0.77	95	52	36	24
0.84	92	32	20	0

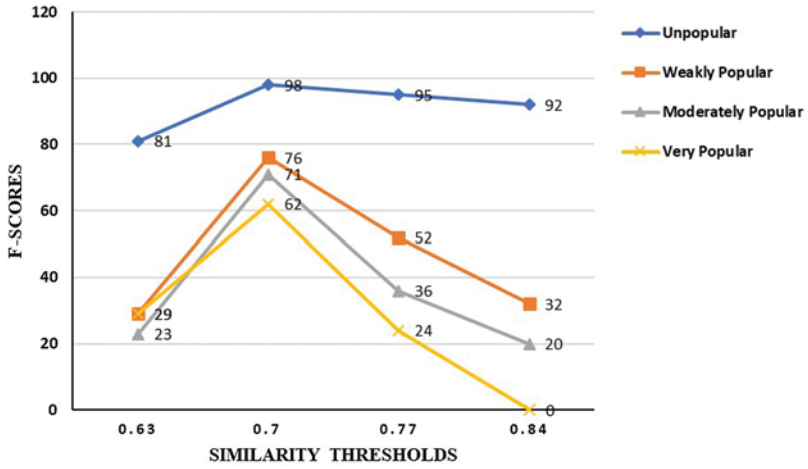

Figure 5: Results of the popularity levels yielded by W2V-PPS at different thresholds

5.4 Evaluation of Matching News Vectors with Each Other

The results produced by the popularity prediction system for different popularity levels depend upon the count of the similar news samples in each class not on which news items are predicted to be similar to the given news sample. In this experiment, we measure the accuracy of matching news vectors with each other. The goal of this experiment is to evaluate the semantic representations generated by the embedding models to check if the vectors of two news items accurately match with each other at a given threshold. To evaluate the accuracy of the models based on matching the similar news items with each other, we selected a set of 100 news samples having at most one similar match from among the news samples. We computed the accuracy on the selected set of samples for all the four embedding models at all the four thresholds. Tab. 5 and Fig. 6 show the accuracies of matching news vectors with each other by different embedding models. It is clear from the results that Cybersecurity News Word2vec outperforms all the other models at the threshold of 0.70.

5.5 Error Analysis

We analyzed the results produced by the proposed W2V_PPS. We noticed that the news items which were not correctly matched with each other contained some identical words which increased the similarity score between the vector representations of the news items even though the complete news headlines were different from each other in overall meaning. In Tab. 6 we give an example from the dataset, the popularity of which was incorrectly computed by the system at the similarity threshold of 0.70. In Tab. 6, 'News item x' is the given news item whose popularity is to be computed by the system. 'News item 1' is the news which is actually similar to the 'News item x,' while 'News item 2' is the news which is incorrectly identified as similar to 'News item x' by the system. We compute the similarity of 'News item x' with 'News item 1' and 'News item 2.' Closer the similarity score to 1, higher is the similarity between the input vectors. Cosine similarity of the 'News item 2' with the 'News item x' yielded by the model is 0.78 which is more than the cosine similarity of 'News item 1' with 'News item x' which is 0.61. Besides cosine similarity, we also experimented with another similarity metric, i.e., euclidean distance. However, it is evident from Tab. 6 that the scores obtained by using euclidean distance are far worse than the scores obtained by the cosine similarity metric.

Table 5: Accuracies of matching news vectors with each other by different models

Similarity thresholds	Models	Accuracy (%)
0.63	Bag of words	15
	TFIDF	15
	Cybersecurity news fasttext	0
	Cybersecurity news word2vec	74
	Pretrained google news word2vec	55
	Pretrained common crawl fasttext	41
0.70	Bag of words	13
	TFIDF	13
	Cybersecurity news fasttext	22
	Cybersecurity news word2vec	**77**
	Pretrained google news word2vec	52
	Pretrained common crawl fasttext	70
0.77	Bag of words	5
	TFIDF	5
	Cybersecurity news fasttext	62
	Cybersecurity news word2vec	46
	Pretrained google news word2vec	23
	Pretrained common crawl fasttext	54
0.84	Bag of words	1
	TFIDF	1
	Cybersecurity news fasttext	63
	Cybersecurity news word2vec	25
	Pretrained google news word2vec	18
	Pretrained common crawl fasttext	29

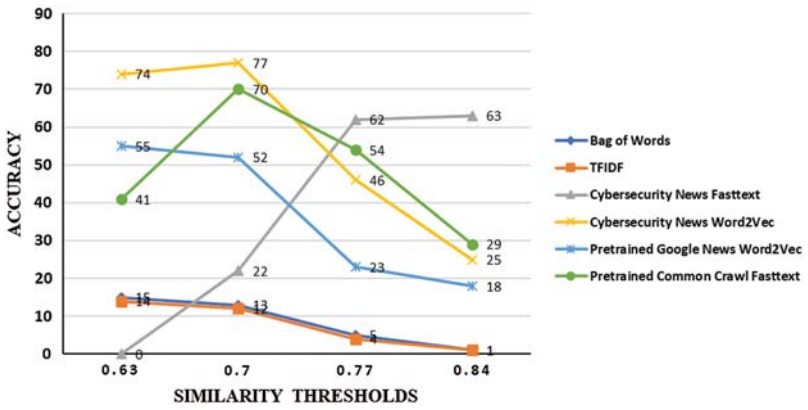

Figure 6: Accuracies of matching news vectors with each other by different models

Table 6: Similarities between the document vectors of the news items

Similarity metrics	News item x	News item 1	News item 2
		Data of virtually all Ecuadorians leaked online	Google estimates 1.5% of web logins exposed in data breaches
Cosine similarities	Data breach exposes	0.61	0.78
Euclidean distance	millions of Ecuadorians	0.004	0.27

'News item 2' is incorrectly computed to be more similar to the 'News item *x*' than the 'News item 1' because some of the words such as *data*, *breach*, and *expose* in both the 'News item *x*' and 'News item 2' are identical. The presence of identical words increases the similarity score between the item vectors while it is clear that both the news items are not semantically similar to each other in actual. There is only one identical word in 'News item *x*' and 'News item 1' which is *data*.

Tab. 7 shows the similarity scores of all the words of 'News item *x*' with all the words of 'News item 1' and 'News item 2' obtained by word2vec model. There are three words in 'News item *x*' and 'News item 1' which are semantically closer to each other in natural language. The word *data* is found in both the news items, hence the cosine value of the word *data* with itself is 1. The word *Ecuadorians* is used in both the 'News item *x*' and 'News item 1,' however, the experiments indicate that the word is not found in the training vocabulary of word2vec. For out of vocabulary words, we set the cosine similarity value to 0. The word *expose* in 'News item *x*' is closer in meaning to the word *leak* in 'News item 1' in natural language. However, the cosine similarity value yielded between these words is 0.35 which indicates very low similarity. This means that semantic representations of these two words have not been learned correctly by the wor2vec model. The other words in 'News item *x*' and 'News item 1' are not closer in meaning to each other in natural language, and hence the cosine values yielded by the model among these words are also low. Hence, as semantic representations of the aforementioned words have not been correctly learned by the word2vec model, the cosine similarity score obtained between the document vectors of 'News item *x*' and 'News item 1' is also low which is 0.61.

Table 7: Cosine similarities yielded by word2vec model between words of news items

Words in news item x	Words in News item 1					Words in News item 2						
	Data	Virtually	Ecuadoreans	Leak	Online	Google	Estimates	Web	Logins	Expose	Data	Breach
Security	0.23	0.03	0	0.015	0.187	0.27	0.12	0.23	0.12	0.15	0.23	0.19
Firm	0.24	0.05	0	0.20	0.18	0.10	0.23	0.15	0.16	0.19	0.24	0.28
Data	1.0	0.12	0	0.25	0.30	0.17	0.10	0.26	0.29	0.19	1.0	0.42
Breach	0.42	0.04	0	0.31	0.29	0.42	0.13	0.10	0.17	0.06	0.42	1.0
Expose	0.19	0.13	0	0.35	0.12	0.08	0.09	0.15	0.18	1.0	0.19	0.06
Million	0.24	0.07	0	0.24	0.29	0.16	0.41	0.18	0.21	0.16	0.24	0.34
Ecuadorians	0	0	0	0	0	0	0	0	0	0	0	0

Since there are three identical words in 'News item *x*' and 'News item 2' i.e., *data*, *breach* and *expose*, the cosine values between these identical words are 1. The presence of high similarity

scores between three of the words plays an influential role in increasing the cosine similarity value between the document vectors of 'News item x' and 'News item 2'.

Hence, in most of the erroneous cases, the presence of identical terms increases the similarity score between the news items even though the contextual meaning of the news items is different from each other.

6 Conclusion and Future Work

This work proposed an automated semantically enriched system to predict the popularity of cybersecurity news. We used word2vec model to represent the text in vector form and used cosine similarity to measure similarity between the vector representations of news articles. We trained word2vec on cybersecurity news dataset collected from different websites. We compared the proposed system with the baseline BoW and TFIDF based systems. We also explored the effect of different embedding models and variations in the threshold. The results showed that the system based on word2vec trained on cybersecurity news data outperformed all the other systems at the similarity threshold of 0.70. The better results of the system based on word2vec reveal the importance of training the embedding model on domain specific data for a particular task. In future work, we aim to improve popularity prediction based on data retrieved from social media. The deep learning based system can be more useful for better accuracy [24].

Acknowledgement: The authors wish to express their appreciation to Higher Education Commission (HEC), Pakistan for supporting the research.

Funding Statement: This research was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0012724, The Competency Development Program for Industry Specialist) and the Soonchunhyang University Research Fund.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Tatar, A., de Amorim, M. D., Fdida, S., Antoniadis, P. (2014). A survey on predicting the popularity of web content. *Journal of Internet Services and Applications, 5(1),* 8. DOI 10.1186/s13174-014-0008-y.
2. Moniz, N., Torgo, L., Eirinaki, M., Branco, P. (2017). A framework for recommendation of highly popular news lacking social feedback. *New Generation Computing, 35(4),* 417–450. DOI 10.1007/s00354-017-0019-x.
3. Guan, X., Peng, Q., Li, Y., Zhu, Z. (2017). Hierarchical neural network for online news popularity prediction. *Chinese Automation Congress*, pp. 3005–3009, IEEE, New York, USA.
4. Liu, C., Wang, W., Zhang, Y., Dong, Y., He, F. et al. (2017). Predicting the popularity of online news based on multivariate analysis. *IEEE International Conference on Computer and Information Technology*, pp. 9–15, IEEE, New York, USA.
5. Khan, A., Worah, G., Kothari, M., Jadhav, Y. H., Nimkar, A. V. (2018). News popularity prediction with ensemble methods of classification. *9th International Conference on Computing, Communication and Networking Technologies*, pp. 1–6, IEEE, New York, USA.
6. Abbar, S., Castillo, C., Sanfilippo, A. (2018). To post or not to post: Using online trends to predict popularity of offline content. *Proceedings of the 29th on Hypertext and Social Media*, pp. 215–219, ACM, New York, USA.

7. Natarajan, S., Moh, M. (2016). Recommending news based on hybrid user profile, popularity, trends, and location. *International Conference on Collaboration Technologies and Systems*, pp. 204–211, IEEE, New York, USA.
8. Rizos, G., Papadopoulos, S., Kompatsiaris, Y. (2016). Predicting news popularity by mining online discussions. *Proceedings of the 25th International Conference Companion on World Wide Web*, Geneva, Switzerland, International World Wide Web Conferences Steering Committee.
9. Keneshloo, Y., Wang, S., Han, E. H., Ramakrishnan, N. (2016). Predicting the popularity of news articles. *Proceedings of the 2016 SIAM International Conference on Data Mining*, Philadelphia, SIAM.
10. Fernandes, K., Vinagre, P., Cortez, P. (2015). A proactive intelligent decision support system for predicting the popularity of online news. *Portuguese Conference on Artificial Intelligence*, Berlin, Germany, Springer.
11. Zhou, J., Wu, G., Tu, M., Wang, B., Zhang, Y. et al. (2017). Predicting the popularity of messages based on big data. *IEEE 2nd International Conference on Big Data Analysis*, pp. 164–168, IEEE, New York, USA.
12. Choudhary, S., Sandhu, A. S., Pradhan, T. (2017). Genetic algorithm based correlation enhanced prediction of online news popularity. *Computational Intelligence in Data Mining*, pp. 133–144, Berlin, Germany: Springer.
13. Khan, M. A., Akram, T., Sharif, M., Javed, M. Y., Muhammad, N. et al. (2019). An implementation of optimized framework for action classification using multilayers neural network on selected fused features. *Pattern Analysis and Applications, 22(4),* 1377–1397. DOI 10.1007/s10044-018-0688-1.
14. Obiedat, R. (2020). Predicting the popularity of online news using classification methods with feature filtering techniques. *Journal of Theoretical and Applied Information Technology, 98(8),* 1163–1172.
15. Xiao, C., Liu, C., Ma, Y., Li, Z., Luo, X. (2020). Time sensitivity-based popularity prediction for online promotion on twitter. *Information Sciences, 525,* 82–92. DOI 10.1016/j.ins.2020.03.056.
16. Yang, Y., Liu, Y., Lu, X., Xu, J., Wang, F. (2020). A named entity topic model for news popularity prediction. *Knowledge-Based Systems, 208(3),* 106430. DOI 10.1016/j.knosys.2020.106430.
17. Hensinger, E., Flaounas, I., Cristianini, N. (2013). Modelling and predicting news popularity. *Pattern Analysis and Applications, 16(4),* 623–635. DOI 10.1007/s10044-012-0314-6.
18. Long, F., Xu, M., Li, Y., Wu, Z., Ling, Q. (2018). XiaoA: A robot editor for popularity prediction of online news based on ensemble learning. *International Conference on Intelligence Science*, Berlin, Germany, Springer.
19. Aswani, R., Chandra, S., Ghrera, S., Kar, A. K. (2017). Identifying popular online news: An approach using chaotic cuckoo search algorithm. *2nd International Conference on Computational Systems and Information Technology for Sustainable Solution*, pp. 1–6, IEEE, New York, USA.
20. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems, 26,* 1–9.
21. Siddiqui, S., Khan, M. A., Bashir, K., Sharif, M., Azam, F. et al. (2018). Human action recognition: A construction of codebook by discriminative features selection approach. *International Journal of Applied Pattern Recognition, 5(3),* 206–228. DOI 10.1504/IJAPR.2018.094815.
22. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics, 5(1),* 135–146. DOI 10.1162/tacl_a_00051.
23. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A. (2017). Advances in pre-training distributed word representations. arXiv preprint arXiv: 1712.09405.
24. Hussain, N., Khan, M. A., Sharif, M., Khan, S. A., Albesher, A. A. et al. (2020). A deep neural network and classical features based scheme for objects recognition: An application for machine inspection. *Multimedia Tools and Applications, 155,* 220. DOI 10.1007/s11042-020-08852-3.