News2vec: News Network Embedding with Subnode Information

Ye Ma[1], **Lu Zong**[1*], **Yikang Yang**[2] **and Jionglong Su**[1]
[1]Mathematical Sciences, Xi'an Jiaotong-Liverpool University, Suzhou, 215028, China
[2]Mathematics and Statistics, Georgetwon University, Washington DC, 20007, US
{ye.ma, lu.zong, jionglong.su}@xjtlu.edu.cn
yy494@georgetown.edu

Abstract

With the development of NLP technologies, news can be automatically categorized and labeled according to a variety of characteristics, at the same time be represented as low dimensional embeddings. However, it lacks a systematic approach that effectively integrates the inherited features and inter-textual knowledge of news to represent the collective information with a dense vector. With the aim of filling this gap, the News2vec model is proposed to allow the distributed representation of news taking into account its associated features. To describe the cross-document linkages between news, a network consisting of news and its attributes is constructed. Moreover, the News2vec model treats the news node as a bag of features by developing the Subnode model. Based on the biased random walk and the skip-gram model, each news feature is mapped to a vector, and the news is thus represented as the sum of its features. This approach offers an easy solution to create embeddings for unseen news nodes based on its attributes. To evaluate our model, dimension reduction plots and correlation heat-maps are created to visualize the news vectors, together with the application of two downstream tasks, the stock movement prediction and news recommendation. By comparing with other established text/sentence embedding models, we show that News2vec achieves state-of-the-art performance on these news-related tasks.

1 Introduction

News, carrying a large amount of information, can often guide public opinions, affect people's behavior and drive the evolution of social events. In the era of information, news text is considered to be a part of big data that is continuously updated. In order to facilitate the performance of downstream NLP tasks, it is rather essential to find an efficient way to represent news as continuous vectors that contain the collective information of its inherited features and the inter-textual knowledge between different news.

The most straightforward approach to embed news is to directly treat it as textual data and apply text/sentence embedding models. To represent texts/sentences as vectors, one of the classic techniques is to perform numerical operations on the word vectors (Mikolov et al., 2013b), which is recognized as a simple but powerful baseline (Conneau et al., 2018). SDAE (Vincent et al., 2010) uses an auto-encoder to compress texts into a low-dimensional vector. Paragraph Vector (Le and Mikolov, 2014) learns the distributed representation of sentences and documents by predicting their contexts, *i.e.*, words in the sentence/document. Additionally, Skip-Thought (Li and Hovy, 2014), FastSent (Hill et al., 2016) and Quick-Though (Logeswaran and Lee, 2018) learn distributed sentence-level representations through the coherence between sentences. BERT (Devlin et al., 2018) proposes a powerful pre-trained sentence encoder which is trained on unsupervised datasets based on the masked language model and next sentence prediction. The major limitation of embedding approaches discussed above is that they produce representations and features that solely describe the contextual information at the document level. In the case of news representation, the connection between different news events is also considered as crucial information that should be incorporated into the news embedding, as well as other labeled features such as the topic category and the polarity of the news.

With the argument that news stories describe events, news event embedding models are developed to offer an alternative solution for the vector representation of news. It is suggested to extract event tuples $E = (Actor, Predicate, Object)$ from headlines and learn vector representations

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 4843–4852,
Hong Kong, China, November 3–7, 2019.

by scoring the correct tuples higher than the corrupted ones (Ding et al., 2015, 2016). Event2vec (Setty and Hose, 2018) is constructed on a classified news event database from Wikipedia[1], where an event network is created by considering events, entities, event types and years as nodes of different types. The Event2vec model thus produces distributed vector representations of news events based on the network embedding mechanisms, which lacks the flexibility of generating vectors for nodes outside the trained network. In addition, the weights of some edges in the event network are determined by objective assumptions since types and years are not comparable.

In this study, the News2vec model is proposed to learn distributed representations of news, with the embedded vectors containing not only the semantic and labeled information, but also the latent connections between different news events. We start by building the news network that connects the news nodes with their corresponding event element nodes in order to create the aggregated news context that describes both the semantic information and the background linkages between different news events (See Figure 1 for a simplified example of the news network). According to Figure 1, the two pieces of news $Alibaba\ reinvested\ Suning\ after\ three\ years$ and $Suning's\ second-half\ profit\ rose\ by\ 5\%$ are connected by sharing the same contextual element $Suning$. The verbal keywords $reinvest$ and $rise$ of the two different news are thus connected through their common node $Suning$ and become the latent contextual element of the other news. Based on the constructed news network, the biased random walk approach (Grover and Leskovec, 2016) is adopted to generate adequate number of connected node sequences which are later used in the network embedding. Inspired by the Subword model (Bojanowski et al., 2016), we further represent each news node as a bag of news features including semantic features, categorical features, time features, *etc.*. The vector representation of the news feature is thus obtained based on the skip-gram architecture (Mikolov et al., 2013a), and the news vectors are computed as the sum of its feature embeddings.

The advantages of the proposed News2vec model are twofold. First, the News2vec embeddings capture both the semantic features and the potential connections between news by constructing the news network. Second, the Subnode model offers an easy solution to create embeddings for unseen (news) nodes outside the trained network, at the same time enriches the node vector with its node attributes (news features). As the experiments suggest, the News2vec model achieves state-of-the-art performance on the news-related downstream tasks, namely the stock movement prediction and news recommendation.

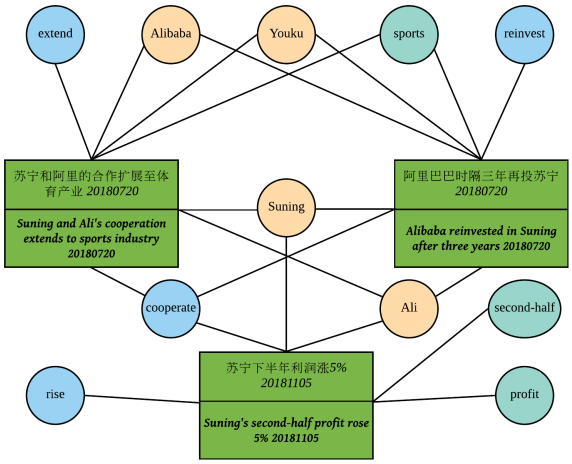

Figure 1: Simplified sample of the news network

2 The News2vec Model

In this section, the formation of the News2vec model is discussed in terms of the news network construction and news embedding based on the Subnode model. According to Figure 2, news elements including entities, actions and nouns, are first extracted from the news titles and texts. Based on these elements and their term frequencyinverse document frequency (tf-idf), the news network is built and sequences of nodes are sampled by the biased random walk. News nodes in these sequences are then represented as bags of extracted news features. The associated vector is thus assigned to each feature by the Subnode model.

2.1 News Network

The News2vec model creates news embeddings based on the news network that connects news with their elements. With the argument that a piece of news often describes one or several events which could be represented as a collection of elements, this study extracts news elements as enti-

[1] https://en.wikipedia.org/wiki/Portal:Currentevents

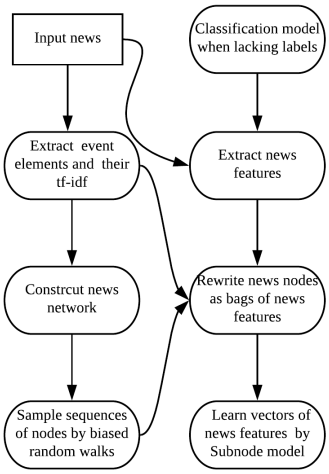

Figure 2: Flow chart of creating news embeddings with News2vec

ties, actions and nouns from news titles and texts, with respect to their tf-idf scores. Specifically, the tf-idf is a well recognized tool to measure the importance of a word/phrase inside the document and to extract the representative elements at the document level. The tf-idf score is formulated as:

$$tfidf_{k,e} = \frac{n_{k,e}}{\sum_w n_{w,e}} \times \log \frac{|D|}{|e : k \in d_e|}, \quad (1)$$

where $n_{k,e}$ denotes the number of the news element k in the news story e, whereas its denominator is the total number of words/phrases in the news document. $|D|$ is the total number of documents, the denominator represents the number of stories containing the element k. In a nutshell, the tf-idf assigns a higher score to the element which appears more frequently in the news story than in others.

After determining the elements of news, the news network is hence constructed by connecting the news nodes $e \in V_e$ of the news title and published time with its element nodes $k \in V_k$. The weights of connective edges (e, k) are computed based on the importance of the element to the news, which is determined by their tf-idf scores. The weight of an edge (e, k) is computed as:

$$W_{e,k} = \frac{tfidf_{k,e}(title)}{Z_1} + \frac{tfidf_{k,e}(content)}{Z_2}, \quad (2)$$

where Z is a normalization constant.

2.2 Network Embedding with Subnode Information

Network embedding

The New2vec network embedding is based on the Node2vec (Grover and Leskovec, 2016) model, with the objective to learn a latent feature vector $F(v)$ for each node $v \in V$ that maximizes the probability of predicting the node $v's$ network neighborhood $N(v)$. The objective function is written as:

$$\max \sum_{v \in V} \log Pr(N(v)|F(v)). \quad (3)$$

Given the feature vector of node v, it is assumed that the prediction probabilities of its neighborhoods are independent from each other, which leads to:

$$Pr(N(v)|F(v)) = \prod_{u \in N(v)} Pr(u|F(v)). \quad (4)$$

To solve the objective function and obtain the optimized node representations, the skip-gram architecture (Mikolov et al., 2013a) is adopted on the basis of network neighboring sequences sampled by the biased random walk. To reduce the computational cost, negative sampling (Mikolov et al., 2013b) is used to replace the softmax classifier by multiple logistic binary classifiers. The optimizer is the Stochastic Gradient Descent (SGD).

Different from the uniform random walk of DeepWalk (Perozzi et al., 2014), News2vec uses the Breadth-first Search (BFS) and the Depth-first Search (DFS) as its search strategies. Suppose that the walk just transitioned from t to v and is now walking to its next step node x, BFS determines the next step with a greater chance of revisiting t, *i.e.*, $d_{tx} = 0$, whilst DFS drives the walk to go deeper and further away from the node t, *i.e.*, $d_{tx} = 2$. In line with the Node2vec model, News2vec defines a search bias α to control the search preference.

$$\alpha_{vx} = \begin{cases} \frac{1}{p}, & if\ d_{tx} = 0 \\ \frac{1}{q}, & if\ d_{tx} = 2 \end{cases} \quad (5)$$

Next, the transition probability of edge (v, x) is defined as the product of the weight of the edge and the bias, *i.e.*, $\frac{\alpha_{vx} \times w_{vx}}{Z}$ where Z is a normalization constant. Explicitly, if $p < q$, the walk is width-first which indicates that there is a greater chance to revisit the original node and sample the

nodes around it, that is, semantically similar news. If $p > q$, the walk is depth-first and tends to go to nodes that are not passed before, which leads to the exploration of news with latent relations. During the random walk, each node is used as a starting vertex for a fixed-length walk to produce a group of sequences. Based on the skip-gram model, we can train vector representations of nodes on these sequences.

Subnode model

Classic skip-gram model assigns a distinct vector to each word, neglecting the morphology information of words. With the purpose of addressing this problem, the Subword (Bojanowski et al., 2016) model represents each word using a bag of character $n-$grams, and the word vector is thus the summation of its character $n-$grams. The major advantage of the Subword model is that it allows the computation of word vectors that are not in the trained corpus by using their character $n-$gram vectors. Inspired by the Subword model, we introduce the Subnode model to solve network embedding problems by offering a solution to create inferential vectors of unseen news nodes outside the trained network, at the same time incorporate node attributes to the node vectors.

The biased random walk produces sequences of news and event element nodes from the news network. The proposed Subnode model expresses each of the news node as a bag of subnode information of its associated features including semantic features such as event elements with high tf-idf scores, text structure features such as words count and paragraphs count, and side information such as published date, news type and emotion. As a result, each news vertex $e \in V_e$ is represented as:

$$G =< entity_A, entity_B, action, \cdots,$$

$$words\ count, paragraph\ count,$$

$$month, day, week, type, sentiment, \cdots >$$

where word count, paragraph count and sentiment are ordinal data rather than real values. Further, the Subnode model represents a news node as the sum of its features, instead of a distinct vector obtained from the other network embedding models. Therefore, the objective function of news network embedding is expressed as:

$$max \sum_{e \in V_e} \log Pr(N(e) | \sum_{g \in G_e} f(g)) \qquad (6)$$

where $f(g)$ denotes the vector representation of a news feature. In addition, we remove nodes with only one edge to reduce the sparsity of the network. As it is mentioned in the previous section, the Subnode vector representations of news features are learned based on the skip-gram architecture by optimizing the updated objective function using SGD with negative sampling (Mikolov et al., 2013b). With an adequately large news network that contains a wide range of features, the Subnode mechanism of News2vec allows the vector representation of an unseen news node directly computed by extracting news features and summing up their corresponding vectors according to the $feature\ to\ vector$ dictionary. See Github[2] for the codes and examples of News2vec.

3 Experiments

Four experiments are implemented and explored in this section. In particular, we first visualize the News2vec embeddings in terms of the news vectors and feature vectors with the dimension reduction plot and the correlation heat maps. Two downstream experiments, *i.e.* the stock movement prediction and news recommendation, are then conducted to examine the News2vec model in comparison with other established text/sentence embedding models. Overall, News2vec achieves state-of-the-art performances in these news-related tasks which demonstrates its validity of addressing potential relevance of news via the network, as well as the integrated consideration of the news features.

3.1 Visualization of News Vectors

In this section, the THUCTC[3] (THU Chinese Text Classification) news data set is used to train the vectors of news features, including the semantic features and four additional features, namely $type$, $words\ count$, $paragraph\ count$ and $sentiment$. We determine the sentiment by counting positive words over negative words. There are 14 types of news in the news corpus and we randomly take 6,000 pieces of news from each type as the training set. In the test set, we additionally extract 1,000 pieces of news from each type. Based on these 84,000 pieces of news in the training set, we learn a vector representation of 128 dimensions for each news feature. The news vectors in the test set are

[2] https://github.com/yema2018/News2vec
[3] http://thuctc.thunlp.org

then computed, whereas their dimension reduction plots are produced using the t-Distributed Stochastic Neighbor Embedding (t-SNE).

Figure 3 shows the dimension reduction results of news vectors in the test set. According to the right panel, a high level of clustering is observed, indicating that News2vec allows the embedding of type information into news vectors after incorporating the type feature. Moreover, the distance between two clusters is in-line with the relevancy between the clustered topics in reality. In particular, neighboring relationships are exhibited between similar topics such as sports & lottery ticket, current affairs & society, stock & finance, and fashion & entertainment, whilst the distance between less relevant clusters, such as constellation & finance, game & education, is relatively larger. Nevertheless, the left panel of Figure 3 shows that news embeddings significantly lose classification information once vectors of the type feature are removed from the news vector, which shows that News2vec enriches the news vector representation by the news network and the Subnode information.

3.2 Correlations of News Features

In this section, we further investigate the vector correlation between three news features, namely the $type$, the $words\ count$ and the $sentiment$. The correlation coefficients (*i.e.*, normalized cosine similarity) heat maps for each feature are displayed in Figure 4.

The correlation heat map of news type features (upper panel) supports the conclusion in Section 3.1. It is observed that vector pairs of close topics such as $type : Stock$ and $type : Finance$ are highly correlated, indicating stock news and financial news share similar contextual elements. Other highly correlated pairs include sports and lottery ticket news, game and technology news, fashion and entertainment news, etc., whilst less relevant topics show low correlations, such as real estate and constellation, sports and real estate. According to the bottom left panel of Figure 4, a significant number of dark blocks are observed in the area from $wc200$ to $wc3000$ ($wc200$ means the news is less than 200 words and $wc2000$ means the news is between 1000 and 2000 words), and there is a clear reduction of correlation once the news exceeds 3,000 words. As it is indicated by the word count correlation heat map, news that has similar numbers of words is more likely to have similar content. The correlation reduction from $wc3000$ to $wc5000$ suggests that there is a group of news with specific content, that is often discussed in long articles. Meanwhile, the right bottom of Figure 4 shows that news with close sentiment levels tends to have higher correlations. As the sentiment moves from more positive to more negative, the correlation decreases. Overall, the correlation heat maps show that the News2vec feature vectors are reliable in terms of expressing Subnode information.

3.3 Stock Movement Prediction

Experimental settings

We use financial news (2009.10.19 to 2016.10.31) from Sohu [4] to predict the daily movement of Shanghai securities composite index. The news feature vectors are trained based on news from 2009.10.19 to 2015.12.31 and news vectors are computed by summing up these feature vectors. There are five additional news features, namely $month$, $week$, $sentiment$, $words\ count$ and day.

The length of the walk is fixed at 100, the context size is 10, return hyper-parameter (p in Equation (5)) and input hyper-parameter (q in Equation (5)) are both set to 1.

Predictive model

The predictive model is the long short term memory (LSTM) network (Hochreiter and Schmidhuber, 1997) with self-attention mechanism (Yang et al., 2016). As shown in 5, the LSTM model has T time steps (*i.e.*, T days) in total. At each time step, the input is the weighted average of news vectors in that day. Weights are initialized at the beginning and updated by the gradient descent. Based on the attention model, news that is closely related to the stock price movement is assigned with higher weights.

In this paper, we use the previous 20 days' ($T = 20$) news to predict the next day's stock index movement. The output layer is a Softmax classfier that indicates whether the index goes UP (rise percent $> 0.33\%$), DOWN (rise percent $< -0.29\%$) or PRESERVE ($-0.29\% <$ rise percent $< 0.33\%$). We employ a rolling window with $size = 20$ and $strider = 1$ to augment the sample. News before 2016 is used as the training set, whilst news after 2016 is used as the test-validation set.

[4] https://www.jianshu.com/p/370d3e67a18f

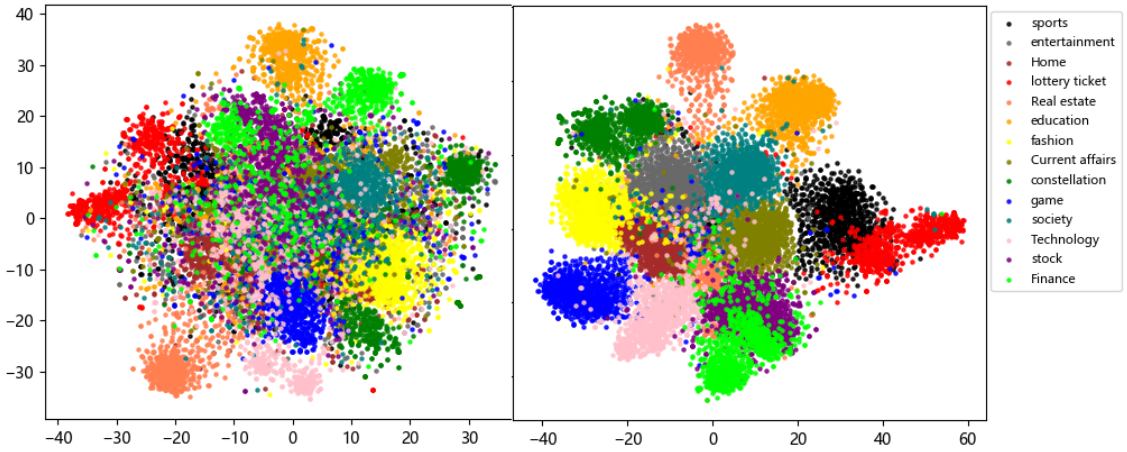

Figure 3: Dimension reduction plots of test set news without and with the type feature

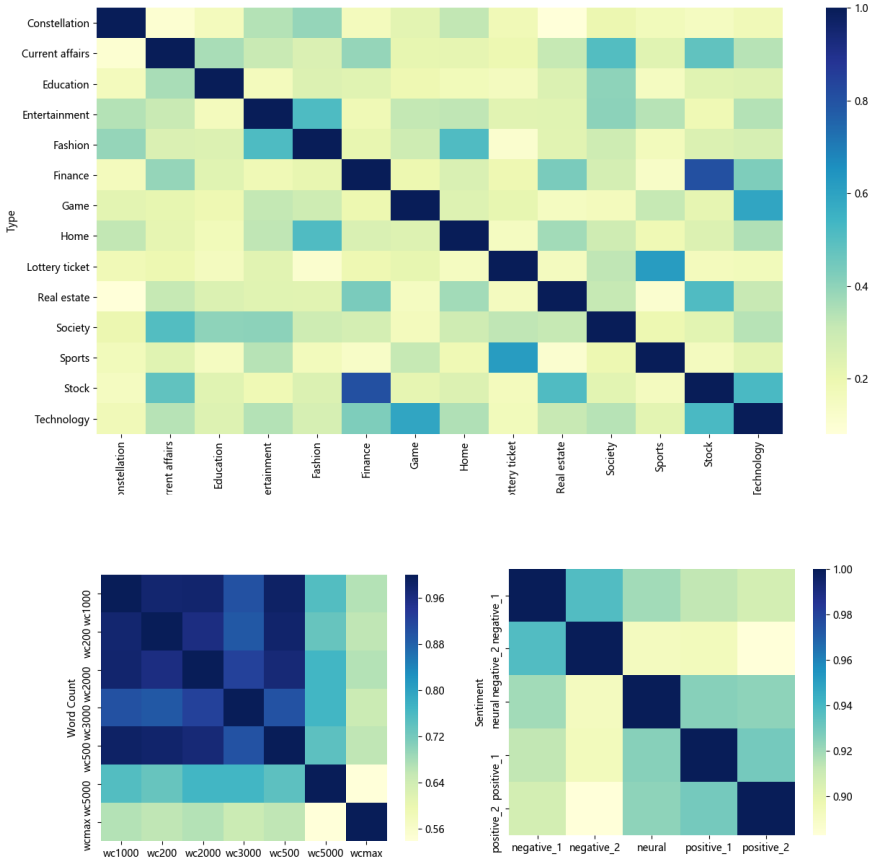

Figure 4: Correlations of news features (Upper panel: news type. Lower left: word count. Lower right: sentiment.

Results

As for the baseline models, we include several established embedding approaches to compute news vectors in the experiment of the stock movement prediction:

Average BOW: News embedding is represented as the average of word vectors in the news titles and bodies (Hu et al., 2018). Word embedding is obtained by training the skip-gram Word2vec model (Mikolov et al., 2013a).

Doc2vec: Paragraph Vector (Le and Mikolov, 2014): News texts are represented as dense vectors by the Paragraph Vector model(Akita et al., 2016).

Event embedding (Ding et al., 2015, 2016): Event tuples are extracted from headlines and represented as vector representations by scoring the correct tuples higher than the corrupted tuples.

In addition, we apply a sentence-level encoder

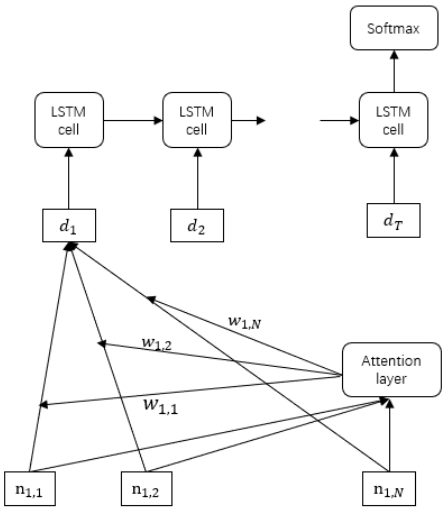

Figure 5: LSTM model with attention mechanism

BERT to encode news headlines as another baseline model:

BERT (Devlin et al., 2018): Average pooling is used to get a fixed representation of a headline based on the Chinese pre-trained model[5].

The proposed News2vec model is implemented separately with and without the five additional features (*i.e.* $month$, $week$, $sentiment$, $words\ count$ and day):

News2vec-: News vectors are the summation of solely the semantic features (*i.e.*, event elements).

News2vec+: News vectors are the summation of all news features including the semantic and the five additional features.

	Acc	MCC
Average BOW($d = 128$)	41.91%	0.1137
Doc2vec($d = 128$)	43.90%	0.1484
Event embedding($d = 128$)	43.76%	0.1439
BERT($d = 764$)	45.83%	0.1442
News2vec-($d = 128$)	44.68%	0.1664
News2vec+($d = 128$)	**46.24%**	**0.1936**

Table 1: Acc and MCC Results of stock movements in the test set

In Table 1, test results are presented according to two measurements, namely the accuracy (Acc) and the Matthews Correlation Coefficient (MCC). We find the News2vec model outperforms BERT and all the other baseline models both in terms of the Acc and MCC, with a lower dimension of 128 (The dimension of BERT embeddings is 764). As the MCC suggests, Doc2vec, Event embedding and BERT produce stock movement predictions of similar qualities as the news vectors generated by the three approaches uniformly limit to the semantic information of news articles. As for news-driven stock price prediction, it is more reasonable to take into account the potential connections between news, such as having similar background story or leading to the same event, and reflect the connection by embed similar values in certain dimensions of the news vectors. In the case of the New2vec+ model, we find that extra features contribute to the improvement of the MCC, indicating the New2vec representations are able to capture the latent connections between news events. After considering all news features, the results are improved by 0.028, showing their positive effect on stock movement predictions. It is worth-mentioning that the New2vec model is performed in an unsupervised manner as all features are extracted from common news text without any artificial label or trained classifier. We believe that there is a large chance to further improve the result once artificial labels such as the polarity are incorporated.

3.4 News Recommendation

Experiment settings

In this section, a news recommendation task is implemented based on the data of news browsing records (2014.2 to 2014.3) using the news embedding models [6]. The obtained data set contains 16214 news browsing sequences in total, which is split into halfs for testing and fine-tuning, respectively. The overall objective is to recommend news based on the content by computing the cosine similarities of the news vectors.

News recommendations are made for each news by selecting the top ten news with the highest vector similarities. The recommendation is considered to be successful as long as the next browse in the sequence belongs to the ten news articles selected. We compute the success rate for each sequence. The average success rate of all sequences is the final evaluation result for the embedding model.

Since the news titles are absent in the data set, event elements are only extracted from news bodies. As a result, we only use the text-level em-

[5] https://github.com/google-research/bert#pre-trained-models

[6] https://github.com/YLonely/web-data-mining

bedding models (*i.e.*, Paragraph Vector and Average BOW) as the baselines to compare with the News2vec model. We use the trained news feature vectors in the previous section as the initial inputs of the new news network (Sohu+rec). Moreover, half of the news sequences are employed to fine-tune these vectors (Sohu+rec+seq). Due to the lack of information, only $sentiment$ and $word\ counts$ are included as the extra news features together with the existing semantic features.

Evaluation of news recommendation

	Success rate
Paragraph Vector	4.31%
Average BOW	3.09%
Sohu (no extra features)	3.36%
Sohu+rec (no extra features)	3.69%
Sohu+rec+seq (no extra features)	6.73%
Sohu	2.95%
Sohu+rec	3.61%
Sohu+rec+seq	**7.53%**

Table 2: Evaluation results of news recommendation

From Table 2, it is observed that the success rate improves significantly after fine-tuning the news vectors with respect to the news browsing sequences. In comparison with Paragraph Vector, News2vec (Sohu+rec) demonstrates no significant advantage without sequence fine-tuning (Sohu+rec+seq), indicating potential connections between news have limited effect on this task. Nonetheless, News2vec achieves state-of-art performance after sequence fine-tuning. On the other hand, the result of Paragraph Vector cannot be further improved by fine-tuning. A possible explanation is that the Paragraph Vector model assigns a distinct vector to each of the news articles, thus inconsistency between news in different news browsing sequences is easily created by the sequence-level fine-tuning. On the other hand, fine-tuned News2vec embeddings carry the information of both the news network and the browsing sequences by treating the browsing sequence as a special sequence generated by the biased random walk. In addition, it is found that with fine-tuning, the additional features tend to have positive effects on the recommendation task (see Sohu to Sohu+rec+seq without extra features and Sohu to Sohu+rec+seq with extra features). As a matter of fact, additional news features, such as the release time and type, are considered to contain important information for news recommendation tasks. Unfortunately, due to the lack of data, only two extra features are included in this experiment.

4 Related Work

In most cases, news embeddings are directly created by sentence/text embedding approaches (Ding et al., 2015; Hu et al., 2018; Vargas et al., 2017; Akita et al., 2016), that are often divided into two groups: the unsupervised and the supervised models. For unsupervised learning, the auto-encoder model is an early solution to compress text data (Vincent et al., 2010). Inspired by word embedding, Paragraph Vector (Le and Mikolov, 2014; Li et al., 2016) represents texts or sentences as vectors based on the co-occurrence relation between words and documents, whilst Skip-Thought (Li and Hovy, 2014), FastSent (Hill et al., 2016) and Quick-Though (Logeswaran and Lee, 2018) are based on the coherence between sentences. For supervised learning-based models, most methods use sentence encoder such as LSTM or Transformer (Vaswani et al., 2017) to encode word vectors into sentence vectors and train the encoder on various NLP tasks (Conneau et al., 2017; Cer et al., 2018).

5 Conclusions

In this paper, we develop the News2vec model that learns the vector representation of news articles by constructing a news network. The Subnode model is further proposed to allow the embedding of unseen (news) nodes outside the existing network, at the same time to enrich the news vector with its associated feature vectors. Compared to other established text embedding models, the News2vec embedding contains not only the information of the contextual relationship between news and its event elements, but also potential connections as the network goes deeper. According to the dimension reduction plots and correlation heat-maps, it is suggested that the news/feature vectors contain adequate information as expected. Two downstream tasks, the news-driven stock movement prediction and news recommendation, show the News2vec embeddings demonstrate the latent connections between news articles, and the integration of news features enhances the model's performance in comparison to the baseline models.

Acknowledgments

The authors would like to acknowledge the support by 2016 Jiangsu Science and Technology Programme: Young Scholar Programme (No. BK20160391).

References

R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara. 2016. Deep learning for stock prediction using numerical and textual information. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, pages 1–6.

P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv e-prints*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder. *CoRR*, abs/1803.11175.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *CoRR*, abs/1705.02364.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *CoRR*, abs/1805.01070.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2327–2333. AAAI Press.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2016. Knowledge-driven event embedding for stock prediction. In *COLING*.

Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 855–864, New York, NY, USA. ACM.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *CoRR*, abs/1602.03483.

Sepp Hochreiter and Jrgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 261–269, New York, NY, USA. ACM.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Bofang Li, Zhe Zhao, Tao Liu, Puwei Wang, and Xiaoyong Du. 2016. Weighted neural bag-of-n-grams model: New baselines for text classification. In *COLING*.

Jiwei Li and Eduard Hovy. 2014. A model of coherence based on distributed sentence representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2039–2048, Doha, Qatar. Association for Computational Linguistics.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *CoRR*, abs/1803.02893.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 701–710, New York, NY, USA. ACM.

Vinay Setty and Katja Hose. 2018. Event2vec: Neural embeddings for news events. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 1013–1016, New York, NY, USA. ACM.

M. R. Vargas, B. S. L. P. de Lima, and A. G. Evsukoff. 2017. Deep learning for stock market prediction from financial news articles. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 60–65.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.