Content-based Dwell Time Prediction Model for News Articles

Heidar Davoudi[1], **Aijun An**[1], **Gordon Edall**[2]
[1]Department of Electrical Engineering and Computer Science, York University, Canada
[2]The Globe and Mail, Canada
{davoudi,aan}@cse.yorku.ca, GEdall@globeandmail.com

Abstract

The article dwell time (i.e., expected time that users spend on an article) is among the most important factors showing the article engagement. It is of great interest to news agencies to predict the dwell time of an article before its release. It allows online newspapers to make informed decisions and publish more engaging articles. In this paper, we propose a novel content-based approach based on a deep neural network architecture for predicting article dwell times. The proposed model extracts emotion, event and entity-based features from an article, learns interactions among them, and combines the interactions with the word-based features of the article to learn a model for predicting the dwell time. We apply the proposed model to a real dataset from a national newspaper showing that the proposed model outperforms other state-of-the-art baselines.

1 Introduction

For online newspapers, it is desirable to predict how user-engaging an article is before publishing it so that editors have an idea about the prosperity of the article. This will help editors select more engaging articles to publish and also make smarter decisions to increase revenue (e.g., displaying more advertisements with an engaging article). Most of the previous studies focus on predicting the page views (i.e., user clicks) as the sole indicator of user engagement and article success (Kim et al., 2016; Ioannidis et al., 2016). However, click-based engagement modeling can be quite noisy (e.g., when a user clicks on a wrong article) and may not show the actual user engagement or satisfaction (Yi et al., 2014). Alternatively, it is shown that the time that a user spends on a page, known as the *dwell time*, is one of the most significant indicators of user engagement (Claypool et al., 2001; Fox et al., 2005; Kim et al., 2014). Thus, we consider dwell time as an engagement measure and design an effective model to predict the dwell time of an article based on its content.

There are some studies on dwell time prediction. Most of them predict dwell time for webpages instead of news articles. Liu et al. (Liu et al., 2010) use regression trees to predict the Weibull distributions of webpage dwell time using keywords and page size. Yi et al. (Yi et al., 2014) predict web content dwell time using support vector regression based on the content length and topic category across different devices. Kim et al. (Kim et al., 2014) use a regression model to estimate the Gamma distributions of page dwell time based on the topic of the page, its length and its readability level. To our knowledge, none of the studies focuses on news articles nor investigates whether high-level features such as events, entities and emotions play an important role in the user engagement of an article measured by dwell time. We believe such high level features are important factors for dwell time prediction.

In this paper we focus on news articles and consider *events*, *emotions* as well as *people* and *organizations* as main contributors to the article dwell time. Both low-level (e.g., word-based) and high level features (e.g., people) are used in our prediction model. However, features such as people and organizations have a very high dimensionality resulting in sparse data representations. In addition, interactions between such features matter. For example, articles mentioning two celebrities (e.g., Prince Harry and Meghan Markle) may be more engaging than articles mentioning only one of them. To address such issues, we propose a model based on the wide and deep neural network architecture (Cheng et al., 2016) which *memorizes* the low order interactions between the sparse features (e.g., people in articles), and at the same

Proceedings of NAACL-HLT 2019, pages 226–233
Minneapolis, Minnesota, June 2 - June 7, 2019.

time *generalizes* word-based content through the deep component. In order to learn the interactions between features, we adopt the factorization machine (Guo et al., 2017), which extracts feature interactions automatically, as the wide component in the proposed model. Our main contributions are as follows. First, we design a novel framework for predicting the dwell time of a news article based on its content. Second, we propose an effective deep neural network model that combines the low-order interactions between high-level factors (i.e., events, emotions and entities) and word-based abstract features for article dwell time prediction. Third, we apply the proposed model to a real dataset from the Globe and Mail[1] and show the effectiveness of the proposed model and the usefulness of event, emotion and entity based features and their interactions for dwell time prediction.

2 Problem Definition

Assume that $\mathcal{D} = \{a_i\}_{i=1}^{N}$ is a set of articles, and $\mathcal{T}_i = \{t_j\}_{j=1}^{N_i}$, is a set of dwell times of article $a_i \in \mathcal{D}$ (based on different users visits), and N_i is the number of visits which article a_i has. To see which type of distribution is most appropriate for modeling article dwell time, we fit the dwell times of articles into different distributions and calculate the average log likelihood among all the articles as the fitness scores. The negative log likelihood of Normal, Exponential and Weibull distributions for our real dataset from the Globe and Mail dataset are 5100.57, 4447.62, and 4306.89 respectively. Therefore, Weibull distribution is selected for modeling article dwell times. Thus, we define the *dwell time* of article a_i, denoted by y_i, as the expected value of the Weibull distribution of dwell times in $\mathcal{T}_i$. We utilize y_i as the target value and build a model to predict it.

PROBLEM STATEMENT: Given a set of articles $\mathcal{D} = \{a_i\}_{i=1}^{N}$ and their respective dwell times, the goal is to learn a model so that it can be used to predict the dwell time of a new article.

3 Detecting High Level Content Factors

3.1 Article-level Emotion Detection

Emotion detection from text has been widely studied in different contexts (Mohammad and Turney, 2013). However, it is not been investigated for the dwell time prediction task. We consider 6 basic emotions (i.e., *happiness*, *sadness*, *disgust*, *anger*, *surprise*, and *fear*) which are widely used in the emotion detection (Ekman, 1992; Agrawal et al., 2018). We utilize a publicly available emotion lexicon (Mohammad and Turney, 2010) as the seed words of different emotions. Given an article $a_i \in \mathcal{D}$, and the word $w \in a_i$, the emotion vector of word w is defined as: $\boldsymbol{em}(w) = [emw_j]$, where emw_j is the average similarity[2] between the pre-trained embedding vector of word w (Mikolov et al., 2013) and those of the seed words of emotion j. The emotion vector for article a_i is calculated as:

$$X_{EM}^{i} = \frac{1}{|\sum_{w \in a_i} \boldsymbol{em}(w)|_1} \sum_{w \in a_i} \boldsymbol{em}(w) \qquad (1)$$

where the denominator is for the scaling purpose.

3.2 Article-level Event Detection

News and events are closely related to each other. Most of the time, a news article reports one central event and a mixture of associated subsidiary events (Chakraborty et al., 2016). The central and subsidiary events manifest themselves in the article content through the event trigger words. Despite the importance of events in news analytics applications (Agrawal et al., 2016), to the best of our knowledge, no study has considered them in article dwell time analysis.

We adapt the method proposed in (Yang and Mitchell, 2016) to extract the events at the article level. The method learns event structures and relations from a corpus and trains a Conditional Random Field (CRF) to extract events. The learned probabilistic models are integrated into a single model to jointly extract events and entities (e.g., people and organizations) from a document. We train the model on the ACE 2005 corpus[3] (Walker et al., 2006). We follow the same setting as (Yang and Mitchell, 2016). We define the event vector for each word w in article a_i as follows: $\boldsymbol{ev}(w) = [evw_j]$, where evw_j is 1 if w is assigned to the $j'th$ event, otherwise 0. The article level event vector X_{EV}^{i} for article a_i is defined as:

$$X_{EV}^{i} = \frac{1}{|\sum_{w \in a_i} \boldsymbol{ev}(w)|_1} \sum_{w \in a_i} \boldsymbol{ev}(w) \qquad (2)$$

We compute the entity vector for word w in a similar fashion: $\boldsymbol{en_k}(w) = [enw_j]$, where enw_j is 1 if w is the $j'th$ instance of entity k (where

[1] https://www.theglobeandmail.com

[2] We use the positive cosine similarity (i.e., max{0,cosine similarity}) as the similarity measure.

[3] https://catalog.ldc.upenn.edu/LDC2006T06

k is a type of entity, i.e., person or organization), otherwise 0. For article a_i, the article level entity vector $X^i_{EN_k}$ is defined as:

$$X^i_{EN_k} = \frac{1}{|\sum_{w \in a_i} \boldsymbol{en_k}(w)|_1} \sum_{w \in a_i} \boldsymbol{en_k}(w) \quad (3)$$

We extract 31 events, 87083 people, and 79143 organizations from the the Globe and Mail dataset.

4 Content-based Correlation Analysis

In this section, we study how different factors of an articles (i.e., entities, emotions, and events) impact the dwell time of the article. We define the *engagement score* of factor c as follows:

$$Score(c) = \frac{1}{df(c)} \sum_i \mathbb{I}[c \in a_i] \times y_i \quad (4)$$

where $\mathbb{I}$ is the indicator function and $df(c)$ is the number of articles containing c. The intuition is that if a factor c appears exclusively in some articles with high dwell time (i.e., y_i), it should have a high engagement score. For example, if *Barak Obama* appears in articles with high dwell time, it should receive a high engagement score.

To investigate the extend to which the engagement score of each article factor could explain the variability of the dwell time of articles, we do a Pearson correlation analysis. In particular, we estimate the predicted dwell time of article a_i by averaging the engagement scores of all individual factors of the same type in the article a_i, and then calculate the Pearson correlation coefficient between an article's actual dwell time and its predicted value. Figure 1 shows the Pearson correlation scores between the actual dwell times and the predicted ones for each type of factor. As illustrated, the emotions (EMO) involved in the articles show the most correlation with the article dwell time. Moreover, location (LOC) and time (TIME) have the least correlation with dwell times. This observation motivates us to use emotion (EMO), event (EVENT), person (PER) and organization (ORG) as the *augmented features* in building the dwell time prediction model.

5 Deep Dwell-time Prediction Model

To learn a dwell time prediction model, we represent an article using both the words in the article and its augmented features (i.e., emotions, events, people and organizations). However, the people and organization features are sparse and high-dimensional. Thus, special attention should

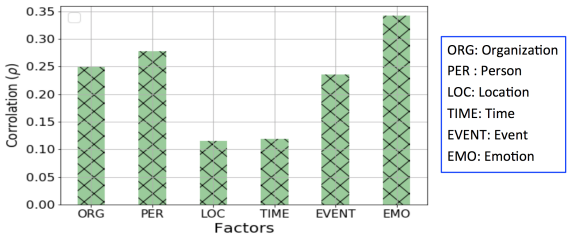

Figure 1: Correlation between the true dwell time and predicted dwell time based on different factors.

be paid to deal with such input features. *Deep neural networks* can learn feature representations and alleviate need for feature engineering by embedding sparse features into a low-dimensional dense space. However, the embedding space may be over-generalized and produce poor results in prediction tasks, when the interactions between high-dimensional features are sparse (Cheng et al., 2016). But such interactions are important for predicting dwell time. For example, an article about two celebrities attending the same event is more likely attracting more readers. Thus, we propose a deep neural network architecture which leverages the augmented features and their interactions in combination with the document (i.e., article) representation to predict the article dwell time.

Inspired by (Guo et al., 2017), we utilized the factorization machine (Rendle, 2010) to capture the augmented feature interactions. However, the proposed model is different from (Guo et al., 2017) in the following aspects: (1) we augment the article content with emotion, event and entity features (2) our model allows multiple factorization machines (each feature is represented with multiple embedding vectors in the factorization layer).

5.1 The Architecture

Figure 2 shows the proposed architecture for the article dwell time engagement prediction task. The architecture consists of two main components: the *deep* and the *factorization machine* components. While the deep competent learns the high order feature interactions and generalizes the article content through a multilayer encoder, the factorization machine captures the low order interactions among the highly sparse augmented features. In particular, suppose that each article is represented by the *TFIDF* (Salton and McGill, 1986) vector X_c, which is fed into the deep component, and augmented vector $X_f = [X_{EV}; X_{EM}; X_{EN}]$, which goes to the factorization machine component, where X_{EV}, X_{EM}, and

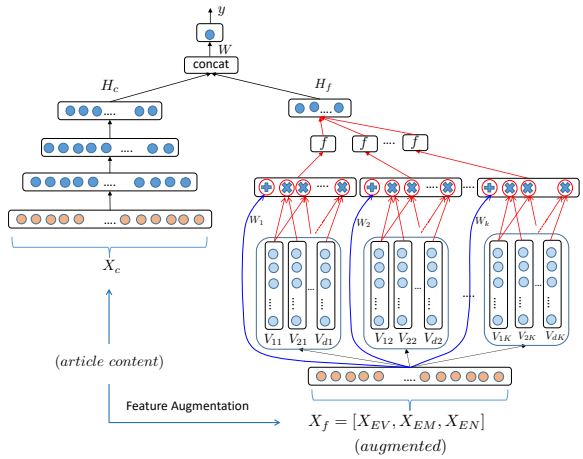

Figure 2: The architecture for article dwell time engagement prediction (left side is deep component and right side is the factorization machine).

$X_{EN} = [X_{EN_{PER}}; X_{EN_{ORG}}]$ are event, emotion, and entity vectors respectively. The whole model is specified by the following equation:

$$H = Concat(Hc, H_f) \quad (5)$$

$$y = WH + b \quad (6)$$

where H_c, H_f are the latent vectors learned by deep and the factorization machine components, H is the concatenation of these two vectors, and W, and b are weight and bias parameters.

5.1.1 Factorization Machine Component

A simple strategy to capture the interactions between features is to learn a weight for each combination of two features. However, this naive approach does not work when the input feature space is sparse. Factorization machine solves the problem by modeling the pairwise feature interactions as the inner product of low dimensional vectors.

The first layer in the factorization machine component is the embedding layer. Given the sparse (augmented) input vector $X_f = [x_i]_{d\times 1}$, it learns multiple vectors $V_{ik} = [v_{ikl}]_{M\times 1}$ $(k = 1 \ldots K)$ for each input dimension, where V_{ik} is the $k'th$ vector for dimension i, and v_{ikl} is the $l'th$ elements of V_{ik}. Then, these factors are fed into the interaction layer to capture the first order and the second order interactions. The interaction layer operation along with the $k'th$ dimension can be formalized as follows:

$$hf_k = f(b_k + \underbrace{W_k \cdot X_f}_{\oplus} + \sum_{i=1}^{d}\sum_{j=i+1}^{d} \underbrace{V_{ik} \cdot V_{jk} x_i x_j}_{\otimes}) \quad (7)$$

where hf_k is the $k'th$ elements of factorization machine component output $H_f = [hf_k]_{K\times 1}$, $W_k = [w_{km}]_{d\times 1}$ (w_{km} is the $m'th$ element of W_k) and b_k are the parameter vector and the bias to be learned and f is the activation function. The $\oplus$ and $\otimes$ symbols in Figure 2 refer to the first order and the second order interaction operations respectively. In fact, factorization machine replaces the interaction weights between feature x_i and x_j with the inner product of respective embedding vectors (i.e., $V_{ik} \cdot V_{jk}$). From modeling perspective, this is powerful since each feature ends up in an embedding space where similar features in this space are close to each other.

5.1.2 Deep Component

In the proposed architecture, the deep component is a dense feed-forward neural network. Each article is vectorized using the *TFIDF* approach (after removing stop words), then is fed into this component. The feed-forward layers convert this sparse vector into low-dimensional dense real-valued vectors.

6 Empirical Evaluation

6.1 Dataset and Set up

All the experiments are conducted on a real dataset from the Globe and Mail dataset. The data collection platform in this company records a timestamp whenever an article page is requested. The difference between two consecutive page click timestamps is used to calculate the articles dwell times. As usual in web analytics the last article in a visit is ignored as we cannot estimate the dwell time for it. Clickstream data is usually noisy. Thus, as a cleaning step, the articles with less than 10 views and dwell time more than 30 minutes are removed resulting in 28502 articles published over period of 2014-01 to 2014-07. Moreover, all the experiments in this section are based on the 10-fold cross validation. We set M and K in the proposed model to 100 and 10 respectively. We used the code in (Pedregosa et al., 2011) with default parameter setting for non-neural networks, and neural network models are implemented using Keras with tensorflow backend (Chollet et al., 2015).

6.2 Baselines

We compare the proposed model with the following baselines including both shallow and deep models as well as Random Forest based models.

Linear Regression (LR): This is a simple baseline used the topics or document vectors as the features and the linear regression method to predict

article dwell times. We extract the articles topics based on the LDA approach (Blei et al., 2003). We set the number of topics to 70 based on the best coherence scores proposed in (Röder et al., 2015). Moreover, we learn the vector representation of each article using the doc2vec method proposed in (Le and Mikolov, 2014). We set the vector size to 100 in all experiments.

Random Forest Regression (RF): Random Forest regression performs well in many applications. It trains an ensemble of uncorrelated decision trees (10 trees in our experiments, which is the default setting in the sklearn code (Pedregosa et al., 2011)), and outputs the average result in the prediction. We used the topic or doc2vec vectors as the input to the Random Forest regression model.

Word Embedding + CNN: We adopt the approach proposed in (Kim, 2014) for the dwell time prediction task. The architecture is comprised of one layer of convolution on top of word vectors pre-trained from an unsupervised neural language model. We use the word vectors[4] trained on 100 billion words of Google News (Mikolov et al., 2013) to initialize the embedding vectors, then fine tuned them in the learning phase. We change the last layer of the architecture (i.e., softmax) to a fully connected (i.e., dense) layer for our task. The final architecture includes convolution, max pooling and fully connected layers.

LSTM + Attention: This is the attention mechanism on top of LSTM layer. The attention layer is designed according to (Raffel and Ellis, 2015). The input of the LSTM are word vectors initialized to pre-trained vectors in (Mikolov et al., 2013). We use a fully connected layer on top of the attention layer to produce the final output.

Multilayer Perception (MLP): This is the multilayer feed-forward network with 3 fully connected (dense) hidden layers. In the model architecture we set 300, 200, and 100 as the hidden layer sizes respectively. This is the deep component in the proposed deep and wide model.

6.3 Evaluation Metrics

We utilize the following metrics to evaluate the performance of different models. Given the actual dwell time y_i and predicted dwell time $\hat{y}_i$ for article a_i ($i = 1, 2, \ldots N$). We calculate the *Mean Square Error* (*MSE*) as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (8)$$

[4] Available at: https://code.google.com/archive/p/word2vec/

Method	MSE	RAE (%)
LR +LDA	4835.74	90.75
LR + Doc2Vec	4857.26	91.21
RF + LDA	4750.10	87.96
RF + Doc2Vec	4566.38	86.44
Word2Vec+CNN	4564.80	85.58
LSTM + Attention	4553.85	90.66
MLP	4122.35	80.79
MLP+Flat Augmented Features (without FM)	4483.34	85.77
Proposed Model (MLP+Augmented Features+FM)	**3883.13**	**78.51**

Table 1: Evaluation of different methods.

Moreover, we calculate the *Relative Absolute Error* (*RAE*) as:

$$RAE = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{\sum_{i=1}^{N}|y_i - \bar{y}_i|} \quad (9)$$

where $\bar{y}_i = \frac{1}{N}\sum_{i=1}^{N} y_i$. Note that $RAE \in [0, \infty)$.

6.4 Experimental Results

Table 1 shows the MSEs and RAEs of different baseline approaches as well as the proposed model. As shown, the proposed model outperforms all the baselines. For shallow (i.e., LR-based) and RF-based models we learn the features using LDA or Doc2Vec approaches and then train the model with Linear Regression (LR) and Random Forest (RF) respectively. As shown, among such models RF+Doc2Vec performs the best.

Among the deep neural network based baselines, we observe that MLP performs better than the other two. One reason could be that our dataset is not very big (with 28502 articles) and as a result the complex models such as CNN and LSTM may overfit to the training data.

To investigate the effect of learning feature interactions with factorization machines, we created another baseline that use MLP with both words and augmented features as input without using factorization machines (denoted as 'MLP + Flat Augmented features" in Table 1). We choose MLP because it is the best among the baselines. As can be seen, the naive approach of adding the augmented features to MLP without using factorization machines leads to poor results.

Table 2 shows the effect of different types of augmented features on the performance of the proposed model. As we observe, using all the augmented features in the proposed model results in the best performance.

6.5 Hyper parameter study

Figure 3 shows the model performance in terms of the number of hidden vectors per feature dimension. We increase the number of hidden vectors

Augmented Features	MSE	RAE (%)
PER	3966.15	79.49
PER+ORG	3963.55	79.36
PER+ORG+EVENT	3933.71	79.10
PER+ORG+EVENT+EMO	**3883.13**	**78.51**

Table 2: Effect of different augmented features.

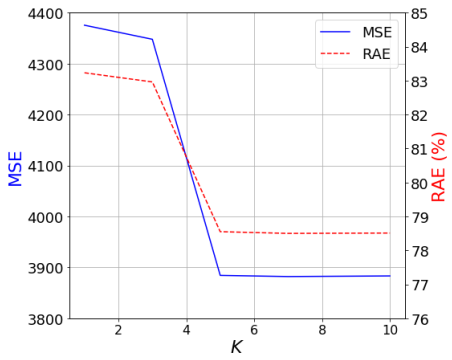

Figure 3: The number of hidden vectors.

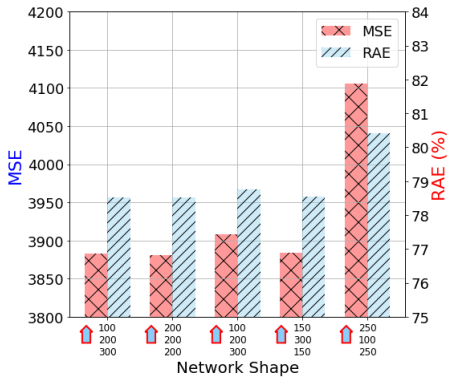

Figure 4: The architecture shapes.

(i.e., K) in the factorization machine component and calculate the errors accordingly. As can be observed, the errors decrease significantly by increasing K from 1 to 5, then becomes stable. This suggests that a value between 5 to 10 would be a good choice for this parameter.

To see the effect of different deep component architecture shapes on the error measures, we keep the number of nodes constant (i.e., 600), and change the number of nodes in the hidden layers. Figure 4 shows the effect of selecting different architectures on the errors. As can be seen, the 250-100-250 is the worst among all architecture and 300-200-100 is slightly better than the others.

In order to study the effect of activation functions on the overall errors, we keep the last layer activation function to ReLU (as it outputs a dwell time value which is always a positive real number) and change the other activation functions to $Tanh$

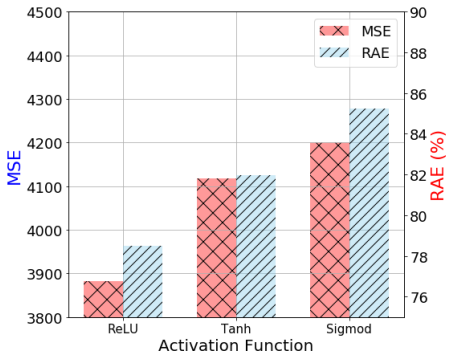

Figure 5: The activation functions.

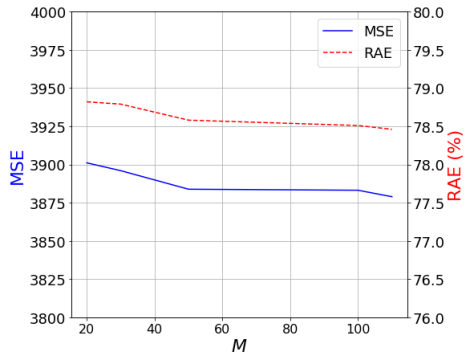

Figure 6: The hidden vector size.

and $Sigmoid$, and then $ReLU$. Figure 5 shows the model errors for different activation functions. Among these activation functions, $ReLU$ gives the best performance and $Simod$ performs considerably worse than the others.

Figure 6 shows the effect of the hidden vector size (i.e., M) of factorization machine component on the overall errors. We observe that errors slightly decrease by increasing hidden vector size form 20 to 40, and then does not show any significant improvement for M between 40 to 100. As such, the proposed model is not sensitive to vector size and this parameter can be set with a value between 40 to 100. Figure 7 shows the prediction errors for different numbers of layers of the deep component. As can be seen, the errors decrease as we increase the number of hidden layers from 1 to 2 and is the best when it is 3.

In order to study the effect of neurons on prediction errors. We start from $300 - 200 - 100$ architecture and increase the hidden layer size by a certain percentage (i.e., 10%, 20%,. . .), then calculate the errors for each architecture. Figure 8 shows the performance of the model for different percentage of node number increase. We observe that the errors remain almost at the same levels

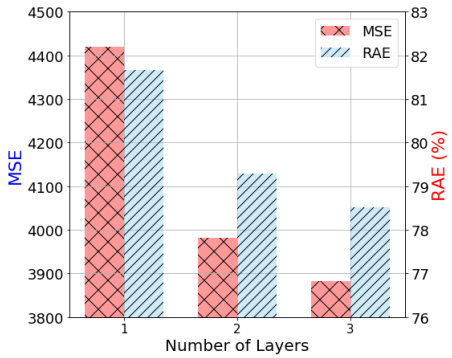

Figure 7: The number of layers.

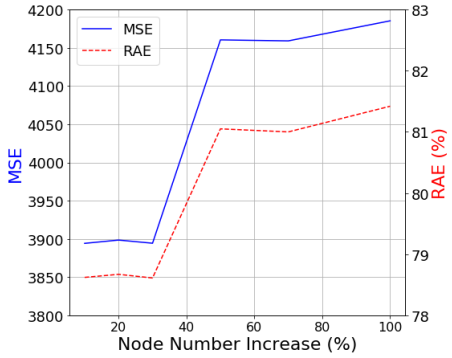

Figure 8: The number of nodes.

when the node numbers in each layer increase by 30%, then starts to get worse from 30% to 100%. This could be due to the overfitting problem.

7 Conclusion

We proposed a novel model to predict the dwell time of a news article based on its content. We first extracted events, emotions, people and organizations from news articles, and then used a deep and wide neural network architecture to learn a prediction model from both the word-based features (via the deep model) and the interactions among the pre-extracted features (via factorization machines). We applied the proposed model to a real dataset from a national newspaper, and showed that using events, emotions, people and organizations and their interactions as features greatly improves article dwell time prediction. The performance of our model is better than using only the deep models for learning abstract features from document representations such as topics, word embedding or TFIDF-based features. As dwell time is a commonly used article engagement measure, the proposed method is of great practical value for news agencies. In addition, the proposed model can be used for other text regression tasks (e.g., predicting revenues from reviews).

Acknowledgments

This work is funded by Natural Sciences and Engineering Research Council of Canada (NSERC), The Globe and Mail, and the Big Data Research, Analytics and Information Network (BRAIN) Alliance established by the Ontario Research Fund-Research Excellence Program (ORF-RE). We would also like to thank Amin Omidvar for his collaboration at the early stage of this work.

References

Ameeta Agrawal, Aijun An, and Manos Papagelis. 2018. Learning emotion-enriched word representations. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 950–961.

Ameeta Agrawal, Raghavender Sahdev, Heidar Davoudi, Forouq Khonsari, Aijun An, and Susan McGrath. 2016. Detecting the magnitude of events from news articles. In *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on*, pages 177–184. IEEE.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Sunandan Chakraborty, Ashwin Venkataraman, Srikanth Jagabathula, and Lakshminarayanan Subramanian. 2016. Predicting socio-economic indicators using news events. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1455–1464. ACM.

Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 7–10. ACM.

François Chollet et al. 2015. Keras. https://keras.io.

Mark Claypool, Phong Le, Makoto Wased, and David Brown. 2001. Implicit interest indicators. In *Proceedings of the 6th international conference on Intelligent user interfaces*, pages 33–40. ACM.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. 2005. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems (TOIS)*, 23(2):147–168.

Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*.

Stratis Ioannidis, Yunjiang Jiang, Saeed Amizadeh, and Nikolay Laptev. 2016. Parallel news-article traffic forecasting with admm. In *SIGKDD Workshop on Mining and Learning from Time Series*. ACM.

Joon Hee Kim, Amin Mantrach, Alejandro Jaimes, and Alice Oh. 2016. How to compete online for news audience: Modeling words that attract clicks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1645–1654. ACM.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Youngho Kim, Ahmed Hassan, Ryen W White, and Imed Zitouni. 2014. Modeling dwell time to predict click-level satisfaction. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 193–202. ACM.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Chao Liu, Ryen W White, and Susan Dumais. 2010. Understanding web browsing behaviors through weibull analysis of dwell time. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 379–386. ACM.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Saif M Mohammad and Peter D Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 26–34. Association for Computational Linguistics.

Saif M Mohammad and Peter D Turney. 2013. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Colin Raffel and Daniel PW Ellis. 2015. Feed-forward networks with attention can solve some long-term memory problems. *arXiv preprint arXiv:1512.08756*.

Steffen Rendle. 2010. Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 995–1000. IEEE.

Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408. ACM.

Gerard Salton and Michael J. McGill. 1986. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57.

Bishan Yang and Tom Mitchell. 2016. Joint extraction of events and entities within a document context. *arXiv preprint arXiv:1609.03632*.

Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. 2014. Beyond clicks: dwell time for personalization. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 113–120. ACM.