CONTENTS

Abstract

Due to the perception of cheap publishing, organizations have been producing humongous content online since the hidden cost of maintenance and usability has always been neglected. This presents the opportunity for automatically maintaing crisp and usable content, especially in news articles. In this paper, we use deep learning to extract features from different classes of content and cluster them under an umbrella topic. For each cluster, we then go on to predict popularity of documents using additional features based on the content only. We conduct our experiments on different news corpuses. Our study also serves to remove information redundancy in multiple documents.

Keywords

Deep learning for text, news, redundancy, content popularity.

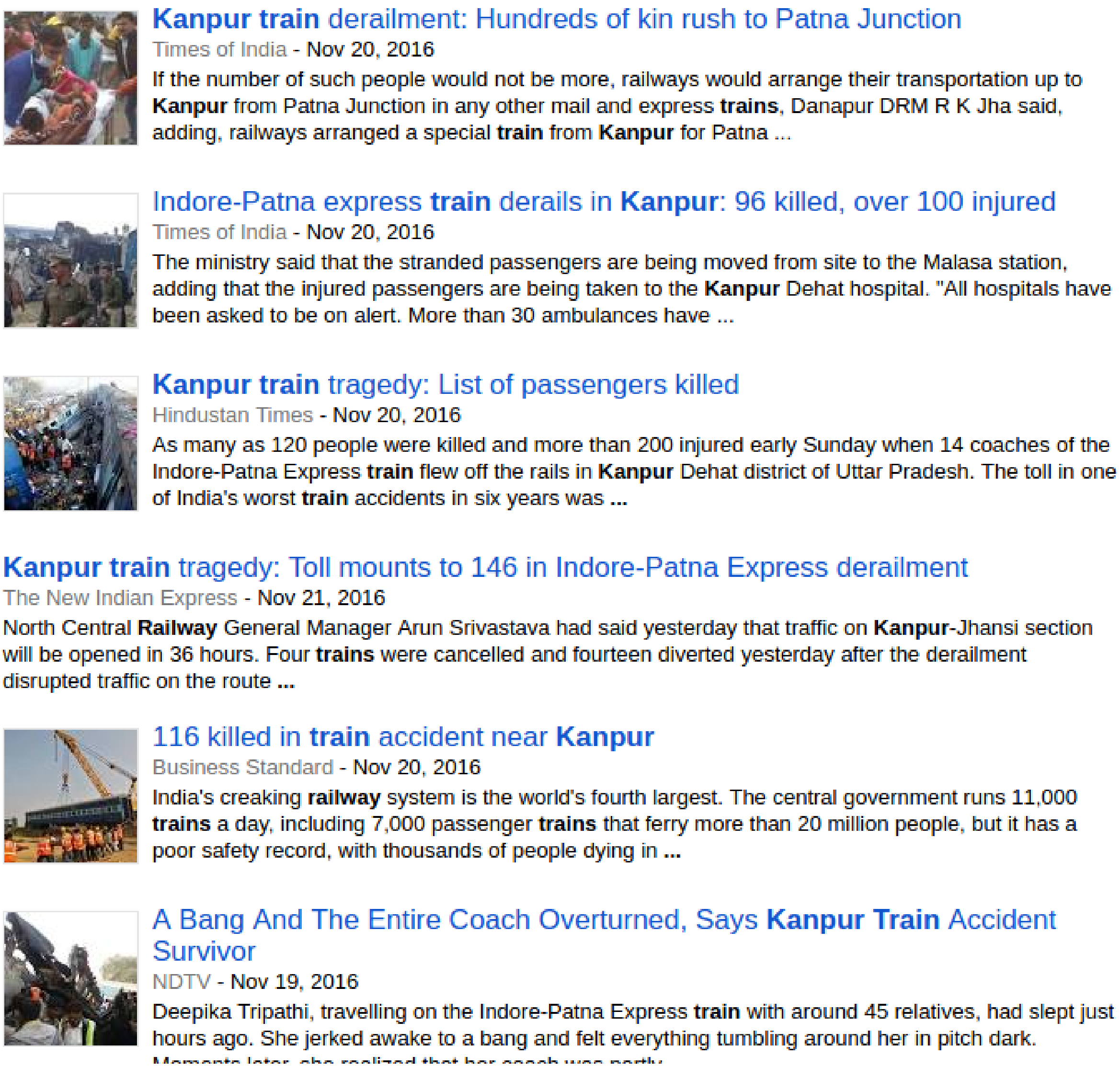
I. Introduction

News articles are very dynamic in nature due to continuously developing nature of the event and parallel reporting of the same, thus they have a very short span of life. The ease and low cost of online content creation and sharing have changed the traditional rules of competition for public attention. News sources now concentrate a large portion of attention on online mediums where they can disseminate their news effectively and to a large population. Due to the time-sensitive post aspect and intense competition for attention in the socially connected digital platform, accurately estimating the extent to which a news article will spread on the web is extremely valuable to journalists, content providers, advertisers and news recommendation systems. However, predicting the online popularity of online news articles is a challenging task. First, context outside the web is often not readily accessible and elements such as local and geographical conditions and various circumstances that affect the population to make this prediction extremely difficult. Furthermore, network properties such as the structure of social networks that are propagating the news, influence variations among members and interplay between different sections of the web add other layers of complexity to this problem. Most significantly, intuition suggests that content of an article must play a significant role in its popularity. Content that resonates with a majority of readers such as a major worldwide event can be expected to garner wide attention while specific content relevant only to a few may not be as successful. Content that is up-to-date and highlights all aspect of that article.

The news data for our study has been collected from AG's news corpus and Financial News Dataset. To generate features for the articles, we have used Character-level Convolution Neural Network. To remove redundant information, we perform specific topic-wise clustering in a certain timeframe. For each cluster, we analyze the contents of new articles and use those for prediction of the popularity prior to publishing. Our work shall also help content writers to remove irrelevant, outdated, trivial and redundant content.

Fig.1 shows the results of search query ”Kanpur Train” on Novermber 26, 2016 at 16:44 IST on Google News. As we can see, outdated news reports are ranked better than latest news reports. This may be because of social influence of outdated and redundant news over different media channels. Our aim here is to rank the latest and the most informative articles at the top.

The rest of the paper is organized as follows. Section II describes the business motivation and opportunities behind the problem. Section III gives a glimpse of the related academic work we have surveyed over the due course. Section IV details the solution framework determined for the problem. After that, in Section V we describe the evaluation criteria for



Kanpur train derailment: Hundreds of kin rush to Patna Junction

Times of India - Nov 20, 2016

If the number of such people would not be more, railways would arrange their transportation up to **Kanpur** from Patna Junction in any other mail and express **trains**, Danapur DRM R K Jha said, adding, railways arranged a special **train** from **Kanpur** for Patna ...

Indore-Patna express **train** derails in **Kanpur**: 96 killed, over 100 injured

Times of India - Nov 20, 2016

The ministry said that the stranded passengers are being moved from site to the Malasa station, adding that the injured passengers are being taken to the **Kanpur** Dehat hospital. "All hospitals have been asked to be on alert. More than 30 ambulances have ...

Kanpur train tragedy: List of passengers killed

Hindustan Times - Nov 20, 2016

As many as 120 people were killed and more than 200 injured early Sunday when 14 coaches of the Indore-Patna Express **train** flew off the rails in **Kanpur** Dehat district of Uttar Pradesh. The toll in one of India's worst **train** accidents in six years was **...**

Kanpur train tragedy: Toll mounts to 146 in Indore-Patna Express derailment

The New Indian Express - Nov 21, 2016

North Central **Railway** General Manager Arun Srivastava had said yesterday that traffic on **Kanpur**-Jhansi section will be opened in 36 hours. Four **trains** were cancelled and fourteen diverted yesterday after the derailment disrupted traffic on the route **...**

116 killed in **train** accident near **Kanpur**

Business Standard - Nov 20, 2016

India's creaking **railway** system is the world's fourth largest. The central government runs 11,000 **trains** a day, including 7,000 passenger **trains** that ferry more than 20 million people, but it has a poor safety record, with thousands of people dying in **...**

A Bang And The Entire Coach Overturned, Says **Kanpur Train** Accident Survivor

NDTV - Nov 19, 2016

Deepika Tripathi, travelling on the Indore-Patna Express **train** with around 45 relatives, had slept just hours ago. She jerked awake to a bang and felt everything tumbling around her in pitch dark. Moments later, she realized that her coach was partly

Fig. 1: Google News results on query Kanpur train on 26 Nov. 2016 at 16:44 IST.

different experiments to be conducted. Finally, we conclude with a summary of this work in Section VI and future possibilities and extensions in Section VII.

II. Business

- **Content Caching and Traffic Management** There is a hidden cost to publishing content, the cost to review and maintain the content. The millions of articles also affect the usability and maintainability of the site. In the long run, it is necessary to tackle redundant, outdated and trivial content which has been cursing the site.
- **Advertising** This work can finds it's application in content-based advertisement alongside news pieces. It will optimize ad-placement logistics and revenues.
- **News Aggregation** With our current event driven clusters knowledge base, we predict the popularity of written articles to be published in that domain. It will allow content writer to write more relevant and less redundant pieces that can make it different from current flowing articles. We have been aggregating up-to-date content rich articles ignoring social backlinks.

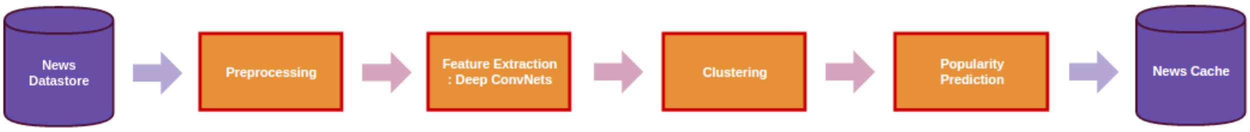


Fig. 2: Solution Framework

- **Trends Forecasting** Since the cache contains most popular pieces from different news events, we are able to show current trends with no redundancy. This data helps in forecasting future trends.

III. Related Work

Previous research on news content analysis for forecasting its popularity prior to publishing takes a lot of factors besides content. Often, factors like social network links, inorganic links undermines content of articles. We believe that content is still the king of any article.

Weber et. al. [1] consider static and dynamic factors to design a framework that does event based clustering of news articles and filters duplicate and less relevant news, making it easier to read the more interesting news in less time.

Traditionally, feature extraction for articles clustering is done using self-design features such as name entity relation, TF-IDF. On the other hand, researchers have found convolutional neural networks [2] [3] to be useful in extracting information from raw signals, ranging from computer vision applications to speech recognition and others Zhang et. al. [4] explored treating the text as a kind of raw signal at the character level and applying temporal (one-dimensional) ConvNets to it. It has been shown that ConvNets can be directly applied to distributed[5][6] or discrete[7] embedding of words, without any knowledge on the syntactic or semantic structures of a language. These approaches have been proven to be competitive to traditional models. This simplification of engineering could be crucial for a single system that can work for different language since characters always constitute a necessary construct regardless of whether segmentation into words is possible. Working on only characters also has the advantage that abnormal character combinations such as spelling mistakes and emoticons may be naturally learned.

Szabo et. al. [8], Tatar et. al. [9] predict popularity from early user reaction to the event which is not sufficient. The problem of prdicting popularity prior to publishing is more interesting and challenging to solve. Bandari et. al. [10] predict popularity of news items only from features extracted from content. We further exploit this work for our purposes.

Finally, we predict the most popular articles in different clusters by different factors such as news source that generates those articles, the subjectivity of the language in the article, factual completeness of the article, name entities mentioned in the article.

IV. Solution Framework

Fig.2 shows a high level overview of the solution framework.

A. Preprocessing

We preprocess the data to make processing more meaningful [1].

- **Filtering** Removal of markup, punctuation and special characters from sentences.

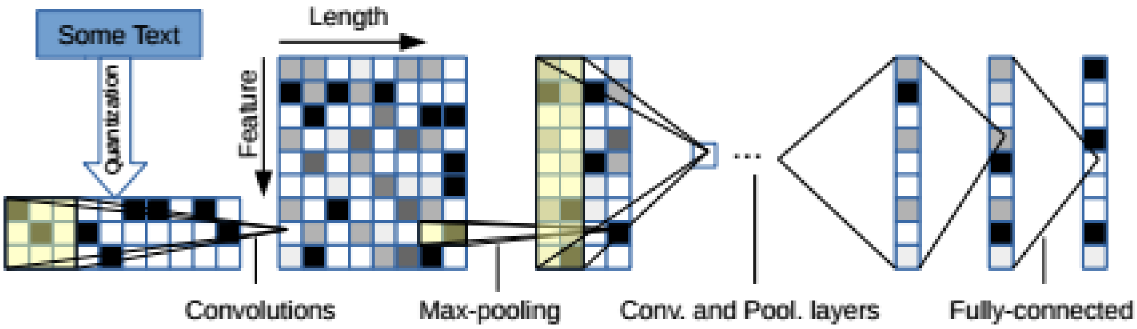


Fig. 3: Deep ConvNets model illustration for Feature extraction.

- **Tokenization** Splitting of text into individual units.
- **Stemming** Reduction of words to their base forms [11].
- **Stopwords removal** Deletion of words that do not convey any special meaning.
- **Pruning** Removal of words that do appear with a low frequency throughout the text.

The result of these preprocessing steps is a set of feature words.

B. Text Understanding

Text understanding consists in reading texts formed in natural languages, determining the explicit or implicit meaning of each element such as words,phrases, sentences and paragraphs, and making inferences about the implicit or explicit properties of these texts[12]. Text understanding can be handled by a deep learning system without artificially embedding knowledge about words, phrases, sentences or any other syntactic or semantic structures associated with a language [4] [13] [3]. ConvNets for text understanding are modular, where gradients are obtained by back-propagation to perform optimization.

Key Modules: It is a temporal convolutional module, which simply computes a 1-D convolution between input and output.

Character Quantization: Our model accepts a sequence of encoded characters as input. The encoding is done by prescribing an alphabet of size m for the input language, and then quantize each character using 1-of-m encoding. Then, the sequence of characters is transformed into a sequence of such m sized vectors with fixed length l. Any character exceeding length l is ignored, and any characters that are not in the alphabet including blank characters is quantized as all-zero vectors.

Model Design: Models shall be tested with different number of hidden units and frame sizes to train model for optimal results.

C. Clustering

In recent years, internet has become a mainstream medium and offers opportunity for large-scale production and distribution. With more news than ever, it has become increasingly difficult to find relevant news. Regardless which approach is taken and which services are used, one may be confronted with multiple news about the same event within the field of interest. The importance of a news event creates the need for a regular detailed coverage and hence, duplicates and redundant pieces. During high-peak of interest to a particular topic, there is no imit to number of duplicates produced. We need to manually filter and review the relevant news pieces. Existing approaches like Weber et. al. [1] cluster news pieces based on similarity of textual content. We intend to use deep learning feature vectors for clustering news items into highly specific cluster from a particular news event. Clustering algorithm k-means[14] does not work because it requires number of clusters beforehand. As the number of clusters will never be fixed, we use Average-link agglomerative clustering. We believe that the cluster should be densely connected to an event and thus, average-link distance.

D. Popularity Prediction

We are motivated to predictp opularity of article beforehand only from content based features and store only a plausible set of articles from each cluster. Bandari et. al. [10] use a supervised classification of category of popularity based on number of tweets. We intend to generate a score for each of the articles unlike categorising them into particular classes.

1) Features: The choice of features is motivated by multiple qustions. Does the source agent reach many readers? Does the language connect with the reader? Has the article became outdated? Do we have some information in the news piece or not? Is the news worthy of a read? These questions helped us in designing following six features.

- **Age** The date of publication of news given by the dataset. We remove few records with missing dates.
- **Text Quality** The ratio of size of document before and after preprocessing.
- **Source Quality** The popularity of source of the content given by initial number of hits provided by the source. If missing, we use the popularity of news agent as a whole. This is log-normalized to account for high range of hits.
- **Subjectivity** This examines whether an article is written in more emotional, touchy tone, where it connects with the reader. We make use of subjectivity classifier from Lingpipe, a natural language toolkit.
- **Named Entities** We hypothesize that well-known named entities will cause a further spread of the article. For instance, articles on Narendra Modi are more likely to be popular among Indian Readers as compared to others. We make use of Stanford CoreNLP[15] to process named entities. We rate entities based on their prominence or past popularity in the media.
- **Factual Density** Previous works like Lex et. al. [16] suggest that density of factual content can be used to measure informativeness of text documents. Zhu et. al. [17] try to eliminate redundancy in news pieces. We utilize the above works to rank pieces based on their information content.

V. Experimental Evaluation

A. Dataset

1) **AG's news corpus** We obtained AG's corpus of news articles on the web[1]. It contains more than 1 million news aricles from more than 2000 sources. We use the source, title, description, rank and pubdate fields for our experiments.
2) **Financial News Dataset** We obtained financial news dataset of Bloomberg and Reuters[2]. It contains 450,341 articles from Bloomberg and 109,110 articles from Reuters. It provides us title, content, date and author of the article.

[1] https://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

[2] https://github.com/philipperemy/financial-news-dataset

3) **20 Newsgroups Dataset** The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. The data is organized into 20 different newsgroups, each corresponding to a different topic.[3]

B. Baseline

1) **News Aggregators** We conduct an internal survey to verify initial results of the pipeline when compared with different news agents and aggregators like Google News, Feedly, Digg etc.
2) **TF-IDF Comparison** We use an obvious baseline based on sum of tf-idf scores of entities in the document to rank it's importance.
3) **Bandari et. al.** We compare the results of our work with previous work by Bandari et. al. [10] which predict popularity of news articles on twitter using features derived only from content prior to publishing.
4) **Clustering Metrics** There exist multiple clustering metrics like Dunn index[18], Davies-Bouldin index [19] etc. [20] to measure how good the clustering is. With this, we test first two modules of the pipeline, namely, feature extraction and clustering.

VI. Conclusion

In this paper, we improve the quality of news cache and recommendations by predicting popularity of articles prior to publishing. The need for the same arises from the stiff competition among different news agencies and aggregators. Through deep convolutional neural networks, we extract features of articles at character-level. To remove redundant information, we make highly specific clusters of news items. Finally, we predict the most popular pieces in differnt clusters to provide the set of most popular articles, which is then used for multiple use-cases in content caching, advertising, forecasting and recommendation. With an initial survey, we ensure inceptive results of the pipeline versus different competitors. Lastly, we compare with different baselines to ascertain quality of our work.

VII. Future Work

1) Information explosion is prevalent not only in news items, but across different content classes. There rises the opportunity to extend the current work for multiple such classes.
2) There are abundant works [8] [9] which when given initial popularity of some content, are better able to forecast popularity. With that notion, we want to adapt the predictions to incoming hits.
3) With growing amount of personalization in different news agents [21] [22], there exists a demand and an opportunity for the same in this work.
4) Growing number of events and articles pose the need for concise summaries of multiple articles. We believe, successful summarization of one cluster can generate concise, clear and helpful news articles.

[3] http://scikit-learn.org/stable/datasets/twenty_newsgroups.html

References

[1] M. Weber and M. H. Lamers, “Finding news in a haystack-event based news clustering with social media based ranking,” in *Cloud and Green Computing (CGC), 2013 Third International Conference on*, pp. 321–326, IEEE, 2013.

[2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] X. Zhang, J. Zhao, and Y. LeCun, “Character-level convolutional networks for text classification,” in *Advances in Neural Information Processing Systems*, pp. 649–657, 2015.

[5] C. N. dos Santos and M. Gatti, “Deep convolutional neural networks for sentiment analysis of short texts.,” in *COLING*, pp. 69–78, 2014.

[6] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.

[7] R. Johnson and T. Zhang, “Effective use of word order for text categorization with convolutional neural networks,” *arXiv preprint arXiv:1412.1058*, 2014.

[8] G. Szabo and B. A. Huberman, “Predicting the popularity of online content,” *Communications of the ACM*, vol. 53, no. 8, pp. 80–88, 2010.

[9] A. Tatar, J. Leguay, P. Antoniadis, A. Limbourg, M. D. de Amorim, and S. Fdida, “Predicting the popularity of online articles based on user comments,” in *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, WIMS ’11, (New York, NY, USA), pp. 67:1–67:8, ACM, 2011.

[10] R. Bandari, S. Asur, and B. A. Huberman, “The pulse of news in social media: Forecasting popularity,” *arXiv preprint arXiv:1202.0332*, 2012.

[11] M. F. Porter, “An algorithm for suffix stripping,” *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[12] P. Linell, *The written language bias in linguistics: Its nature, origins and transformations*. Routledge, 2004.

[13] X. Zhang and Y. LeCun, “Text understanding from scratch,” *arXiv preprint arXiv:1502.01710*, 2015.

[14] M. Telgarsky and A. Vattani, “Hartigan’s method: k-means clustering without voronoi.,”

[15] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, “The stanford corenlp natural language processing toolkit.,” in *ACL (System Demonstrations)*, pp. 55–60, 2014.

[16] E. Lex, M. Voelske, M. Errecalde, E. Ferretti, L. Cagnina, C. Horn, B. Stein, and M. Granitzer, “Measuring the quality of web content using factual information,” in *Proceedings of the 2Nd Joint WICOW/AIRWeb Workshop on Web Quality*, WebQuality ’12, (New York, NY, USA), pp. 7–10, ACM, 2012.

[17] X. Zhu and T. Oates, “Finding story chains in newswire articles using random walks,” *Information Systems Frontiers*, vol. 16, no. 5, pp. 753–769, 2014.

[18] J. C. Dunn, “Well-separated clusters and optimal fuzzy partitions,” *Journal of cybernetics*, vol. 4, no. 1, pp. 95–104, 1974.

[19] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.

[20] F. Kovács, C. Legány, and A. Babos, “Cluster validity measurement techniques,” Citeseer.

[21] I. Ilievski and S. Roy, “Personalized news recommendation based on implicit feedback,” in *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge*, NRS ’13, (New York, NY, USA), pp. 10–15, ACM, 2013.

[22] J. Bao and M. F. Mokbel, “Georank: an efficient location-aware news feed ranking system,” in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 184–193, ACM, 2013.