WikiNews Generator : In pursuit of concise news

Qin Wang and Cong Lin

Department of Information Technology and Electrical Engineering, ETH Zurich

{qwang, linco}@student.ethz.ch

1. Introduction

In this day and age, everything is about information. First, the amount of information we can access has grown unprecedentedly in the last few decades. From the traditional media like CNN and Bloomberg, to the new online media like Twitter and Wiki News. At the same time, for most of us, we are overwhelmed by the mass information flow. Second, the information could be highly bias based on source’s political view. As an example, towards the news “legalize same sex marriage in the USA”, FOX news wrote “Gay marriage: Why Supreme Court got it wrong” [1], while CNN wrote “Supreme Court rules states must allow same-sex marriage” [2]. Third, fake news not only confuse us as audience but also can create turmoil in political world.

Therefore, in pursuit of the conciseness, neutrality and fact, we propose the news article processing pipeline to generate concise, neutral and real news from various news sources from the internet. We start from crawl source news article from various source like CNN, Reuters and Wiki News etc. At the same time, we train a Fake News detection classifier using the LIAR dataset by William Y. Wang [3], using the method proposed by the same paper, in which the classifier was trained hybridly by metadata and the articles with convolutional Neural Networks. [4] So we filter the fake news articles by the binary classifier we trained. Then do clustering the news articles so the articles describe the same events/topics could be in the same cluster. In the end, we generate the abstract like articles as a summary of the event. The figure showed the pipeline of our method.

1. Related works
2. WikiNews Dataset

We crawled WikiNews articles for our training and evaluation for clustering and summary task. Every WikiNews article has several source articles talking about the same news event. Only news event with more than three source articles from whiltelisted websites are taken into account in the dataset. We collect 1.2K news events from WikiNews.

The dataset is available at http://qin.ee/wikinews.json

1. News clustering

After we get the news article, we started another processing pipeline to do clustering of the news article. The processing pipeline shows in the figure.

4**.1 Pre-processing**

At first, when we extract features, we want to abandon the features that are homogeneous in each document to improve results and speed of clustering. So we exclude the common stop words in English such as: “a”, “an”, “by” etc. Second, in most of language include English, the word often has several morphological variants. [5] For example, “fishing”, “fisher”, “fished” all can be “fish”. Also for some stemmer, “unhappy”, “happiness” could be the same. In our work, we use SnawBall stemmer by MF Porter. [6]

**4.2 Generate features**

We generated the features for clustering algorithms in two ways.

The first method is based on the term frequency-inverse document frequency (tf-idf) model by Salton and McGill [7]. We count word frequency by documents and transfer the document strings to frequency matrix. Words occur in one documents but not in another receive better score.

The second method, by build a dictionary of all words that occurs in the database and count the words occurrence in each document. It will give as a sparse matrix. Then we do singular value decomposition for the sparse matrix for dimension reduction for the final matrix.

**4.3 Clustering**

As we’ve already know the source of our data, for a same event, it normally has 3 sources, and each media, for example, CNN would normally only cover one event with one article. Suppose we have N documents, the number of clusters should be around N/3. In the clustering step, we used K-means and Hierarchical Agglomerative Clustering (HAC). Then for K-means we pick the exact number of cluster using elbow method.

**4.4 Evaluation**

The data we crawled from the internet are collected by events, there we have the label of the clustering. We evaluate the clustering result by 3 matric, homogeneity score, completeness score and V measure score.

High homogeneity score means the clusters contain only data points which are member of a single class. High completeness score means all data points are member of a given class are elements of the same clusters. V measure consider both homogeneity and completeness. [8]

**4.5 Results**

|  |  |  |
| --- | --- | --- |
| Matric | Count Victorizer and SVD | TFIDF |
| Homogeneity Score | 0.845 | 0.923 |
| Completeness Score | 0.880 | 0.917 |
| V measure Score | 0.862 | 0.920 |
| Number of Clusters | 1440 | 1500 |

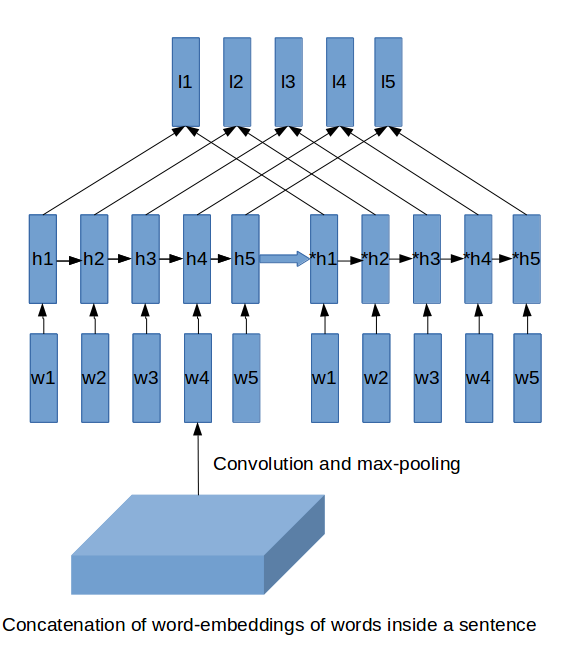
1. News Summarization

After news articles are clustered to different news event, generating summaries based on them – the so-called “multi-document summarization” - in the new task. Before 2015, most multi-document summarization are term-frequency based methods. Later, people start using neural networks to tackle the problem and achieved better results than the traditional methods. In our work, we implement both methods, and try to see if the generation result has a big difference. In both models, news summarization means sentence level extraction, which means the task was turned into a classification problem, where every sentence in the input is given a label by the model to decide whether or not to include it in the summary.

There are also models doing word level extraction, however, since each word is dependent on the previous prediction, the models can easily break if one prediction is not meaningful. In this paper, only sentence level extraction is discussed.

**5.1 Traditional SumBasic Summarizer**

SumBasic was first introduced by Nenkova and Vanderwende (2005).[10] The original work compared how many of the top frequency words appeared in summaries and came to the conclusion that word frequency is one of the factors that affects whether a certain part of the article should be included in the summary. The original algorithm have the following steps. The probability distribution over words are computed, then based on the word distributions, sentence weights are computed as the average probability of the words in the sentence. The sentence with top weight in the input is then selected as part of the summary and we update the probability of every word in that sentence by self-multiplication. By recursively choosing the sentence with top weights, we reach a summary with desired length.

**5.2 Neural Network Summarizer**

A LSTM based neural network summarizer is introduced in this section. Our work mainly follows the architecture from Cheng and Lapata(2016). The network is mainly constructed by four layers. The CNN, max-pooling layer and a sum layer convert the list of word representations of each sentence to 2D sentence representations. Then the LSTM encoder learns the document representation and pass it into the LSTM decoder layer. Finally we have a attention-like layer that concatenate the output of both LSTMs and use a softmax to determine the prediction label. We use the cross entropy loss for our training.

**5.3 Experiments**

We trained our neural model with DailyMail dataset provided by Cheng and Lapata(2016). The dataset contains 170k news article/summary pair. The training takes roughly 40 hours on a Nvidia GTX 1070 to converge. We also use 100 dimension word2vec pretrained embeddings as a strating point.

**5.4 Results**

We briefly report our result of the Neural network in the following table, all evaluations are based on the test set of DailyMail dataset:

**5.5 Comparison between two models**

Summaries for our WikiNews dataset are generated by the above two algorithms. For each news event, we feed the models with three news articles and get a summary. Since there is no simple way to judge which generation is better, and relative papers often ask volunteers to rate the generation, we publish our summaries online (http://qin.ee/summary\_comparison) to ask the reader decide which one has better generation result. As a (extremely strong) baseline, we also put the original summary written by WikiNews authors.

1. Demo

Based on our above work, we here introduce a simple demo that collects news articles, clusters them into different news events and generates summaries based on them. The demo first collects news topics on Google News, and then collects a list of news articles based on these topics. We then do clustering and make sure each article is assigned to the right class. Summaries are then generated based on these articles.

The demo page is updated every 24 hours to reduce computation intensity. Readers can access the demo on http://news.qin.ee

1. Conclusion

Reference:

[1]http://www.foxnews.com/opinion/2015/06/30/same-sex-marriage-is-only-beginning.html

[2]http://edition.cnn.com/2015/06/26/politics/supreme-court-same-sex-marriage-ruling/

[3] Wang, William Yang. "" Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection." arXiv preprint arXiv:1705.00648 (2017).

[4] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

[5] N. Sandhya, Y. Sri Lalitha, V. Sowmya, K. Anuradha, A. Govardhan, Analysis of

stemming algorithm for text clustering, IJCSI International Journal of Computer

Science Issues 8 (5) (2011) 352–359.

[6] Porter, M. F., and Richard Boulton. "Snowball stemmer." On line http://www. snowball. tartarus. org.[Visited 15/11/2005] (2001).

[7] Salton, Gerard, and Michael J. McGill. "Introduction to modern information retrieval." (1986).

[8] Rosenberg, Andrew, and Julia Hirschberg. "V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure." EMNLP-CoNLL. Vol. 7. 2007.

[9] Cheng, Jianpeng, and Mirella Lapata. "Neural summarization by extracting sentences and words." *arXiv preprint arXiv:1603.07252* (2016).

[10] Nenkova, Ani, and Lucy Vanderwende. "The impact of frequency on summarization." Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005 101 (2005).

Figures

