NLP course. Topical extractive summarization

Kristina Zheltova

December 2020

1 Introduction

Topical extractive summarization is directed toward extract sentences most relevant to a given topic. There are many approaches to create a summary from the source text. However, only a few works about extractive summarization using any information about topics. This work proposes a graph-based Biased LexRank approach combines with topic modeling to create a topic-based summary.

2 Related work

In the work [Nallapati et al., 2016] authors present Recurrent Neural Network based on sequence model for extractive summarization. The work focuses only on sentential extractive summarization of single documents using neural networks. Extractive summarization is treated as a sequence classification problem wherein, each sentence is visited sequentially in the original document order and a binary decision is made in terms of whether or not it should be included in the summary. GRU based RNN was used as the basic building block of the sequence classifier.

In the [Xu et al., 2020] work the discourse-aware neural extractive summarization model was built upon BERT. To perform compression with extraction simultaneously and reduce redundancy across sentences, authors take Elementary Discourse Unit (EDU) as the minimal selection unit (instead of sentence) for extractive summarization. Extractive summarization is formulated as a sequential labeling task, where each EDU is scored by neural networks, and decisions are made based on the scores of all EDUs.

The work [Mihalcea and Tarau, 2004] describes TextRank – a graph-based ranking model for text processing and sentence extraction. The basic idea is that when one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex.

In the [Yasunaga et al., 2017] the first step is producing sentence relation graph. Given a relation graph, our summarization model applies a Graph Convolutional Network (GCN), which takes in sentence embeddings from Recurrent

Neural Networks as input node features. then sentence salience estimations are obtained via a regression on top, and it allows extract salient sentences in a greedy manner avoiding redundancy.

In the [Erkan, 2006] authors propose a graph-based sentence ranking algorithm for extractive summarization. This method is a version of the LexRank algorithm extended to the focused summarization task of DUC 2006. As in LexRank, the set of sentences in a document cluster is represented as a graph, where nodes are sentences and links between the nodes are induced by a similarity relation between the sentences. Then authors perform the ranking of the sentences according to a random walk model defined in terms of both the inter-sentence similarities and the similarities of the sentences to the topic description.

The [Cui et al., 2020] paper proposes a graph neural network (GNN)-based extractive summarization model, enabling to capture intersentence relationships efficiently via graph-structured document representation. Also, a joint neural topic model (NTM) to discover latent topics is used.

3 Model description

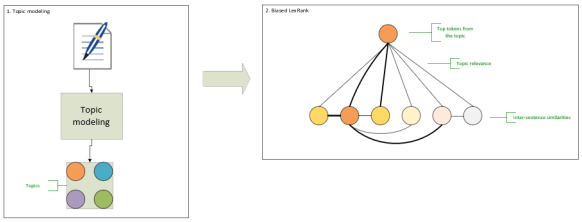

Figure 1: A variant of the model pipeline

The main idea consists of using word embeddings for top-tokens from topic modeling (ie. words with max probability) as a topic description for biased LexRank algorithm [Erkan, 2006]. After that, we can perform random walking and extract summary sentences. Such adopted Biased LR can be computed as

$$\text{LR(u|t)} = \text{d} * \cos(\text{u,t}) + (1 - \text{d}) * \sum_{v \in adj[u]} \frac{cos(u,v)}{\sum_{z \in adj[v]} cos(v,z)} * LR(v|t)$$

where t is a vector of the topic, u is a vector of a sentence, cos is a cosine similarity.

4 Dataset

4.1 The construction of a query-focused summarization corpus

In the [Zhu et al., 2019] authors first take the highlighted statement as the summary. Its supporting citation is expected to provide an adequate context to derive the statement, thus can serve as the source document. On the other hand, the section titles give a hint about which aspect of the document is the summary's focus. Therefore, they use the article title and the section titles of the statement to form the query. Given that Wikipedia is the largest online encyclopedia, massive query-focused summarization examples can be automatically constructed.

4.1.1 Dataset preparation

The original dataset was taken from Russian Wikipedia and contains 10.400 pages from the Personalities category. Each headline is considered as a short topic description. When the topic modeling process is done, headlines are matched with topics by hand. Some of the headlines with unclear sense or with low frequency in the corpus are ignored. After that dataset contains 7.333 short texts with one or more assigned topics. Manual matching of headlines and topics was performed for headlines with a frequency of at least 50.

5 Experiments and results

5.1 Evaluation metrics

Precision is a proportion of sentences really belonging to a given topic regarding all sentences that are related to this topic. Precision is calculated for each document independently.

5.2 Summary generation with BigARTM

The topic model is trained with such parameters:

- max features of CountVectorizer = 2000
- number of topics = 50
- tau of SmoothSparsePhiRegularizer = -2
- tau of DecorrelatorPhiRegularizer = 1e+5
- number of collection passes during the training = 35

Each topic is represented as a list of 15 top tokens and is converted into an average vector of w2v embeddings vectors. By the way, each sentence is also

converted to the w2v representation, and the Biased LexRank algorithm works with these vectors.

The series of experiments was carried out with varying the parameter d (damping factor):

Num	d	Avg precision
1	0	0.6974
2	0.1	0.7134
3	0.2	0.7344
4	0.3	0.7524
5	0.4	0.7706
6	0.5	0.7855
7	0.6	0.7962
8	0.7	0.8031
9	0.8	0.7936
10	0.9	0.8127
11	1	0.8165

6 Conclusion

The Biased LexRank algorithm combined with BigARTM model to create a topic-based summary was implemented via Python 3.

7 Discussion

This work is only the first step to create more complex extractive summarization model. Next steps could include further exploration of the topic models instead BigARTM which work with word embeddings like NVDM. Also, neural models with graph attention could be used.

References

[Cui et al., 2020] Cui, P., Hu, L., and Liu, Y. (2020). Enhancing extractive text summarization with topic-aware graph neural networks. *arXiv preprint arXiv:2010.06253*.

[Erkan, 2006] Erkan, G. (2006). Using biased random walks for focused summarization.

[Mihalcea and Tarau, 2004] Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

[Nallapati et al., 2016] Nallapati, R., Zhai, F., and Zhou, B. (2016). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents.

[Xu et al., 2020] Xu, J., Gan, Z., Cheng, Y., and Liu, J. (2020). Discourse-aware neural extractive text summarization.

[Yasunaga et al., 2017] Yasunaga, M., Zhang, R., Meelu, K., Pareek, A., Srinivasan, K., and Radev, D. (2017). Graph-based neural multi-document summarization. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 452–462, Vancouver, Canada. Association for Computational Linguistics.

[Zhu et al., 2019] Zhu, H., Dong, L., Wei, F., Qin, B., and Liu, T. (2019). Transforming wikipedia into augmented data for query-focused summarization.