Key Phrase Extraction Using Recurrent Neural Network

Aakib Jabbar[1], **Preeti Sondhi**[2]

Abstract: ***In natural language processing, key phrase extraction is a necessary activity that helps map documents to a collection of emblematic phrases. For many uses, such as publication ranking, query-based engines, such as Google Scholar, etc., these sentences can be used. A small set of terms, key phrases and keywords which define the meaning of the document are described by extracting keywords. Keyword search enables efficient search of large document sets. Text categorization techniques can be applied to assign relevant key-phrases to new documents. In the training materials, a predefined list of key-phrases from which all key-phrases for new documents are chosen is issued.For each key-phrase, the training data defines a set of documents associated with it. For each key-phrase, standard machine learning techniques are used to construct a "classifier" from the training materials, using those applicable to it as positive examples and the remainder as negative examples. If a new text is given, the classifier of each key-phrase will process it. The LSTM algorithm is used to classify key phrases in this research work. Our LSTM network has been trained with 5 types of data, i.e. sports, entertainment, technology, politics, and industry. The LSTM network classifies text from the categories listed.***

Keywords: Key phrases, Neural Networks, RNN, LSTM etc.

1. Introduction

Key phrases are a collection of terms that represent a document's main topic of interest. It plays critical roles in document description, text mining and web content retrieval. Since it is closely linked to a document, it represents the content of the document and serves as an example of the document in question. To understand the main contents of the text, it is important to extract the ideal key phrases.

Knowledge is now one of the most significant and powerful weapons in the modern world. We get an increasingly large amount of data or information from different sources, such as emails, web pages, electronic records, etc., every moment.But all the sources do not meet the readers' needs of the user as it is more difficult to locate the relevant details from a large amount of paper. For the very fast increasing content, it is very very difficult for a human being to find out the summery or extract the key topics from a wide body of text. Automatic extraction of keywords offers an accurate and efficient way for broad documents to be summarized. The terms that can quickly extract the main problems or subjects discussed in a text document are key phrases. It is very useful for efficiently classifying, clustering, and summarizing text documents.Keyword extraction allows the reader to discover a simple description of documents by extracting the most relevant words from a file. So, Keywords offers a description of a document that leads to an improved method of retrieving information. For several purposes, key phrases are essential and useful tools to remove.

For example, i) key phrases include a description that lets readers make faster decisions on whether the article is worth reading in-depth, ii) increase the efficiency of document indexing, iii) allow readers to quickly locate an article related to a particular topic or issue, and iv) enable a search engine to make the search more accurate for readers.**Recurrent**

Neural Network for Key phrase Extraction:
A recurrent neural network (RNN) is a type of artificial neural network used primarily for speech recognition and the processing of natural language (NLP). In deep learning and in the creation of models that mimic neuron activity in the human brain, RNN is used.

Recurrent networks are designed to identify patterns originating from sensors, financial markets, and government agencies in data sequences, such as text, genomes, handwriting, spoken word, and numerical time series data.

Except that a memory-state is added to the neurons, a recurrent neural network looks similar to a traditional neural network. A basic memory will be used in the computation.

A type of deep learning-oriented algorithm which follows a sequential approach is the recurrent neural network. We often presume, in neural networks, that each input and output is dependent on all other layers. These types of neural networks are referred to as recurring because they conduct mathematical computations sequentially.

The recurrent neural will perform the following.
First of all, the recurrent network transforms independent activations to dependent ones. It also assigns all layers the same weight and bias, which decreases the complexity of parameter RNNs. And it offers a standard medium for memorizing past outputs by supplying the next layer with the previous output as an input.

All three layers are combined into a single recurring unit with the same weights and biases.
For the current status measurement—
$\mathbf{h_t = f(h_{t-1}, X_t)}$
Where h_t= **current state**
H_{t-1}= **previous state**
X_t= **input state**

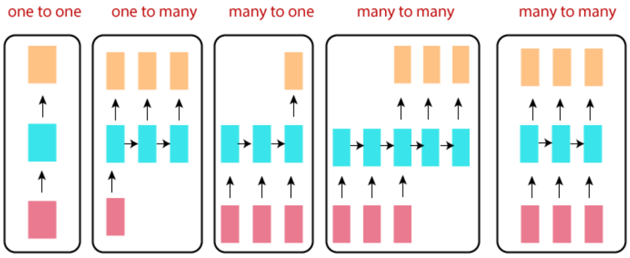
To apply the activation function tanh, we have-
$\mathbf{ht = tanh\ (W_{hh}h_t\text{-}1 + W_{xh}X_t)}$
Where:

W_{hh} = **weight of recurrent neuron and,**
W_{xh} = **weight of the input neuron**
The formula for calculating output:
$\mathbf{Y_t = W_{hy}h_t}$

Training through RNN

- A single time-step input is taken by the network.
- Using the current input and the previous state, we can determine the present state.
- Now, the present state for the next state by means of ht-1
- There are n numerous phases, and all the data can be joined in the end.
- After all the steps are completed, the final step is to measure the performance.
- Finally, the error is determined by measuring the difference between the real output and the estimated output.
- In order to change the weights and achieve a better result, the error is propagated back to the network.

Types of RNN
The key explanation for the more exciting recurrent networks is that they allow us to operate over vector sequences: input sequence, output sequence, or, in the most general case, both. Some examples may be more concrete:



In the picture above, every rectangle represents vectors, and arrows represent functions. The input vectors are red, the output vectors are blue, and the RNN state is kept in green.

One-to-one:
It is also known as **Simple Neural Networks**. This deals with a fixed input size to the fixed output size, where it is independent of previous information/output.
Example: Classifying pictures.

One-to-Many:
It deals with a fixed information size as an input and provides a data sequence as an output.
Example: Image Captioning takes the image as the input and generates a word expression.

Many-to-One:
It takes an information sequence as an input and outputs a fixed output size.

Example: analysis of feelings where any expression is categorized as reflecting a positive or negative feeling.

Many-to-Many:
It takes an information sequence as an input and outputs a fixed output size.
Example: analysis of feelings where any expression is categorized as reflecting a positive or negative feeling.

Bidirectional Many-to-Many:
Input and output synced sequence. Note that there are no pre-specified limits on the sequences of lengths in each case since the recurrent transformation (green) is fixed and can be applied as many times as we like. Example: Video classification in which we want to mark each video frame.

Advantages of Recurrent Neural Network

- A series of data can be modelled by RNN such that each sample can be presumed to be based on previous ones.
- In order to expand the active pixel neighbourhood, a recurrent neural network is also used with convolutional layers.

Long short-term memory (LSTM) RNN in Tensor flow
An artificial recurrent neural network (RNN) architecture used in the field of deep learning is long short-term memory (LSTM). **SeppHochreiter** and **Jurgenschmidhuber** suggested it in 1997. LSTM has feedback links, unlike normal feed-forward neural networks. Not only single data points (such as images), but even whole sequences of data can be processed (such as speech or video).

LSTM, for instance, is an application for tasks such as un-segmented, related **recognition of handwriting** or **recognition of expression.**

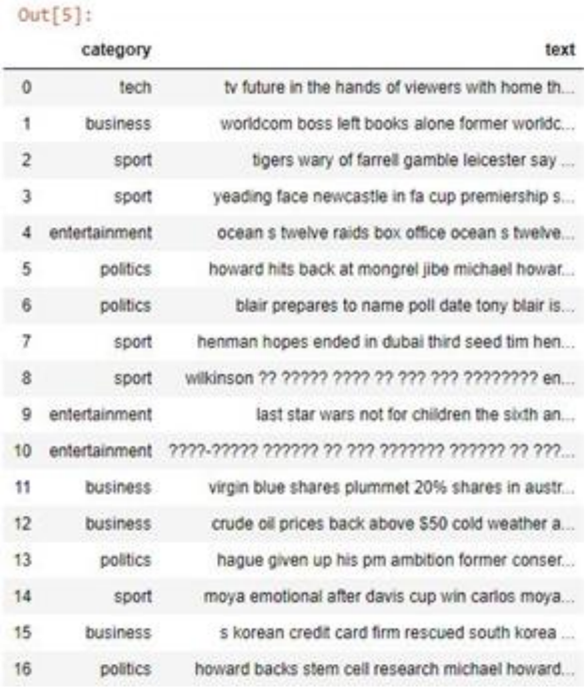
A general LSTM unit consists of a cell, an input gate, an output gate, and a gate to forget. Over arbitrary time periods, the cell remembers values, and the flow of information into and out of the cell is governed by three gates. LSTM is well suited for classifying, processing and predicting the uncertain length of the time series given.

2. Result and Discussion

This analysis uses the LSTM algorithm to classify key phrases. We trained our LSTM network in this study with 5 data types, i.e. sports, culture, tech, politics, and industry. The LSTM network classifies text from the categories listed.

We create and begin with an embedding layer with a tf.keras. Sequential model. One vector per word is stored by an embedding layer. It transforms sequences of word indices, when called, into sequences of vectors. Words with similar definitions also have similar vectors after practicing. For an LSTM layer, the Bidirectional wrapper is used, which propagates the input through the LSTM layer forwards and backwards and then concatenates the outputs.This helps LSTM to consider dependencies in the long term. To do classification, we then fit it to a dense neural network. In place of tahn function, we use relu because they are very good alternatives from each other.

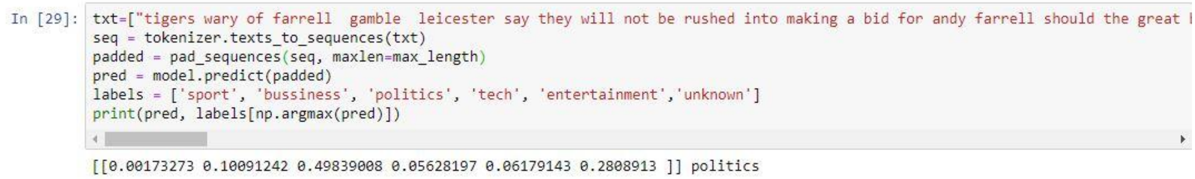
First read csv file



Out[5]:

	category	text
0	tech	tv future in the hands of viewers with home th...
1	business	worldcom boss left books alone former worldc...
2	sport	tigers wary of farrell gamble leicester say ...
3	sport	yeading face newcastle in fa cup premiership s...
4	entertainment	ocean s twelve raids box office ocean s twelve...
5	politics	howard hits back at mongrel jibe michael howar...
6	politics	blair prepares to name poll date tony blair is...
7	sport	henman hopes ended in dubai third seed tim hen...
8	sport	wilkinson ?? ????? ???? ?? ??? ??? ???????? en...
9	entertainment	last star wars not for children the sixth an...
10	entertainment	????-????? ?????? ?? ??? ??????? ?????? ?? ???...
11	business	virgin blue shares plummet 20% shares in austr...
12	business	crude oil prices back above $50 cold weather a...
13	politics	hague given up his pm ambition former conser...
14	sport	moya emotional after davis cup win carlos moya...
15	business	s korean credit card firm rescued south korea ...
16	politics	howard backs stem cell research michael howard...

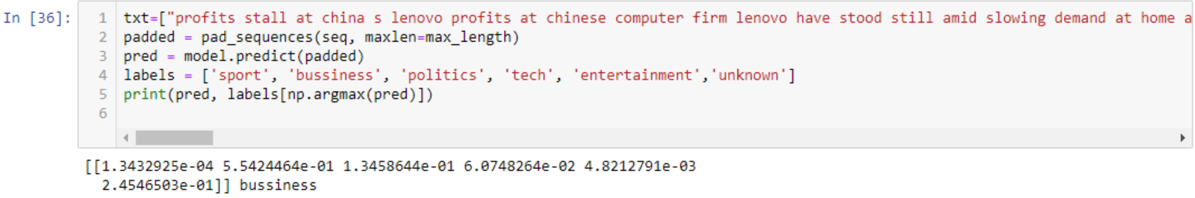
LSTM classify i.e the input data related to politics.



```
In [29]: txt=["tigers wary of farrell  gamble  leicester say they will not be rushed into making a bid for andy farrell should the great b
         seq = tokenizer.texts_to_sequences(txt)
         padded = pad_sequences(seq, maxlen=max_length)
         pred = model.predict(padded)
         labels = ['sport', 'bussiness', 'politics', 'tech', 'entertainment','unknown']
         print(pred, labels[np.argmax(pred)])

         [[0.00173273 0.10091242 0.49839008 0.05628197 0.06179143 0.2808913 ]] politics
```

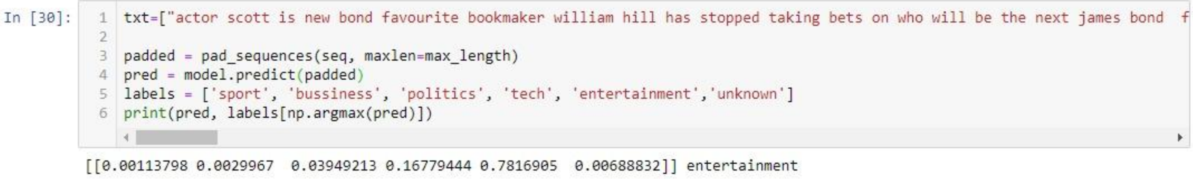
LSTM classify i.e the input data related to Bussiness.



```
In [36]: txt=["profits stall at china s lenovo profits at chinese computer firm lenovo have stood still amid slowing demand at home a
         padded = pad_sequences(seq, maxlen=max_length)
         pred = model.predict(padded)
         labels = ['sport', 'bussiness', 'politics', 'tech', 'entertainment','unknown']
         print(pred, labels[np.argmax(pred)])


         [[1.3432925e-04 5.5424464e-01 1.3458644e-01 6.0748264e-02 4.8212791e-03
           2.4546503e-01]] bussiness
```

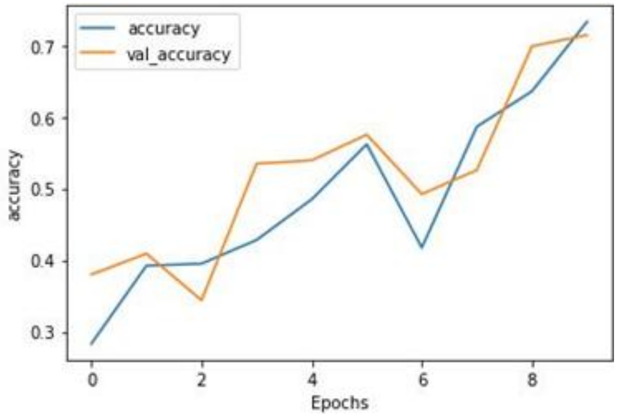
LSTM classify i.e the input data related to Entertainment.

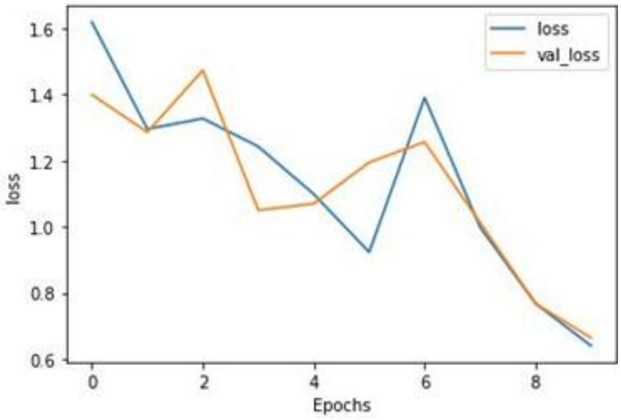


```
In [30]: txt=["actor scott is new bond favourite bookmaker william hill has stopped taking bets on who will be the next james bond  f

         padded = pad_sequences(seq, maxlen=max_length)
         pred = model.predict(padded)
         labels = ['sport', 'bussiness', 'politics', 'tech', 'entertainment','unknown']
         print(pred, labels[np.argmax(pred)])

         [[0.00113798 0.0029967  0.03949213 0.16779444 0.7816905  0.00688832]] entertainment
```





3. Conclusion

In recent years, the importance of the method of text summarisation has increased due to the large amount of data available on the Internet. It is possible to break text summarisation into extractive and abstractive approaches. A method of extractive text summarization produces a summary consisting of terms and phrases based on linguistics and statistical features from the original text, while a method of abstractive text summarization rephrases the original text to produce a summary consisting of new phrases. This analysis uses the LSTM algorithm to classify key phrases. We trained our LSTM network in this study with 5 data types, i.e. sports, culture, tech, politics, and industry.The LSTM network classifies text from the categories listed.

References

[1] D. Suleiman and A. A. Awajan, "Deep learning based extractive text summarization: approaches, datasets and evaluation measures," in Proceedings of the 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), pp. 204–210, Granada, Spain, 2019.View at: Publisher Site | Google Scholar

[2] Q. A. Al-Radaideh and D. Q. Bataineh, "A hybrid approach for Arabic text summarization using domain knowledge and genetic algorithms," Cognitive Computation, vol. 10, no. 4, pp. 651–669, 2018.View at: Publisher Site | Google Scholar

[3] Y. Ying, T. Qingping, X. Qinzheng, Z. Ping, and L. Panpan, "A Graph-based Approach of Automatic Keyphrase Extraction," in Procedia Computer Science, 2017, vol. 107, pp. 248–255.

[4] F. Xie, X. Wu, and X. Zhu, "Efficient sequential pattern mining with wildcards for keyphrase extraction," Knowledge-Based Syst., vol. 115, pp. 27–39, 2017.

[5] Q. Wang, V. S. Sheng, and X. Wu, "Document-specific keyphrase candidate search and ranking," Expert Syst. Appl., vol. 97, pp. 163–176, 2018.

[6] S. Štajner and G. Glavaš, "Leveraging event-based semantics for automated text simplification," Expert Syst. Appl., vol. 82, pp. 383–395, 2017.

[7] S. Siddiqi and A. Sharan, "Keyword and Keyphrase Extraction Techniques: A Literature Review," Int. J. Comput. Appl., vol. 109, no. 2, pp. 18–23, 2015.

[8] J. Rafiei-Asl and A. Nickabadi, "TSAKE: A topical and structural automatic keyphrase extractor," Appl. Soft Comput. J., vol. 58, pp. 620–630, 2017.

[9] E. Papagiannopoulou and G. Tsoumakas, "Local word vectors guiding keyphrase extraction," Inf. Process.Manag., vol. 54, no. 6, pp. 888–902, 2018.

[10] J. Hu, S. Li, Y. Yao, L. Yu, G. Yang, and J. Hu, "Patent Keyword Extraction Algorithm Based on Distributed Representation for Patent Classification," Entropy, vol. 20, no. 2, p. 104, 2018.

[11] Kathait, S.S., Tiwari, S., Varshney, A. and Sharma, A. "Unsupervised Key-phrase Extraction using Noun Phrases", International Journal of Computer Applications, 162(1), 2017.

[12] Gadag, Ashwini I., and B. M. Sagar. "N-gram based paraphrase generator from large text document", In Computation System and Information Technology for Sustainable Solutions (CSITSS), International Conference on, pp. 91-94. IEEE, 2016.

[13] Shirakawa, Masumi, Takahiro Hara, and ShojiroNishio. "N-gram idf: A global term weighting scheme based on information distance", In Proceedings of the 24th International Conference on World Wide Web, pp. 960- 970. International World Wide Web Conferences Steering Committee, 2015.

[14] Chatterjee, Niladri, and NehaKaushik. "RENT: Regular Expression and NLP-Based Term Extraction Scheme for Agricultural Domain", In Proceedings of the International Conference on Data Engineering and Communication Technology, pp. 511-522. Springer Singapore, 2017.

[15] Nesi, Paolo, Gianni Pantaleo, and GianmarcoSanesi. "A Distributed Framework for NLP-Based Keyword and Keyphrase Extraction From Web Pages and Documents", In DMS, pp. 155-161. 2015.

[16] Onan, Aytuğ, SerdarKorukoğlu, and HasanBulut. "Ensemble of keyword extraction methods and classifiers in text classification", Expert Systems with Applications 57 pp. 232-247, 2016.

[17] C. Sun, L. Lv, G. Tian, Q. Wang, X. Zhang, and L. Guo, "Leverage label and word embedding for semantic sparse web service discovery," Mathematical Problems in Engineering, vol. 2020, Article ID 5670215, 8 pages, 2020.

[18] E. Egonmwan and Y. Chali, "Transformer-based model for single documents neural summarization," in Proceedings of the 3rd Workshop on Neural Generation and Translation, pp. 70–79, Hong Kong, 2019.View at: Publisher Site | Google Scholar

[19] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," 2019, http://arxiv.org/abs/1908.08345.View at: Google Scholar