Generic Summary Generation to Produce a Highlighted Version of Documents

Juan E. Junco
Earlham College
Richmond, Indiana
Jejunco20@earlham.edu

ACM Reference Format:
Juan E. Junco. 2018. Generic Summary Generation to Produce a Highlighted Version of Documents. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

ABSTRACT

Identifying the most relevant text information from online texts has become a standard task now that data is easily accessible. Consequently, summary generation has gained significant importance by reducing the amount of text a person has to read to get meaningful information from the text. The underlying field for this process is Natural Language Understanding (NLU), a sub-field of Natural Language Processing (NLP). NLU enables algorithms to understand and rewrite a text based on the original information. In this proposal, I will explore how a Convolutional Neural Network (CNN) can create a new approach for visualizing the critical aspects of a text. Instead of generating a shorter text, the output will be a version with the essential elements highlighted.

KEYWORDS

Summary Generation, Natural Language Processing (NLP), Machine Learning (ML), Convolutional Neural Networks (CNN), Artificial Intelligence (AI).

1 INTRODUCTION

The definition of summary generation I will use in this paper resembles the definition stated by H. Lie, he refers to summary as a condensation of the main ideas in an article and defines it as a text reduced to its main points [9]. The massive number of digital texts on a single topic increases the difficulty of identifying relevant information. 'Summarization is important in some context to help people understand facts or to gain knowledge." [16]. Summary generation utilizes ML, statistical analysis, and NLP strategies [12]. In this paper, I will explore how using tools for summary generation along with a CNN allow the creation of new output. The work was inspired by Yadav et al. (2018) [8] and is guided by the research question, Is a CNN effective to generate a highlighted document version for users that identifies the most relevant parts of the text that convey the document's meaning?

The idea that key portions of a document can instead be highlighted has been present for quite some time. Among the motivations to do so, researchers argue that, highlights appear within their context (unlike a summary), and the impact of 'bad" highlights is of much lower consequence than 'bad" summaries [18]. The importance of the proposed software is that it highlights the most relevant sentences in a text. The software would allow the user to identify the fundamental ideas of a text using the author's words. Moreover, this approach avoids the process of sentence generation or reconstruction to generate the summary that is used in some of the work surveyed for this review. This approach uses the same methods of summary generation though a CNN of Yadav et al. and combines it with Spala et al. methods to highlight specific parts of the text. Currently, most methods for summary generation extract relevant sentences or words and categorize them based on statistical analysis, and finally reach the point of summary reconstruction. In addition, every relevant sentence is glued together using punctuation marks and connectors to build a coherent summary.

There are different types of summaries. According to Berger and Mittal [2], summarization is a field divided into two different ways of generating the desired extraction of information: Generic summaries and query-relevant summaries. This project will focus on the former type of summary. Generic summaries allow understanding of information without considering what specific pieces the reader might be hoping to extract from the article. In other words, it is much like 'the abstract in a paper, designed to distill salient points" [2]. The outcome of such analysis is a new text that semantically joins relevant aspects into a new text. This kind of summary allows users to understand the main points of the text independent of any query.

When presented with a summary, the user is faced with a choice: either rely on the summary or take the long approach and skim through the full text. As a solution, this study aims to address that limitation by successfully identifying the general ideas of a text and using them to generate a highlighted version of the text that spotlights the key points. The user can now identify the same information as in the generated summary but relying on the veracity of the words written by the author. The models will be evaluated using the accuracy Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scores as a measurement for relevance of text as suggested by the work by Baldwin et. al [1].

The remainder of this paper is organized into four sections. Section 2 provides a brief overview of the related work. Section 3 describes the design of the project. Finally, section 4 provides the timeline of the project next semester.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

2 BACKGROUND AND RELATED WORKS

This section introduces existing research on two widely-used methods for text summarization and similar work that also highlights significant aspects of a text. The summarization methods explored are the statistical approach and the convolutional neural networks (CNN). The work surveyed for this proposal generates a new summarized piece of text using one of those two methods, but very few generate a highlighted version. The generation of that document will be the objective in my senior capstone project.

In this paper, we will use generic summaries, but there is a direct contrast that generates a summary based on the user's query. 'User Focused Abstracts, i.e., abstracts relating information in the document to a particular user interest"[11]. The summary reflects the importance of individual pieces of the text that match the text query made by the user. Basically, it spotlights the most important aspects of the text based on what the user is looking for.

This section will briefly explore related work that attempted to highlight the most important aspects of a text. Then, explore the utilization of a NN and how efficient it is to generate summaries based on sentences from the text in a categorized way. Finally, statistical analysis has a broad range of operations to achieve a hierarchy of either words or sentences in the text. When the latter approach is implemented correctly, it has a similar and, in some cases, higher accuracy than using a NN.

2.1 Highlighting Text

Spala et al. implemented a survey study that presented two different sets of human candidates with the same text version. One group showed an unannotated text, and annotators were asked 'to highlight sentences that would make document comprehension easier and faster for another naive reader"[17]. The second group was presented with two already highlighted document versions. Participants had to vote on every highlight displayed on the document. Once they reached the document's end, they had to rate the two versions of the highlights. Spala et al. used the interaction of human users to come up with a correct highlighted version of documents. Spala et al. is an excellent example of how a highlighted version of a document is possible but has yet to be fully automatized.

In a second approach, Turney tested different benefits of extracting keyphrases from documents, among them the benefit of having a highlighted version of the text [19] . P. Turney treated a document as a set of phrases, which a learning algorithm learns to classify as positive or negative examples of key phrases. Turney's first set of experiments applied the C4.5 decision tree induction algorithm and the second set of experiments applied a GenEx algorithm specifically for this task. Highlighting was included by a direct comparison between the GenEx algorithm and Verity's Search 97 text retrieval system [6]. Verity's Search 97 produces a summary with highlighted key phrases embedded in the sentences. Turney utilized a search-based optimization technique based on Genetics and Natural Selection principles. This lies outside of all the other summarization techniques surveyed in this proposal; the aim of P. Turney was not specifically to produce readable documents for the user but test the importance of key phrases.

2.2 Statistical Approaches

Statistical approaches summarize a document using statistical features of the sentence, such as title, the location and, term frequency, assigning weights to keywords (keywords are words of the title in the text) and then calculating the score of the sentence and selecting the highest-scoring words into the summary [3].

Baldwin and Morton [1] implemented an information retrieval algorithm based on the probabilistic analysis. The main objective was to find all the possible sentences in the text until the query was 'covered," meaning that a sentence contains information related to the query. The method used two features, first finding the probability for a word in the document to match the word on the query. Therefore words were assigned a number based on how similar they were to words on the query. The words with the highest calculations were further analyzed with NLP techniques, including:

- Entity Recognition
- Tokenization
- Sentence Detection
- Part of Speech Tagging
- Morphological Analysis
- Parsing
- Argument Detection

Baldwin and Morton categorized the sentences using coreference chains, pieces of text with any common sub-sequence of words from the query. After categorizing sentences based on resemblance with conference chains, those with the highest probability were organized in ascending order and added to the final summary. The following query showcases the possible sentence selected from a text using their algorithm:

Query: What evidence is there of paramilitary activity in the U.S.?

Summary: Last month, the extremists used rocket-propelled grenades for the first time in three attacks on police and paramilitary units

Munot and Govilkar performed a comparative study using other statistical approaches [12]. In the study they show how 'abstractive text summarization method generates a sentence from a semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate. Such a summary might contain words not explicitly present in the original." Their software identifies items the original text and then, produces a new document using fewer words.

Software that uses NLP techniques has a very effective rate of summary (the appropriateness of the sentences it produces as output). In both articles mentioned above, the testing showed more than 90 percent accuracy with human generated summaries. Each sentence in the text is compared with the reference summary and measures the overlapping percentage of words between them. In terms of the implementation of such software, both articles classified sentences and use mathematical analysis to break down the measured categories.

2.3 Neural Networks

'An artificial neural network consists of an input layer of neurons (or nodes, units), hidden layers of neurons, and a final layer of output neurons" [20]. In summary generation, the input is a preprocessed version of the text that can be given to a neural network. A NN is a powerful pattern classification tool. Pattern classification is important to summary generation, as the association of related words allows the reduction in length of a text [14].

Kaiaikhah et al. [8] by creating different features and then manipulating the hidden layers to output a summarized version of the text. The point of uniqueness of this research lies in turning sentences into vectors. Each sentence was evaluated using seven different criteria. Each criterion became the input of the NN, meaning that the information had already normalized to some extent. Kaiaikhah et al. first used already categorized summaries to train the NN. The hidden layer calculations allowed the NN to decide which feature had the most impact on the summarization process by pruning. Every feature that contains the highest resemblance to the query is gathered together for the last stage of the process, the document reconstruction.

A second relevant example of using a NN to produce a summary was proposed by Sinha et al. [15] implemented a successful Neural Network (NN) model that generated very accurate summaries of online texts with an output accuracy of 95 percent compared to human-written summaries. The main point of their approach is to optimize news articles. News articles encapsulate the topic in the title, but more than the title is usually needed to understate the story's context. The NN requires a numerical representation of the input to perform calculations. Sinha et al. used a vector representation of words. They fed the sentences as input to the word2vec [4] model that provides vector representation for words of the English language. Once the calculations are done, there is a final activation function in the output layer that allows the NN to gather sentences with the highest measure of belief to be included in the summary, in other words the highest numbers are belived to be added to the summary. Finally, after extracting the relevant points, the evaluation method to test the accuracy that Sinha et al. used was ROUGE scores. This technique compares a human-generated text summary with the output summary of the NN using n-grams. The summaries had an average accuracy score of 95 percent compared to human-performed summaries. A second test is related to the length of the summary. It should contain fewer sentences than the original text. The summary length in terms of the number of sentences is fixed and known before summary generation.

3 DESIGN AND IMPLEMENTATION

This project focuses on generating an annotated version of a document based on the most relevant ideas contained. The main point would be the generation of a new output that could potentially be useful for the average user. Moreover, this study focuses on extraction of sentences from the text insted of any further analysis that could match the words in a query with the expected output. In this section I will discussed the Data set to be used, the model to implement and finally, the evaluation techniques for the capstone project.

3.1 Research Data

TREC (Text Research Collection Volume) [7] has been used across several papers that deal with text summaries. TREC is an effort to advance the state of the art in effective document detection (information retrieval) and data extraction from large, real-world data collections. The content is divided between three different disks. The first disk contains material from the Wall Street Journal, (1986, 1987, 1988, 1989), the AP Newswire (1989), the Federal Register (1989). The second disk contains information from the same sources, but from different years. The third disk contains more information from the Computer related articles, plus material from the San Jose Mercury News (1991), more AP newswire (1990) and about 250 megabytes of formatted U.S. Patents.

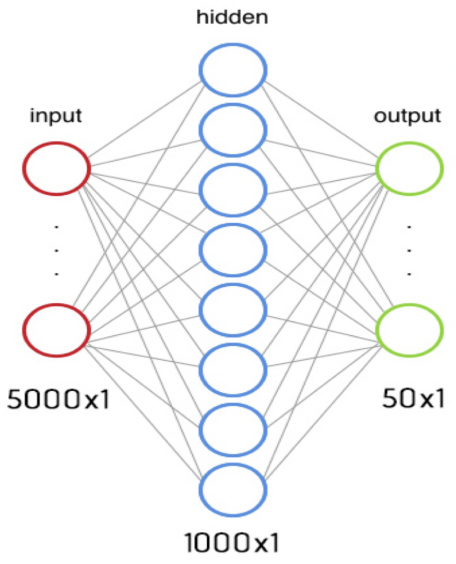

Figure 1: Neural Network

Neto et al. used the TREC Collection in a NN approach [13]. In their paper they performed two series of experiments, first implementing a Neural Network and second implementing statistical grouping of words. For both cases they used the same source of data from the TREC project, testing their approaches using material from the Wall Street Journal [9]. Later in the paper they go into detail about other possible sources of information they plan to use to further test their proposed approaches, one of them being magazines about computers.

3.2 Summary Generation Model

This section will detail a step by step procedure to follow in order to produce a new type of summary.

3.2.1 Gather information. The first step is digging deeper into the TREC library's disks and identifying how to extract documents. I will look for the most recent disks and extract information. This project will use the TREC library, a method already used by other summarization researchers like Neto et al. This is a crucial step as it will be the skeleton of the whole research and the input for the NN.

3.2.2 Training and Pruning. Building a Neural Network could be quite a challenging task from scratch. Some software opens the

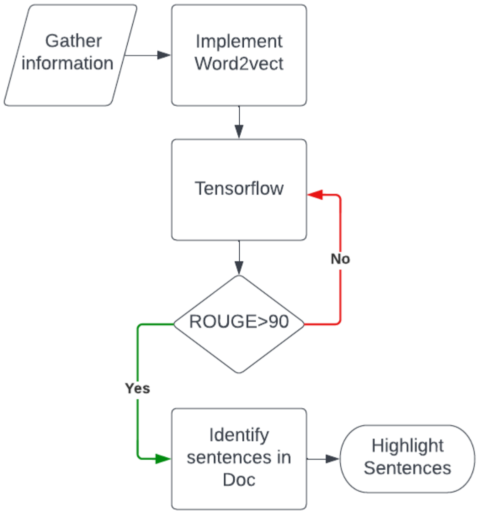

Figure 2: Framework of the Project

possibilities of deep learning by showing a visual representation of the underlying operations of a Neural Network. To create the CNN that will process the data, I will use the dependency of word2vec in TensorFlow. TensorFlow is an open-source deep learning software library for defining, training, and deploying machine learning models [5]. This choice is due to the significant benefits of representing an algorithm in a graph. TensorFlow allows the definition of nodes as a representation of operations.

This process will replicate the methods used by Kaiaikhah et al. [8] by creating different features and then manipulating the hidden layers to output hierarchized sentences of the text in question. Even if a NN would be able to accomplish the task with sufficient cleaning of the initial data, a CNN will allow software to independently translate words into useful numeric data. the This will be the step I will do rather soon, as it will require the implementation of a NN using a software that might need time debugging. The second stage will be using the ROUGE scores dependency to test the accuracy and train it until the success rate outputs similar results to Kaiaikhah et al. The main difference would be the implementation of word2vec as the way for the CNN to transform data into numeric values and the way to display the output, instead of a generated text a highlighted version of the text.

3.2.3 Highlighted Version. Once all the sentences have passed all the different tests and are usually ready for the text reconstruction part of the process, I will take those sentences and loop through the text while assigning each sentence an ID number. Once one of the IDs matches the text produced by the NN, highlight it. For this, I will have the original document as input and read it line by line, comparing it with an array containing all expected sentences. Once there is a match, I will use the library 'termcolor" from Python to highlight that portion. Once the software parses the document, I will return the new highlighted version.

3.3 Evaluation methods

A significant portion of the Neural Net based summaries used testing associated with the accuracy of a generated summary is done by using already annotated version of documents and then compare the extracted items of both. For example, Kaiaikhah [15] "selected 25 different news articles. The human reader and all three modified networks summarized the 25 news articles, independently." The average accuracy of the discretized real-values into intervals neural network NJ was 96 percent summaries." The second most common approach was given by the ROUGE scores. ROUGE stands for Recall-Oriented Understudy for Gisting Evalua- tion. It includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. The measures count the number of overlapping units such as n-gram, word sequences, and word pairs between the computer- generated summary to be evaluated and the ideal summaries created by humans [10]. ROUGE scores will be based on n-grams, avoiding human involvement and using a widely accepted approach.

4 TIMELINE

Date	Work
Winter Break	Data collection from TREC
Week 1	Learn word2vec
Week 2	Implement NN
Week 3	Implement NN
Week 4	Implement NN and test Python code
Week 5	ROUGE Test
Week 6	Debugging
Week 7	End to end output
Week 8	Use untrained Data
Week 9	Second Draft
Week 10	Record Video
Week 11	Work on poster
Week 12	Third Draft
Week 13	Demonstration

5 ACKNOWLEDGEMENTS

I would like to thank David Barbella and Sofia Lemons for providing me detailed feedback to improve this topic proposal paper.

REFERENCES

[1] Breck Baldwin and Thomas S. Morton. 1998. Dynamic coreference-based summarization. In *Proceedings of the third conference on empirical methods for natural language processing*. 1–6.
[2] Adam Berger and Vibhu O Mittal. 2000. Query-relevant summarization using FAQs. In *Proceedings of the 38th annual meeting of the association for computational linguistics*. 294–301.
[3] Saeedeh Gholamrezazadeh, Mohsen Amini Salehi, and Bahareh Gholamzadeh. 2009. A comprehensive survey on text summarization systems. In *2009 2nd International Conference on Computer Science and its Applications*. IEEE, 1–6.
[4] Yoav Goldberg and Omer Levy. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* (2014).
[5] Peter Goldsborough. 2016. A tour of tensorflow. *arXiv preprint arXiv:1610.01178* (2016).
[6] Ed Gordon. 1996. Verity agent technology: Automatic filtering, matching and dissemination of information. *Vine* (1996).
[7] Donna Harman and Mark Liberman. 2012. Tipster complete. https://catalog.ldc.upenn.edu/LDC93T3A
[8] Khosrow Kaikhah. 2004. Automatic text summarization with neural networks. In *2004 2nd International IEEE Conference on'Intelligent Systems'. Proceedings (IEEE Cat. No. 04EX791)*, Vol. 1. IEEE, 40–44.

[9] Danny H Lie. 1998. Sumatra: a system for automatic summary generation. *Carp Technologies* (1998).

[10] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013

[11] Inderjeet Mani. 2001. *Automatic summarization*. Vol. 3. John Benjamins Publishing.

[12] Nikita Munot and Sharvari S Govilkar. 2014. Comparative study of text summarization methods. *International Journal of Computer Applications* 102, 12 (2014).

[13] Joel Larocca Neto, Alex A Freitas, and Celso AA Kaestner. 2002. Automatic text summarization using a machine learning approach. In *Brazilian symposium on artificial intelligence*. Springer, 205–215.

[14] Phil Picton. 1994. What is a neural network? In *Introduction to Neural Networks*. Springer, 1–12.

[15] Aakash Sinha, Abhishek Yadav, and Akshay Gahlot. 2018. Extractive text summarization using neural networks. *arXiv preprint arXiv:1802.10137* (2018).

[16] Yong SP, Ahmad IZ Abidin, and YY Chen. 2006. A neural-based text summarization system. *WIT Transactions on Information and Communication Technologies* 37 (2006).

[17] Sasha Spala, Franck Dernoncourt, Walter Chang, and Carl Dockhorn. 2018. A Comparison Study of Human Evaluated Automated Highlighting Systems. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*.

[18] Sasha Spala, Franck Dernoncourt, Walter Chang, and Carl Dockhorn. 2018. A Web-based Framework for Collecting and Assessing Highlighted Sentences in a Document. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Santa Fe, New Mexico, 78–81. https://aclanthology.org/C18-2017

[19] Peter D. Turney. 2002. Learning to Extract Keyphrases from Text. https://doi.org/10.48550/ARXIV.CS/0212013

[20] Sun-Chong Wang. 2003. Artificial neural network. In *Interdisciplinary computing in java programming*. Springer, 81–100.