Implementation of Extractive Text Summarization using Word Frequency in Python

Ahmad Farhan Al Shammari
Department of Computer and Information Systems
College of Business Studies, PAAET
Kuwait

ABSTRACT
The goal of this research is to develop an extractive text summarization program using word frequency in Python. The steps of text summarization process are: preprocessing text, word-tokenization, creating bag of words, calculating word frequency, sentence-tokenization, calculating sentence score, calculating average score, and making summary. The developed program was examined on an experimental text from Wikipedia. The program performed the steps of text summarization and provided the required summary.

Keywords
Artificial Intelligence, Machine Learning,Text Summarization, Natural Language Processing, Tokenization, Word Frequency, Sentence Score, Summary, Python.

1. INTRODUCTION
With the great advances in information technology and internet applications, the amount of data is growing very fast. Processing data is more becoming a crucial job. More efficient methods are strongly needed to process data, analyze it, and extract information from it.

Machine Learning (ML) is playing a big role in solving this problem. ML is a branch of Artificial Intelligence (AI) that studies computer algorithms to improve the performance of computer applications.

One of the important applications in ML is text summarization. It is a common field between ML and Natural Language Processing (NLP), where it applies the algorithms of ML and the methods of NLP to process human language.

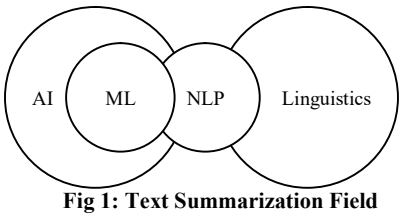

Fig 1: Text Summarization Field

2. LIREATURE REVIEW
The research in text summarization started in the late fifties. The first paper was published in 1958 by Luhn [2]. He applieda frequency-based method to automatically summarize technical papers and magazine articles. He used"word frequency"tomeasure theimportanceof wordsand sentences in the text. The sentences ofhigh scores are extracted to make the summary.

Reviewing literature of text summarization shows that research based on word frequency is one of the most widely used approaches in text summarization (Lloret& Palomar [1], and Nenkova&McKeown [4]).

Many researchers likeSparck Jones [10], Salton & McGill [11], and Salton & Buckley [12] have developeddifferent weighting methodsbased on word frequency. One of the famous methods is Term Frequency-Inverse Document Frequency (TF-IDF) which is the most widely used method in search engines, digital libraries, and information retrieval systems.

Orasan[5]accomplished a comparative study between different weighting methods. He found thatweighting methods based on word frequencycan provide informative summary.

The fundamental concepts of text summarization are explained in the following section:

Text Summarization:
Text summarization isa process that provides a brief and useful summary from the text. The purpose of text summarization is to save the time of reading by reducing the size of text.

Types of Text Summarization:
Text summarization is divided into two main types: Extractive and Abstractive.

In the extractive type:the summary consists of sentences that are "selected" from the text. It is the traditional approachthatis used more because of its simplicity.

In the abstractive type:the summary consists of sentences that are "generated" from the text. It is used less because of its complexity.

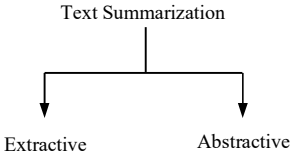

Fig2: Types of Text Summarization

In this research, the extractive approach is used.

Preprocessing Text
The raw text should be "cleaned" from the unwanted characters or wordsfor examplepunctuationcharacters andstopwords.

Tokenization
Tokenization is the process of breaking thetext into smaller units or "tokens" such as words and sentences.

The resulting words and sentences are processed using someweighting methods, such as word frequency and sentence score.

Bag of Words

Bag of words (BoW) is the set of important words in the text, without repetition.

Word Frequency:

Word frequency is the number of times a word occurs in the textdivided by the number of words in the text.It is calculated by the following formula:

$$Freq(w_i) = \frac{f(w_i)}{Nw} \qquad (1)$$

Where:$f(w_i)$ is the number of times the word (w_i) occurs in the text, and Nw is the total number of words in the text.

Sentence Score:

Sentence scoreis the sum of word frequenciesfor the words in the sentencedivided by the number of words in the sentence. It is calculated by the following formula:

$$Score(s_j) = \frac{\sum Freq(w_i|s_j)}{Nw_i|s_j} \qquad (2)$$

Where:$\sum Freq(w_i|s_j)$is the sum of word frequenciesforthe words in the sentence (s_j), and $Nw_i|s_j$ is the number of words in the sentence (s_j).

Python:

Python [9] is a general high-level programming language. It is simple, easy to learn, and powerful. It is the best choice for many programmers, especially in the field of machine learning.

Python provides additional libraries for processing numbers, plots, texts, and images, such as: numpy [6], pandas [8], matplotlib [3], NLTK [5], and scikit [13].

In thisresearch,the standard functions of Python are only usedwithout any additionallibrary.

3. RESEARCH METHODOLOGY

Extractive text summarization is done by the following steps:(1) preprocessingtext, (2) word-tokenization, (3) creating bag of words, (4) calculating wordfrequency,(5)sentence-tokenization, (6) calculating sentence score, (7) calculating average score, and (8) making summary.

1. Preprocessing Text
2. Word-Tokenization
3. Creating Bag of Words
4. Calculating Word Frequency
5. Sentence-Tokenization
6. Calculating Sentence Score
7. Calculating Average Score
8. Making Summary

Fig 3: Steps of Extractive Text Summarization

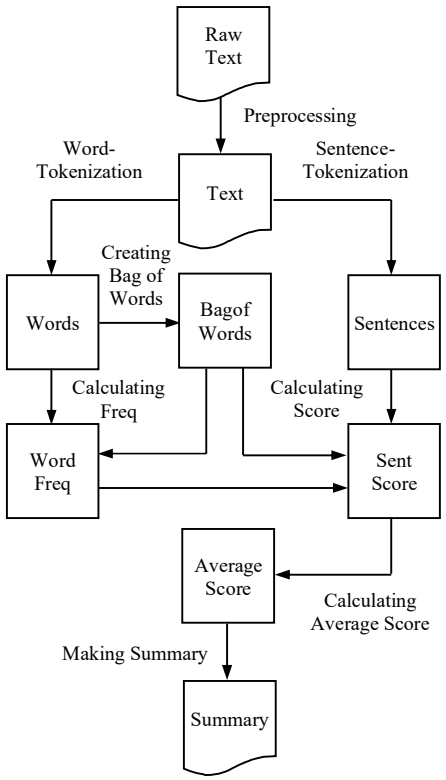

Fig4: Flowchart of Extractive Text Summarization

The steps of text summarization are explained in the following section:

1.Preprocessing Text:

The text is preprocessed to clean it from the unwanted characters and words. It is done bythe following steps:

1.1 Converting Text into Lower Case:

The text is converted into lower case form. It is done by the following code:

```
raw_text = "..."
text = raw_text.lower()
```

1.2 Removing Punctuation Characters:

Punctuation characters are removed from the text. It is done by the following code:

```
letters = "abcdefghijklmnopqrstuvwxyz"
for c in text:
if (c not in letters):
       text = text.replace(c,"")
```

1.3InitializingStopwords:

Stopwordsare words that have no importance in the text. For example: in English language, they are about (150) words. The stopwordslist isinitializedby the following code:

```
stopwords = ["i", "we", "am", "he", "she",
"it", "is", "are","the",
             "they", "that", "this",... ]
```

2. Word-Tokenization:

The text is split into words. It is done by the following code:

```
words = []
for word in text.split()
    if (word not in stopwords):
        words.append(word)
```

3. Creating Bag of Words:

Bag of words is the set of important words without repetition. It is created by the following code:

```
bag_of_words = set(words)
```

4. Calculating Word Frequency:

The word frequency holds the frequencies of the words.

Word	Frequency
w_1	*freq*(w_1)
w_2	*freq*(w_2)
w_3	*freq*(w_3)
…	…
w_n	*freq*(w_n)

Fig 5: Word Frequency

Where:*freq*(w_i) is the frequency of word (w_i). It is calculated by the following code:

```
Nw = len(words)
freq = {}
for word in bag_of_words:
    freq[word] = words.count(word)/Nw
```

5. Sentence-Tokenization:

The text is split into sentences. It is done by the following code:

```
sentences = []
for sent in text.split("."):
    sent = sent.replace(".","")
    if (sent != ""):
        sentences.append(sent)
```

6. Calculating Sentence Score:

The sentence score holds the scores of the sentences.

Sent	Score
s_1	*score*(s_1)
s_2	*score* (s_2)
s_3	*score* (s_3)
…	…
s_m	*score* (s_m)

Fig6: Sentence Score

Where:*score*(s_j) is the score of sentence (s_j). Itis calculated by the following code:

```
score = {}
for sent in sentences:
    Nws = 0
    sum = 0
    for word in sent.split():
        if (word not instopwords):
            sum += freq[word]
            Nws += 1
        score[sent] = sum/Nws
```

7. Calculating Average Score:

The average score is the average value of the sentence scores. It is calculated by the following code:

```
Ns = len(score)
sum = 0
for sent in score:
    sum += score[sent]
average = sum/Ns
```

8. Making Summary:

The sentences are "filtered"according to the averagescore. If the sentence score is above the average score, then the sentence is added to the summary. The summary is created by the following code:

```
summary = ""
for sent in score:
    if (score[sent] >= average):
                summary += sent + "."
```

4. RESULTS AND DISCUSSION

Thedeveloped program was tested on an experimental text from Wikipedia [7]. The program performed the steps of text summarization and provided the required results. The program output is shown in the following section:

Words:

The important words are extracted from the text. They are shownin the followingview:

```
Words:
0 abstract
1 abstraction
2 abstraction
3 abstractive
4 abstractive
5 abstractive
...
```

Bag of Words:

The bag of words is created from the list of words. It is shown in the following view:

```
Bag of Words:
0 abstract
1 abstraction
2 abstractive
3 addition
4 analogous
5 applied
...
```

Word Frequency:

The word frequency is calculated for all words. It is shown inthe following view:

```
Word Frequency:
0 abstract        : 0.004878048780487805
1 abstraction     : 0.00975609756097561
2 abstractive     : 0.014634146341463415
3 addition        : 0.004878048780487805
4 analogous       : 0.004878048780487805
5 applied         : 0.004878048780487805
...
```

Sentences:

The sentences are extracted from the text. They are shown in the following view:

```
Sentences:
0 :text summarization is the process ...
1 :in addition to text images and ...
2 :text summarization finds the most ...
3 :there are two general approaches ...
4 :extraction based summarization ...
5 : examples of extracted content ...
...
```

Sentence Score:

The sentencescore is calculated for allsentences. It is shownin the following view:

```
Sentence Score:
0 :0.26341463414634153
1 :0.0878048780487805
2 :0.45853658536585384
3 :0.15609756097560978
4 :0.23902439024390243
5 : 0.3121951219512196
...
```

Average Score:

The average score is calculated from the sentence score. It is shown in the following view:

```
Average Score = 0.22195121951219518
```

Summary:

The summary consists of the important sentenceswhich have scorevaluesabove the average score. It is shown in the following view:

```
Summary:
text summarization finds the most informative sentences in a document various methods of image summarization are the subject of ongoing research with some looking to display the most representative images from a given collection or generating a video,video summarization extracts the most important frames from the video content.
examples of extracted content include key phrases that can be used to tag or index a text document or key sentences including headings that collectively comprise an abstract and representative images or video segments as stated above.
abstractive based summarization abstractive summarization methods generate new text that did not exist in the original text.
```

5. CONCLUSION

Text summarization is an important field of machine learning. The purpose of text summarization is to provide a short and useful summary of the text. The importantsentences are extracted from the text to make the summary.

In this paper, the researcher developed an extractivetext summarization programusing word frequency in Python. The developed program performed the steps of text summarization: preprocessing text, word-tokenization, creating bag of words, calculating word frequency, sentence-tokenization, calculating sentencescore, calculating average score, and making summary.

The program successfully provided the required results: words, bag of words,word frequency, sentences, sentence score, average score, and summary.

In the future, more research will be done to examine the different methods of extractive text summarization in othercultures and languages for example Arabic.

6. REFERENCES

[1] Lloret, E. & Palomar, M. (2012). "Text Summarization in Progress: A Literature Review". Artificial Intelligence Review. 37, 1-41.

[2] Luhn, H. P. (1958). "The Automatic Creation of Literature Abstracts". IBM Journal of research and development, 2(2), 159-165.

[3] Matplotlib: https://www. matplotlib.org

[4] Nenkova, A., &McKeown, K. (2012). "A Survey of Text Summarization Techniques". In Mining Text Data, Springer, 43-76.

[5] NLTK: https://www.nltk.org

[6] Numpy: https://www.numpy.org

[7] Orasan, Constantin. (2009). "Comparative Evaluation of Term-Weighting Methods for Automatic Summarization". Journal of Quantitative Linguistics. 16, 67-95.

[8] Pandas: https:// pandas.pydata.org

[9] Python: https://www.python.org

[10] Salton, G. (1989). "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer". Addison- Wesley Publishing Company, USA.

[11] Salton, G. & McGill, M. (1983). "Introduction to Modern Information Retrieval". McGraw Hill Book Co, New York.

[12] Salton, G., & Buckley, C. (1988). "Term-Weighting approaches in Automatic Text Retrieval". Information Processing and Management, 24(5), 513-523.

[13] SciKit: https://scikit-learn.org

[14] Sparck Jones, K. (1972). "A Statistical interpretation of Term Specificity and its Application in Retrieval". Journal of Documentation, 28(1), 11-21.

[15] Wikipedia: https://en.wikipedia.org