Implementation of Text Summarization using Word Frequency in Python

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

ABSTRACT
The goal of this research is to develop a text summarization program using word frequency in Python. The purpose of text summarization is to provide a short and useful summary of the text. The word frequency is used to measure the importance of words and sentences in the text. The basic steps of text summarization are explained: preprocessing text, creating list of words, creating bag of words, creating word frequency, creating list of sentences, creating sentence score, sorting sentence score, calculating average score, and making summary. The developed program was tested on an experimental text from Wikipedia. The program successfully performed the basic steps of text summarization and provided the required results.

Keywords
Artificial Intelligence, Machine Learning, Natural Language Processing, Text Mining, Text Summarization, Word Frequency, Python, Programming.

1. INTRODUCTION
The rapid development of Information and Communications Technology (ICT) is enabling the volume of data to grow very fast. Processing large amounts of data is becoming a crucial issue. Computer systems need more powerful methods to process data, analyze it, and extract information. Actually, machine learning is playing a key role in processing data more quickly and efficiently.

Machine Learning (ML) is a branch of Artificial Intelligence (AI) which is focused on the study of computer algorithms to improve the performance of computer programs.

Text summarization is one of the important applications of machine learning. It is a common field between ML and Natural Language Processing (NLP). Therefore, it applies both the methods of ML and the techniques of NLP to process human language.

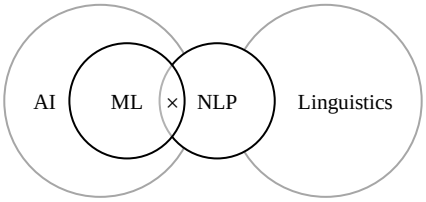

Fig 1: Field of Text Summarization

2. LIREATURE REVIEW
The review of literature revealed the major contributions in the field of text summarization [1-19]. The research started in the late fifties. In 1958, Hans Luhn published the first paper in text summarization [20] to automatically summarize technical papers and magazine articles. He used the "word frequency" to measure the importance of words and sentences in the text. Then, the sentences of high scores are selected and added to the summary.

The word frequency is a simple and powerful weighting method. It has been used widely in the research of text summarization [6, 7].

Orasan [4] conducted a survey to compare between the different weighting methods. He found that programs based on word frequency can provide an informative summary.

Gerard Salton [21] introduced the "Vector Space Model" (VSM) to represent text as a vector of numbers or weights, as shown in the following view:

$$Text = (weight_1, weight_2, \dots , weight_n)$$

Over time, researchers continued to develop new weighting methods based on word frequency. Salton and others developed various formulas for different purposes [22-26].

Sparck Jones [27, 28] suggested the "Term Specificity" to overcome the limitation of word frequency. This led to the development of TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF is the most widely used method in many applications like search engines and digital libraries.

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

Text Summarization:
Text summarization is the process that provides a short and meaningful summary of the text. It helps to reduce the size of text and save the time of reading.

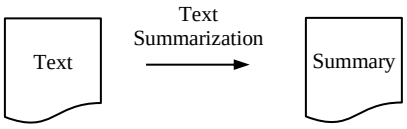

Fig 2: Concept of Text Summarization

Types of Text Summarization:
The types of text summarization are divided into two main groups: Extractive and Abstractive.

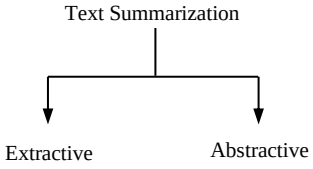

Fig 3: Types of Text Summarization

In the extractive type: the summary consists of the original sentences that are selected from the text. It is the traditional approach and is more frequently used because of its simplicity.

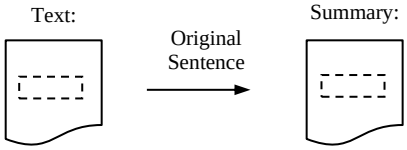

Fig 4: Extractive Text Summarization

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

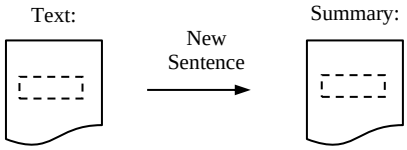

Fig 5: Abstractive Text Summarization

In this research, the extractive approach is applied.

Text Summarization System:

In the text summarization system; the input is the text given by the user. Then, the system will process the text, calculate the weights of words based on their frequency in the text, and calculate the sentence scores. The output is the summary.

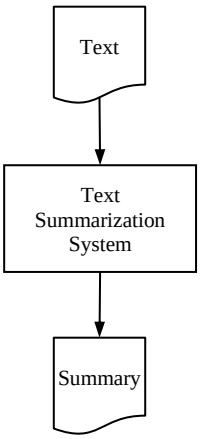

Fig 6: Diagram of Text Summarization System

Preprocessing Text:

The text should be "cleaned" from the unwanted characters and words, for example: punctuation symbols and stopwords.

List of Words:

The text is split into words as shown in the following view:

$$List\ of\ Words = [\ word_1, word_2, \dots , word_n\ ]$$

Bag of Words:

The bag of words (BoW) is the set of words without repetition as shown in the following view:

$$Bag\ of\ Words = (\ word_1, word_2, \dots , word_m\ )$$

Word Frequency:

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

$$freq(w_i) = \frac{Nw_i}{Nw} \qquad (1)$$

Where: Nw_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.

List of Sentences:

The text is split into sentences, as shown in the following view:

$$List\ of\ Sentences = [\ sentence_1, sentence_2, \dots , sentences_k\ ]$$

Sentence Score:

Sentence score is the sum of word frequencies for the words in the sentence divided 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 frequencies of the words in the sentence (s_j), and $Nw_i|s_j$ is the number of words in the sentence (s_j).

Python:

Python [29] is a general high-level programming language. It is simple, easy to learn, and powerful. It is the most preferred programming language by the developers of machine learning applications.

Python provides additional libraries like: Numpy [30], Pandas [31], Matplotlib [32], NLTK [33], and SK Learn [34].

In this research, the standard functions of Python are applied without using any additional library.

3. RESEARCH METHODOLOGY

The basic steps of text summarization are: (1) preprocessing text, (2) creating list of words, (3) creating bag of words, (4) creating word frequency, (5) creating list of sentences, (6) creating sentence score, (7) sorting sentence score, (8) calculating average score, and (9) making summary.

1. Preprocessing Text
2. Creating List of Words
3. Creating Bag of Words
4. Creating Word Frequency
5. Creating List of Sentences
6. Creating Sentence Score
7. Sorting Sentence Score
8. Calculating Average Score
9. Making Summary

Fig 7: Steps of Text Summarization

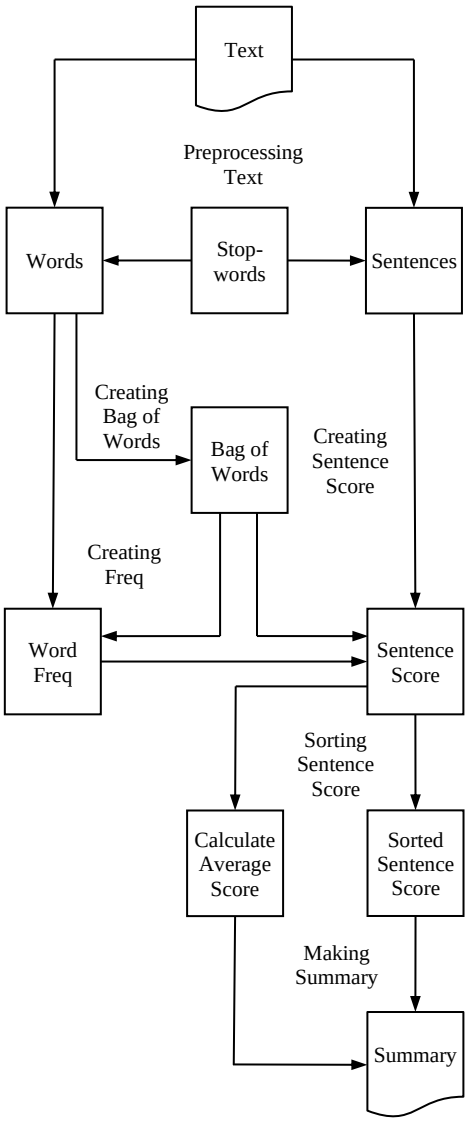

Fig 8: Flowchart of Text Summarization

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

1.Preprocessing Text:

The text is cleaned from the unwanted characters and words. It is done by the following steps:

1.1 Converting Text into Lower Case:

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

```
text = text.lower()
```

1.2 Removing Punctuation:

Punctuation symbols (like: !@#$ …) 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," ")
```

1.3 Removing Stopwords:

The stopwords (like: I, am, is, are, …) are removed from the text. It is done by the following code:

```
stopwords = ["i", "am", "is", "are", "we",
             "he", "she", "it", "the",
             "this", "they", "that", ... ]
for word in text:
    if (word in stopwords):
        text = text.replace(word,"")
```

2. CREATING LIST OF WORDS:

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

```
words = text.split()
```

3. CREATING BAG OF WORDS:

The bag of words is the set of words. It is done by the following code:

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

4. CREATING WORD FREQUENCY:

The word frequency holds the frequencies of words.

Word	Frequency
w_1	$freq(w_1)$
w_2	$freq\ (w_2)$
w_3	$freq\ (w_3)$
…	…
w_n	$freq(w_n)$

Fig 9: Structure of Word Frequency

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

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

5. CREATING LIST OF SENTENCES:

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

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

6. CREATING SENTENCE SCORE:

The sentence score holds the scores of sentences.

Sent	Score
s_1	$score(s_1)$
s_2	$score\ (s_2)$
s_3	$score\ (s_3)$
…	…
s_m	$score\ (s_m)$

Fig 10: Structure of Sentence Score

Where: $score(s_j)$ is the score of sentence (s_j). It is done by the following code:

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

7. SORTING SENTENCE SCORE:

The sentence score is sorted in reverse order by the score value. In Python, sorting a list is simply done by using the (sorted) function as shown in the following code:

```
sorted_list = sorted(list, reverse=True)
```

However, sorting a dictionary is more complicated than a list because the structure of dictionary is composed of paired (key, value) items.

8. CALCULATING AVERAGE SCORE:

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

```
Ns = len(score)
sum = 0
for sent, value in score.items():
    sum += value
average = sum / Ns
```

9. MAKING SUMMARY:

The summary consists of the sentences that have scores above the average score. They are selected and added to the summary in their natural order. It is done by the following code:

```
summary = ""
for sent, value in score.items():
    if (value >= average):
        summary += sent + ". "
```

4. RESULTS AND DISCUSSION

The developed program was tested on an experimental text from Wikipedia [35]. The program performed the basic steps of text summarization and provided the required results. The program output is shown in the following section:

List of Words:

The list of words is shown in the following view:

```
List of Words:
        abstract
        abstraction
        abstraction
        abstractive
        abstractive
        ...
```

Bag of Words:

The bag of words is shown in the following view:

```
Bag of Words:
        abstract
        abstraction
        abstractive
        addition
        analogous
        ...
```

WORD FREQUENCY:

The word frequency is shown in the following view:

```
Word Frequency:
        abstract        0.0048780488
        abstraction     0.0097560976
        abstractive     0.0146341463
        addition        0.0048780488
        analogous       0.0048780488
        ...
```

List of Sentences:

The list of sentences is shown in the following view:

```
List of Sentences:
        1       text summarization is the ...
        2       in addition to text image ...
        3       text summarization finds  ...
        4       there are two general app ...
        5       extraction based summariz ...
        6       examples of extracted con ...
        7       for text extraction is an ...
        8       other examples of extract ...
        9       abstractive based summary ...
        10      this has been applied mai ...
        ...
```

Sentence Score:

The sentence score is shown in the following view:

```
Sentence Score:
        1       0.263414634
        2       0.087804878
        3       0.458536585
        4       0.156097561
        5       0.23902439
        6       0.312195122
        7       0.282926829
        8       0.170731707
        9       0.302439024
        10      0.068292683
        ...
```

The following chart shows a visual representation of the sentence score:

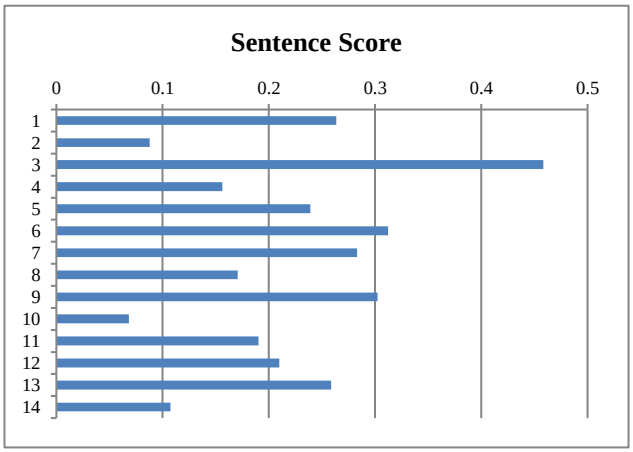

Fig 11: Chart of Sentence Score

Average Score:

The average score is shown in the following view:

```
Average Score = 0.2219512195
```

Sorted Sentence Score:

The sorted sentence score is shown in the following view:

```
Sorted Sentence Score:
        3               0.458536585
        6               0.312195122
        9               0.302439024
        7               0.282926829
        1               0.263414634
        13              0.258536585
        5               0.239024390
        12              0.209756098
        11              0.190243902
        8               0.170731707
        4               0.156097561
        14              0.107317073
        2               0.087804878
        10              0.068292683
```

The following chart shows a visual representation of the sorted sentence score:

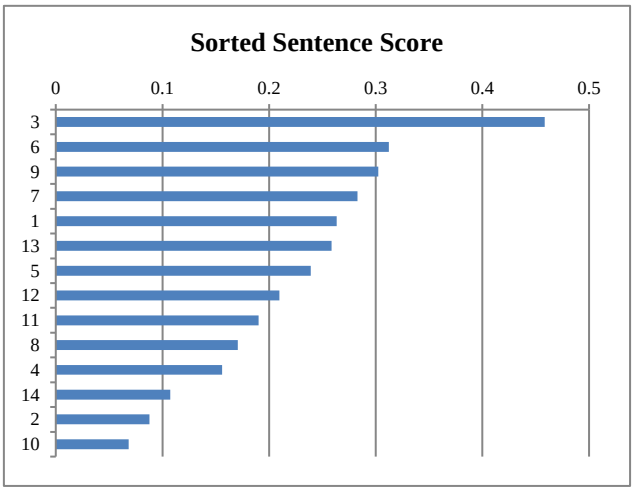

Fig 12: Chart of Sorted Sentence Score

Summary:

The summary consists of the sentences that have sentence scores above the average score. They are displayed in their natural order, as shown in the following view:

```
Summary:
        1               0.263414634
        3               0.458536585
        5               0.239024390
        6               0.312195122
        7               0.282926829
        9               0.302439024
        13              0.258536585
```

Finally, the summary is shown in the following view:

```
Summary:
text summarization is the process of shortening a set of data computationally to create a subset a summary that represents the most important or relevant information within the original content.
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.
extraction based summarization here content is extracted from the original data but the extracted content is not modified in any way.
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.
for text extraction is analogous to the process of skimming where the summary if available headings and subheadings figures the first and last paragraphs of a section and optionally the first and last sentences in a paragraph are read before one chooses to read the entire document in detail.
abstractive based summarization abstractive summarization methods generate new text that did not exist in the original text.
such transformation however is computationally much more challenging than extraction involving both natural language processing and often a deep understanding of the domain of the original text in cases where the original document relates to a special field of knowledge.
```

In summary, the program output clearly verifies that the developed program performed the basic steps of text summarization and provided the required results.

5. CONCLUSION

Text summarization is one of the important applications of machine learning. The purpose of text summarization is to provide a short and useful summary of the text. It helps to reduce the size of text and save the time of reading. The word frequency is used to measure the importance of words and sentences in the text.

In this research, the author developed a text summarization program using word frequency in Python. The developed program performed the basic steps of text summarization: preprocessing text, creating list of words, creating bag of words, creating word frequency, creating list of sentences, creating sentence score, sorting sentence score, calculating average score, and making summary.

The program was tested on an experimental text from Wikipedia and provided the required results: list of words, bag of words, word frequency, list of sentences, sentence score, sorted sentence score, average score, and summary.

In future work, more research is certainly required to improve and develop the current methods of text summarization. In addition, they should be more investigated in different domains and languages such as Arabic.

6. REFERENCES

[1] Sammut, C., & Webb, G. I. (2011). "Encyclopedia of Machine Learning". Springer.

[2] Torres-Moreno, J. M. (2014). "Automatic Text Summarization". John Wiley & Sons.

[3] Das, D., & Martins, A. F. (2007). "A Survey on Automatic Text Summarization". Language Technologies Institute. Language Technologies Institute.

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

[5] Gupta, V., & Lehal, G. S. (2010). "A Survey of Text Summarization Extractive Techniques". Journal of Emerging Technologies in Web Intelligence, 2(3), 258-268.

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

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

[8] Saranyamol, C. S., & Sindhu, L. (2014). "A Survey on Automatic Text Summarization". International Journal of Computer Science and Information Technologies, 5(6), 7889-7893.

[9] Dalal, V., & Shelar, Y. (2015). "Survey of Various Methods for Text Summarization". International Journal of Engineering Research & Development, 11(3), 57-59.

[10] Saziyabegum, S., & Sajja, P. S. (2016). "Literature Review on Extractive Text Summarization Approaches". International Journal of Computer Applications, 156(12).

[11] Kumar, Y.J., Goh, O.S., Basiron, H., Choon, N.H., & Suppiah, P.C. (2016). "A Review on Automatic Text Summarization Approaches". J. Comput. Sci., 12, 178-190.

[12] Gambhir, M., & Gupta, V. (2017). "Recent Automatic Text Summarization Techniques: A Survey". Artificial Intelligence Review, 47, 1-66.

[13] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). "Text sSummarization Techniques: A Brief Survey". arXiv preprint arXiv:1707.02268.

[14] Bharti, S. K., & Babu, K. S. (2017). "Automatic Keyword Extraction for Text Summarization: A Survey". arXiv preprint arXiv:1704.03242.

[15] Zerari, N., Aitouche, S., Mouss, M. D., & Yaha, A. (2017). "Automatic Text Summarization: A Review". In Proceedings of the 9th International Conference on Information, Process, and Knowledge Management.

[16] Klymenko, O., Braun, D., & Matthes, F. (2020). "Automatic Text Summarization: A State-of-the-Art Review". In Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEI), 1, 648-655.

[17] Mridha, M. F., Lima, A. A., Nur, K., Das, S. C., Hasan, M., & Kabir, M. M. (2021). "A Survey of Automatic Text Summarization: Progress, Process and Challenges". IEEE Access, 9, 156043-156070.

[18] Yadav, A.K., Maurya, A.K., Ranvijay, & Yadav, R.S. (2021). "Extractive Text Summarization Using Recent Approaches: A Survey". Ingénierie des Systèmes d Inf., 26, 109-121.

[19] Cajueiro, D.O., Nery, A.G., Tavares, I., Melo, M.K., Reis, S.A., Weigang, L., & Celestino, V.R. (2023). "A Comprehensive Review of Automatic Text Summarization Techniques: Method, Data, Evaluation and Coding". ArXiv, abs/2301.03403.

[20] Luhn, H. (1958). "The Automatic Creation of Literature Abstracts". IBM Journal of Research and Development, 2(2), 159-165.

[21] Salton, G., Wong, A., & Yang, C. S. (1975a). "A Vector Space Model for Automatic Indexing". Communications of the ACM, 18(11), 613-620.

[22] Salton, G., Yang, C. S., & Yu, C. T. (1975b). "A Theory of Term Importance in Automatic Text Analysis". Journal of the American Society for Information Science, 26(1), 33-44.

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

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

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

[26] Salton, G., Singhal, A., Mitra, M., & Buckley, C. (1997). "Automatic Text Structuring and Summarization". Information Processing & Management, 33(2), 193-207.

[27] Sparck Jones, K. (1972). "A Statistical Interpretation of Term Specificity and Its Application in Retrieval". Journal of Documentation. 28(1), 11–21.

[28] Sparck Jones, K. (2004). "IDF Term Weighting and IR Research Lessons". Journal of Documentation, 60(5), 521-523.

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

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

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

[32] Matplotlib: http://www.matplotlib.org

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

[34] SK Learn: https://scikit-learn.org

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