Extraction based approach for text summarization using k-means clustering

Ayush Agrawal , Utsav Gupta

Abstract- This paper describes an algorithm that incorporates k-means clustering, term-frequency inverse-document-frequency and tokenization to perform extraction based text summarization.

Index Terms- Information retrieval, k-means clustering, Natural Language Processing, text summarization.

I. Introduction

In recent years, natural language processing (NLP) has moved to a very firm mathematical foundation. Many problems in NLP, e.g., parsing [2] word sense disambiguation[3], and automatic paraphrasing [4]have benefited significantly by the introduction of robust statistical techniques.[1] In this paper we will discuss an unsupervised learning technique for a NLP problem. With the recent increase in the amount of content available online, fast and effective automatic summarization has become more important. The need for getting maximum information by spending minimum time has led to more efforts being directed to the field of summarization. So in this paper we will discuss an unsupervised learning technique for the Automated Summarization problem.

Text summarization is the process of automatically creating a compressed version of a given text that provides useful information for the user. The information content of a summary depends on user's needs. Topic-oriented summaries focus on a user's topic of interest, and extract the information in the text that is related to the specified topic. On the other hand, generic summaries try to cover as much of the information content as possible, preserving the general topical organization of the original text. There are basically two methods of summarization: extraction and abstraction. Extractive summarization produces summaries by choosing a subset of the sentences in the original document(s). This contrasts with abstractive summarization, where the information in the text is rephrased. Although summaries produced by humans are typically not extractive, most of the summarization research today is on extractive summarization. Purely extractive summaries often give better results compared to automatic abstractive summaries. This is due to the fact that the problems in abstractive summarization, such as semantic representation, inference and natural language generation, are relatively harder compared to a data-driven approach such as sentence extraction. In fact, truly abstractive summarization has not reached to a mature stage today.

In this paper, we employ extraction based techniques to generate automatic summaries from a document. Early research on extractive summarization is based on simple heuristic features of the sentences such as their position in the text, the overall frequency of the words they contain, or some key phrases indicating the importance of the sentences [5,6 ,7]

The approach used in this paper is an unsupervised learning technique which is accomplished as a part of three step process.

- Tokenization of Document
- Computing Score for each Sentence
- Applying Centroid Based Clustering on the Sentences and extracting important Sentences as part of summary.

II. Tokenization of Document

Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. The Major idea is to break the entire document into a list of sentences.[8]

2.1. Methods for tokenization:

Typically, tokenization occurs at the word level. However, it is sometimes difficult to define what is meant by a "word". Often a tokenizer relies on simple heuristics, for example:

- All contiguous strings of alphabetic characters are part of one token; likewise with numbers
- Tokens are separated by whitespace characters, such as a space or line break, or by punctuation characters.
- Punctuation and whitespace may or may not be included in the resulting list of tokens.

In languages that use inter-word spaces (such as most that use the Latin alphabet, and most programming languages), this approach is fairly straightforward.

However, even here there are many edge cases such as contractions, hyphenated words, emoticons, and larger constructs such as URIs (which for some purposes may count as single tokens). For instance consider the text "New York-based", which a naive tokenizer may break at the space even though the better break is (arguably) at the hyphen.

III. Computing Score for each Sentence

Each sentence is given an importance score and this acts as a goodness measure for the sentence. The scores can be used to order sentences and pick most important sentences. The probability of a sentence to be present in the summary is proportional to its score. Each sentence is represented by a set of features and the score is a function of the weighted sum of the individual feature values.[9]

The features we have used are:

3.1. TF-IDF sum: The goodness of a sentence is usually represented by the importance of the words present in it. TF-IDF is a simple but powerful heuristic for ranking the words

according to their importance. This feature is the sum of the TF-IDF scores of the individual words of the sentence.

- TF:Term Frequency, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization:

  TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

- IDF:Inverse Document Frequency, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

  $IDF(t) = \log_e^{(\text{Total number of documents / Number of documents with term t in it})}$ [10, 11, 12, 13]

- Tf-idf weighting:We now combine the definitions of term frequency and inverse document frequency, to produce a composite weight for each term in each document. The *tf-idf* weighting scheme assigns to term a weight in document given by

Tf-idf = tf * idf

In other words, tf-idf assigns to term a weight in document that is

1. highest when term occurs many times within a small number of documents (thus lending high discriminating power to those documents);
2. lower when the term occurs fewer times in a document, or occurs in many documents (thus offering a less pronounced relevance signal);
3. lowest when the term occurs in virtually all documents. [8,9,13]

3.2. Sentence length: This feature is the number of words present in the sentence. Longer sentences usually contain more information about the documents.

IV. Centroid Based Clustering

K-means is an unsupervised learning algorithm that solves the well known clustering problem. The procedure classifies a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroid are chosen to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When all points have been classified, we re-calculate k new centroid as new centers of the clusters resulting from the previous step. After we have these k new centroid, a new association is generated between the same data set points and the nearest new centroid. The k centroid change their location in each step until no more changes occur. Although the K-means algorithm will always terminate, it does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The K-means algorithm can be run multiple times to reduce this effect.[14]

The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum.[15]

This algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function is :

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

Where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the *n* data points from their respective cluster centres.[16,17]

The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated. [18]

Our approach

The major idea is to divide the entire document into sentences. Each sentence can be considered as a point in the Cartesian plane.

Each sentence is then broken into tokens and the tf-idf score is computed for each token in the sentence.

Tf_t = f(t,d) / f(d) where , t is a token

d represents the document

f(t,d) represents frequency of t in d

f(d) represents frequency of every term in d

$idf_t = \log_{10}^{(N / f(t,d))}$

where , N is the number of sentences in the document

$$\text{tf-idf}_t = Tf_t * idf_t$$

Score for each sentence is computed by summing up the tfd-idf score for every token in the sentence and normalizing it by using the sentence length.

$Score(X) = \sum_t$ tf-idf $_t$ / |X|
where , X represents a sentence in the document
t is a term in X
|X| represents length of X

These sentence scores are used to represent the sentences as unique coordinates in the single dimensional Cartesian plane.

These coordinates are used as input data for unsupervised k-means clustering algorithm. Simulating the algorithm on the input data generates k cluster centers.

Now we classify each sentence into different clusters based on the scores computed for each sentence.

Finally, we pick the cluster with maximum number of sentences and generating the summary by producing the sentences in the same order in which they appear in the original document.

This approach gives a precise summary because the densest cluster which is returned by the k-means clustering algorithm consists of the sentences with highest scores in the entire document. These sentence scores are computed by summing up the tf-idf scores of individual terms in the sentence and normalizing it by using the length of the sentence.tf-idf takes into account the case of stop words and unique words. Thus the sentences in the most dense cluster are the ones which are contextually closer to the abstract idea of the document.

Choice of k for the clustering algorithm

The length of the summary should change depending on the length of the document. If we choose a very large value for k , then the clusters are sparse and the summaries are not coherent. On the contrary, if the value of k is very small then the clusters are very dense and the summary is not so concise. Hence, the value of k should dynamically vary according to the length of the document.

After running the simulations on various documents and different values of k , we formulated a function to determine the value of k.

```
if N <= 20:
   k = N - 4
else :
   k = N – 20
```

where, N is the number of sentences in the document

Evaluation

As the approach described in the paper is an extraction based, it is necessary for the resulting summary size to be around 35% - 50% of the original text size because the resulting summary if is smaller then 35- 50% size of original size of text will be small in size but not concise in meaning and the summary will also be not that coherent when compared to human written abstractive summary. To evaluate our approach we ran simulation on some text samples and compared it with other existing summarizer which uses an extraction based technique

Text	Number of Sentences in Original Passage	Number of Sentences in summary made by our approach	Number of Sentences in summary made by autosummarizer.com	% of size of text in summary by our approach	% of size of text in summary by our autosummarizer.com
1	24	10	4	41.66	16.66
2	20	6	5	30	25
3	30	14	8	46.66	26.66
4	33	16	6	48.48	18.18
5	18	9	5	50	27.77
6	14	6	3	42.85	21.43

Figure 1 : Results of Evaluation of our approach on various sample text

Our approach when compared with a human written abstractive summary produces a good result. Whereas, other extraction based technique does not produce a similar result when compared to human written summary.

Summary written by human for a certain document -

New Caledonia is an island archipelago slightly north the Tropic of Capricorn, approximately 1,500 km from Australia and New Zealand. 80% of New Caledonia's 200,000 residents live on the main island, Grand Terre. The economy is sustained mostly by mining and tourism. Through the mining sector has been down lately, tourism is doing well and many wind surfers, scuba divers, snorkelers, and golfers flock to Pacific islands. The capital city of Noumea has a charming French atmosphere and lots of boutiques, museums, and restaurants. The zoo, botanical gardens, and aquarium are a must see. New Caledonia offers a wide variety of hotel facilities and though French is the lingua franca, English is also widely understood - particularly in touristy places.

Summary generated by our approach -

New Caledonia a cluster of islands in the southwestern Pacific Ocean.New Caledonia's economy is based mainly on

tourism and mining.About 25% of the world's known nickel resources are found in New Caledonia.The botanical and zoological gardens in Noumea are first rate.A range of accommodations are available throughout the territory.You five-star hotels/resorts or simple tribal lodgings in Melanesian villages.

Summary generated by autosummarizer.com -

The main island, known as Grand Terre, is home to over 160,000 people - over 80% of the population of this French colony. The capital city, Noumea, has a distinctly French ambience and offers many shops, museums, and restaurants with various French, Indonesian, Chinese, Italian, Mexican, and Japanese food. Though French is the official language island there are about 30 local languages, English is also widely spoken in areas which are heavily touristed.

V. Conclusion

In this paper, we proposed an automatic text summarization approach by sentence extraction using an unsupervised learning algorithm. In particular, the K- means algorithm for creating groups of similar sentences was used. Then, from the groups of sentences, the most representative sentence was selected for composing the summary. The proposed approach, in contrast to supervised methods, does not need large amount of golden samples for training. Therefore, our proposed approach is more independent from language and domain. According to experimental results we demonstrate that the proposed approach obtains more favourable results than others state-of-the-art approaches using graph based techniques and supervised learning algorithms [19, 20, 21]; ranking our proposed approach in second place, very close to the first place.

References

[1] Erkan ,G. & Radev, D. (2004). LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research 22 (2004) 457-479

[2] Collins, M. (1997). Three generative, lexicalised models for statistical parsing. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics.

[3] Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics.

[4] Barzilay, R., & Lee, L. (2003). Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In Proceedings of HLT-NAACL

[5] Baxendale, P. (1958). Man-made index for technical litterature - an experiment. IBM J. Res. Dev., 2 (4), 354{361.

[6] Edmundson, H. (1969). New Methods in Automatic Extracting. Journal of the Association for Computing Machinery, 16 (2), 264{285.

[7] Luhn, H. (1958). The Automatic Creation of Literature Abstracts. IBM Journal of Research Development, 2 (2), 159{165.

[8] Manning, C. D.; Raghavan, P.; Schutze, H. (2008). Introduction to Information Retrieval. Cambridge University Press

[9] K. Sparck Jones. "A statistical interpretation of term specificity and its application in retrieval". Journal of Documentation, 28 (1). 1972.

[10] G. Salton and Edward Fox and Wu Harry Wu. "Extended Boolean information retrieval". Communications of the ACM, 26 (11). 1983.

[11] G. Salton and M. J. McGill. "Introduction to modern information retrieval". 1983

[12] G. Salton and C. Buckley. "Term-weighting approaches in automatic text retrieval". Information Processing & Management, 24 (5). 1988.

[13] H. Wu and R. Luk and K. Wong and K. Kwok. "Interpreting TF-IDF term weights as making relevance decisions". ACM Transactions on Information Systems, 26 (3). 2008.

[14] J. B. MacQueen (1967): "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, 1:281-297

[15] Mahajan, M.; Nimbhorkar, P.; Varadarajan, K. (2009). "The Planar k-Means Problem is NP-Hard". Lecture Notes in Computer Science 5431: 274–285.

[16] E.W. Forgy (1965). "Cluster analysis of multivariate data: efficiency versus interpretability of classifications". Biometrics 21: 768–769.

[17] J.A. Hartigan (1975). Clustering algorithms. John Wiley & Sons, Inc

[18] Hartigan, J. A.; Wong, M. A. (1979). "Algorithm AS 136: A K-Means Clustering Algorithm". Journal of the Royal Statistical Society, Series C 28 (1): 100–108. JSTOR 2346830

[19] Villatoro-Tello, E., Villaseñor-Pineda, L., Montes-y-Gómez, M.: Using Word Sequences for Text Summarization. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 293–300. Springer, Heidelberg (2006)

[20] Chuang, T.W., Yang, J.: Text Summarization by Sentence Segment Extraction Using Machine Learning Algorithms. In: Proc. of the ACL 2004 Workshop, Barcelona, España (2004)

[21] Neto, L., Freitas, A.A., Kaestner, C.A.A.: Automatic Text Summarization using a Machine learning Approach. In: Proceedings of the ACL 2004 Workshop, Barcelona, España (2004)

Authors

First Author – Ayush Agrawal, ayush.agrawal464@gmail.com
Second Author – Utsav Gupta, utsavgupta7@gmail.com