International Journal of Scientific & Engineering Research Volume 8, Issue 5, May-2017
ISSN 2229-5518

Using NLP for Article Summarization

Nishit Mohanan, Johny Johnson, Pankaj Mudholkar

Abstract— Summarization is the process of reducing a block of text by extracting the most important points in a text document, resulting in a summary of the original document. This is a part of Machine Learning and Data Mining. The crux of summarization is to find the subset of a text or article which contains the information of the entire set of data. There are two techniques for summarization, the first is extraction-based summarization which extracts certain key sentences from the text using various algorithms like text rank. The second is Abstraction-based summarization where the text is analyzed and rewritten or rephrased to achieve a text of shorter length, but this technique requires natural language generation which itself is an emergent field and not widely used. Summarization can be used in various fields and has various applications, news sites can use them to provide a short summary of the entire article, and it can be used to save time by obtaining the necessary information without spending too much time reading the article. This paper reviews the use of NLP for article summarization.

Index Terms— Artificial Intelligence, Algorithms, Automatic evaluation, Data Mining, NLP, Machine Learning, Summarization.

1 Introduction

Natural language processing is a field of artificial intelligence, computer science and computational linguistics concerned with the interactions between computers and human languages. NLP is related to the area of human to computer interaction. Many challenges in NLP involve: natural language understanding, enabling computers to derive meaning from human or natural language input; and others involve natural language generation. Research on NLP was started around 1950s, Alan Turing published an article "Computing Machinery and intelligence which proposed the Turing Test.Real progress has been slow in NLP and after ALPAC research which tool 10 years to complete failed to fulfill their goals funding and research in this field had been reduced. Recently NLP has gainedtraction due to faster processers and a focus on machine learning. There are several tasks which can be done using NLP like Summarization, Machine Translation, Natural language generation and understanding. One of the major tasks which is done using NLP is Summarization.

2 Algorithms used for NLP

Many different types of machine learning algorithms are used for NLP. Some of the earliest algorithms used decision tree which used if-then rules. Current methods use statistical models which attach weights for each input. Such models are used in Automatic Learning Algorithms which can produce better results after increasing the amount of data which is used to train the system.Statistical natural language processing uses stochastic, probabilistic and statistical methods, especially to resolve difficulties that arise because longer sentences are highly ambiguous when processed with realistic grammars, yielding thousands or millions of possible analyses.

- Nishit Mohanan *is currently pursuing MCA at Thakur Institute of Management Studies, Career Development and Research (TIMSCDR),Mumbai, India. Email:* nishitmohanan@gmail.com
- Johny Johnson *is currently pursuing MCA at Thakur Institute of Management Studies, Career Development and Research (TIMSCDR),Mumbai, India. Email:* johnsonjohny1993@gmail.com
- *Prof. Pankaj Mudholkar is Asst. Professor at Thakur Institute of Management Studies, Career Development and Research (TIMSCDR), Mumbai, India. E-mail:* mudholkarpankaj@gmail.com

Methods include Corpora and Markov Chains.

Markov model is a stochastic model used to model randomly changing systems, it is assumed that future states depend only on the current state not on the events that occurred before it. Hidden Markov Chains can also be used in NLP.

3 Types of Evaluation

There are different methods to evaluate NLP.
Some of the methods are:

3.1 Intrinsic and extrinsic evaluation:

Intrinsic systems are isolated systems and its performance is characterized by the standards set by the evaluators. Extrinsic systems consider the NLP system in a more complex setting as either an embedded system or a precise function for a human user

3.2 Black-box vs glass-box evaluation:

Black-box evaluation requires the user run an NLP system on a sample data set and to measure the number of parameters related to the quality of the process, such as reliability, speed and most importantly, the quality of the result, such as the accuracy of data annotation.
Glass-box evaluation looks at the algorithms that are implemented design of the system, the resources it uses, like vocabulary size or expression set cardinality.
Given the complexity of NLP problems, it is often difficult to predict performance only on the basis of glass-box evaluation but this type of evaluation is more informative with respect to error analysis or future developments of a system.

All tables and figures will be processed as images. You need to embed the images in the paper itself. Please don't send the images as separate files.

3.3 Manual and Automatic evaluation:

Automatic procedures are defined by comparing its output with a standard output. The cost of reproducing the standard output can be high, therefore automatic evaluation of the same input data can be high. But bootstrapping automatic evaluation on the same input can be repeated without incurring huge additional costs. However for many NLP tasks the precise definition of a standard can be difficult to define. In man-

ual evaluation human judges evaluate the quality of a system or its sample output, based on a set number of criteria.

4 STANDARDIZATION

An ISO subcommittee is working to standardize and ease interoperability between lexical resources and NLP programs.The subcommittee is called ISO/ TC37/ SC4 and is a part of ISO/ TC37.Most ISO standards related to NLP are still under construction.

5 NLP METHODS

By and large, there are two ways to deal with programmed rundown: extraction and abstraction. Extractive strategies work by selecting a subset of existing words, expressions, or sentences in the first content to frame the synopsis. Interestingly, abstractive strategies fabricate an inward semantic representation and after that utilization characteristic dialect era methods to make a synopsis that is nearer to what a human may produce. Such an outline may contain words not expressly display in the first. Scrutinize into abstractive techniques is an undeniably imperative and dynamic research region, however because of many-sided quality requirements, research to date has concentrated fundamentally on extractive strategies. In some application areas, extractive synopsis bodes well. Cases of these incorporate picture gathering rundown and video outline.

5.1 EXTRACTION BASED SUMMARIZATION

In this method, the programmed framework removes objects from the whole gathering, without changing the articles themselves. Cases of this incorporate key phrase extraction, where the objective is to choose singular words or expressions to "tag" an archive, and report synopsis, where the objective is to choose entire sentences (without changing them) to make a short passage outline. Correspondingly, in picture gathering outline, the framework extricates pictures from the accumulation without changing the pictures themselves.

5.2 ABSTRACTION BASED SUMMARIZATION

Extraction procedures simply duplicate the data regarded most vital by the framework to the outline (for instance, key provisos, sentences or passages), while abstraction includes rewording segments of the source record. By and large, abstraction can consolidate a content more unequivocally than extraction, yet the projects that can do this are harder to create as they require utilization of characteristic dialect era innovation, which itself is a developing field.

5.3 AIDED SUMMARIZATION

Machine taking in strategies from firmly related fields, for example, data recovery or content mining have been effectively adjusted to help programmed synopsis.

Aside from Fully Automated Summarizers (FAS), there are frameworks that guide clients with the undertaking of outline (MAHS = Machine Aided Human Summarization), for instance by highlighting hopeful sections to be incorporated into the synopsis, and there are frameworks that rely on upon post-handling by a human (HAMS = Human Aided Machine Summarization).

6 SYSTEMS AND APPLICATIONS OF SUMMARIZATION

There are extensively two sorts of extractive synopsis errands relying upon what the rundown program concentrates on. The first is bland outline, which concentrates on acquiring a non specific rundown or unique of the gathering (whether archives, or sets of pictures, or recordings, news stories and so forth.). The second is question applicable synopsis, once in a while called inquiry based outline, which abridges objects particular to an inquiry. Rundown frameworks can make both question significant content outlines and non specific machine-produced synopses relying upon what the client needs.

A case of a synopsis issue is report outline, which endeavors to consequently deliver a theoretical from a given archive. At times one may be keen on creating a synopsis from a solitary source record, while others can utilize numerous source reports (for instance, a bunch of articles on similar point). This issue is called multi-record rundown. A related application is outlining news articles. Envision a framework, which consequently pulls together news articles on a given subject (from the web), and succinctly speaks to the most recent news as a rundown.

Picture accumulation rundown is another application case of programmed outline. It comprises in selecting a delegate set of pictures from a bigger arrangement of images.[1] A rundown in this setting is helpful to demonstrate the most illustrative pictures of results in a picture accumulation investigation framework. Video rundown is a related space, where the framework consequently makes a trailer of a long video. This additionally has applications in buyer or individual recordings, where one might need to skirt the exhausting or monotonous activities. Also, in reconnaissance recordings, one would need to extricate essential and suspicious action, while disregarding all the exhausting and repetitive edges caught.

At an abnormal state, outline calculations attempt to discover subsets of articles (like arrangement of sentences, or an arrangement of pictures), which cover data of the whole set. This is likewise called the center set. These calculations show ideas like assorted qualities, scope, data and representativeness of the synopsis. Question based outline methods, furthermore display for relevance of the rundown with the inquiry. A few systems and calculations which normally show synopsis issues are TextRank and PageRank, Submodular set capacity, Determinantal point handle, maximal marginal relevance (MMR) and so forth.

7 KEYPHRASE EXTRACTION

The undertaking is the accompanying. You are given a bit of content, for example, a diary article, and you should create a rundown of watchwords or key[phrase]s that catch the essential themes talked about in the content. On account of research

articles, numerous writers give physically relegated catchphrases, yet most content needs previous keyphrases. For instance, news articles once in a while have keyphrases joined, however it is helpful to have the capacity to naturally do as such for various applications examined underneath. Consider the example content from a news article:

"The Army Corps of Engineers, rushing to meet President Bush's promise to protect New Orleans by the start of the 2006 hurricane season, installed defective flood-control pumps last year despite warnings from its own expert that the equipment would fail during a storm, according to documents obtained by The Associated Press".

A keyphrase extractor might select "Army Corps of Engineers", "President Bush", "New Orleans", and "defective flood-control pumps" as keyphrases. These are pulled directly from the text. In contrast, an abstractive keyphrase system would somehow internalize the content and generate keyphrases that do not appear in the text, but more closely resemble what a human might produce, such as "political negligence" or "inadequate protection from floods". Abstraction requires a deep understanding of the text, which makes it difficult for a computer system. Keyphrases have many applications. They can enable document browsing by providing a short summary, improve information retrieval (if documents have keyphrases assigned, a user could search by keyphrase to produce more reliable hits than a full-text search), and be employed in generating index entries for a large text corpus.

Depending on the different literature and the definition of key terms, words or phrases, highly related theme is certainly the Keyword extraction.

8 Supervised Learning Approach

Starting with the work of Turney, numerous analysts have drawn closer keyphrase extraction as a managed machine learning issue. Given a report, we build a case for each unigram, bigram, and trigram found in the content (however other content units are likewise conceivable, as talked about beneath). We then process different elements depicting every case (e.g., does the expression start with a capitalized letter?). We accept there are known keyphrases accessible for an arrangement of preparing records. Utilizing the known keyphrases, we can dole out positive or negative names to the cases. At that point we take in a classifier that can segregate amongst positive and negative cases as an element of the components. A few classifiers make a paired characterization for a test case, while others allocate a likelihood of being a keyphrase. For example, in the above content, we may take in a decide that says phrases with introductory capital letters are probably going to be keyphrases. In the wake of preparing a learner, we can choose keyphrases for test reports in the accompanying way. We apply similar illustration era methodology to the test archives, then run every case through the learner. We can decide the keyphrases by taking a gander at double order choices or probabilities came back from our educated model. On the off chance that probabilities are given, a limit is utilized to choose the keyphrases. Keyphrase extractors are for the most part assessed utilizing exactness and review. Exactness measures what number of the proposed keyphrases are really right. Review measures what number of the genuine keyphrases your framework proposed. The two measures can be joined in a F-score, which is the consonant mean of the two ($F = 2PR/(P + R)$). Coordinates between the proposed keyphrases and the known keyphrases can be checked in the wake of stemming or applying some other content standardization.

Outlining an administered keyphrase extraction framework includes settling on a few decisions (some of these apply to unsupervised, as well). The principal decision is precisely how to produce cases. Turney and others have utilized all conceivable unigrams, bigrams, and trigrams without interceding accentuation and subsequent to evacuating stopwords. Hulth demonstrated that you can get some change by selecting cases to be arrangements of tokens that match certain examples of grammatical feature labels. In a perfect world, the component for creating illustrations delivers all the referred to named keyphrases as competitors, however this is frequently not the situation. For instance, on the off chance that we utilize just unigrams, bigrams, and trigrams, then we will never have the capacity to separate a known keyphrase containing four words. In this way, review may endure. Be that as it may, producing excessively numerous illustrations can likewise prompt low accuracy.

We additionally need to make highlights that portray the cases and are sufficiently educational to permit a taking in algorithm to segregate keyphrases from non-keyphrases. Regularly highlights include different term frequencies (how often an expression shows up in the present content or in a bigger corpus), the length of the case, relative position of the principal event, different boolean syntactic components (e.g., contains all tops), and so on. The Turney paper utilized around 12 such elements. Hulth utilizes a lessened arrangement of elements, which were discovered best in the KEA (Keyphrase Extraction Algorithm) work got from Turney's fundamental paper.

At last, the framework should give back a rundown of keyphrases for a test record, so we need an approach to constrain the number. Outfit techniques (i.e., utilizing votes from a few classifiers) have been utilized to create numeric scores that can be thresholded to give a client gave number of keyphrases. This is the system utilized by Turney with C4.5 choice trees. Hulth utilized a solitary twofold classifier so the learning algorithm certainly decides the fitting number.

When cases and components are made, we require an approach to figure out how to anticipate keyphrases. For all intents and purposes any directed learning algorithm could be utilized, for example, choice trees, Naive Bayes, and control enlistment. On account of Turney'sGenEx algorithm, a hereditary algorithm is utilized to learn parameters for an area particular keyphrase extraction algorithm. The extractor takes after a progression of heuristics to distinguish keyphrases. The hereditary algorithm enhances parameters for these heuristics as for execution on preparing records with known key expressions.

9 Unsupervised Approach: TextRank

Another keyphrase extraction algorithm is TextRank. While managed strategies have some pleasant properties, such as having the capacity to deliver interpretable tenets for what highlights describe a keyphrase, they likewise require a lot of preparing information. Numerous records with known keyphrases are required. Besides, preparing on a particular area has a tendency to redo the extraction procedure to that space, so the subsequent classifier is not really compact, as some of Turney's outcomes illustrate. Unsupervised keyphrase extraction evacuates the requirement for preparing information. It approaches the issue from an alternate edge. Rather than attempting to learn unequivocal elements that describe keyphrases, the TextRank algorithm abuses the structure of the content itself to decide keyphrases that seem "focal" to the content similarly that PageRank chooses critical Web pages. Review this depends on the idea of "eminence" or "proposal" from interpersonal organizations. Along these lines, TextRank does not depend on any past preparing information by any stretch of the imagination, but instead can be keep running on any discretionary bit of content, and it can deliver yield essentially in light of the content's inherent properties. Accordingly the algorithm is effortlessly versatile to new areas and dialects.

TextRank is a broadly useful diagram based positioning algorithm for NLP. Basically, it runs PageRank on a chart extraordinarily intended for a specific NLP errand. For keyphrase extraction, it manufactures a diagram utilizing some arrangement of content units as vertices. Edges depend on some measure of semantic or lexical closeness between the content unit vertices. Not at all like PageRank, the edges are normally undirected and can be weighted to mirror a level of similitude. Once the diagram is built, it is utilized to frame a stochastic network, joined with a damping component (as in the "irregular surfer display"), and the positioning over vertices is acquired by finding the eigenvector comparing to eigenvalue 1 (i.e., the stationary circulation of the arbitrary stroll on the chart).

The vertices ought to relate to what we need to rank. Conceivably, we could accomplish something like the managed techniques and make a vertex for each unigram, bigram, trigram, and so forth. In any case, to keep the diagram little, the creators choose to rank individual unigrams in an initial step, and afterward incorporate a second step that consolidations exceedingly positioned nearby unigrams to frame multi-word phrases. This has a decent symptom of permitting us to deliver keyphrases of subjective length. For instance, on the off chance that we rank unigrams and find that "best in class", "regular", "dialect", and "preparing" all get high positions, then we would take a gander at the first content and see that these words show up successively and make a last keyphrase utilizing every one of the four together. Take note of that the unigrams put in the chart can be sifted by grammatical feature. The creators found that modifiers and things were the best to incorporate. In this manner, some etymological learning becomes an integral factor in this progression.

Edges are made in light of word co-event in this utilization of TextRank. Two vertices are associated by an edge if the unigrams show up inside a window of size N in the first content. N is normally around 2–10. Therefore, "regular" and "dialect" may be connected in a content about NLP. "Normal" and "preparing" would likewise be connected in light of the fact that they would both show up in similar string of N words. These edges expand on the thought of "content union" and the possibility that words that show up close to each other are likely related definitively and "prescribe" each other to the peruser.

Since this technique essentially positions the individual vertices, we require an approach to edge or create a set number of keyphrases. The strategy picked is to set a tally T to be a client indicated part of the aggregate number of vertices in the diagram. At that point the top T vertices/ unigrams are chosen in view of their stationary probabilities. A post-handling step is then connected to combine adjoining occasions of these T unigrams. Thus, possibly pretty much than T last keyphrases will be created, yet the number ought to be generally corresponding to the length of the first content.

It is not at first clear why applying PageRank to a co-event chart would create valuable keyphrases. One approach to consider it is the accompanying. A word that seems numerous times all through a content may have a wide range of co-happening neighbors. For instance, in a content about machine taking in, the unigram "learning" may co-happen with "machine", "directed", "un-administered", and "semi-managed" in four unique sentences. In this way, the "learning" vertex would be a focal "center" that associates with these other altering words. Running PageRank/ TextRank on the chart is probably going to rank "adapting" exceedingly. Essentially, if the content contains the expression "directed grouping", then there would be an edge amongst "administered" and "order". In the event that "characterization" seems a few different spots and along these lines has numerous neighbors, its significance would add to the significance of "regulated". On the off chance that it winds up with a high rank, it will be chosen as one of the top T unigrams, alongside "learning" and most likely "order". In the last post-handling step, we would then wind up with keyphrases "regulated learning" and "directed order".
To put it plainly, the co-event chart will contain thickly associated districts for terms that show up regularly and in various settings. An irregular stroll on this diagram will have a stationary circulation that doles out substantial probabilities to the terms in the focuses of the bunches. This is like thickly associated Web pages getting positioned profoundly by PageRank. This approach has additionally been utilized as a part of record rundown, considered beneath.

10 Conclusion and Future Directions

In this review different application and methods were researched related to summarization of articles and text. As of now summarization and the entirety of NLP is still in its early stages of research and no single method is perfect due to the nuanced nature of human language. Further most NLP and summarization has been done for the English language while summarization of non-English languages still in very early stages. In the near future machine learning based aided summary will make huge advances in this field especially due to

the development of faster methods and various libraries related to NLP like NLPTK, Tensor Flow etc.

REFERENCES

[1]Rada Mihalcea and Paul Tarau, 2004: TextRank: Bringing Order into Texts, Department of Computer Science University of North Texas.

[2] Proceedings of the 14th Iberoamerican Conference on Pattern Recognition: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP '09),

[3]Güneş Erkan and Dragomir R. Radev: LexRank: Graph-based Lexical Centrality as Salience in Text Summarization

[4] Yatsko, V. et al Automatic genre recognition and adaptive text summarization. In: Automatic Documentation and Mathematical Linguistics, 2010, Volume 44, Number 3, pp.111-120.

[5] CS838-1 Advanced NLP:
Automatic SummarizationAndrew GoldbergMarch 16, 2007.

[6] Hui Lin, Jeff Bilmes. "A Class of Submodular Functions for Document Summarization", The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT) , 2011.

[7]Sebastian Tschiatschek, RishabhIyer, Hoachen Wei and Jeff Bilmes, Learning Mixtures of Submodular Functions for Image Collection Summarization, In Advances of Neural Information Processing Systems (NIPS), Montreal, Canada, December - 2014.

[8] Carbonell, Jaime, and Jade Goldstein. "The use of MMR, diversity-based reranking for reordering documents and producing summaries." Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1998.

[9]Hui Lin, Jeff Bilmes. "Learning mixtures of submodular shells with application to document summarization", UAI, 2012,

[10]Sarker, Abeed; Molla, Diego; Paris, Cecile (2013). "An Approach for Query-focused Text Summarization for Evidence-based medicine". Lecture Notes in Computer Science. 7885: 295–304