UNIVERSITY OF GOTHENBURG

Automatic topic extraction from research articles using N-gram analysis

Bachelor of Science Thesis in the Programme
Software Engineering and Management

MAOMAO CHEN
MAOYI HUANG

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
Göteborg, Sweden, June 2016

Automatic topic extraction from research articles using N-gram analysis

MAOMAO CHEN
MAOYI HUANG

Examiner: Christian Berger
Supervisor: Hang Yin and Alessia Knauss

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden June 2016

Abstract

Identifying the topic of an article can involve a lot of manual work. The manual processes can be exhaustive when it comes to a large volume of articles. In order to tackle this problem, we propose an automated topic extraction approach, which is able to extract topics for a large number of articles with a consideration to efficiency. To support the automatic topic extraction, our research focuses on existing N-gram analysis, which only calculates the words appearing frequency in a document. But in our research, we apply our customized filtering standards to improve the efficiency. And also to eliminate the irrelevant or noncritical phrases as many as possible. By doing that, we can make sure that our final selected keyphrases to each article are unique labels, which can represent the core idea of each specific article. In our case, we choose to focus on the research papers within the autonomous vehicle domain because the research papers are highly demanded in our daily life. Since most of the research papers are available only in PDF format, we need to process the PDF format files into the editable file types such as TXT. In order to realize the automation, we have selected a large number of autonomous vehicle-related articles to test our proposed idea. Then we observe the result and compare it with the manual topic extraction result to evaluate our approach.

Keywords *-- automatic topic extraction, N-gram, keyphrase, frequency statistic*

Acknowledgment

We want to especially thank this research's supervisor, Hang Yin and Alessia Knauss, Department of Computer Science and Engineering, who provided us various assistances in the study process and guided our thesis orientation and structure. Also, we are thankful for our examiner, Christian Berger, associated professor at Chalmers/University of Gothenburg, Department of Computer Science and Engineering, for reading and providing feedback for our thesis. Finally, we extend our thanks to Chalmers/University of Gothenburg, Department of Computer Science and Engineering, our course lecturer Imed Hammouda.

Contents

1. Introduction

1.1 Overview

Nowadays due to the rapidly growing pace of the information technology, it is critical to achieving the information retrieval task. For retrieving information in a research paper, the topic would be the most interested part for us. So how to elicit the interested information and filter out the irrelevant data has become a more critical issue nowadays. If we look at this issue a little further, there are numerous electronic documents including the research studies, e-mails, E-books etc. shared on the Internet and the numbers are still increasing. Everybody prefers to spend less time in checking through an entire document to know the topic so they can decide whether or not they want to read more. Unfortunately, right now most of the people still have to do manual checking with the document because the current technology mainly just offers the user to give their own keywords to filter papers. If there were a way to help the user to perform automatic checking for the document, it would save much time for them. Thus, we have an idea of implementing an automated topic extraction approach for the research papers, so that can help the readers to quickly check through documents and have a knowledge about the topics. Then they can decide the documents to read without taking too much effort.

Topic extraction (also as keyphrase extraction) is a proven idea that is able to achieve this task. There are a considerable number of researchers that have thought about this subject and made their contributions to the topic. There are some insightful articles about models or frameworks proposed by them. For example, from Zhiyuan et al (2010) [1]'s work, they introduce an idea of classifying approaches for topic extraction into two categories: the supervised and unsupervised principles. The supervised principle can be illustrated in Turney's work [2], which is a model to determine whether a candidate topic is a key topic. This type of extraction can be commonly found on the online digital library search engine. The inconvenient part of this approach is that it requires human labeling which demands the users to come up with their own keywords and then uses the input to match the whole database to see if there is any label fit. However, if the task is to collect a big number of articles then the user needs to spend considerable time to check if the searched result is actually focusing on his keywords instead of just briefly mentioning it for several times. Which means that the unsupervised principle can be quite time-consuming under this circumstance, hence we put more focus on the supervised approach [12].

As in Mihalcca and Tarau's [4] suggestion, the graph-based ranking algorithms have drawn a lot of attention and had its success for the supervised approach. This approach is a way of deciding on the importance of a vertex within a graph by taking into account global information recursively computed from the entire graph [4]. The HITS algorithm [12] and Google's PageRank [12] are two noticeable examples that have been acknowledged by the wide range of users. However, they are mainly used in citation analysis, social networks and the web-link extraction [12]. Hence, it has its shortcomings if we intend to deal with the topic extraction, which can help to summarize the text's concept

In order to make the topic extraction process efficient and reliable, here we propose an automated topic extraction approach that based on n-gram analysis to perform the job. To make the process efficient, we bring in an idea of emerging a semi-customized blacklist and whitelist to filter the irrelevant result after n-gram analysis. Moreover, we take an adequate number of training samples to make sure the quality of blacklist and whitelist. In the end, we will also perform a manual topic extraction process for evaluating our automated approach's efficiency and accuracy.

Our study is aimed to help the readers to efficiently label the selected documents so they can decide whether or not it is his/her desirable topic. We also make an endeavor to make the topic extraction process as automated as possible, which means the reader should spend the least effort in identifying the topic but the topics should be delivered to the reader automatically. Apart from that, the result should show a good illustration in terms of efficiency and reliability.

1.2 Research Objective

So in order to achieve the topic extraction automation, we propose an integrated way to identify the topic of numerous research studies by using N-gram analysis with the help of a blacklist and whitelist, then utilizing the result to label each document that we have processed. By doing this, we aim at finding a practically useful solution for automated topic extraction task, The application should handle the task well in assisting the scholars or any people who intend to find his/her desired article from an extensive number of random articles. Then we will have a deep understanding about the accuracy and efficiency of the N-gram based method, depending on analyzing the results from automatic and manual topic extraction methods

RQ1: "How can N-gram analysis help in the automation of topic extraction from research papers?"

RQ2: "How can we evaluate our automated topic extraction approach?"

In general, the sections of this thesis are structured as follows:
The background section describes related studies in the area of topic extraction and n-gram based techniques. The third section explains the research method that was used in conducting this study. In section 4, it illustrates the environmental settings for our design research study. And section 5 depicts the processes including our way to developing the solution as well as our plan to implement it under the defined environment. After that in section 6, we illustrate the process of how we plan to evaluate the solution. Then we conduct a detailed discussion on the results of the previous section in section 7 and section 8 is the conclusion.

2. Background

2.1 Keyphrase Extraction

With the development of scientific research, relevant research articles are emerging exponentially. How to effectively seek and manage information becomes an important research issue. At that time, we put forward the concept of keyphrases to help organize, manage and retrieve documents. Keyphrases are expressions, either single words or phrases, describing the most important ideas or main topics of a document [5]. As a representative summary of the document, keyphrases have been utilized extensively to acquire core information. As well as extracting high-quality keyphrases can benefit various natural language processing (NLP) applications, such as summarization [6], information retrieval (IR) [7], question answering (QA) [6], document classification etc [8]. Especially for the query or topic independent summarization system, it's a must-needed module [8].

2.2 Manual Extraction VS. Automatic Extraction

Keyphrase extraction is divided into two methods, manual and automatic. Authors usually assign Keyphrases in articles of journals and books. However, most manual keyphrase extraction is not consistent. Although information specialists are highly skilled, it is difficult for a team to classify content consistently and unambiguously, even if they are following a standard template [9]. The existence of subjective opinions is a problem that cannot be ignored in the manual method. Moreover, the manual process is a very tedious and time-consuming project.

With the volume of documents and information significantly increases, and more and more risks and costs，it is clear that the drawbacks of manual extraction becoming more acute. Based on this observation, people began to tend to use automatic extraction method to solve the problem. Jones and Paynter [10] describe that listing documents related to a primary document's keywords is a good solution to classify articles by different types and using keyword anchors as hyperlinks between documents enable a user to quickly access related material.

But it does not mean the automatic keyphrase extraction is better than the manual keyphrase extraction in certain aspects. The former may significantly improved the efficiency than the latter, but not necessarily in accuracy. Moreover, current keyphrase technology still has much room for improvement [11].

2.3 N-gram

An n-gram is a contiguous sequence of n items from a given sequence of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The n-grams typically are collected from a text or speech corpus [12]. Specifically, An N-gram is a sequence of N words, for example, a 2-gram (or bigram) is a two-word sequence of words like "please turn", "turn your", or "your homework", and a 3-gram (or trigram) is a three-word sequence of words like "please turn your", or "turn your homework" [13]. Using N-gram to estimate the probability of the last word in one sentence, and also to assign probabilities to the entire sequences [13].

The underlying mathematics of the N-gram was first proposed by Markov (1913). The early 1980s to the 1990s, n-gram technique is widely used for text compression, checking spelling errors, accelerating string search, and document language identification. In the 90s, the technology gets new applications in the field of automatic natural language processing, such as automatic classification, automatic indexing, automatic generation of hyperlinks, document retrieval and text language unseparated cut grading. On the other hand, n-gram also widely used in probability, such as communication theory, computational linguistics (for instance, statistical natural language processing), computational biology (for instance, biological sequence analysis) and data compression [12]. The most useful feature in N-gram is the automatic classification of natural language. As you can imagine, now it is an era of information overload, humanly relies on artificial information identification and classification has become unrealistic and the automatic classification of the natural language is becoming a reality.

Currently, many companies have developed a number of N-gram-based tasks, such as Google and Microsoft. Here is a publicly available web scale n-gram model by Microsoft: http://research.microsoft.com/en-us/collaboration/focus/cs/web-ngram.aspx.

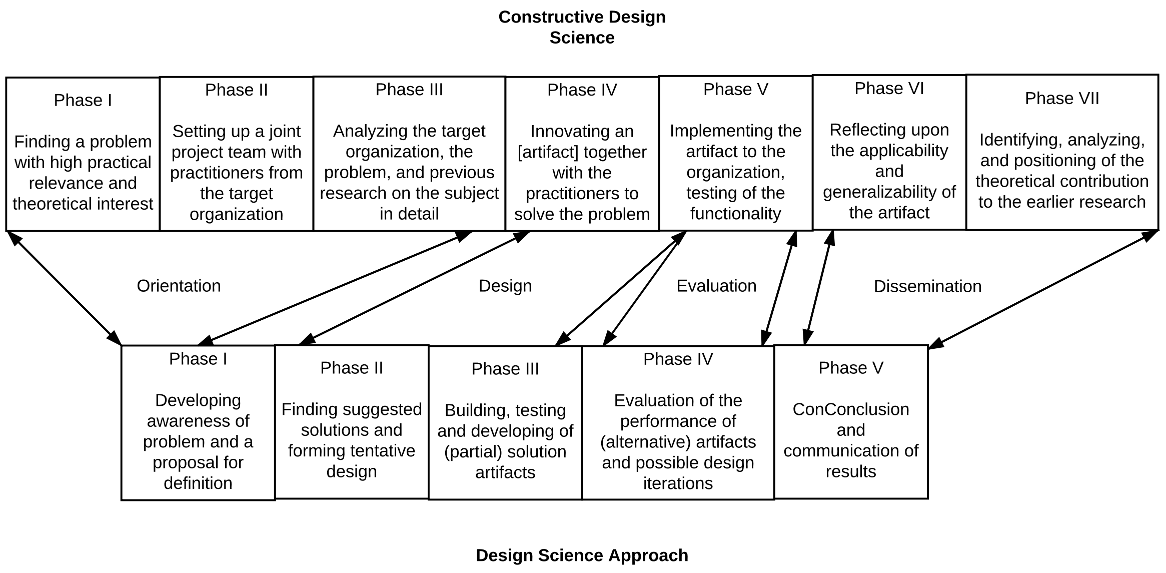
2.4 Related Studies

Before we begin to study, understanding and reading the existing related researches are of great help to us. Some of them introduced the implementation of the keyphrases extraction with other methods. Kazi and Vincent present a survey of the state of the art in automatic keyphrase extraction, examining the major sources of errors made by existing systems and discussing the challenges ahead [14]. In this study, not only included the keyphrases extraction understanding also introduced a lot of different keyphrase extraction approaches. These approaches are mainly divided into two major categories: supervised approaches [1] and unsupervised approaches respectively. This concept has also mentioned in [15]. Research on supervised approaches to keyphrase extraction has focused on two issues: task reformulation and feature design. The goal of keyphrase extraction is to identify the most representative phrases for a document. In other words, if a candidate phrase c1 is more representative than another candidate phrase c2, c1 should be preferred to c2 [14]. Hence, to determine whether a candidate term of the document is a keyphrase, we should do analysis based on statistical and linguistic features. For the supervised keyphrase extraction approach, a document set with human-assigned keyphrases is required as training set. However, human labeling is time-consuming [15]. Existing unsupervised approaches to keyphrase extraction can be categorized into four groups: graph-based ranking, topic-based clustering, simultaneous learning and language modeling. The most instructive for our research is topic-based clustering, which involves grouping the candidate keyphrases in a document into topics, such that each topic is composed of all and only those candidate keyphrases that are related to that topic [14]. Learning KeyCluster and Topical PageRank (TPR) in the survey from Kazi Saidul Hasan and Vincent Ng helps us design the research methodology. KeyCluster underlying hypothesis is that each of these clusters corresponds to a topic covered in the document, and selecting the candidates close to the centroid of each cluster as keyphrases ensures that the resulting set of keyphrases covers all the topics of the document. But this way has a potential drawback: by extracting keyphrases from each topic cluster, it essentially gives each topic equal importance. In practice, however, there could be topics that are not important and these topics should not have keyphrase(s) representing them. At that time, Zhiyuan et al. (2010) propose TPR, an approach that overcomes the aforementioned weakness of KeyCluster [16]. It runs TextRank multiple times for a document. By running TextRank once for each topic, TPR ensures that the extracted keyphrases cover the main topics of the document. The final score of a candidate is computed as the sum of its scores for each of the topics, weighted by the probability of that topic in that document. Hence, unlike KeyCluster, candidates belonging to a less probable topic are given less importance. Stuart et al. (2010) not only shows the development process of different keyphrases extraction methods but and introduces how to generate better stop list [17].

We also found some articles about n-gram technology based keyphrases extraction. However, these articles will be relatively less. Niraj and Kannan present an automatic Keyphrase extraction technique for English documents of the scientific domain [18]. The devised algorithm uses n-gram filtration technique, which filters sophisticated n-grams along with their weight from the words of the input document. To develop n-gram filtration technique, they have used (1) LZ78 data compression based technique, (2) a simple refinement step, (3) a simple Pattern Filtration algorithm and, (4) a term weighting scheme. The entire system is based on statistical observations, simple grammatical facts, heuristics, and lexical information of English language. Kamal present a new method for Bengali keyphrase extraction that has several steps such as extraction of n-grams, identification of candidate keyphrases and assigning scores to the candidate keyphrases [19]. Jinkai et al. (2012) introduced an improved text feature extraction algorithm based on N-gram theory [20]. The algorithm is based on algorithm ideas to deal with text processing and feature extraction making the text features more accurately. The results can be applied to information processing fields, such as text search and web mining. Most of these articles are about N-gram arithmetic analysis, rather than n-gram algorithm in python language. That deepened our research interest because the very similar articles with our research did not exist.

3. Methodology

In this study, we decide to follow a design science research approach. Generally the design science research approach consists of two main stages: the "construct" and "evaluation". In construct stage, it contains the major steps from identify the problem domain to outputting the first solution. The evaluation part plays a role in ensuring the developed solution can satisfy the need to address the target problem as well as fulfilling all the necessary detailed requirements. To explain this method thoroughly, we attach a graph in Figure 1:



Constructive Design Science
Phase I Finding a problem with high practical relevance and theoretical interest
Phase II Setting up a joint project team with practitioners from the target organization
Phase III Analyzing the target organization, the problem, and previous research on the subject in detail
Phase IV Innovating an [artifact] together with the practitioners to solve the problem
Phase V Implementing the artifact to the organization, testing of the functionality
Phase VI Reflecting upon the applicability and generalizability of the artifact
Phase VII Identifying, analyzing, and positioning of the theoretical contribution to the earlier research
Orientation
Design
Evaluation
Dissemination
Phase I Developing awareness of problem and a proposal for definition
Phase II Finding suggested solutions and forming tentative design
Phase III Building, testing and developing of (partial) solution artifacts
Phase IV Evaluation of the performance of (alternative) artifacts and possible design iterations
Phase V ConConclusion and communication of results
Design Science Approach

Figure 1. The process outlines and main activities in the CRA and DSR (adapted and rephrased from Lukka, 2003; 2006 [CRA]; Vaishnavi and Kuechler, 2004 [DSR]) [21]

As shown in the graph, the main concept that we would like to explain is the artifact. The artifact can be existed in various forms: a program, a principle, an application or even an algorithm. But no matter which forms it chooses to appear, it must be able to address the problem or aimed at addressing the problem. In other words, the artifact is also the solution to the previously analyzed problem.

So when we look back to the graph, we can observe that those numerous phases can be summed up into 4 stages mainly. First, it would be the orientation stage. In general, this stage means to get familiar with the challenge or the problem domain. So later it may increase the varieties of choices for us to address the problem. In practice, it consists of three smaller phases including identifying the practical benefits in solving this problem and also finding out the theoretical background, forming up a team and searching for any related information. After this, we come to the design stage, where we build, test and develop the artifact (solution) in order to tackle the problem. But before that, it usually encourages us to look up for the similar cases, because it is highly likely that the people who have studied the same or similar topics can bring valuable experiences, which can guide us in the right direction. Then it is the evaluation stage where we implement the correlated tests to assess the performance of our artifact. The guideline of making evaluations to the solution should follow the principle that whether or not the solution makes any effort or how it can help to tackle the problem. If the answer is yes then it means the solution is accepted then we will see the actual performance. Otherwise, we will return to the stage where the problem that failed the test occurs.

If the evaluation is passed, then it is the final stage: dissemination. In this stage, we mostly reflect on the previous stages and analyze the findings from the entire process. The result will be used as contributions to the earlier research studies.

To apply this research strategy in our thesis, we plan to do the following steps:

- Orientation:
    - Understand the idea of topic extraction automation
    - Identify the feasibility of achieving automatic topic extraction
- Design:
    - Propose to use N-gram analysis to address the issue
    - Check through related literatures about N-gram analysis and topic extraction
    - Build, develop and test the N-gram analysis
- Evaluation:
    - Use related metrics to evaluate the performance of the N-gram analysis
- Dissemination:
    - Reflect on the process to see what could be done better and discuss about the merits and demerits of the solution

One thing we need to point out is that the stages or phases we listed above are all carried out iteratively if it is necessary. This means if there is any mistake spotted in any step, it can always return to the previous step to correct the mistake. This is one of the many advantages of using design science research. By taking this measure, we just need to analyze in which step it caused the failure and then we can start from there instead of starting from the initial stage all over again.

4. Environment

4.1 File type selection

The file type is selected as the Portable Document Format (PDF) since it is widely accepted research paper work format. It supports a large number of data types including texts, formulas, images and table graphs. Although other types of documents such as Microsoft Office can also achieve the same goal, the lack of security measure to preserve the data integrity makes them highly unstable. So under that consideration, we decided to use PDF document as the available data set type to evaluate the development and implementation of our topic extraction process.

4.2 Domain selection

In this paper, we select the autonomous vehicle as the domain to control the boundary of our data set. As one of the most popular topic at the moment, the autonomous vehicle field has attracted a tremendous number of researchers to spend their effort on this subject. The varieties of the regarded topic can range from the effectiveness of a small sensor to building an intelligent autopilot system. Therefore analyzing the topics under this domain enables us with a great number of research studies to select. So it will ensure us to have an adequate size of data set for both development and evaluation.

4.3 File Conversion

The beginning phase of this study is the file conversion. As we mentioned before the domain is aimed at the research papers in autonomous vehicle area, so the selected file type is mainly targeted at the Portable Document Format (PDF). However, due to the unique and fixed layout of PDF document, we find it is tough to directly extract information from it. Therefore, the file type needs to be transitioned into the editable type of formats such as txt, doc or any Microsoft office document type.

Admittedly the Microsoft Office document can be an accepted choice since it supports the image and table transplant which slightly edges out the txt file type. But our purpose is to essentially extract the representative key labels of research papers so the inability of converting image and table can be ignored for our result. So we choose to use TXT file to analyze the data because it is easier for text processing.

- **PDFMiner**

One major task for us to process the PDF document is to convert it to an “editable” file type. Currently, there are libraries available to access PDF files (PDFbox from Apache for instance), also, many Linux distributions provide command-line tools to work with PDFs. In the various options of conversion choices we select PDFminer [15] for this task. One of the advantages of using this tool is that the PDFminer is accurate in converting the PDF files and it preserves the original content as much as possible, which reduces the chances that we can get abnormal results, which are possibly caused by inappropriate conversion operations. Also, PDFMiner is designed in python language, which is adaptive with our programming language application as well. There is also one extra point for using PDFMiner is that it enables to read the PDF document page by page, which also facilitates our work in further stages.

5. Implementation

The N-gram statistical sequence plays a vital role in this research. It handles the main part of our solution, which is to calculate the word's frequency of the document based on the selected phrase length. In n-gram language model, it mainly supports up to 5 grams (words length) for text processing. The term grams indicate one complete English word. The words’ length is represented as unigram to five-gram [20], which means from a single word to a 5 words phrase. Since the unigram, four-gram and five-gram all have their significant drawbacks: the result derived from unigram is lack of uniqueness [22] and the chance of obtaining results from four-gram or -five-gram is relatively rare [23], in order to fit our possible results with our research goal which is to bring a reliable solution, we would skip the implementation on the aforementioned gram choices. As for the trigram, it shares a vast amount of similarity with the bigram and also it has the potential risk of losing shorter sized keyphrases, we would also not implement trigram in this case. Therefore, the development and implementation of our proposed solution will mainly operate on bigram. To illustrate the minor difference from unigram to trigram, we will also provide a comparison chart from in appendix A. It shows the result of extracted topics in three types of a gram from the same PDF document. Hence, we can observe the difference between each gram and have a direct knowledge of each gram’s performance

5.1 The sequence length selection (N-gram selection)

In N-gram statistic sequence there are usually 5 types of grams used. They are unigram, bigram, trigram, fourgram, and fivegram. From the result that we have observed, some grams such as bigram and trigram can have relatively similar results. However the unigram’s result can be unpredictable while the fourgram and fivegram tend to lack its standing point in most of the articles. Here we show different result from unigram to trigram. Under the consideration of reducing complexity, we only show the high-frequency words from one PDF. Here we attach the comparison between unigram to trigram with the same data set in appendix A.

After we conducted the comparison we found out that, the unigram sequence words are too simple and general, which makes it challenging to label the article's topic. For example, a single word like "design", "pattern" etc. It is imprecise to use them label one research paper without putting them together. As for trigram, it seems like that it added one more word, which makes the sequence more specific and detailed. However, many results showed that a lot of, many trigrams are received in a format of one bigram and one decorative word. There is also another condition that happened to us which is after we implement the blacklist to remove those redundant words. The application can barely find high-frequency trigrams in every article. So, due to the above-mentioned reasons we determined to select the bigram as our final evaluation sequence.

5.2 Traverse the document with N-gram

We initiate this step to prepare for the later frequency checking process. At first we use space as a delimiter to separate each word in the converted TXT file and remove all types of illegal signs such as ",", ".", "?" etc. By doing this, the whole content of the document would be words that are only connected by spaces. In addition to that, the application will also find the location of reference and skip that content. The content involving the "@" will also be removed since it mostly indicates the e-mail address for the author so it should be removed in order to keep the content clean. The e-mail address will also be replaced by a space.
Until this step, the processed content is stored into a list L1 and waiting to be matched. Then there will be a new empty list L2 created, it will start to select the words that will be used for matching later from the first letter of the document until it reaches the second space then the selection process is over. Here we set the stopping point at the second space because we are using bigrams, which are supposed to have only one space in between. The stopping point can be flexible by increasing or decreasing the number of spaces to pass by so that we are able to check another type of N-grams.

Then the first phrase p is extracted and will be put into L2. Here all of the content in L1 will check the phrase from first space until the third space and see if it matches with the selected phrase L2L0 (the first element in L2). If the match evaluation succeeded then we increase the value by 1 for the selected phrase counter. And then check the next phrase by increasing the starting point and ending point both by one space until it reaches the end of the document, which has no space to be found. When every round of checking is done, the selected phrase will be recorded with its frequency and they are stored in a new list Lf. And the elements in this list will be sorted by descending order from highest frequency phrase to the lowest frequency one.

After the first round we remove the element in L2L0 and replace it with an upcoming phrase, which starts from the end of first space, and ends with the next second space.
By implementing this measure, we make sure not to omit any phrase in the document, although occasionally it may bring in some noises to interfere with our results. Since it happens in very few times and even if it happens, it does not affect much with the overall accuracy so this concern can be ignored in this case.

5.3 Blacklist

After implementing N-gram analysis with the article, the application gives us a list of high-frequency phrases (since we use bigram). However, the accuracy level at this stage is still very limited since many English common words can be a huge interference here. For example, the words such as “we”, “think”, “after”, “then” have a high chance to appear in many places in an academic setting paper. Therefore eliminating those type of words can be the first step to improving the accuracy.
The initial version of the Blacklist would be the English stop words from [24]. We have thought about coming up with our own blacklist from the scratch, but all of our concluded words are also included on this website so it is more comprehensive to utilize a complete list.

To implement the blacklist, at first we need to establish a list that contains the entire primitive blacklist words that should be filtered out. In our case, we take all of the words from the site as mentioned in the previous paragraph, and store them into the established empty list Lb, and then we take each element of Lb to compare with the acquired n-gram list. If the e1 is identically contained in an element of list Lf, then it indicates that this element may contain the irrelevant results hence it will be disposed from the list.

In Appendix B, we store the latest version of our selected blacklist phrases in a table.

5.4 Whitelist

The whitelist is a further step to eliminate those types of irrelevant topics, which are related to the research study, but they are not representative enough to be the “label” for the study. After we filtered out the irrelevant phrases that could mislead the topic of the research study. Most of the phrases left should be at least connected with our chosen topic (autonomous vehicle) in some way. But still, there will be phrases that are related to the topic but not critical one. It means that there could be some overly general phrases but they are not unique or specific enough to label the concept or main idea of the research study.

By implementing the white list, we are able to proceed to eliminate the “related but insignificant” words, which still has relevancy to our autonomous vehicle domain context further to improves the accuracy of the labels. In order to accomplish this, we need to implement the N-gram analysis again but now we implement it page by page instead of on the whole article. The idea behind that is after observation, we found that if a phrase almost exists in every page or section of the article, it is usually a general concept or a triggering point that enlightened the author to write such a paperwork, but it tends not to be the exact topic or technology that the author is explaining about. So to filter out these types of phrases, it can help greatly in preserving the true critical phrases.

To implement the whitelist page by page, we need to invoke some functions from PDFminer to parse the PDF document again. The reason for this is that the previously converted TXT file has been modified after the N-gram analysis hence it does not have the same layout as the original document has. So some parts of PDFminer such as PDFparser, PDFResourceManager, PDFDevice, PDFAggregator, and PDFInterpreter will be reused here to parse the original document again. And then use a similar frequency checking method. However, unlike what we do in the first n-gram checking, this time, we need to create one list from page 1 to the last page, we suppose as the page n: Lp1 - Lpn. Then we will get a list of high-frequency phrase for every page.

The criterion to determine whether or not the element should be placed in whitelist is that we calculate the average number of pages called “Pa”. And then we check if the high-frequency

phrase in page n also appears in other pages. If yes then we increase the value for this high-frequency phrase (for example we call element 1 as "e1") counter by 1: CLp1e1++. And if the final value of CLp1e1 > Pa, then e1 is evaluated as a phrase that should be placed in the whitelist. And we apply this rule to any other element inside the each page's list.
After this is done, we utilize the Lw to filter the elements in Lf, following the same rule that how we filter the blacklist.

After we implement the rule of selecting whitelist candidate, we will apply the rule on every file in the training set. Then we make a collection of the filtered whitelist topics from each file. In this stage, we will create a new list to contain all of those filtered whitelist topics as the same function the blacklist has. However, we will perform a relevancy check before we actually store them into the newly created list because there could be some whitelist candidates that are less typical to be applied as a general rule for other documents. The guideline to select the whitelist phrase is that the suitable instance is that it should not be a unique identifier to any specific study, but also it should have some relevancy to the autonomous vehicle to some extent. When this stage is done, it means that the construct part in this round is over. Then next should be the evaluation stage.

In appendix C, we also provide a table containing all of our observed whitelist phrases in it.

- **Example**

File name: file_example.pdf
File type: PDF
File title: enhance threat assessment and vehicle stability to ensure the active safety for autonomous vehicles
Default Keywords: Active safety, automated vehicles, global chassis control, predictive control, threat assessment, vehicle stability.

In addition, we display more details with implementing the solution in table below:

OS	IDE	Programming Language	External Library	Input file format	Output file format
Ubuntu 14.04LTS	Pycharm Community edition	Python	PDFminer	PDF	TXT

The n-gram analysis serves as the fundamental concept of our topic extraction idea. It consists of several parts including the frequency checking, the blacklist filtration, and the whitelist filtration. The first step of initializing the N-gram analysis is to get the raw hand of data which is the frequency checking without any conditions. The only thing we need to specify is the gram size. As we discussed before, we will display the result for one randomly selected document. We will display the same result list after each step so that we can compare the difference between implementing each filtration technique. In below we display the result:

✶ High frequency phrases extraction:

Bigram	Count (times)
of the	72
the vehicle	59

in the	37
the driver	30
..	20
esc system	14
1 and	14
in Fig	13
roadway departure	10
predictive prevention	8
critical situation	8

Table 1.

As we can see in table 1, we have a list of the Bigram's counting times with its name. We choose not to show the entire result because it will be exhaustive and it can take up too much space in the study. In order to classify the different relevance of the gram, we assign different colors to present its level of relevancy here. From least relevancy to the highest:

Red - Orange- Blue - Green

So after the n-gram analysis, we easily identify that the ".." in the result should never be shown in the result and it needs to be disposed of right away. As well as the orange group, they are obviously weakly connected with any possible topic that this report could be related to. So here we implement the blacklist to filter the result and then we would have a much cleaner list:

- BLACKLIST Filtration:

Bigram	**Count (times)**
esc system	14
roadway departure	10
predictive prevention	8
critical situation	8
ieee transactions	7
intelligent transportation	7
active safety	7
departure avoidance	7
vehicle control	6

Table 2.

In table 2 we can see that the blue colored result still remains on the list. We call these words the whitelist grams. These grams seemingly look like that they can fall into the autonomous vehicle category, or they are somewhat related to the topics of the study, however they tend to be more abstract and pointless than the actual grams, this can be easily observed by comparing the green colored words with the blue colored words, with a purpose of illustrating the topics of the study.

So in order to further filter these grams out, we implement the whitelist in this stage.

✶ WHITELIST Filtration

Bigram	Count (times)
roadway departure	10
predictive prevention	8
active safety	7
departure avoidance	7
vehicle control	6

Table 3.

So as we can see in table 3, after the whitelist filtration, the grams left in this list have a concrete meaning and each gram actually represents a part of the topic for the study.

Also, if we compare these 5 words together with the file topic keywords, which are concluded by the author of this paperwork, we can surprisingly find that most of our automated words match with the keywords defined by the author. For the complete results, it is stored in appendix D.

For a closer look at the program, we provide a Github link to the application so it can be downloaded and tested by the users themselves. The Github link can be found in appendix F.

6. Evaluation

To evaluate our solution, we conducted two main types of evaluation tests. One is the evaluation for the precision checking and the other one is for the running speed comparison. The purpose of running these two types of tests is to examine the actual precision rate and the running time consumption by comparing our automated results to the manual results.

Before we conduct the evaluation, first we specify the preparations for conducting such evaluation. There will be test set contains a number of documents to test our solution. The documents inside the test set all have the same domain and file type, only the topics are different from each other. And also in order to compare our automated result, we have a manual topic extraction result to be used. As we know that the human labeling takes more time but it is more accurate than the automated approach, so we believe that comparing our automated result with the manual result will give us a good accuracy rate.

6.1 Validation for the efficiency

In order to evaluate the performance of our solution, we need to make sure the result of manual extraction as unbiased as possible, we conduct the manual checking process twice by each of the group member, and then summarize our results by saving the agreements and discussing on the disagreements. In the end, we will get a commonly approved manual result by us. Admittedly if we can involve in more people to take part in the manual checking process and collect their results to combine with ours will provide us a more convincible result, however due to the time consideration and our selected research paper difficulty, we believe that the result obtained by two people are still reliable and accurate enough to be the actual topics of the corresponding research studies. Additionally nowadays many authors are asked to add their own keywords/keyphrases to their works [12] so the reliability of our selected topics should be trustworthy.

What's more to the manual topic extraction can help us to evaluate our automated solution is that we need to utilize the manual result to perform the "precision test" to our solution. Since our research study is highly attached to the topic of identifying the relevancy between our automated results to the manual result under information retrieval context, we decide to use the "precision and recall" [25] as the standard relevancy evaluation approach [25]. That would be the most relevant and effective evaluation to our topic to explain the precision calculation in conceptual formula ①**.**

The "precision and recall" measures two aspects under the information retrieval context: Precision a.k.a Positive Predictive Value (PPV) is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search while Recall a.k.a sensitivity is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents. Here in our case, since our default setting is to preserve the high frequency phrases from n-gram analysis, all the retrieved documents here would automatically be assumed as Positive value. Thus the Recall concept will always have the 1.0 score so we will not calculate the sensitivity for our result.

In our report, we apply this theory to check if the automated result is aligned with our discussed manual result. Before we bring in more specific details about this approach, we would like to make a concept list for all the possible issuing elements:

- ➢ P = Positive, the assumed relevant result
- ➢ N = Negative, the assumed irrelevant result
- ➢ TP = True positive, the correctly identified result
- ➢ FP = False positive, the incorrectly identified result
- ➢ TN = correctly rejected result (0 in our case)
- ➢ FN = Incorrectly rejected result (0 in our case)
- ➢ Precision (PPV) = as in Positive Predictive Value, is the fraction of retrieved instances that are relevant.
- ➢ Recall = also as sensitivity, is the fraction of relevant instances that are retrieved (As aforementioned it is 1.0 in our case which means all relevant documents are retrieved).

So in our data set, the positives are the automated result because they are the expected outcomes by implementing our approaches. There will be no negatives in this case since we would already filter out the irrelevant results by selecting to obtain the result with high frequency. The reason of it is that we already know that the low-frequency phrase would be irrelevant to our topic in this case. Hence, there is no point to make an observation of the irrelevant value again. Next, we would like to introduce the actual formula here to illustrate how we will carry out the precision test with our research topic in equation ②.

The true positive value is that we compare our automated result to the manual result, if one element from automation is semantically equal to the other element from manual labeling, then that automation element is truly positive. Ideally, we will receive more extracted topics from the automation rather than the manual work, so we will take the size of the manual result as the total documents number. Then we make an observation to see how many automated results fit with the manual results. If any of them matches then it will be considered as true positive, otherwise it will become false positive value (FP). Then we will use the above formula to calculate the precision for each result for the test set. The Test outcome positive is a set, which consists both the true positive and also the false positive.

$$① \; \boldsymbol{Precison} = \frac{|\{\boldsymbol{relevant\ document}\} \cap \{\boldsymbol{retrieved\ documents}\}|}{|\{\boldsymbol{retrieved\ documents}\}|}$$

$$② \; \boldsymbol{Precision\ (PPV)} = \frac{\sum \boldsymbol{True\ positive}}{\sum \boldsymbol{Test\ outcome\ positive}}$$

- Example

To answer this question, we implement the "precision and recall" method. The purpose of this approach is to measure the relevancy of a set of retrieved documents. It exactly fits to be an effective evaluation approach for us. However, as we have mentioned before, the recall rate is not calculated for our result so we mainly lay emphasis on the precision test.

In our result, typically we will receive a long list of phrases after the implementation of our automatic topic extraction process. Under the consideration of possible future use, which we are supposed to provide the user with several clear and significant topics for the file instead of throwing them with a long exhausting list to read. We decided to only accept the extracted topics with highest frequency.

So we have selected 5 automated topics for each paper, and then we compare each file's automated result with the manual result. The automated topics are identified as the positive value, and the manual result are used to decide whether or not the automated topics are true positive or false positive. And then according to the formula, we use the true positive topics to divide with the sum of true positive and false positive topics then we can get the precision rate our automated results for this paper. Based on the precision rate we have collected for each paper, we set up 3 relevancy ranges to categorize them from high to low.

High Relevancy: 66.7% - 100%
Medium Relevancy: 33.4% - 66.6%
Low Relevancy: 0% - 33.3%

And by analyzing the distribution we can speculate the performance of our approach. In below we show the data for the test set in table 4:

Precision type (PT):	Precision Rate (PR):	Number of Files (NoF):	Precision Level (PL):
1	100%	28	High
2	75%	3	High
3	66.7%	19	High
4	50%	22	Medium
5	33.4%	14	Medium
6	25%	4	Low

7	0%	10	Low

Table 4.

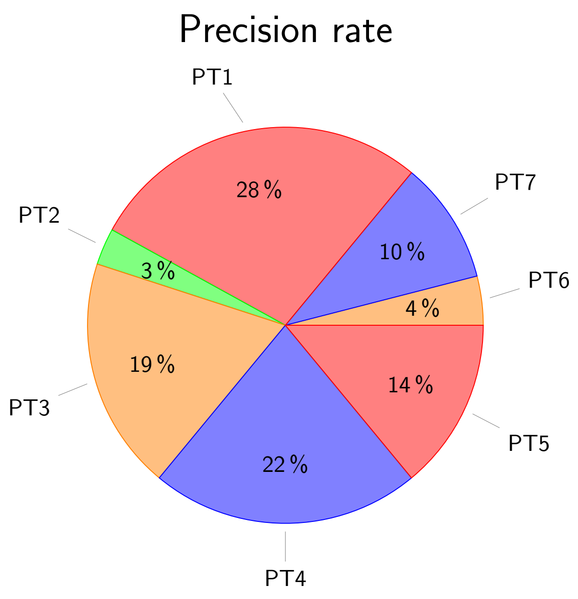


Figure. 2.

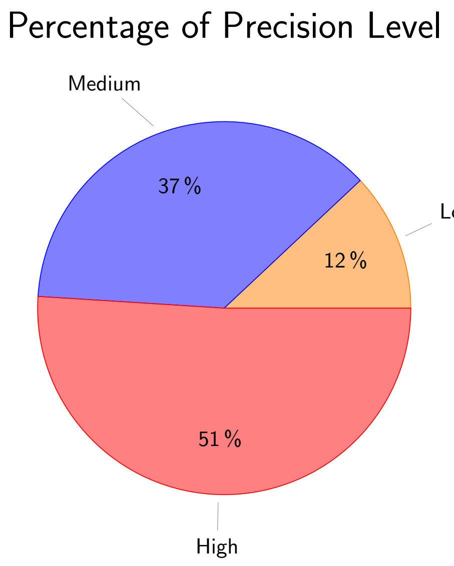


Figure. 3

As shown in the graph above, the chart Figure.2 illustrates the percentage of each Precision Type in the entire test set. It is clear to see that the 100% precision rate takes the most percentage of the first graph, and then follows by the third precision group, which is 66.7%. It means that our solution can be most effective in those two groups.

In Figure.3, it displays the percentage by each Precision Level, we can see that the high Precision Level group also takes the major part by slight over the half percentage, and then the medium PL takes the next major part by around 36% while the low precision group takes the least.

In explanation of this table, we notice that the type of precision rate does not have many changes when it comes to the variety. One of the reasons that let this happen is as we mentioned before, we have only selected 5 high frequency automated results and unfixed number of manual results (tend to be less than five).

6.2 The speed of automated approach

We also intend to make an observation to the application's running speed. We selected 21 papers from the test set. The papers are divided into three different groups based on its content size. The number of each group is evenly distributed and content size is ranging from one or two pages to a large size, which could be over 10 pages. The topic of the evaluation paper is randomly selected. The result for each file's processing time in 3 file size groups is shown below:

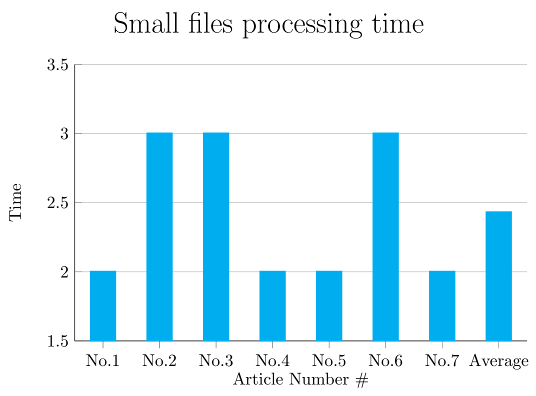


Figure 4.

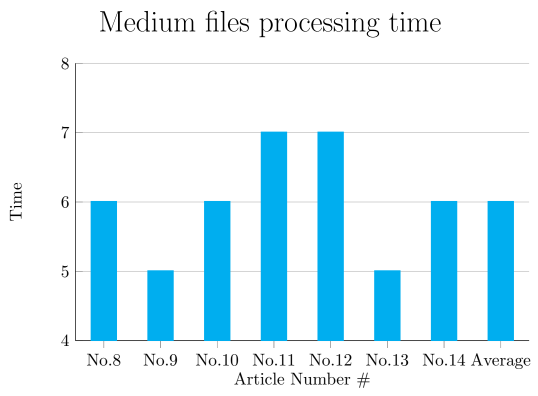


Figure 5.

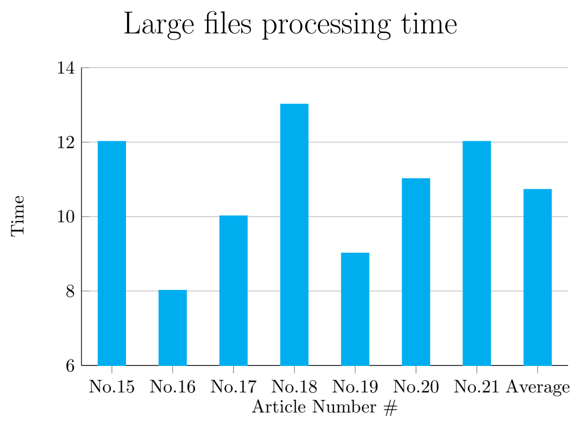


Figure 6.

So after we run the application on the selected documents and record the result. We found that the average time is between several seconds to less than 15 seconds as the maximum. In Figure.4, the small file group is averaging around 2.4 seconds while the medium group is averaging around 6 seconds in Figure.5, the large file group takes the most time which spending around 11 seconds for the processing time in Figure.6.

7. Discussion

For the application running time, we can observe that the processing time increases with the file size. However, the processing time did not exceed 15 seconds even when the largest file size is selected. This result tells us that our automated approach can keep in a relatively stable and fast processing speed for most of the documents.

The next thing we would like to talk about is the precision of using text-based background setting. Using a text-based strategy is not a brand-new idea. In some similar works such as [14], [26], the researcher has proposed to use the text-based topic extraction idea for broadcasting news speech. Although in those studies they do not emerge with the idea of establishing blacklist and whitelist as we did. Instead, they use their trained phonetically balanced sentences and dialogue read by 50 male speakers and over ten thousands of utterances then to combine with n-gram language models. For their approach, they can reach a prominent result of less than 30% error rate [3] for the word checking accuracy. This shows us that the statistical n-gram language models definitely have its edge in a text-related world

such as the broadcast speech field, but it is not that much different than the academic research studies since both of them tend to have a clear topic and several key labels.
So to compare their result with our result, it is obvious to find out that our approach even has a higher checking efficiency with a 51% high PR and 36.7% medium PR. The reason is because of the idea of implementing a blacklist and whitelist filter to double eliminate the irrelevant as much as possible, and also we conducted this process by iterations so the irrelevant results are controlled at the maximum level of our competence in the given time.

Besides the efficiency of our topic extraction model, we also have presented the result for how our approach performed when it comes to the time-saving part. From the statistics that we have explained, the automated approach surpasses the manual checking speed by a big margin. Regardless of the volume of the document, the automated extraction almost ensured to out speed the manual extraction. However one thing from the graph that we can notice is that the automation speed is aligned with the document's size. E.g. when the document has a small content the extraction performs in a blink of an eye while the document has a large size of content then the extraction speed is dragged down in an obvious but not dramatically way. However, the manual checking speed seems not to be influenced by the volume of the document so it keeps in a relatively steady level. However, even for the largest size document, the automation performing speed still double leading the manual checking speed.

7.1 Validity threats

The major issue of our approach is to build a perfect blacklist and universal whitelist. As we mentioned before, the blacklist is meant to filter the irrelevant result for our study, namely most of the English common words would be our primary target to fix with. Also we have observed different scenarios such as the reference information overlapping (the author or other parts of the reference repeatedly appear in the main content), this issue is fixed by ignoring the whole reference section when we perform the topic extraction. One thing that is not well considered in our model is the ability to address the abbreviation issue for the user. Because right now it is very popular that people like to define their own abbreviations to describe their topic, and those acronyms tend to be extracted by our model since they have a high opportunity to be talked many times in the study. Hence this can create a problem for other users because they can be unfamiliar with the acronyms and it may not help them to comprehend the topic of the study. One assumption for why this scenario could happen is that we mainly used only one or two types of grams to extract the labels. It limited the chance of discovering the acronyms that could have a different length than our selected gram. This issue sometimes is not that critical since our topic extraction takes not only one word or phrase for the user but a set of n-gram words. So in most of the time, the user can try to understand the topic of the study by viewing the obtained results as a whole. But if the author has defined multiple acronyms in the study, then it is highly likely to extract the true explanatory n-gram words to illustrate the study.

Another validity threat is the algorithm for the whitelist. Right now we have two rules to complete the whitelist. One is to semantically judge the phrase to see if it is too abstract or too general to describe the topic. The other one is to filter out the high-frequency phrases, which keep on appearing on every page. The second measure is strong with capturing whitelist words because if the words appear on every page of the document with high frequency, it indicates that this phrase might be only invoked to explain a theory rather than being used as a development part of the study. The result that we presented earlier showed how effective our whitelist approach is. Nevertheless, we still feel that it is possible to miss some whitelist words by implementing the previous algorithm. Because sometimes in the study, the author tends to mention the whitelist words in some certain sections. For example, "autonomous vehicle" (The field name itself can be in whitelist because it is too general) in the introduction, related work, discussion and conclusion. So this phrase does not appear on every page so it will not be considered in the whitelist. But still it does appear in many pages. Due to some technique difficulties, it is infeasible to realize the ideal approach in the given period of time.

7.2 Future works

We have several future outlooks with our topic extraction process. The first good thing improve is the ability to recognize acronyms as aforementioned. As we know that it is quite common for the scholar or research to define his or her own abbreviations to simplify the workload. If our approach is able to recognize it, then the reader will have a higher chance to find his/her desirable articles without acquiring the full knowledge to that article.

And also in the future, we intend to improve the accuracy for the whitelist selection. In order to achieve that, we think it can be realized by altering the criteria for selecting the "high-frequency phrase, which appears on every page of the document" into "if the high-frequency phrase appears in a certain level of ratio" then we put it in the whitelist. By that it means we can set up more evaluation condition to determine whether or not the high-frequency phrase should be placed into the whitelist. One condition might be that if the high-frequency phrase appears every 2-3 pages of the document, or the high-frequency phrase do not appear regularly but it shows up in over 60% of the total pages. Then we can consider putting those words into the whitelist. We believe that the accuracy level can be greatly improved by enforcing these measures.

8. Conclusion

Based on the result that we have gained, we can say that text-based topic extraction approach can be very powerful and time-saving if the evaluation conditions are set up right. Our automated topic extraction process is able to perform topic extraction in a fast period of time, while still keeps the accuracy to an accepted level. Although the automated approach cannot always ensure to extract the highly accurate labels, we believe that our approach can be a good start for the topic extraction field. With continuous development over time, the approach can be completed to identify more accurate labels if the blacklist and whitelist are actively updated and carefully selected. For the minimal task of achieving the automation topic extraction, our approach has its edge in processing a number of PDF documents and presents the potentially interested topics for the reader. The reader saved the work to do the reading for articles that they are unsure if it is their target.

Moreover, we also believe that if we can find and build better-defined conditions for the topic extraction approach when it comes to the irrelevant information filtering, our approach can even bring up more accurate and fast result. As more reasonable conditions are applied, it is more likely for our topic extraction to reach a much higher accuracy rate for more topics.

Referennces

[1] M. W. Berry and J. Kogan. "Text mining: applications and theory". *In Proc. of the Natural language processing Conf. Computer science.* pp.1–20. 2010.

[2] P. D. Turney. "Learning algorithms for keyphrase extraction". Institute for Information Technology. National Research Council of Canada. Ottawa, Ontario, Canada. October 1999.

[3] K. Tumer and J. Ghosh. "Estimating the Bayes error rate through classifier combining". *In Proc. of the 13th International Conference on Pattern Recognition*. Vol. 2. pp. 695–699. Vienna, Austria. 1996.

[4] Z. Liu, W. Huang, Y. Zheng and M. Sun. "Automatic Keyphrase Extraction via Topic Decomposition". *In Proc. of the Empirical Methods in Natural Language Processing. Computational Linguistics*. pp. 366–376. MIT, Massachusetts, USA. October 2010.

[5] E. Pianta and S. Tonelli. "KX: A flexible system for Keyphrase eXtraction". *In Proc. of the 5th International Workshop on Semantic Evaluation. Computational Linguistics.* pp. 170–173. Uppsala, Sweden. July 2010.

[6] E. D'Avanzo and B. Magnini. "A keyphrase-based approach to summarization: the LAKE System at DUC-2005". *DUC Workshop at the Human Language Technology Conf. HLT/EMNLP'05.* Vancouver, B.C., Canada. 2005.

[7] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin and C. G. Nevill-Manning. "Domain-Specific Keyphrases Extraction". *In Proc. of the Sixteenth International Joint Conference on Artificial Intelligence*. pp. 668–673. San Francisco, CA, USA. 1999.

[8] P. Bhaskar, K. Nongmeikapam and S. Bandyopadhyay. "Keyphrase Extraction in Scientific Articles: A Supervised Approach". *In Proc. of the COLING 2012: Demonstration Papers*. pp. 17–24. Mumbai. December 2012.

[9] E. V. Rheenen. (2016). *Manual versus auto-classification.* [Online]. Available: https://www.xillio.com/blog/manual-versus-auto-classification

[10] S. Jones and G. W. Paynter. "Automatic extraction of document keyphrases for use in digital libraries: Evaluation and applications". *ACM digital library. American Society for Information Science and Technology archive*. Vol. 53. Issue 8. pp. 653–677. August, 2002.

[11] S. N. Kim, M. Y. Kan. "Re-examining Automatic Keyphrase Extraction Approaches in Scientific Articles". *In Proc. of the 2009 Workshop on Multiword Expressions. ACL-IJCNLP 2009.* pp. 9–16. Suntec, Singapore. August 2009.

[12] P. D. Turney. "Learning Algorithms for Keyphrase Extraction". Information Retrieval 2(4). 2000. 303-336.

[13] K. S. Hasan and V. Ng. "Automatic Keyphrase Extraction: A Survey of the State of the Art". *In Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics*. Vol. 1. pp. 1262-1273. 2014.

[14] Z. Liu, W. Huang, Y. Zheng, and M. Sun. "Automatic keyphrase extraction via topic decomposition". *In Proc. of the 2010 Conference on Empirical Methods in Natural Language Processing. Computational Linguistics.* pp. 366–376. MIT, Massachusetts, USA. 2010.

[15] Y. Shinyama. (2007). *PDFMiner: Python PDF parser and analyzer.* [Online] Available: http://www.unixuser.org/~euske/python/pdfminer/

[16] N. Kumar and K. Srinathan. "Automatic keyphrase extraction from scientific documents using N-gram filtration technique". *In Proc. of the eighth ACM symposium on Document engineering*. Pp. 199–208. New York, NY, USA. 2008.

[17] K. Sarkar. "An N-Gram Based Method for Bengali Keyphrase Extraction". *The series Communications in Computer and Information Science*. Vol. 139. pp. 36–41. Patiala, India. March 2011.
[18] J. Yu, Y. Wang and H. Chen. "An improved text feature extraction algorithm based on N-gram". Modem Computer. Vol. 23. 2012.
[19] J. Dumoulin. "Smoothing of ngram language models of human chats". *In Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conf*. pp. 1–4. 2012.
[20] H. Wang and B. Lang. "Online Ngram-enhanced topic model for academic retrieval". *In Proc. of the Sixth International Conf. ICDIM*. pp. 137–142. 2011.
[21] K. A. Piirainen and R. A.Gonzalez. "Constructive Synergy in Design Science Research: A Comparative Analysis of Design Science Research and the Constructive Research Approach". *Liiketaloudellinen Aikakauskirja*. Vol. 3-4. pp. 206–234. 2014.
[22] W. Zhou, B Yuan, Z. Miao, W. Zhu and W. Liu. "Detecting errors in Chinese spoken dialog system using ngram and dependency parsing". *In Proc. of the IET 2nd International Conf. Wireless, Mobile and Multimedia Networks (ICWMMN 2008)*. pp. 532–535. Beijing, China. 2008.
[23] O. Serban, G. Castellano, A. Pauchet, A. Rogozan and J. Pecuchet. "Fusion of Smile, Valence and NGram features for automatic affect detection". *In Proc. of the 2013 Humaine Association Conf. Affective Computing and Intelligent Interaction (ACII)*. pp. 264–269. 2013.
[24] A. Z. Broder, S. C. Glassman, M. S. Manasse and G. Zweig (1997). *Syntactic Clustering of the Web*. [Online] Available: http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-TN-1997-015.pdf
[25] B. Bhatta. "Research Methods in Remote Sensing". *Springer Dordrecht Heidelberg New York London. Springer Science & Business Media*. pp.50–125. 2013.
[26] T. Nguyen and M. Kan. "Keyphrase extraction in scientific publications". *In Proc. of the 10th International Conference Asian Digital Libraries*. pp. 317–326. 2007.

Appendix A:

- **Unigram**

Label	Design	Pattern	Execution	Tactical	Level	Vehicle	Commander	Individual	Components
Count	13	13	10	7	17	6	6	5	5
Frequency	0.328448711	0.328448711	0.252652855	0.176856998	0.429509853	0.151591713	0.151591713	0.101061142	0.101061142

- **Bigram**

Label	design pattern	execution level	tactical level	vehicle commander	individual component	component commands
Count	13	10	7	6	5	4
Frequency	0.328448711	0.252652855	0.176856998	0.151591713	0.101061142	0.083032143

- **Trigram**

Label	a design pattern	the Execution level	the Tactical level	the Vehicle Commander	of individual components	control of individual	individual components within	mission specification system	into appropriate component
Count	13	10	7	6	6	4	3	3	3
Frequency	0.328448711	0.252652855	0.176856998	0.151591713	0.151591713	0.101061142	0.075795856	0.075795856	0.075795856

Appendix B (Blacklist):

a	added	anybody	because	believe	co	doesn\'t	enough
about	adj	anyhow	been	below	com	doing	especially
above	affected	anymore	before	beside	come	done	et
after	affecting	anyone	being	besides	comes	don\'t	et-al
again	affects	anything	below	between	contain	down	etc
against	after	anyway	between	beyond	containing	downwards	even
all	afterwards	anyways	both	biol	contains	due	ever
am	again	anywhere	but	both	could	during	every
an	against	apparently	by	brief	couldnt\'d	darpa	everybody
and	ah	approximately	b	briefly	cid	de	everyone
any	all	are	back	but	cmd	dolan	everything
are	almost	aren	be	by	did	dl	everywhere

aren\’t	alone	arent	became	can\’t	didn\’t	each	ex
as	along	arise	because	cannot	do	e	except
at	already	around	become	could	does	each	er
a	also	as	becomes	couldn\’t	doesn\’t	ed	era
able	although	aside	becoming	c	doing	edu	few
about	always	ask	been	ca	don\’t	effect	for
above	am	asking	before	came	down	eg	from
abst	among	at	beforehand	can	during	eight	further
accordance	amongst	auth	begin	cannot	date	eighty	f
according	an	available	beginning	can\’t	did	either	far
accordingly	and	away	beginnings	cause	didn\’t	else	few
across	announce	awfully	begins	causes	different	elsewhere	ff
act	another	al	behind	certain	do	end	fifth
actually	any	be	being	certainly	does	ending	first
followed	g	had	heres	hed	i	immediate	five
follows	gave	hadn\’t	herself	hence	i\’d	immediately	fix
for	get	has	him	her	i\’ll	importance	itself
former	gets	hasn\’t	himself	here	i\’m	important	i\’ve
formerly	getting	have	his	hereafter	i\’ve	in	ieee
forth	give	haven\’t	how	hereby	if	inc	ion
found	given	having	how\’s	herein	in	indeed	io
four	gives	he	h	heres	into	index	iv
from	giving	he\’d	had	hereupon	is	information	ins
further	go	he\’ll	happens	hers	isn\’t	instead	ii
furthermore	goes	he\’s	hardly	herself	it	into	k
fig	gone	her	has	hes	it\’s	invention	keep
feb	got	here	hasn\’t	hi	its	inward	keeps
his	gotten	here\’s	have	hid	itself	is	kept
hither	howbeit	j	haven\’t	him	let\’s	isn\’t	kg
home	however	just	having	himself	i	it	km
how	hundred	million	he	no	id	itd	know
l	me	miss	no	nobody	ie	it\’ll	known
largely	more	ml	nor	non	if	its	knows
last	most	more	not	none	i\’ll	obviously	k|k
lately	mustn\’t	moreover	n	nonetheless	im	of	seven
later	my	most	na	noone	says	off	several
latter	myself	mostly	name	nor	sec	often	shall
latterly	m	mr	namely	normally	section	oh	she
least	made	mrs	nay	nos	see	ok	shed
less	mainly	much	nd	not	seeing	okay	she\’ll
lest	make	mug	near	noted	seem	old	shes
let	makes	must	nearly	nothing	seemed	omitted	should
lets	many	my	necessarily	now	seeming	on	shouldn\’t
like	may	myself	necessary	nowhere	seems	once	show

liked	maybe	mu	need	ne	seen	one	similar
likely	me	mi	needs	predominantly	self	ones	similarly
line	mean	respectively	neither	present	selves	only	since
little	means	resulted	never	previously	sent	onto	six
\’ll	meantime	resulting	nevertheless	primarily	showed	or	slightly
look	meanwhile	results	new	probably	shown	ord	so
looking	merely	right	next	promptly	showns	other	some
looks	mg	run	nine	proud	shows	others	somebody
ltd	might	thereto	ninety	provides	significant	otherwise	somehow
something	stop	thereupon	though	put	significantly	unlikely	someone
sometime	strongly	there\’ve	thoughh	too	under	until	somethan
sometimes	sub	these	thousand	took	until	unto	useful
somewhat	substantially	they	throug	toward	up	up	usefully
somewhere	successfully	theyd	through	towards	u	upon	usefulness
soon	such	they\’ll	throughout	tried	un	ups	uses
sorry	sufficiently	they\’re	thru	tries	under	us	using
specifically	suggest	they\’ve	thus	truly	unfortunately	use	usually
specified	sup	think	til	try	unless	used	very
specify	sure	this	tip	trying	unlike	was	v
specifying	sg	those	to	ts	while	wasn\’t	value
still	w	thou	together	twice	who	we	various
went	want	whence	wherever	two	who\’s	we\’d	\’ve
were	wants	whenever	whether	what\’s	whom	we\’ll	very
werent	was	where	which	when	why	we\’re	via
we\’ve	wasnt	whereafter	while	when\’s	why\’s	we\’ve	viz
what	way	whereas	whim	where	with	were	vol
whatever	we	whereby	whither	where\’s	won\’t	weren\’t	vols
what\’ll	wed	wherein	who	which	would	what	vs
whats	welcome	wheres	whod	whom	wouldn\’t	y	you
when	we\’ll	whereupon	whoever	whomever	youre	yes	you\’d
wish	wont	wouldnt	whole	whos	yours	yet	you\’ll
with	words	www	who\’ll	whose	yourself	you	you\’re
within	world	z	&	why	yourselves	youd	you\’ve
without	xi	zero	,	widely	you\’ve	you\’ll	your
yourselves’	x	/	.	willing	yi	your	yours
							yourself

Appendix C (Whitelist):

actual,parameter	based,method	decision,maker
area,extracted	based,models	design,choices
automated,driving	based,driving	design,decision
automated,mode	car,guidance	design,model
automated,vehicle	car,sharing	design,strategy
automatic,vehicle	car,test	development,processes
automation,driver	center,point	driver's,manual
automation,levels	change,request	driving,cars
automation,phase	communication,based	driving,mode
automation,science	communication,systems	driving,period
autonomous,car	computer,society	driving,task
autonomous,guided	context,relevant	dual,tree
autonomous,units	control,systems	maximum,velocity
autonomous,vehicles	correct,identification	mobile,autonomous
autonomously,driving	current,frame	mobile,robots
autonomous,robots	customized,approach	model,updated
original,image	controlled,vehicle	driving,strategy
unmanned,vehicle	highest,priority	estimation,concept
short,term	highly,automated	evaluated,vehicle
solution,exists	highway,system	expected,sizes
success,rate	intelligent,systems	experimental,design
system,development	intelligent,transportation	experimental,scenario
system,model	intelligent,vehicles	principal,component
systems,technology	vehicle,data	proposed,method
semi,autonomous	vehicle,guidance	public,road
test,vehicle	vehicle,model	public,transport
transport,systems	vehicle,number	publication,citation
transportation,systems	vehicle,position	real,data
type,systems	vehicle,states	real,time
uncorrelated,effects	vehicles,communicates	relaxation,problem
urban,car	feature,function	relevant,road
urban,challenge	front,vehicle	road,urban
ground,vehicle	fully,automated	lead,car
guided,vehicle	making,system	leader,vehicle
journal,content		lead,vehicle

Appendix D (Evaluation result):
https://docs.google.com/spreadsheets/d/17vRNwPMDByTdLTzBJqQaIBdwq6SkWMBku30Iorv9rdg/pubhtml?gid=1295569155&single=true

Appendix E

Github:
1. https://github.com/maoyi/topic.git (For cloning by Github or checkout by SVN)
2. git@github.com:maoyi/topic.git (For using SSH key and a passphrase from an account)