TOWARDS APPLICATION OF TEXT MINING FOR ENHANCED POWER NETWORK DATA ANALYTICS - PART II: OFFLINE ANALYSIS OF TEXTUAL DATA

Yushi Chen[1], Jelena Ponocko[1,], Nikola Milosevic[2*], Goran Nenadic[2], Jovica V. Milanovic[1]*

[1]*Electrical Energy and Power Systems Group, University of Manchester, Manchester, United Kingdom*
[2] *School of Computer Science, University of Manchester, Manchester, United Kingdom*
jelena.ponocko@manchester.ac.uk

Keywords: Text mining, text categorisation, text summarisation, data analytics, power distribution network

Abstract

Text mining is a subdivision of data mining technologies used to extract useful information from unstructured textual data. In recent years, power distribution networks have become more complex due to the versatile consumer demand and integration of distributed energy resources. This has led to the need for enhanced data processing and analysis, i.e., data analytics, in distribution system studies. This paper for the first time explores the feasibility of application of text mining methods as a part of power system data analytics. The focus is on identifying and describing the steps that need to be taken for the knowledge extraction from large offline textual document collections and on demonstrating the effectiveness of the whole process if undertaken by a power system engineer, i.e., a non-specialist in the area of text mining.

1 Introduction

With the increasing involvement of new actors, market rules and generation and load technologies the, electrical power distribution system is becoming increasingly more complex This has led, among the other, to the need for efficient extraction, classification and processing of existing information and knowledge about the network and methods used for network maintenance, operation and control. In order to facilitate efficient information extraction text mining methodologies offer potentially feasible solution.

Text mining is a process of discovering the underlying knowledge from textual data and condensing it into certain types of structured information [1]. Critical factor during the information extraction process is the way of quantification of useful information. A general concern is what type of information is to be obtained and to what level sparse knowledge in a document can be concentrated and analysed. This paper attempts to answer these questions from the power network point of view. At this preliminary stage of application of text mining in general power system area, it was assumed that the research papers are the major sources of informative textual data in the area so the extractive or abstractive mining [2] is the primary task. By referring to conventional text mining process, a set of scenarios is developed to achieve high effectiveness in knowledge discovery. The application of text mining, i.e., the extraction of the key information at both paper and sentence level, from offline document collections is illustrated by focusing on electrical power distribution system. The methodology illustrated in this paper can be seen as an extension to the web crawling of on-line textual data, described in Part I of the paper. Based on case studies, different approaches to extract useful text information are investigated. The entire text mining process is developed and illustrated on a large multi-document case (several hundreds of research papers) and relevant advantages and shortcomings of the methods applied discussed in detail.

2 Text Mining Process

2.1 Source Data

Portable document format (pdf) is extensively used to present flat documents with fixed layout. Due to prevalence of digital libraries, pdf is becoming one of the most frequently-used formats for research papers. In this paper, the source data is given in a form of sets of research papers from electrical power engineering conferences. All the papers are in English and stored in offline databases. Web crawling and multi-lingual processing are thus not taken into consideration.

2.2 The Flow of Text Mining Process

A complete text mining process is graphically illustrated in Fig. 1. In the figure, blue (dark shaded) cells represent the key factors/categories that are used in the process and white cells with numbers are the steps taken for text mining. As it can be observed in the chart, the first step of text mining is text pre-processing, starting with a collection of text documents in pdf. Format conversion is a necessary step, as well as removal of redundant text sections, such as reference list and conference name in the headers. Only plain text within the main body of the paper passes to the next step.

Step 2, term extraction, is based on the assumption that each document in the corpora (structured set of texts) could be represented by a set of terms [3]. Two

measures are used together at this stage for document representation and feature quantisation: i) term frequency (TF) and ii) inverse document frequency (IDF) [4]. TF refers to the frequency of a term being mentioned in corpus and IDF is used to identify those terms that are concentrated in a few documents but not in entire corpus. Combing these two measures (TF-IDF), some representable phrases could be generated, namely term candidates (TC).

Step 3 aims to discover potential categories of terms based on human judgement, followed up by lexicon establishment for each category. During the process, TCs are re-filtered and phrases that contain higher level of information are selected in order to ease manual work. These phrases are referred to as critical terms (CT). Based on the lists of CT, key terms (KT) are obtained to form lexicons and each is given appropriate weighting (set manually by the user).

The final step, the output of the whole process, contains two ways to realise the information extraction tasks. It categorises papers in different topics and provides summaries of multiple documents for each category.

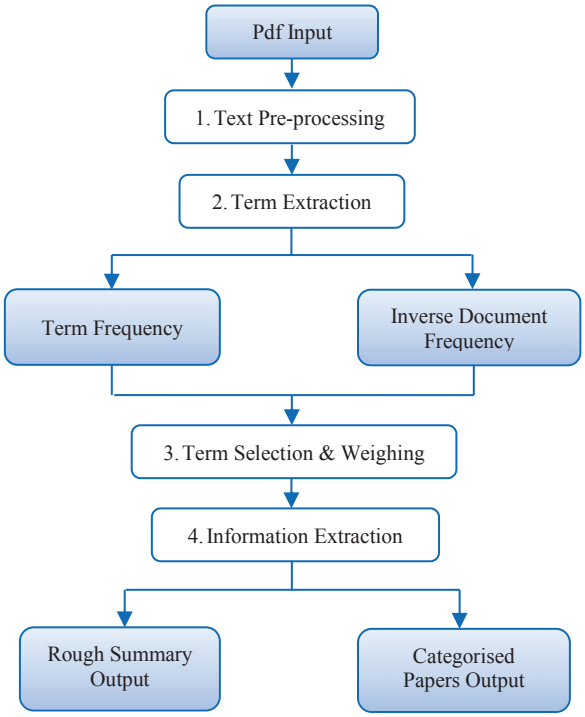

Figure 1 Flowchart of the Text Mining Process

3 Text Mining Methodologies

3.1 Text Pre-processing

The primary task of text pre-processing is to transform the source data into txt format which is easier to compile. While pdf document usually contains additional information like graphs and formulas, a txt file could only store plain text which is beneficial for text mining.

3.1.1 Format conversion:

In order to achieve accurate text extraction from source data, format conversion from pdf to txt is needed. Considering the typical layout of a scientific paper, there are a few requirements for the tools used for format conversion: i) The converter should be able to recognise the two-column structure of scientific papers and reorganise the text content in the correct order; ii) Headers and footers should be removed since they may lead to splitting of paragraphs or sentences; iii) Graphs and formulas that are of no value in terms of text mining should be eliminated in txt files; iv) Only desktop tools should be used, as source data are available off line.

Considering the requirements above, *PDFMiner* is chosen as format converter [5]. It is a powerful desktop tool for converting pdf files to other formats such as txt and html. In order to realise an automated process for large document collections, *Windows PowerShell ISE* [6] is used. This software provides the users with full access to system operation and document management. By running the *pdf2txt.py* in a loop, the process of multi-document format conversion could then be accomplished automatically.

The format conversion using these two tools together has turned out to be effective and efficient. But drawbacks still exist. Due to the mismatch of the encodings of textual data and *pdf2txt.*py which is not the usual case, some words in pdf documents might be converted to messy codes and the underlying information is not likely to be regained. This may affect the accuracy of the following text mining process but most of texts could be converted correctly. The other problem is caused by the texts contained in graphs and formulas such as textual annotations which could not be removed. This results in longer extracted sentences which increases the workload for text analysis and therefore longer processing time. Both problems are fairly common in text pre-processing as 100% of successful conversion could never be achieved without manual involvement of the user.

3.1.2 Noise Removal:

The task of eliminating the noise from the data is necessary before the main text mining steps can be undertaken. Since the source data used to illustrate this methodology consists of numerous research papers, the intention is to make the process automated. There are two types of intrinsic issues entangled with research papers in terms of text mining – these are the layout structure and content. Layout type noise refers to headers, footers and annotations and content type noise comprises non-essential sections which contain no information of interest, such as references list.

The layout issue could be addressed easily when all the documents in the set come from the same source and thus have the same labels for header and footer. Conference papers and company reports for instance could have this type of noise removed by adding noisy

words or phrases in a so called *stop-word list*. Regarding the content type interference, customary format rules for the scientific papers could be constructive since they all have "Abstract", "Introduction", "Conclusions" and "References" as their subheadings of sections. These headings still remain in their initial positions after the format conversion. Among these, the 'References' section is of no use because there is no intention to analyse linked papers. Alternatively titles that are contained in the references could cause confusion and reduced accuracy. Therefore, it is essential to remove this section.

In the multi-document case which is presented here, *Replace Pioneer* tool [7] is used since it is equipped with a batch runner to deal with multiple files. This is a professional text/binary replace tool with hundreds of function templates. For reference removal, the function that extracts the text content in between two words is used and these two words refer to "Abstract" and "References". One side effect is that the titles of documents are also eliminated and the later text mining process will only include the main body of source texts. Since query-based extractive method is used, the removal of titles will not matter too much. After the pdf documents are all automatically converted to txt files and "cleaned" from noise, the text files are ready as the input for the term extraction.

3.2 Term Extraction

Term extraction is a preliminary process to the analysis of text files using various types of measures. In this paper, this process is carried out with a large collection of txt files as input and lists of TC and their TF-IDF as output.

Following the assumption that documents are representable at term level, some phrases that are mentioned in the corpus are viewed as features of text documents and should be later selected as TC according to their TF-IDF. These phrases are not random combinations of adjacent words but semantically meaningful terms. TF-IDF [8] is a collection-dependent measure for multi-document text mining and is actually a product of TF and IDF [9], i.e. both factors are taken into consideration. The IDF and TF-IDF are calculated using (1) and (2) [10]:

$$\mathrm{IDF} = \log \frac{\mathrm{N}}{\mathrm{M}} \tag{1}$$

$$TFIDF = \mathrm{TF} \times \mathrm{IDF} \tag{2}$$

where N is the total number of text files and M is the number of documents that contain the term. Thus, to calculate TF-IDF for a term, both the frequency of its occurrences at the level of entire corpus and its document frequency of appearance should be derived first. Apparently, these two measures could not be calculated at the same time, so the process is accomplished step by step, starting with determining the TF first.

3.2.1 Term Frequency:

To create a list of TC and calculate their TF, application software should be involved to realise the measurement automatically. Many devices are available for both TC generation and TF calculation but most of them have restrictions on the size of corpus to deal with. *FlexiTerm* [11], is superior to others in this aspect. It is an open source program written in Java and web access is not required. In terms of its functionality, the primary issue *FlexiTerm* has addressed is to parse the text files at sentence level and extract the phrase structures inside as TC, i.e. to create a list of TC. On this basis, TF of each phrase is derived using string matching techniques. Apart from major functions, *FlexiTerm* could also accomplish the task of stemming paronymous words into one term automatically. A stop-word list can be used for removing layout noise generated from extra information such as header and footer of the source texts.

With regards to the result, the output file of *FlexiTerm* could be stored in the format of html, csv and txt. Taking csv as example, the result in the output file is typically displayed in columns which contain the information of ranking, TC and TF (Table 1).

Table 1 The Output Excel File (csv) of FlexiTerm
(Data are derived based on the processing of papers taken from PowerTech 2015 conference)

No of TC	Rank	Term Representation	TF
1	1	power system	3149
2	2	reactive power	1923
3	3	active power	1359
……	……	……	……
195	138	load curve	109
197	138	system load	109
198	139	reference voltage	108
……	……	……	……
k	*m*	Term *k*	3

In the table, cells in white colour present the original output data and blue (dark shaded) cells are added additionally for illustration. TCs are usually presented as meaningful two-word phrases and they are already ranked according to their TF. In the list there will be *k* terms which are morphologically different to each other and corresponding number of TF. The first term has an extremely high TF and in the lower rankings, more terms tend to share the same value of TF. Therefore the last term usually has a ranking value of *m* which is much smaller than *k*.

This application software makes the process of TC generation almost completely automated and in the meantime, their TF are calculated accordingly. Because of its multi-functionality, the effectiveness of the list of TC is enhanced, i.e. fewer terms tend to interfere with each other. Nevertheless the process is time-consuming. The relation between the process time and the number of input texts doesn't grow linearly but exponentially. Whereas the process time for one file is only a few seconds, it will take around four hours for *FlexiTerm* to

produce an output for a 13MB corpus, which equals to a hundred text files converted from five-page pdf documents. This problem turns out to be a common side effect of TF calculators and could only be solved with new algorithm established. Since the creation of TC could not be realised in advance, the measurement of IDF was time demanding. (*Note:* It should be pointed out that the process could take significantly less time than if the tools were used by trained computer science and text mining specialists. The study, instead, was performed by beginners in the area, i.e. electrical power engineers without any prior experience in text mining. One of the aims of this feasibility study was, after all, to investigate the feasibility and limitations of performing these studies by untrained professionals without any background in text mining.)

3.2.2 Inverse Document Frequency:

Similarly to TF, the calculation of IDF also requires involvement of computers since the number of TC could be a few hundred times larger than the number of text files and similar calculations are repeated for every term in the list. This process is carried out with string matching techniques which will also be used in the step of information extraction. The algorithm is shown in Fig. 2.

```
Input: R is the set of prediction rules (ranking tables)
       D is the set of documents
Output: F is the output file (table of ranking, TC, TF, M,
       IDF and TF-IDF)
Function: String Matching (R, D)

F=∅
For each rule r in the prediction rule base R do
        For each example d ∈ D do
                If r refers to d (R, D)
                        Add information to F
Return F
```

Figure 2 The Algorithm of Calculating IDF and TF-IDF (Modified from [1])

Python program uses the string matching function to calculate the number of files that mention the specific TC, which is in fact the value earlier defined by M as the number of documents that contain a specific term. IDF and TF-IDF of the term could consequently be calculated based on the equations (1) and (2). Additional open source functions are also used for the complete automation of TF-IDF calculation.

Results have revealed that some terms could have a zero value of M, meaning that none of the text files have ever mentioned the corresponding terms, which makes no sense. This type of unreasonable phenomena is not prominent but it should not be ignored due to its potential influence on category discovery, i.e., the definition of the threshold of TF-IDF for categorisation. The major cause of this is related to several small-scale failures of accurate TC detection and therefore the abnormal terms are usually messy characters and meaningless phrases, namely random combinations of words.

The problem caused by messy characters could be manually fixed if the original terms could be recognised but the mistake in the latter case is irrecoverable. For instance, the phrase "coefficients" is a messy term which probably refers to "coefficients". *FlexiTerm* calculates its accurate value of TF but expresses it in a wrong way. By manually correcting the spelling of the term, its M value could be derived normally just as the table is indicated. On the other hand, the term like "ieee trans" is of no meaning due to the mistaken analysis of phrase structure and clearly the phrase could not be corrected back to what it should be.

Therefore, these recognisable messy terms are manually corrected before calculating IDF and the meaningless terms are eliminated after the calculation. The process could also be simplified by reducing workload. Obviously, there is no need to derive IDF and TF-IDF for all of TC since only terms with rather high occurrence are qualified for the next step of selecting useful terms. The average value of TF is used as a threshold and those terms with smaller values are considered to be useless for document representation.

Once TF-IDF and M are calculated, the process of text mining will step to the selection and weighing of terms.

3.3 Term Section and Weighting

This step aims to connect TC with term categories and is carried out manually by the user. CTs which refer to phrases that are representative for a specific topic of power distribution networks (e.g. demand side management - DSM) and thus qualified for category detection should be derived first. Correspondingly, there will be "key terms" (KT) which are still related to some topics but not as informative as CT – these are commonly derived based on CT. For each topic, a category name, i.e. a collective phrase is used as the name index (NI) and NI is not included in TC but given by researchers.

Compared to the previous steps, term selection and weighting is more case-dependent and different strategies are applied to ensure the precision and rationality. Case studies are therefore introduced here to illustrate the process. The experimental process is based on the pdf documents taken from power system conferences, PowerTech 2015 [13] and Cired 2015 [14]. The tasks are to figure out the major themes of the conference with categories unidentified and to derive some kind of information based on the conference papers in respect to a given topic or the topic derived such as Demand Side Management (DSM). Text mining process will be carried out exactly in the same way for both conferences and results will be evaluated together for comparison.

3.3.1 Category Discovery

Regarding the categorisation task, the first step is to derive potential categories. In Fig. 3, a flow chart of the procedures for determination of categorises is shown. *N*

is the total number of source text files; M is the number of documents that contain a specific term; n is the assumed number of categories; k is a variable that describes the number of categories derived during rounds of iterative process (initiated as 0); a is a measure used to describe the deviations of the number of papers among different categories. For example, for a set of around 500 papers, 10 is taken as a relatively appropriate number of categories considering the workload and overlaps among lexicons for different categories. Each category hence has 50 papers in average. The initial value for a is adopted to be 0.1, which means that the deviations in number of papers among categories range up to ±10% (45 to 55 papers). The range is enlarged by new ±10% in each iteration by increasing a by an increment of 0.1, to allow bigger deviation in the number of papers. Variable b is an iterative number used to record the progress of the process.

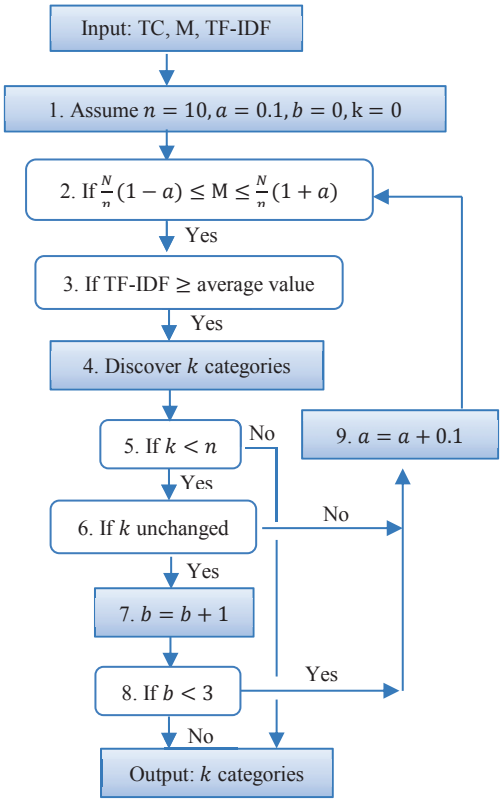

Figure 3 Flowchart of Category Discovery

TF-IDF was supposed to be the only measure to determine CTs but there still exist some extreme cases where TF-IDF are biased by either exceptionally large TF or IDF. M, as the number of documents that contain a certain term, is thus added as another measure to help constraint the range of TF and IDF and make TF-IDF useful. This perception is based on the assumption that documents that belong to the same category will share the similar set of terms (CTs), which makes the M values of these terms more or less similar to the number of documents in the category. A term that is a CT should have M value in an acceptable range. TC with nonbiased TF-IDF could be obtained in step 2. In step 3, the common filtering of TF-IDF is done, where the threshold is set as the average value of TF-IDF and those generalised terms with TF-IDF value smaller than the threshold are filtered out. First-round CTs are then derived and ranked in descending order of TF-IDF. Experts' knowledge in the area of power networks should be involved to group CTs into categories and assign a NI to each category (step 4). The process will then be finished if k is larger or equal to 10 (step 5), which is unlikely in the first round, leading to step 6. The comparison of the values of k obtained before and after iteration is intended to record the number of rounds resulting with no new categories found (step 7). Therefore, in the first iteration, step 6 will directly pass to step 9 and then back to step 2. For the following iterations, process will not end until there are three consecutive iterations with no new categories discovered (step 8) – this is adopted as the boundary number of iterations after which the process stops. Otherwise, the process is continued to enlarge the floating range and more CTs will be extracted (step 9).

Parameters a, b and n all have high dependence on the size of the document collection and should be changed according to the actual situation. Since categories are determined with lists of CT, the process will be continued to extract KT and add weightings to build up lexicons.

3.3.2 Lexicon Establishment

To realise the tasks of knowledge discovery, lexicons for categories or topics should be created and more categorised terms are required other than CT and NI. Therefore, the process here is to extract potential KTs for each category and to manually give them weightings based on their correlations to the topics. An algorithm is developed regarding the difference between words in a phrase. Taking DSM as example (Fig. 4), "demand" and "management" are the words that are by themselves keywords, while "side" is less meaningful.

Similarly, each CT could be divided into two or three words with different importance. According to inherent characteristics of phrases, one or two keywords will be found from each term to be analysed as the input (CT or topic NI). More terms could be found from the lists of TC as they contain keywords and more keywords could be then detected. A tree structure could therefore be built. In Fig. 4, a fraction of the tree structure starting from DSM (NI) is shown. Based on "demand" and "management", more terms such as "demand consumption" and "system management" could be found. Clearly, there is no keyword found in the latter term and "consumption" is observed as a new keyword for next round of KT extraction. As the list of TC is considerably large in length, the tree structure actually expands more quickly in horizontal direction.

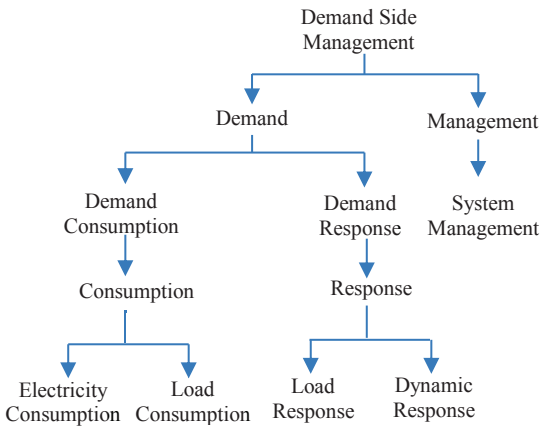

Figure 4 Tree structure of NT Developed for DSM (Data are derived based on the real process on papers taken from PowerTech 2015)

This extraction process is usually repeated three to four times until no keywords are found. The list of KT at this stage has not yet been compressed, so duplicates and terms that are completely irrelevant with the given topic should be removed manually. After this cleaning process, KT left in the list will be assessed with weightings from 5 to 1, following criteria:

5: Terms that are strongly related to the given topic
4: Terms that are not likely to be mentioned by other topics
3: Terms that could be mentioned by other topics
2: Generalised terms and concepts
1: Terms that are more likely to be mentioned by other topics

Table 2 the Lexicons with weighting Based on CT (Data are derived based on the real processing of papers taken from PowerTech Eindhoven 2015)

Low Carbon Technologies	W	Demand Side Management	W	Network Control	W
thermal storage	5	demand response	5	frequency droop	5
li-ion battery	5	demand profile	5	primary frequency response	5
storage potential	4	electricity demand	4	droop control	4
battery capacity	4	load demand	4	synchronous area	4
water heater	3	load profile	3	control strategy	3
heat output	3	electricity consumption	3	control action	3
energy storage device	2	actual demand	2	system state	2
energy storage technology	2	electrical load	2	grid frequency	2
renewable resources	1	residential load	1	phase angle	1
thermal rating	1	local demand	1	synchronous generators	1

Based on this standard, each of NT will be weighted by experts and all of them together constitute a lexicon of a certain category. Table 2 presents parts of lexicons with weights (W) for three topics: DSM, low carbon technologies (LCT) and network control (NC). With weightings given, feedbacks from experts are given and the importance of NT in respect to a given topic can be quantified, which will contribute to the query-based information extraction.

3.4 Term Analysis

In order to figure out the effectiveness of different types of knowledge power systems area, term analysis can be divided into two subtasks: paper categorisation and sentence summarisation. These two will be discovered using a query-based method, for both cases - with unknown categories and with a pre-defined topic.

Query-based information extraction is based on a simple rule that sentences in the corpus will be given scores based on the terms they contain [15]. The score of a sentence in this paper consists of the sum of the weightings of KT it contains. Higher score indicates strong correlation to the topic and lower score means that the sentence is less informative. Since a static rule is used, the output could become fuzzy if the source texts are "contaminated" with noise.

3.4.1 Paper Categorisation

Paper categorisation is to categorise conference papers into defined topics, each paper assigned to a score. If a KT included in one of the lexicons is mentioned in a document, its weighting will be added as score of the paper for the specific category. By going through the lexicons, the scores of a paper for different categories could be derived and each is actually the sum of weightings of a set of terms. String matching function is again used to detect the existence of KT in the document.

The approach is straightforward and it can actually distinguish papers from each other without manual reading and only by calculating the score of a paper. For each paper, a high score indicates strong relevance to the topic and a low mark means little relevance. Some papers could have relatively high scores only for one topic while some others could be related to multiple categories. Both cases are common since some research studies focus on one point and some are aiming at broader area. A uniformed classification standard is used: for each category, papers with top 5% of scores could be considered to have strong bond with the corresponding topic. This is because each category is considered to contain 10% papers and half of them with higher scores should be relevant.

Table 3 illustrates an output file of paper categorisation. Each paper, identified with an index, has a certain score for each of the topics (LCT, NC, DSM and DSM-2). The scores in DSM-2 are derived based on the tree structure given in Fig. 4. Even though DSM and DSM-2 are linked to the same papers, ranking of these papers would not be the same. As seen in the table, paper 476038 has very high score in two topics – LCT and DSM, which means that it probably covers a wider area.

Based on some other papers having relatively high score in both of these two topics, it can be potentially concluded that there is a lot of research done in the cross-field of these areas.

Table 3 Scores of Papers for Different Categories

Paper Index	LCT	DSM	DSM-2	NC
461949	6	42	34	0
463367	26	39	30	9
476038	44	35	37	0
469002	1	35	35	0
443365	13	33	26	0
460412	13	33	33	0
459224	11	33	44	8
470791	0	33	34	0
474573	17	32	29	2
476533	26	31	25	8

3.4.2 Sentence Summarisation

Sentence summarisation in this paper refers to the pure extraction of sentences containing important terms, which is also a query-based text mining process. Similar scoring system to paper categorisation is used. Sentence summarisation is here realised based on the results of papers categorised in the same topic. Summaries for different topics comprise the sentences in the papers that show strong relevance. Other than string matching function, sentence splitter from the Natural Language Toolkit [16] is also used for cutting texts into sentences.

Table 4 Scores of Sentences Extracted for DSM

Sentences	Paper Index	Score
OPP typical daily **load profiles**: The average **load curves** of the DH clusters showed significant differences in the daily **energy consumption** probably as a result of household size and appliances (Fig.	461949	15
Clusters found for the OPP dataset (Mondays to Fridays): (a) average **load curves** of DH cluster, (b) average **load curves** of DHnorm cluster (logarithmic axis display, grey areas indicate the times of on-peak rate of the two-stage TOU tariff for winter), (c) distribution of normalized daily load curves (colour coded, each line is a daily **load curve** of a household), ND: Number of data points.	461949	15
Implications for the **Load Profile** of Offices Figure 7. Shows predicted **electricity demand profiles** for a week in winter (upper) and in summer (lower).	469002	13
These are used to generate synthetic **load curves**, corresponding to **thermal loads** such as air conditioning and electric water heaters, typical appliances, and PHEVs.	443365	8
For example, the annual load curve Pn[k] of a household n is divided in daily **load curves** Pd[k] with d = 1.	461949	5

Table 4 presents an example of the sentence summarisation output, with the terms recognised as features in bold. The highest scored sentence regarding DSM topic comes from the paper which also had the highest score for DSM (Table 3). In the third sentence, in one phrase 'electricity demand profiles', two terms are detected: 'electricity demand' and 'demand profiles', which made the score higher. Apparently, string matching techniques used in the program should be improved and same suggestion applies for paper categorisation. Instead of searching term by term in a loop, the match search should also consider possible combinations of terms. Moreover, the sentence extraction is based on the top 20 papers selected during the paper categorisation, which is why errors could be lower if the accuracy of paper categorisation was higher.

4 Evaluation and Results

Criteria have been established for assessing two types of information obtained: categorised papers and summaries that consist of sentences for different categories. For paper categorisation, the actual relevance of a paper to a category is the major parameter for result evaluation and here it is measured with scores from 0 to 3. If any terms with weightings of 3 and above in the corresponding lexicon are mentioned in all of the three important segments of paper, i.e., title, abstract and keywords, a paper is considered to have 3 hits which is a score of 3.

Figure 5 presents effectiveness of document ranking for two conferences: PowerTech 2015 (PT) and Cired 2015 (CD), and two categories: DSM and DSM-2. The common characteristic of the four curves is that the average accuracy tends to decrease as the paper ranking drops. Also, comparing the two conferences, the highest scored papers do not necessarily have the same relevance to the topic of DSM. In addition, there is no absolute relationship between the accuracies of results obtained from a single term and groups of terms (DSM-2 and DSM, respectively). It is clearly case-dependent and either method could derive useful information with certain level of accuracy.

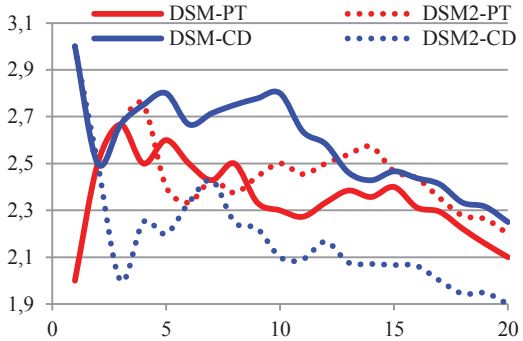

Figure 5 Average Effectiveness of Top 20 Papers for DSM in Different Conferences

Similarly, each sentence will be weighed with scores according to three types of performance: understandable on its own without referring to the original source paper, conclusive (but not descriptive) information and original in perception. Figure 6 illustrates the performance of top 20 sentences for different categories (DSM and DSM-2) in PowerTech 2015 conference (PT). As shown in the figure, the relevance to the given topic is not

directly related to the scores obtained via query-based summarisation. The information extracted appears to be chaotic in most cases, which indicates the need for enhanced summarisation with complex queries, taking into account sentence position and length.

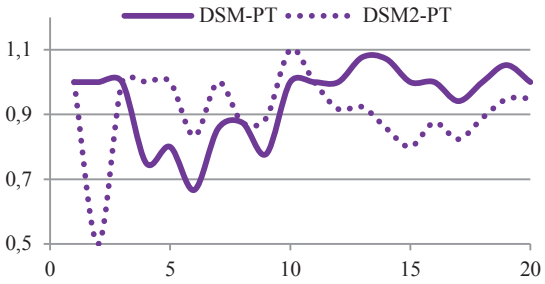

Figure 6 Average Effectiveness of Top 20 Sentences for DSM

5 Conclusion

A complete text mining process on offline conference papers has been explored within the framework of the traditional text mining applications. Starting with text pre-processing, format conversion and noise removal, sets of textual documents were ready for feature extraction. In the next steps, classification of terms was performed and lexicon established for each class (category). Paper categorisation and sentence summarisation were accomplished by query-based method and string matching functions. Finally, the results were evaluated with certain standards and paper categorisation turned out to be more accurate and informative than sentence summarises.

With this series of applications, text mining process framework for power system data analytics could be formulated and performance would be much better if each step of text mining could be accomplished with minimum imported noise. More tools and methods should be used to eliminate interference given the characteristics of power system research papers. Software applications should be involved in the step of term selection and weighing to promote the process efficiency while maintaining effectiveness. Enhanced methods such as abstractive summarisation should be implemented for the step of sentence level information extraction. Further explorations on evaluation system should be also undertaken. Opposite to manual processing of text (i.e. reading), computer-based evaluation is of high efficiency and the output would show higher reliability if the training algorithm was enhanced. The entire process still requires involvement of experts in the area, whose feedbacks are critical to modify ways to discover knowledge from text.

6 Acknowledgements

This research is partly supported by the EU Horizon 2020 project “Nobel Grid”, contract number 646184.

7 References

[1] M. Rajman, “Text mining: Natural language techniques and text mining applications,” *Proc. 7 th IFIP*, vol. 1998, 1997.

[2] J. C. K. Cheung, “Comparing Abstractive and Extractive Summarization of Evaluative Text : Controversiality and Content Selection,” 2008.

[3] C. Feilmayr, “Text Mining-Supported Information Extraction: An Extended Methodology for Developing Information Extraction Systems,” *2011 22nd Int. Work. Database Expert Syst. Appl.*, pp. 217–221, 2011.

[4] A. Aizawa, “An information-theoretic perspective of tf-idf measures,” *Inf. Process. Manag.*, vol. 39, no. 1, pp. 45–65, 2003.

[5] “PDFMiner,” *Unixuser.org*, 2014. [Online]. Available: http://www.unixuser.org/~euske/python/pdfminer/. [Accessed: 20-Apr-2016].

[6] “Microsoft PowerShell,” *Msdn.microsoft.com*. [Online]. Available: https://msdn.microsoft.com/en-us/powershell.

[7] “Replace Pioneer Homepage - Batch search/replace/convert/rename/split text/binary/web file,” *Mind-pioneer.com*. [Online]. Available: http://www.mind-pioneer.com/.

[8] F. El-Ghannam and T. El-Shishtawy, “Multi-Topic Multi-Document Summarizer,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 6, pp. 77–90, 2013.

[9] C. Ludwig, “Text Retrieval,” vol. 24, no. 5, pp. 1–21, 2007.

[10] M. A. Fattah, “A hybrid machine learning model for multi-document summarization,” *Appl. Intell.*, vol. 40, no. 4, pp. 592–600, 2014.

[11] I. Spasić, M. Greenwood, A. Preece, N. Francis, and G. Elwyn, “FlexiTerm: a flexible term recognition method.,” *J. Biomed. Semantics*, vol. 4, no. 1, p. 27, 2013.

[12] N. Kanya and S. Geetha, “Information Extraction - a text mining approach,” *Information and Communication Technology in Electrical Sciences (ICTES 2007), 2007. ICTES. IET-UK International Conference on*. IET, pp. 1111–1118, 2007.

[13] “PowerTech Eindhoven 2015 - Towards Future Power Systems and Emerging Technologies,” *PowerTech Eindhoven 2015*. [Online]. Available: http://powertech2015-eindhoven.tue.nl/.

[14] “CIRED 2015 • International Conference on Electricity Distribution,” *Cired2015.org*, 2015. [Online]. Available: http://www.cired2015.org/.

[15] F. Pembe and T. Güngör, “Automated Querybiased and Structure-preserving Text Summarization on Web Documents,” *... Innov. Intell. Syst. ...*, 2007.

[16] S. Bird, "NLTK: the natural language toolkit," in *Proceedings of the COLING/ACL on Interactive presentation sessions*, 2006, pp. 69-72.