# Preprocessing Methods for Unstructured Healthcare Text Data

**Naresh Patel K M, Kiran P**

***Abstract*: At present, the amount unstructured text data is increasing exponentially from the past periodically. Information retrieval (IR) from these unstructured text data is challenging. As the data users foresee for particular/specific outcomes. Retrieval of the significant outcomes depends on the fashion, how they are associated/indexed. Unstructured text data like clinical data containing more health information requires challenging preprocessing methods, which also help to reduce the size of the dataset so that it will optimize the performance of the IR system. In this paper, we have proposed the pre-processing methods such as Data collection, Data Cleaning, Tokenization, Stemming, Removal of Stop words which will efficiently help the data users to find the specific patterns from the unstructured text data.***

***Keywords*: Information Retrieval (IR), Tokenization, Stemming, Stop words, Unstructured Text Data.***

## I. INTRODUCTION

Mining unstructured text data is a novel and interesting area in research that tries to figure out the crisis of overwhelming of the information with the aid of Natural Language Processing, Machine Learning and Information Retrieval systems which are the data mining techniques. Unstructured text data is reasonably a narrative data which includes discharge records, clinical notes, pathology reports, surgical records. Unstructured text cache more of valuable clinical data and also don't have a typical structural form and there exists several errors, like improper use of grammar, spelling errors, native dialects, and linguistics ambiguities that raises the complications in data analysis and processing [1]. So, before implementing any technique on the raw text data, the data need to be preprocessed. Raw text data is highly subjected to noise, inconsistency and missing values which affects the quality of information retrieval system. Text preprocessing reduces the size of the text documents. Total word counts of stop words in a text documents account to 20-30%. Stemming can scale back to maximum extent up to 40-50%. On the opposite side, it also enhances the potency and effectives of the IR system from the unstructured text data, as Stop words aren't helpful for looking out or mining the text and that may mislead the retrieval system, similar words in text document can be matched by using stemming method [7].

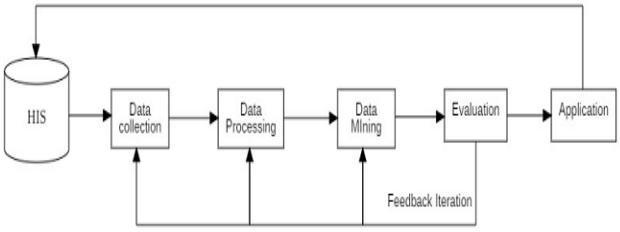


**Fig.1. Methodology of Data Preprocessing**

## II. HEALTHCARE DATA PREPROCESSING

### A. Data collection

In advance to the preprocessing of the data, we need to gather unstructured health care text data which is a challenging task. Healthcare data are generally inadequate, conflicting or lack certain errors. The raw healthcare data are inadequate to take right decisions. The data collected by healthcare organization might lack a typical structural form and there exists several errors, like improper use of grammar, spelling errors, native dialects, and linguistics ambiguities, that raises the complications in data analysis and processing. So, proper data collection is a fundamental and vital step to acquire the fine data which are suitable [2]. Data can be gathered from two different data sources, which are grouped as primary and secondary data sources [11].

Figure 1 illustrates two methods of healthcare data collection.

**Primary data:** Data that are gathered freshly for the primary time.

*Examples*-Observations, survey, interview and focus groups.

**Secondary data:** Data that are already collected, analyzed by someone else.

*Examples*- Internet, library, Research articles.

In healthcare sector an ample of data generated rapidly which in turn lack of information. There are plenty of methods by which data may be collected for research and healthcare management [2].

**Revised Manuscript Received on December 22, 2019.**

**Naresh Patel K M.,** Assistant Professor, CSE Department, BIET, Davangere, Karnataka, India.

**Dr. Kiran P**, Assoc. Professor, CSE Department, RNSIT, Bengaluru, India.

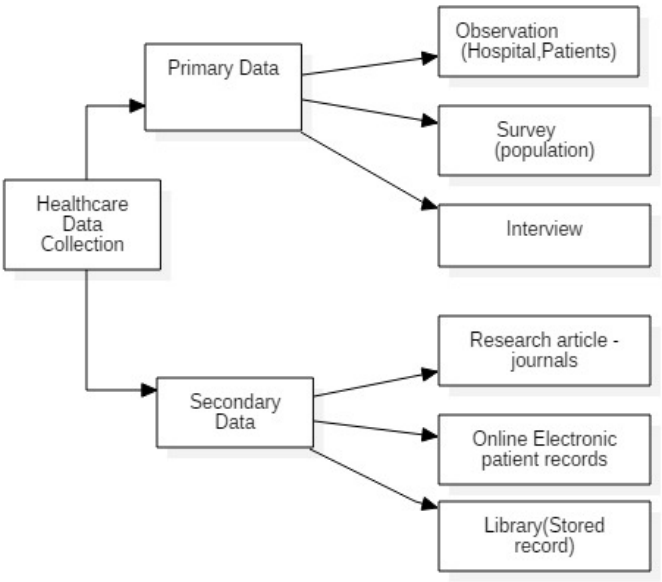


**Fig. 2. Healthcare Data Collection Methods**

## B. Data Cleaning

Data cleaning is an important aspect which has an direct impact on the accuracy of the derived models and conclusions.

### 1. Default Preprocessing

While collecting the data some data characters or values will be lost due to manual errors, sometimes due to system failure. For default data, we can neglect the lost data, where the default values can be filled manually, it can also be used with the attribute averages, use of mostly liked values to fill the defaults, or we can also retrieve from the other data sources available.

In case, while collecting the patient information, if the name of the operation is not present, then that input can be neglected, suppose if the information like bed number is not present, then that input should not be neglected. Suppose the input data is comparatively minimal, then in such cases the default values could be filled manually. But, when dealing with maximal datasets with larger default values, it is not possible to fill the default values manually. In extension to this, it is very time-consuming and also costs more, so it cannot be applied most of the times. Suppose the data is distributed evenly and costs less, then default values can be placed with attribute averages. Likewise, for the default values optimum value can be found using machine learning methods. Even though the inference may give nearly a more variation in severe conditions, these methods are more capable of dealing with the default values. In addition, we can retrieve the missed data attribute that will exist in some other data sources [1].

### 2. Noise Preprocessing

Noise in data preprocessing hints to an irregular attribute value in an data source [5].

The preprocessing of noise data possibly includes the techniques like regression, binning, outlier analysis and retrieval from other data sources. The binning method is a process of smoothing the organized data values by investigating the values that are around. The basic criterion in a binning method is the capacity of sub-box. The process of modifying the noise value by creating the function model that will fit the data attribute value is called regression method. The process of building clusters by using clustering method is called outlier analysis. The attribute values of the data points are identical which are within the same cluster, however attribute values of the data points among distant clusters include a more deviations.

### 3. Inconsistent data Preprocessing

In this method there may exists inconsistencies in several homologous data or sources, like inconsistencies in recorded values and measurement units. The inconsistencies in the data may be revised by identifying the correlation in data and fetching the data from distant data sources [1].

## C. Tokenization

Tokenization is a process of segregating all the words, numbers, characters and unnecessary punctuations etc. in the input document are called as tokens [9]. This method not only generates the tokens at the same time it also identifies the frequencies of these tokens which appears in the input document by counting the number of times it appears in the document. At the same time the infrequent words which have less frequency value can be removed as they are not useful in information retrieval system which is done by another preprocessing method called stop word removal.

For example, the document containing the information like "There are 3 main preprocessing methods for unstructured text data namely tokenization, stop word removal and stemming".

Suppose the tokenization method is applied on this document, the output of the system is to segregate the characters or words or numbers like there, are, 3, main etc. as tokens by using python code to do this text preprocessing operations.

NLTK- Natural Language Toolkit with its NLP libraries can be used perform text preprocessing like tokenization, stop word removal and stemming.

Input:

| "There are 3 main preprocessing methods for unstructured text data namely tokenization, stop word removal and stemming". |
|---|

**Fig. 3. Input document**

On applying the tokenization process to the above input document the output will be established like:

**Output:**

**Words=**there<1>are<1>main<1>preprocessing<1>methods<1>for<1>unstructured<1>text<1>data<1>namely<1>tokenization<1>,<1>stop<1>word<1>removal<1>and<1>stemming<1>

**Numbers=**3<1>

The values within the angular braces show the frequency of the words/numbers within the given document. For example "there", "are", "3", "main" occurs one time in the document so their frequency is 1.

## D. Stop Word Removal

As total word counts of stop words in a text documents, account to 20-30%, so dimensionality of the text can be reduced by removing those stop words from the text.

The most frequent words within the text documents are pronouns, prepositions, articles that don't offer the meaning in the documents. Usually all the text documents contain these type of stop words which gives least preference for analysis and makes the text to look bulky. The words like the, is, what, when, in, a, with etc. are evaluated as stop words and should be eliminated from the text because they are not treated as keywords in text mining applications.

For example, the document containing the information like "There are 3 main preprocessing methods for unstructured text data namely tokenization, stopword removal and stemming".

Suppose the stop word removal process is applied on this document then the output of the process is to remove such words which are not considered as keywords.

**Input:**

"There are 3 main preprocessing methods for unstructured data namely tokenization, stop word removal and stemming".

**Fig. 4. Input document**

On applying the stop word removal process to the above input document the output will be established like:

**Output:**

**Words=** preprocessing unstructured tokenization stop word stemming

### E. Stemming

Stemming could be a method of checking out the roots/stems in a word by removing inflection through dropping extra characters typically suffix. Stemming models can be categorized as truncating and statistical stemmers where the results are often used to find commonalities across the massive dataset. Sometimes it may also do a over stemming where the words like 'universe' and 'university' are reduced to the constant root 'univers'. Generally stemmer can formed like, when the user enters the word like 'repetition' in IR query, the system will retrieve 'repeated',' repetitively'.

**Input:**

"There are 3 main preprocessing methods for unstructured data namely tokenization, stop word removal and stemming".

**Fig. 5. Input document**

On applying the stemming process to the above input document the output will be established like:

**Output:**

**Words=**preprocess unstructure token stop word stem

Stemmers can be categorized as:

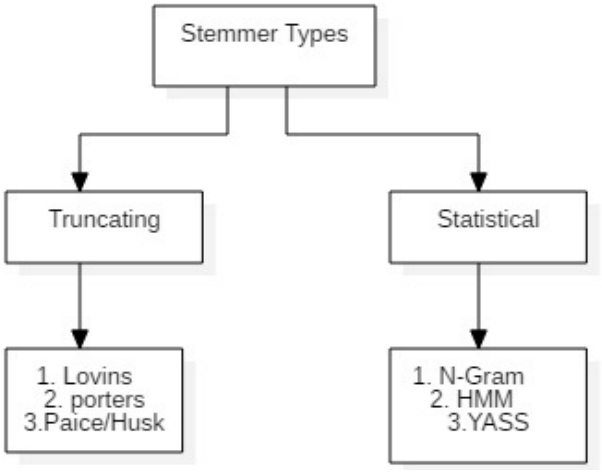


**Fig. 6. Stemmer Types**

**1.Truncating stemmers (Affix removal)**

As the name itself says these stemmers perform removing of suffixes and prefixes of a word. The most basic stemmer is Truncate (n) stemmer that truncates a word at the nth symbol i.e. keep n letters and take away the next. Words are kept as they are and unbroken if they are shorter than n [8].

### A. Lovins Stemmer

It is an efficient stemmer sujjested by Lovins in 1968 which has a listing of 294 suffixes, 29 conditions and 35 transformation rules which are used for maximum match. Here it removes the longest words suffix. It can remove ultimately 1 suffix from a word and can also handle many uneven plurals like index and indices. It also eliminates double lettered words like 'embedded' as 'embed'. Sometimes several suffixes don't seem to be accessible within the table endings, so it is unreliable and fails offenly. Lovins stemmer is heavier and larger because of its vast ending lists.

### B. Porter Stemmer

Among the different stemmer models Porter stemmer is a well known and repeatedly used stemmer that performs suffix stripping to form roots/stems which has 5 steps and 60 rules inside every step. The suffix will be removed if the rule is accepted and therefore the further step will be applied. At the end of fifth step ultimate resultant stem will be returned [6]. For example, Porter stemmer giving a root/stem for a word 'exams' by simply removing the 's' after 'exam'. Porter stemmer is lighter than Lovins because of less number of steps and less error rate.

Some rules:

1. <suffix> ⟶ <new suffix>

Exams ⟶ Exam
Ovens ⟶ Oven

2. <condition><suffix> ⟶ <new suffix>

Overseed ⟶ Oversee

Here, the rule says (m>0) EED ⟶ EE means, suppose the word has minimum one vowel, one consonant and ends with EED, then modify the ending to EE.

### C. Paice/Husk Stemmer

The Paice/Husk stemmer was recomended by Chris D. Paice, in 1990. It is an iterative stemmer which follows 120 rules, where each rule states for the elimination of the ending. It is an assertive type and may over stem. It can be implemented easily but not competent enough compared with the other stemmers and also more number of words will conflates to incorrect words.

## 2.Statistical Stemmers

Statistical stemmers are competent and suitable approaches in information retrieval system. They do not require language expertise but these stemmers can handle statistical information from a bulk of data to study complex words.

### A. N-Gram

N-Gram gathers the similar pair of words. N-Gram can be definedas uni-gram for n=1, bi-gram for n=2, tri-gram for n=3 etc. which produces a pair of successive letters. So the name is N-Gram. In this method it finds the association measure between the pair of words which depends on the common unique digrams. To find an association measure it uses Dice's coefficient. On finding the unique digrams for a word-pair, then a similarity measure is figured out on them. Similarity measure on Dice's coefficient is explained as in equation (1):

$$S = \frac{2C}{A + B} \tag{1}$$

where, A and B are estimated as the number of unique bi-grams in first and second words respectively and C is estimated as the number of unique bi-grams shared by both A and B.

**Table-1: Similarity Measure for Bi-grams and Tri-grams for words.**

| Steps | Example word: Similar and Dissimilar | | |
|---|---|---|---|
| | Word Calculation | Bi-grams | Tri-grams |
| 1 | Unique N-gram for word 1 | *S SI IM MI IL LA AR R* | **S *SI SIM IMI MIL ILA LAR AR* R** |
| 2 | Unique N-gram for word 2 | *D DI IS SS SI IM MI IL LA AR R* | **D *DI DIS ISS SSI SIM IMI MIL ILA LAR AR* R** |
| 3 | A=Unique N-gram for word 1 | 8 | 9 |
| 4 | B=Unique N-gram for word 2 | 11 | 12 |
| 5 | C=Shared unique words | 7 | 7 |
| 6 | Dice Coefficient | 0.73 | 0.66 |

In the above example * indicates the padding space. The Dice's Coefficient for the word 'similar' is 0.73. Such a similarity measures will be found for each pairs of terms in the database. Hence, the result of the Dice's Coefficient says that the stem for these pair of words lies in first 8 unique Bi-grams [5].

### B. YASS

The acronym of YASS is Yet Another Suffix Striper. It is a statistical further it is a corpus based language independent stemmer which operates without any linguistic input. YASS can perform better than porter and lovins stemmers in terms of maximum number of documents retrieved from the languages with poor resources. YASS defines the set of string distance measures and a lexicon for a given document, which are clustered with the distance measures and recognize the equivalence categories. The similarity between the two words can be measured by using string distance measure by calculating the distance between the two strings and also maps a pair of string 'a' and 'b' to a real number 'r'. Lesser the value of 'r' higher the similarity between 'a' and 'b. The logic of measuring these distances is to seek out long matching prefixes and to penalize the initial mismatch.

### C. HMM

The acronym of HMM is Hidden Markov Model. It represents a finite state automata concepts. The probability functions are governed by the transitions between the states. HMM stemmer is an extension of a Markov process where the observation is a probabilistic function of a state. A symbol will be produced by a new state at each transition with a given probability. It is supported unsupervised learning which will not require the previous linguistic data of the dataset. The likelihood of every path are computed by Viterbi coding, where this coding is be used to find the most probable paths. In this stemmer it considers the word as a integration of 2 sub sequences as a prefix and a suffix. Here the states are splited into two disjoint sets where the initial states are the stems, later states as suffixes or stems. We can do some assumptions while using this stemmer.

a. Treating the initial states as stem.
b. If there is a transition from suffix state to a stem state then we can say it is a null probability.
c. The final state always belongs to both states.

## III. RESULTS AND DISCUSSIONS

Every method has its own limitation. Here in the stemming methods we can identify two types of errors that usually occurs:

1. **Under stemming-** This type of error exists when we need to stem two different words to a same root which are of same conflation class, but they are not.
2. **Over stemming-** This type of error exists when we need to stem two different words to a same root which are of different conflation class, but they are.

The limitations of two stemmers, Truncating methods stemmer and Statistical methods stemmers are briefed out in the Table 2 and Table 3 respectively.

**Table-2: Truncating methods.**

| Truncating methods | |
|---|---|
| **Stemmers** | **Limitations** |
| Lovins Stemmer | 1. It is unreliable which is not able to form words from the stem repeatedly.<br>2. It is time consuming as it has 294 suffixes, 29 conditions, 35 transformation rules.<br>3. Even it has a list of 294 suffixes sometimes suffixes are not available. |
| Porter Stemmer | 1. It forms over stemming.<br>2. It takes more time.<br>3. It has 5steps and 60 rules to generate the stem.<br>4. Stems generated are not always the real words. |
| Paice/Husk Stemmer | 1. It forms over stemming.<br>2. It is heavy stemmer |

**Table-3: Statistical methods.**

| Statistical methods | |
|---|---|
| **Stemmers** | **Limitations** |
| N-Gram Stemmer | 1. Need more room for indexing and creating the N-Grams.<br>2. It is not a time efficient stemmer. |
| Yass Stemmer | 1. It requires more computing power.<br>2. It becomes difficult to determine a threshold for making clusters. |
| HMM Stemmer | 1. It forms over stemming.<br>2. It is a complex method. |

## IV. CONCLUSION

As there is a huge amount of data generated in the healthcare sector which always lacks in the potential information that makes difficult in making decisions and analysis. Because of this the researchers find difficulty in collecting data from primary and secondary data sources. As in real world the data suffers from inconsistency, noisy data which may lead to inaccurate decisions and analysis. So, healthcare data need to be preprocessed. In order to clean the data several preprocessing methods are applied which plays a vital role in making decisions and analysis and also by these preprocessing methods we can reduce the size of the data. This paper gives the comparison of the truncating and statistical methods. Even though these methods may not give the 100% output, they are enough for mining the unstructured text document.

## REFERENCES

1. Wencheng Sun et al., "Data Processing and Text Mining Technologies on Electronic Medical Records", Journal of Healthcare Engineering, Volume8,April-2018. Available: https://doi.org/10.1155/2018/4302425
2. Uma K, M., Hanumanthappa, "Data Collection Methods and Data Pre-processing Techniques for Healthcare Data Using Data Mining", International Journal of Scientific & Engineering Research, Volume 8, Issue 6, June-2017.
3. Akrivi Krouska et al., "The effect of preprocessing techniques on Twitter sentiment analysis", International conference on Information, Intelligence Systems and Applications, July 2016.
4. S P Ruba Rani, B Ramesh, M Anusha, Dr. J G R Sathiaseelan, "Evaluation of Stemming Techniques for Text Classification", International Journal of Computer Science and Mobile Computing A Monthly Journal of Computer Science and Information Technology, Volume 4, Issue. 3, March 2015.
5. Lai K H, Maxim T, Goss F R, "Automated misspelling detection and correction in clinical free-text Records", Journal of Biomedical Informatics, 2015.
6. Dr. S Vijayarani, Ms. J Ilamathi, Ms. Nithya "Preprocessing Techniques for Text Mining-An Overview", International Journal of Computer Science & Communication Networks, Volume 5(1), 7-16.
7. Vairaprakash Gurusamy, Subbu Kannan, Dr. S Kannan "Preprocessing Techniques for Text Mining", Conference Paper, October 2014.
8. Deepika Sharma, "Stemming Algorithms: A Comparative Study and their Analysis", International Journal of Applied Information Systems", Foundation of Computer Science FCS, New York, USA, Volume 4 – No.3, September 2012. Available: www.ijais.org.
9. J. Qui, C Tang, "Topic Oriented Semi-Supervised Document Clustering", In Proceedings of SIGMOD, Workshop on Innovative Database Research, pp- 57-52, 2007.
10. Erhard Rahm, "Data Cleaning: Problems and Ap-proaches", University of Leipzig, Germany.
11. C.R. Kothari, Research Methodology: methods and Techniques, 2004.

## AUTHORS PROFILE

**Naresh Patel K M.,** working as Assistant Professor in CSE Department, BIET, Davangere from past 7 years. Area of Intrest: Data Mining, Privacy Preserving Data Mining and Privacy Preserving Data Publishing. Registered under VTU, Belgavi as a Research Scholar in 2016 with the Research Center RNSIT, Bengaluru.

**Dr. Kiran P**, BE, M.Tech, Ph.D, working as Assoc. Professor in CSE Department, RNSIT, Bengaluru from past 18 years. His research interests include Cryptography, Big Data Analytics, Data Mining, Privacy Preserving Data Mining and Privacy Preserving Data Publishing.