Role of Natural Language Processing for Text Mining of Education Policy in Rajasthan

Pooja Jain and Shobha Lal

Abstract The knowledge of education policy will bring an array of new growth, but it has necessitated an improved type of human–machine intercommunication, in which the machine enhances a thoughtful and interactive intelligence. Natural language processing (NLP), a part of artificial intelligence (AI), is the competence of a computer program to comprehend spoken and written human language (https://www.linguamatics.com/what-text-mining-text-analytics-and-natural-language-processing; Zhang and Segall in IJITDM 7(4):683–720, (2008)) [1, 2]. After being thoughtful about it, in mining, one should have sagacity for the predetermination of policy (Bhardwaj in Int J Eng Res Technol (IJERT) 1(3), 2012; Maes in Commun ACM 7:30–40, 1994) [3, 4]. Using NLP, this provides a quick way of extracting information about education policy. This paper focuses on manipulating NLP commands after data collection using unstructured interviews about the attitude of NLP and then filling out a website questionnaire form to collect the satisfaction result. Coding is executed to get the required data using Python and NLP. During the analysis of feedback at colleges in Jaipur, Rajasthan, it is divulged about the satisfaction of using NLP commands, so it is observed that NLP creates a convenient way of mining. The goal behind this text mining is to identify the importance of NLPs in getting data into an integrated form. Lastly, in the execution phase, it narrates the process to obtain cognition for extricating data about policies for gratification.

Keywords NLP · Education policy · Unstructured data mining · Web text mining · Execution

P. Jain (✉) · S. Lal
Jayoti Vidyapeeth Women's University, Jaipur, Rajasthan, India
e-mail: Pooja1981jain@yahoo.com

S. Lal
e-mail: dean.fet@jvwu.ac.in

A. Mishra et al. (eds.), *Advances in IoT and Security with Computational Intelligence*, Lecture Notes in Networks and Systems 756,
https://doi.org/10.1007/978-981-99-5088-1_14

1 Introduction

Text mining is one of the AI techniques. It enlists NLP and converts unstructured text into data analysis format. Data on the web is mainly in unstructured format [5, 6]. Unstructured data is inputted into models to get predictions. NLP is a sub-part of data science that consists of processes for intelligently processing, interpreting, and getting knowledge from text data. NLP and its components can be used to organize large amounts of data, perform various automated tasks, and solve a variety of problems. Important tasks of NLP are text classification, text matching, and co-reference resolution. Text mining is a technique for reviewing the records of a large group to find knowledge from the data. It is broadly useful for getting knowledge [7–10]. This uncovers documentation of large amounts with interrelationships. To process the text, text mining can be used with NLP. Text mining produces structured data that can be incorporated into databases [11–15].

1.1 Interpretation with ML for NLP

Python is a highly regarded and machine-friendly programming language in the artificial intelligence world. It works on a variety of data science topics such as ML, NLP, and more. It has a path for every stage of the data science process [16]. A query in Python extracts the data for cleaning and sorting. NLP simulates making devices more intelligent to search the web. It allows machines to read the text and reply accordingly. It encompasses both natural language understanding and natural language generation [17]. A search engine like Google provides every type of required data due to NLP. Understanding the meaning of text can be accomplished by using machine learning for NLP. NLP turns unstructured text into usable data. It can be categorized as supervised machine learning (SML) and unsupervised machine learning (USML). If the model is put into other text, it is considered SML, and a set of algorithms that extricate meaningful data is viewed as UML. Classification and regression are two categories of SML. Through classification, it can be used for fraud detection, image classification, customer retention, diagnostics, etc. On the other hand, commercial prophecy, climate foretelling, business conjecture, evaluating expectations of life, citizens' development projections, etc., are manipulated by regression. So classification problems are applied to train a model to predict qualitative goals. After predicting a number, the relationship between dependent and independent variables is discovered in regression. And when the datasets, like unlabeled data, are loaded into the model for analysis and clustering without being predefined, clustering, association, and dimensionality reduction (generalization) are the ways to describe UML. Customer segmentation, targeted marketing, recommender systems, etc., are included in clustering. Market basket analysis, customer clustering in retail, price bundling, assortment decisions, cross-selling, and others are contained in the association, whereas meaningful compression, structure discovery, feature elicitation, and

big data visualization, etc., are operated using dimensionality reduction. This is an unsupervised technique where the unlabeled groups of similar entities are processed as image compression, recognizing forgery newscasts, unsolicited processes, advertising mechanisms, systematizing web marketing, associating crooked or delinquent tasks, recording surveys, and others are solved by it [18].

1.2 About Education Policy 2020

National Education Policy 2020 includes nearly 2 lakh suggestions from 2.5 lakh gram panchayats, 6600 blocks, 6000 urban local bodies, and 676 districts. By 2030, this new policy aims to universalize education from pre-school to the secondary level. There is a strong emphasis on foundational literacy. Vocational education will begin in Grade 6 with internships, and until Grade 5, it will be taught in the parent's native language. According to NEP 2020, it has been dividing the 10 + 2 system into the 5 + 3 + 3 + 4 format. Flexibility in a higher education curriculum will be added [19–21]. Medical education will be mingled with Ayurveda, Naturopathy, Unani, Homoeopathy, Siddha, and vice versa at the undergraduate level, according to the education policy [22].

2 Methodology

In the execution, it is initialized as "Education Policy Websites." For applying web text mining, NLP commands are used. If we get a satisfactory result from policy extraction using NLP, we will stop the execution. Otherwise, it will be continued with the same technique for a worthy result (Fig. 1).

2.1 Code and Executed Screenshots of Python for NLP

NLP is applied for cleaning and summarizing text, tokenizing sentences and words, getting the frequency of words, etc. There are some steps in text mining for deriving meaningful information when manipulating NLP with Python code [23] (Figs. 2, 3, 4, 5, 6, 7, 8, 9).

```
#Installing NLTK (Natural Language Toolkit)
C:\Users\HP\AppData\Local\Programs\Python\Python39>python
>>> import nltk
>>> nltk.download()
Showing info https://raw.githubusercontent.com/nltk/nltk_data/
gh-pages/index.xml
#Working with tokenization in NLP
```

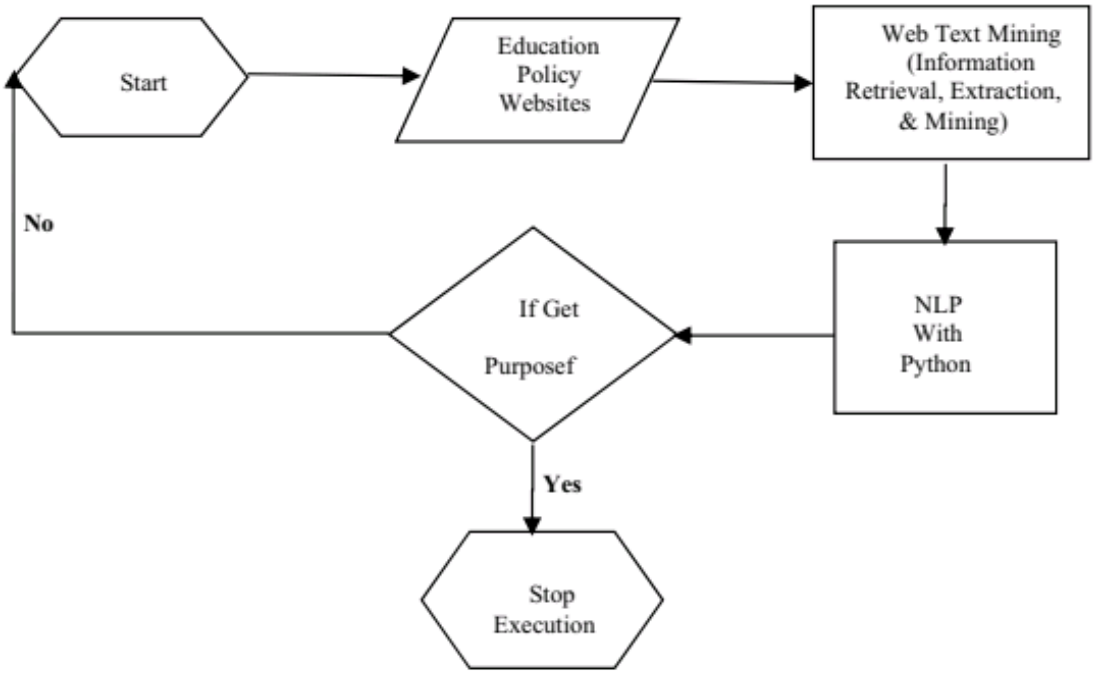

Fig. 1 Execution flow of text mining

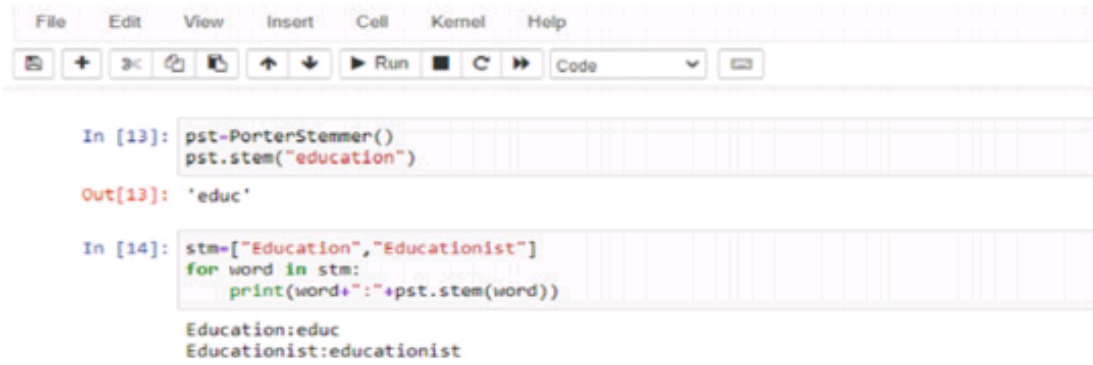

Fig. 2 Stemming for text mining

```
>>> Education_Policy="According to NEP 2020, it has been dividing
the 10+2 system into the 5+3+3+4 format. Flexibility in a higher
education curriculum will be added."
>>> token=word_tokenize(Education_Policy)
>>> token
['According', 'to', 'NEP', '2020', ',', 'it', 'has', 'been',
'dividing', 'the', '10+2', 'system', 'into', 'the', '5+3+3+4',
'format', '.', 'Flexibility', 'in', 'a', 'higher', 'education',
'curriculum', 'will', 'be', 'added', '.']
# Locating the frequency distinct in the tokens
>>> from nltk.probability import FreqDist
>>> fdist=FreqDist(token)
>>> fdist
FreqDist({'the': 2, '.': 2, 'According': 1, 'to': 1, 'NEP': 1,
'2020': 1, ',': 1, 'it': 1, 'has': 1, 'been': 1, ...})
```

```
In [32]: from nltk.stem import LancasterStemmer
         lst = LancasterStemmer()
         stm = ["giving"]
         for word in stm :
          print(word+ ":" +lst.stem(word))

         giving:giv

In [33]: stm=["Education","Educationist"]
         for word in stm:
             print(word+":"+pst.stem(word))

         Education:educ
         Educationist:educationist

In [37]: from nltk.stem import WordNetLemmatizer
         lemmatizer = WordNetLemmatizer()

         print("Nationalist :", lemmatizer.lemmatize("National"))
         print("Educations :", lemmatizer.lemmatize("Education"))

         Nationalist : National
         Educations : Education
```

Fig. 3 Stemming and lemmatization for text mining

```
In [45]: import os
         from nltk import word_tokenize
         from nltk.corpus import stopwords
         sw = set(stopwords.words('english'))
         Education_Policy = "National Education Policy 2020 includes nearly 2 lakh suggestions from 2.5 lakh
         Education_Policy1 = word_tokenize(Education_Policy.lower())
         print(Education_Policy1)
         stopwords = [n for n in Education_Policy1 if n not in sw]
         print(stopwords)

         ['national', 'education', 'policy', '2020', 'includes', 'nearly', '2', 'lakh', 'suggestions', 'from'
         anchayats', ',', '6600', 'blocks', ',', '6000', 'urban', 'local', 'bodies', ',', 'and', '676', 'dist
         ',', 'the', 'new', 'policy', 'aims', 'to', 'universalize', 'of', 'education', 'from', 'pre-school',
         '.', 'there', 'is', 'a', 'strong', 'emphasis', 'on', 'foundational', 'literacy', '.', 'vocational',
         n', 'in', 'grade', '6', 'with', 'internships', '.', 'up', 'to', 'at', 'least', 'grade', '5', 'shoulc
         he', 'mother', 'tongue', '.', 'according', 'to', 'nep', '2020', ',', 'it', 'has', 'been', 'dividing'
         f', '10+2', 'system', 'into', 'the', '5+3+3+4', 'format', '.', 'flexibility', 'of', 'higher', 'educa
         l', 'be', 'added', '.']
         ['national', 'education', 'policy', '2020', 'includes', 'nearly', '2', 'lakh', 'suggestions', '2.5',
         s', ',', '6600', 'blocks', ',', '6000', 'urban', 'local', 'bodies', ',', '676', 'districts', '.', '2
         'aims', 'universalize', 'education', 'pre-school', 'secondary', 'level', '.', 'strong', 'emphasis',
         '.', 'vocational', 'education', 'begin', 'grade', '6', 'internships', '.', 'least', 'grade', '5', 't
```

Fig. 4 Removing stop words for text summarization

```
>>> fdist1=fdist.most_common(9)
>>> fdist1
[('the', 2), ('.', 2), ('According', 1), ('to', 1), ('NEP', 1),
('2020', 1), (',', 1), ('it', 1), ('has', 1)]
# Opening a jupyter notebook
```

```
In [75]: text = word_tokenize("National Education Policy 2020 includes nearly 2 lakh
         nltk.pos_tag(text)

Out[75]: [('National', 'NNP'),
          ('Education', 'NN'),
          ('Policy', 'NN'),
          ('2020', 'CD'),
          ('includes', 'VBZ'),
          ('nearly', 'RB'),
          ('2', 'CD'),
          ('lakh', 'JJ'),
          ('suggestions', 'NNS'),
          ('from', 'IN'),

In [77]: text = nltk.Text(word.lower() for word in nltk.corpus.brown.words())
         text.similar('National')

         the state new a american general this federal one present first
         community social economic world two time house school christian

In [78]: tagged_token = nltk.tag.str2tuple('policy/NN')
         tagged_token
         tagged_token[0]

Out[78]: 'policy'

In [72]: tagged_token[1]

Out[72]: 'NN'

In [80]: nltk.corpus.brown.tagged_words()
         nltk.corpus.brown.tagged_words(tagset='policy')

Out[80]: [('The', 'UNK'), ('Fulton', 'UNK'), ('County', 'UNK'), ...]

In [81]: print(nltk.corpus.nps_chat.tagged_words())
         nltk.corpus.conll2000.tagged_words
         nltk.corpus.treebank.tagged_words()

         [('now', 'RB'), ('im', 'PRP'), ('left', 'VBD'), ...]

Out[81]: [('Pierre', 'NNP'), ('Vinken', 'NNP'), (',', ','), ...]
```

Fig. 5 Classifying words using POS-tagging, tagged token and Brown Corpus

```
File  Edit  View  Insert  Cell  Kernel  Help
Run  Code

In [1]: import re
        txt="National Education Policy 2020 includes nearly 2 lakh suggestions from 2.5 lakh
        r=re.search("^National.*Districts$",txt)
        if r:
            print("It found")
        else:
            print("Not found")

        It found

In [2]: r=re.findall("i",txt)
        print(r)

        ['i', 'i', 'i', 'i', 'i', 'i', 'i', 'i']
```

Fig. 6 Importing re (regular expression) module for finding

```
In [3]: r=re.findall("rajasthan",txt)
        print(r)
        if r:
            print("It found")
        else:
            print("Not found")

        []
        Not found

In [4]: r=re.search("\s",txt)
        print("From starting whit-space character is located at:",r.start())

        From starting whit-space character is located at: 8

In [5]: r=re.search("rajasthan",txt)
        print(r)

        None

In [6]: r=re.split("\s",txt)
        print(r)

        ['National', 'Education', 'Policy', '2020', 'includes', 'nearly', '2', 'lakh', 'sugge
        anchayats,', '6600', 'Blocks,', '6000', 'Urban', 'Local', 'Bodies,', 'and', '676', 'D

In [7]: r=re.split("\s",txt,1)
        print(r)

        ['National', 'Education Policy 2020 includes nearly 2 lakh suggestions from 2.5 lakh
        ocal Bodies, and 676 Districts']

In [8]: r=re.sub("\s","Rajasthan",txt)
        print(r)

        NationalRajasthanEducationRajasthanPolicyRajasthan2020RajasthanincludesRajasthannearl
        nsRajasthanfromRajasthan2.5RajasthanlakhRajasthanGramRajasthanPanchayats,Rajasthan660
        anRajasthanLocalRajasthanBodies,RajasthanandRajasthan676RajasthanDistricts

In [9]: r= re.search("li", txt)
        print(r)

        <re.Match object; span=(21, 23), match='li'>

In [10]: x = re.search(r"\bN\w+", txt)
         print(x.group())

         National
```

Fig. 7 Finding, searching, splitting, replacing patterns

```
In [61]: from bs4 import BeautifulSoup
         from bs4.element import Comment
         from nltk.tokenize import sent_tokenize, word_tokenize
         import urllib.request

         def visible_tag(element):
             if element.parent.name in ['style', 'script', 'head', 'title', 'meta', '[document]']:
                 return False
             if isinstance(element, Comment):
                 return False
             return True
```

Fig. 8 Importing beautiful soup and nltk.tokenize

```
def html_text(body):
    soup = BeautifulSoup(body, 'html.parser')
    texts = soup.findAll(text=True)
    visible_texts = filter(visible_tag, texts)
    return u" ".join(t.strip() for t in visible_texts)
```

```
html = urllib.request.urlopen('https://education.rajasthan.gov.in/content/raj/education/secondary
html_data = html_text(html)
html_data = (f'"{html_data}"')
print(f"Sentence Tokenization: ")
print(sent_tokenize(html_data))
```

```
Sentence Tokenization:
['"            Government of Rajasthan  Secondary Education                    Toggle navigation
About Us    down-arrow   Department at a glance    Administrative Structure        Orders/Notific
u   Orders/Notifications/Circulars    down-arrow  AB Section(HM/Principal)    ACP(Assured Career I
S/Death    NOC(No Objection Certificate)    DPC(Departmental Promotion Committee)    Fixation/Lea
ansfer/Deputation/APO Orders    Legal Orders(WRIT)    Others    C Section(Lecturers)    ACP(Assu
irement/VRS/Death    NOC(No Objection Certificate)    DPC(Departmental Promotion Committee)    RP
ion/APO Orders    Legal Orders(WRIT)    Fixation/Leave Orders    Others    F Section(Other Grade
Leave Orders    Transfer/Deputation/APO Orders    6D/3B Orders    Court Case    Others    Second
```

Fig. 9 Importing sent_tokenize() and word_tokenize() from nltk.tokenize package using Beautiful Soup

```
C:\Users\HP\AppData\Local\Programs\Python\Python39>jupyter
notebook
[W 14:47:40.293 NotebookApp] Terminals not available (error was No
module named 'winpty.cywinpty')
[I 14:47:40.543 NotebookApp] Serving notebooks from local direc-
tory: C:\Users\HP\AppData\Local\Programs\Python\Python39
[I 14:47:40.543 NotebookApp] Jupyter Notebook 6.2.0 is running at:
[I 14:47:40.543 NotebookApp] http://localhost:8888/
?token=85319cedbe702cff61e821a7e71b767c23e5c6db032d48ef
[I 14:47:40.559 NotebookApp] or http://127.0.0.1:8888/
?token=85319cedbe702cff61e821a7e71b767c23e5c6db032d48ef
[I 14:47:40.559 NotebookApp] Use Control-C to stop this server and
shut down all kernels (twice to skip confirmation).
[C 14:47:40.637 NotebookApp]
To access the notebook, open this file in a browser: file://
/C:/Users/HP/AppData/Roaming/jupyter/runtime/nbserver-1700-
open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=85319cedbe702cff61e821a7e71b767c
23e5c6db032d48ef
or http://127.0.0.1:8888/?token=85319cedbe702cff61e821a7e71b767c
23e5c6db032d48ef
[W 14:49:57.733 NotebookApp] 404 GET /undefined/undefined (::1)
22.060000ms referer=None
[I 14:53:45.992 NotebookApp] Creating new file in
[I 14:53:46.054 NotebookApp] Creating new notebook in
[I 14:53:46.443 NotebookApp] Creating new notebook in
[I 14:53:46.683 NotebookApp] Creating new notebook in
[W 14:53:46.939 NotebookApp] 404 GET /undefined/undefined (::1)
29.570000ms referer=None
```

```
[I 14:53:46.943 NotebookApp] Creating new notebook in
[I 14:53:51.419 NotebookApp] Kernel started: d18dbe85-4850-45b2-
a71f-534acdb74e99, name: python3


#Using urllib.request for fetching URLs
>>> import urllib.request
>>> response = urllib.request.urlopen('https://www.rajasthanshi
ksha.com/')
>>> html = response.read()
>>> print(html)
b'<!doctype html ><html class="ie8" lang="en"><html lang="en-US">
'
'
Continue URL page source display


#Installing beautifulsoup4, a Python package to drag data from HTML
and XML files
>>> import bs4 as bs
>>> parsed_article=bs.BeautifulSoup(html,'lxml')
>>> text = parsed_article.get_text
>>> print(text)
<bound method PageElement.get_text of <!DOCTYPE html>
<html class="ie9" lang="en"> <head></head><body>
'
Continue URL page source display
,
</body></html>
```

3 Results

It is the scheme of problem-solving arrangements after loading, examining, and executing the data [24–27]. It is viewed that NLP grants the way of interconnecting SML with users. To the cognition of education policy, NLP offers tokenization, stemming, lemmatization, and classifying words using POS-tagging, urllib.request, re-module, and beautifulsoup4 for text mining and summarization [28, 29] (Fig. 10).

By using unstructured interviews and the responses to a website questionnaire form, it has been proven that it is easy to get satisfaction after using NLP through a generated online website's address, https://drpoojajain.in/Chartreport.aspx?aa=51E7F0C352CE201B50C8EC347DE68701AC365347058EEF911308281EE25E4DBE392E3B51E564391A2B48F363740728D4F43F52596D548B65B9FB54ED49AC83882F6C08EF.

This analysis tends toward the mining of web pages using NLP. It was focused on a more appropriate way of doing text mining. Unstructured data was collected through educational websites for summarization [30, 31]. The whole work made up an interactive platform.

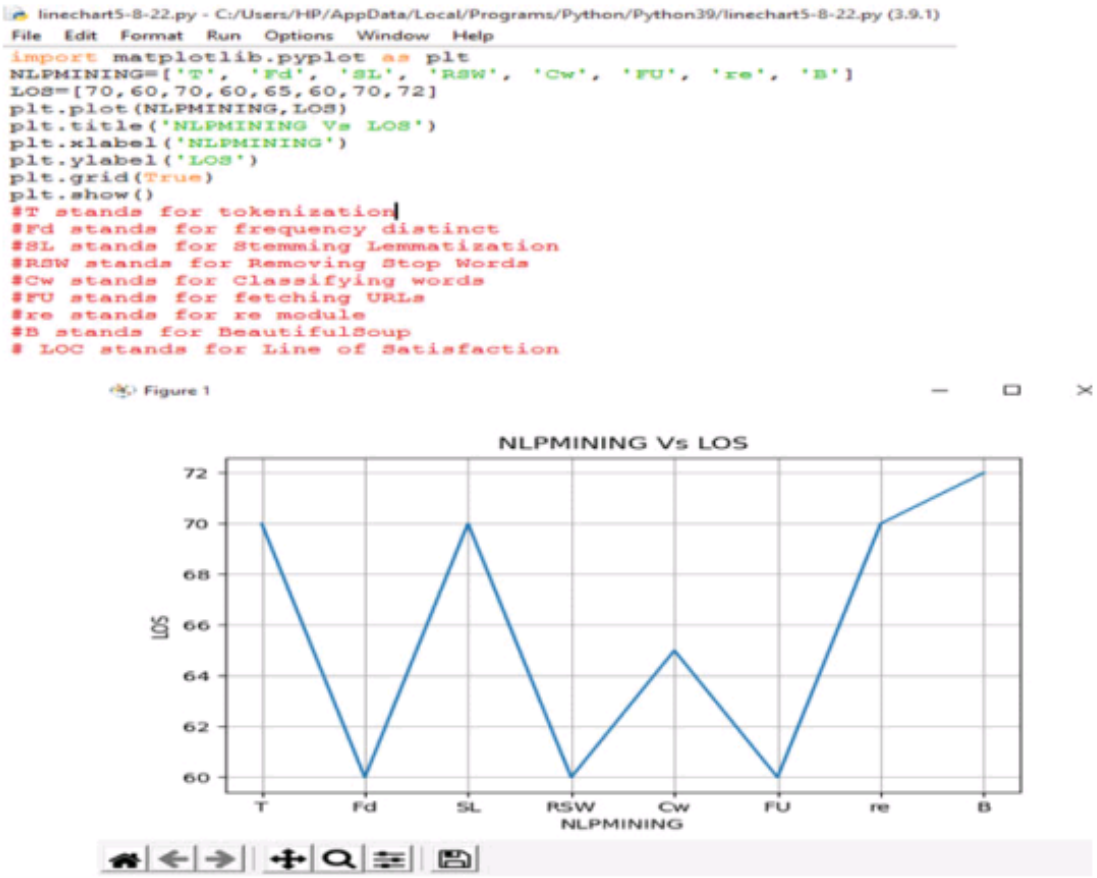

Fig. 10 NLPMINING versus line of satisfaction

4 Conclusion

In this culmination of research, a reader will keep an expeditious technique to extricate about education policy. NLP has reached the level of an interdisciplinary area of AI. Web text mining serves a valuable task in finding relevant information from big data. Thus, several web codes of NLP show how to mine the web contents of policy and can be mined for appropriate data using Python. It will be beneficial for the user to obtain mining and summary as per requirements. The principle of this paper is to accumulate the generated description of the complicated text and present a way to bring out the fruitful product, and it will enhance the knowledge of users for making tools for mining techniques.

References

1. https://www.linguamatics.com/what-text-mining-text-analytics-and-natural-language-processing
2. Zhang Q, Segall RS (2008) Web mining, a survey of current research, techniques, and software. IJITDM 7(4):683–720

3. Bhardwaj B (2012) Extracting data through web mining. Int J Eng Res Technol (IJERT) 1(3). ISSN: 2278-0181
4. Maes P (1994) Agents that reduce work and information overload. Commun ACM 7:30–40
5. Shahmoradi L (2014) Structure-based web pages clustering. Int J Sci Eng Res 5(4). ISSN: 2229–5518
6. https://www.datamation.com/big-data/structured-vs-unstructured-data.html
7. Tsuyoshi M, Saito K (2006) Extracting user's interest for web log data. IEEE 343–346. ISBN: 0-7695-2747-7
8. Malarvizhi R, Saraswathi K (2013) Web content mining techniques tools & algorithms—a comprehensive study. IJCTT 4(8). ISSN: 2231-2803
9. https://shodhganga.inflibnet.ac.in:8443/jspui/handle/10603/334941
10. Shetty S, Hans V (2019) Education for skill development and women empowerment. EPRA Int J Econ Bus Rev Peer Reviewed J 7(2). e-ISSN: 2347–9671|, p-ISSN: 2349-0187
11. Inamdar SA, Shinde GN (2011) Web data mining using an intelligent information system design. Int J Comput Tech Appl 280–283. ISSN: 2229–6093
12. Saini S, Pandey HM (2015) Review on web content mining techniques. Int J Comput Appl 118(18) (0975–8887)
13. Khalili A (2008) A semantic web service-oriented model for project management. In: IEEE 8th international conference on computer and information technology workshops. CIT Workshops, pp 667–672
14. Fedak G (2009) BitDew: a data management and distribution service with multi-protocol file transfer and metadata abstraction. J Netw Comput Appl 32(5) [Next Generation Content Networks, pp 961–975]
15. Mebrahtu A, Srinivasulu B (2017) Web content mining techniques and tools. IJCSMC 6(4):49–55. ISSN: 2320-088X
16. Barfourosh AA, Motahary Nezhad HR, Anderson ML, Perlis D (2002) Information retrieval on the world wide web and active logic: a survey and problem definition
17. https://searchenterpriseai.techtarget.com/definition/natural-language-processing-NLP
18. Barba P (2020) Machine learning (ML) for natural language processing (NLP), September 29, 2020. https://www.lexalytics.com/lexablog/
19. https://www.rajasthanshiksha.com/download-national-edcuation-policy-2020-pdf/
20. https://www.rajras.in/education/
21. https://economictimes.indiatimes.com/jobs/making-skilling-part-of-education-system-a-challenging-task/articleshow/67633636.cms?from=mdr
22. https://www.mid-day.com/amp/mumbai/mumbai-news/article/new-education-policy-will-suffocate-medical-education-22945232
23. https://www.kdnuggets.com/2018/11/text-preprocessing-python.html
24. Sharma P, Bhartiya R (2012) An efficient algorithm for improved web usage mining, vol 3, issue 2, pp 766–769. ISSN: 2229-6093
25. Beniwal R, Tanwar V (2014) Evaluation of web personalization. IJIRST 1(6). ISSN: 2349-6010
26. Ameen A, Khan KUR, Rani BP (2012) Semantic web personalization: a survey. Inf Knowl Manage 2(6). ISSN: 2224-5758
27. Yadav M, Mittal P (2013) Web mining: an introduction. Int J Adv Res Comput Sci Softw Eng 3(3). ISSN: 2277 128X
28. https://towardsdatascience.com/gentle-start-to-natural-language-processing-using-python-6e46c07addf3
29. https://machinelearningmastery.com/clean-text-machine-learning-python/
30. Vijayarani S, Suganya E, Prakathambal M (2018) Web log files in web usage mining research—a review, vol 5, issue 2. ISSN: 2394-2320
31. Ratnakumar AJ (2010) An Implementation of web personalization using web mining techniques. J Theor Appl Inf Technol [2005–2010 JATIT]