KEYWORD EXTRACTION FROM THE NEWS FEED OR BLOGPOSTS WITH IBM CLOUD

1.Introduction

a) Overview: Keyword extraction is a text analysis technique that automatically extracts the most used and most important words and expressions from a text. It helps summarize the content of texts and recognize the main topics discussed.

Keyword extraction uses machine learning artificial intelligence (AI) with natural language processing (NLP) to break down human language so that it can be understood and analysed by machines. It’s used to find keywords from all manner of text: regular documents and business reports, social media comments, online forums and reviews, news reports, and more.

And In this project flask frame have been used to create an app and run the application from browser itself. Like taking input from user like text from user and extracting minimum no of keywords from the text file and printing the keywords on to the keywords.html file.

Mainly in this project we are using rapid api which helps us to extract keywords and remove non keywords. When connecting an API to a project or application, you must have an API key to authenticate your request. Creating an app within RapidAPI generates an API key (X-RapidAPI-Key) specific to that application. You can view analytics based on the API calls you make using this app key.

Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

b). Purpose: Purpose of the project is to extract keywords from blog posts as well as from any text file. When we want to understand key information from specific documents, we typically turn towards keyword extraction. Keyword extraction is the automated process of extracting the words and phrases that are most relevant to an input text.

It is a text analysis technique. We can obtain important insights into the topic within a short span of time. It helps concise the text and obtain relevant keywords. It saves the time of going through the entire document. Example use-cases are finding topics of interest from a news article and identifying the problems based on customer reviews and so. One of the techniques used for Keyword Extraction is TF-IDF (Term Frequency – Inverse Document Frequency)]

- Keywords themselves can be useful, particularly in formulating a response to “What are people most frequently talking/asking about?”

- Keywords can help you focus in on smaller sets of individual records in order to learn more about them and begin to answer particular questions about user needs and goals
- Keywords in combination with analysis of smaller sets of individual records can help you identify gaps in your understanding of users that can help focus subsequent research efforts

2. Literature Survey

In 1995, J.D. Cohen proposed an approach to draw index terms from text [4]. It doesn‘t use any stop list, stemmer, or any language and domain-specific component, allowing for easy application in any language or domain with slight modification. The method utilizes n-gram counts, which results in a function similar and more general than a stemmer. In 2002, M. Orton et al. demonstrated that important words of a text have a tendency to attract each other and form clusters [5]. He argues that the standard deviation of the distance between successive occurrences of a word is such a parameter to quantify this self-attraction.

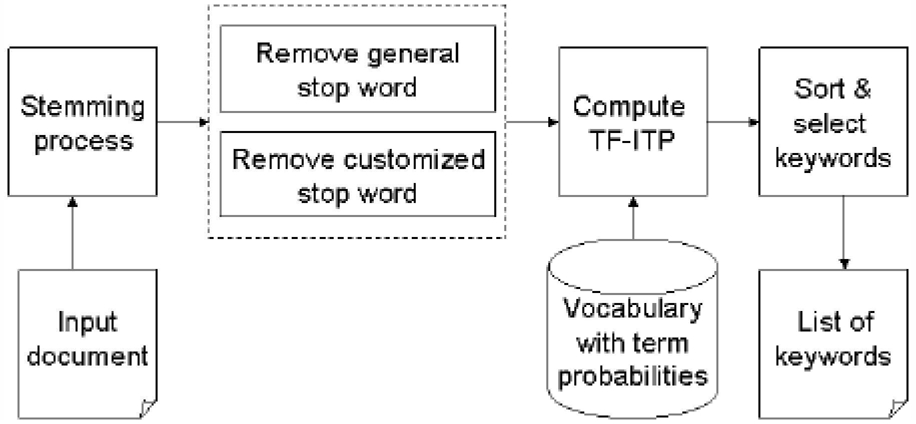
P. Carpena et al. proposed to automatically extract keywords from literary texts through a generalization of the level statistics analysis of quantum disordered systems [7]. They consider frequencies of the words along with their spatial distribution along the text, and is based on the observation that important words are significantly clustered whereas irrelevant words are distributed randomly in the text. No reference corpus is needed in this approach and it is especially suitable for single documents for which no priori information is available.

a)Existed System: The existed system is not able work on all type of data like blog posts must considering only text files whereas coming to this project it is about to extract keywords from blog posts also and also not been deployed with IBM cloud and also not used any frameworks like flask ,rapid api. Not able to extract keywords accurately.

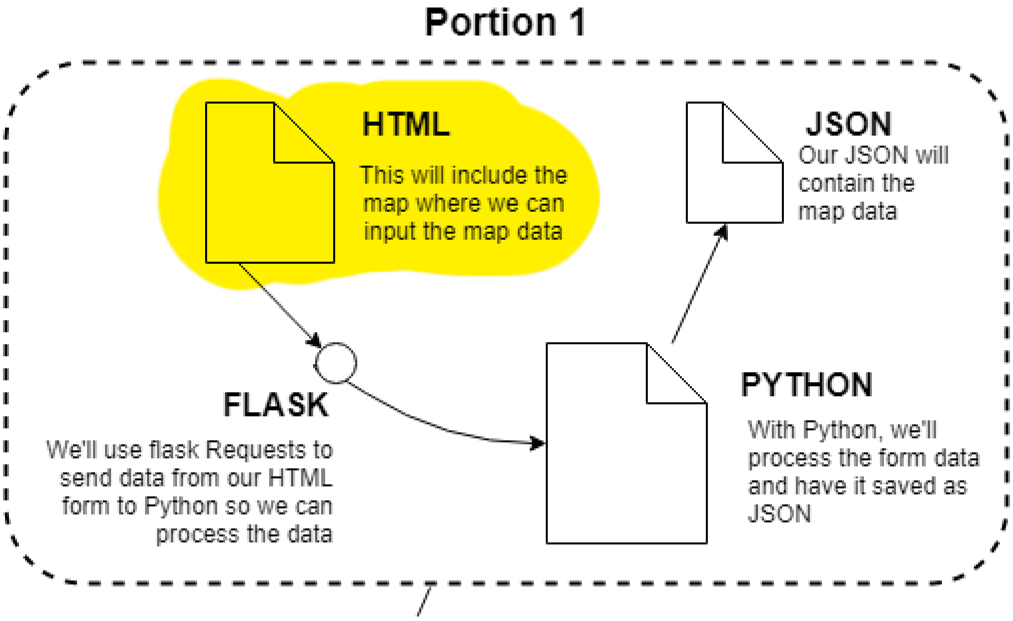
b) Proposed System: Keyword extraction is the automated process of extracting the most relevant words and expressions from the text. With more than 290 billion emails sent and received on a daily basis, and half a million tweets posted every single minute, using machines to analyse huge sets of data and extract important information is definitely a game-changer. We are building a flask application by using a keywords extraction API. This API uses advanced Natural Language Processing techniques to extract the most important 'Keywords' and 'Phrases' from a LIST of text or URL that you provide. It can take any type of text as input and will perform best on URLs corresponding to news, blog, content, etc. It can also take MULTIPLE text or URLs as input and you can specify the number of keywords that need to be extracted.

3. Theoretical Analysis:

a)Block Diagram:



b) **Hardware / Software designing**


Portion 1
HTML
This will include the map where we can input the map data
JSON
Our JSON will contain the map data
FLASK
We'll use flask Requests to send data from our HTML form to Python so we can process the data
PYTHON
With Python, we'll process the form data and have it saved as JSON

4. Experimental Investigations:

Experimental Investigations refers to researched some research papers which helped us to implement with help of flask framework as well as usage of rapid API. And also using IBM cloud which is a good platform which helps to deploy model.

IBM Watson on the IBM Cloud helps to transform businesses, enhancing competitive advantage and disrupting industries by unlocking the potential within unstructured data. Fundamental to providing a strong foundation for companies wanting to leverage Watson

Services, IBM uses best-in-class security and compliance processes that allow for successful execution of challenging workloads.

IBM Cloud Deployment Services (ICDS) reduces complexity by integrating multicloud environments with a single orchestration platform. That easily integrates with existing tools and architectures and is technology agnostic.

ICDS improves DevOps through automated patterns and workflows, boosts operational efficiency and reduces services deployment time. Zero touch IT helps with resource management and tracks governance issues with a self-service portal. Get virtualization and container orchestration with cloud native infrastructure support by building your own platforms.

5. Flowchart

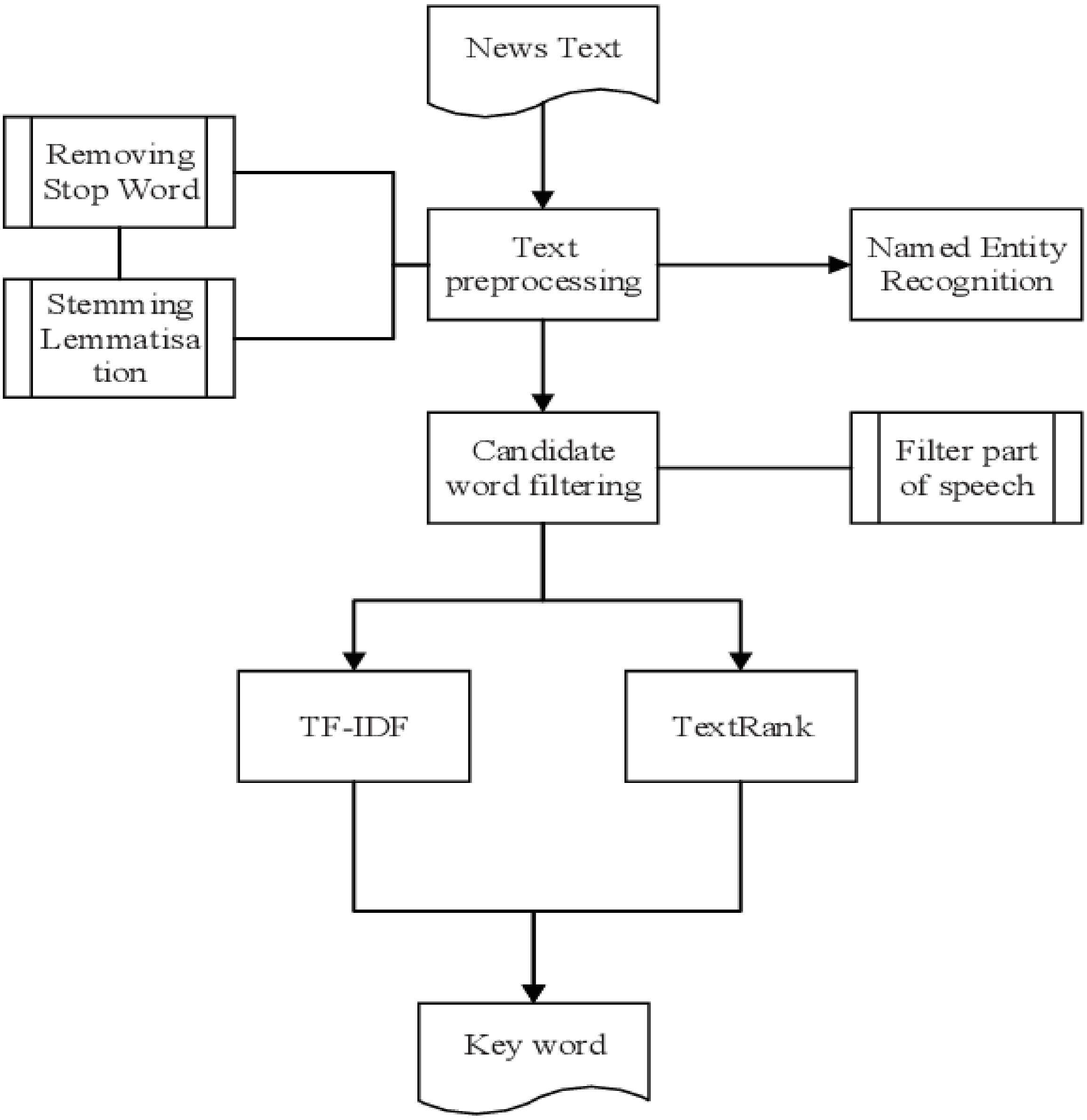


Fig. 2.   Keyword Extraction Flow Chart

6.Result:

Gives output a set of keywords on the keywords.html file of required count of keywords .And also flowchart from IBM cloud platform on deploying the model. UI pages with a nice layout with different buttons which helps to take input from user a text file and extract the data and print in the json format.

7. Advantages & Disadvantages:

Advantages: Advantages refers to deploying the model and also no need of more amount of python code and also using API can easily find out keywords. Helps to find the keywords to find the recommended systems like finding the most watched or most buying products so helps for suggestions in the recommended systems. Keywords mainly helps to find a video or a product based on keywords .

Disadvantages: Disadvantages are less in number as it is deploying the model with IBM cloud helps to reduce error in finding the keywords which or not. Suggesting fake videos or fake products or fake movies etc. Mainly in Recommendation systems.

8. Applications:

1. Movie Recommendation System

2. Sentiment Analysis

3.Youtube video statistics.

4. finding required documents

5. Groceries recommendation system

9. Conclusion

By the end of the project we have implemented the project successfully without any errors with a beautiful UI and developed an app for taking text as input and find the keywords. Mainly this project consists of home.html, extractor.html, keywords.html and extractor.py these are the files which helped to find the keywords and deployed with IBM cloud.

10. Future Scope

This can be implemented in more detail like adding url which will redirect to that blog and extract the text and find the keywords and place it in json file and print them on keywords.html file. Using a logic to find the keywords rather than using RAPID API.

11. Bibliography

1. https://towardsdatascience.com/build-a-keyword-extraction-api-with-spacy-flask-and-fuzzywuzzy-4909d7ffc105

2.https://www.researchgate.net/publication/313469342_Totally_automated_keyword_extraction

3. https://ieeexplore.ieee.org/abstract/document/9076460

4. https://paperswithcode.com/task/keyword-extraction

5.https://www.researchgate.net/publication/319979502_An_empirical_study_of_important_keyword_extraction_techniques_from_documents

6. Feather, J. and S. P., International encyclopedia of information and library science. London & New York: Routledge, 1996

12. Appendix

a) Source code

Extraction.py

```
from flask import Flask, request, render_template

import re

import requests



app = Flask(__name__)



@app.route('/')

def home():

    return render_template('home.html')



@app.route('/extractor')

def extractor():

    return render_template('extractor.html')
```

```
@app.route('/keywords', methods=['POST'])

def keywords():

    sen = request.form['text']

    num = request.form['wordnum']

    keyword = check(sen, num)

    return render_template('keywords.html', keyword=keyword)


def check(sentence, num):

    url = "https://textanalysis-keyword-extraction-v1.p.rapidapi.com/keyword-extractor-text"

    payload = "text=" + sentence + "&wordnum=" + num

    headers = {

        'content-type': "application/x-www-form-urlencoded",

        'x-rapidapi-key': "bdf00f69d0mshf71b715e665de84p187496jsn103deb88c72c",

        'x-rapidapi-host': "textanalysis-keyword-extraction-v1.p.rapidapi.com"

        }


    response = requests.request("POST", url, data=payload, headers=headers)

    return response.json()['keywords']


if __name__ == "__main__":

    app.run(debug="")
```

home.html

```
<!DOCTYPE html>
```

```html
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
<!--[if IE 7]>         <html class="no-js lt-ie9 lt-ie8"> <![endif]-->
<!--[if IE 8]>         <html class="no-js lt-ie9"> <![endif]-->
<!--[if gt IE 8]>      <html class="no-js"> <!--<![endif]-->
<html>


<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Home</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css" rel="stylesheet">
</head>


<body>
    <nav class="bg-gray-800">
        <div class="max-w-7xl mx-auto px-2 sm:px-6 lg:px-8">
            <div class="relative flex items-center justify-between h-16">
                <div class="absolute inset-y-0 left-0 flex items-center sm:hidden">
                    <!-- Mobile menu button-->
                    <button type="button"
                        class="inline-flex items-center justify-center p-2 rounded-md text-gray-400 hover:text-white hover:bg-gray-700 focus:outline-none focus:ring-2 focus:ring-inset focus:ring-white"
                        aria-controls="mobile-menu" aria-expanded="false">
                        <span class="sr-only">Open main menu</span>
                        <!--
                      Icon when menu is closed.

                      Heroicon name: outline/menu

                      Menu open: "hidden", Menu closed: "block"
                    -->
                        <svg class="block h-6 w-6" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24"
                            stroke="currentColor" aria-hidden="true">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
                                d="M4 6h16M4 12h16M4 18h16" />
                        </svg>
                        <!--
                      Icon when menu is open.

                      Heroicon name: outline/x

                      Menu open: "block", Menu closed: "hidden"
                    -->
                        <svg class="hidden h-6 w-6" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24"
                            stroke="currentColor" aria-hidden="true">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
                                d="M6 18L18 6M6 6l12 12" />
                        </svg>
```

```html
                </button>

            </div>

            <div class="flex-1 flex items-center justify-center sm:items-stretch sm:justify-start">

                <div class="flex-shrink-0 flex items-center">

                    <p class="text-white font-medium text-lg">Keyword Extraction</p>

                </div>

            </div>

            <div class="absolute inset-y-0 right-0 flex items-center pr-2 sm:static sm:inset-auto sm:ml-6 sm:pr-0">

                <div class="hidden sm:block sm:ml-6">

                    <div class="flex space-x-4">

                        <!-- Current: "bg-gray-900 text-white", Default: "text-gray-300 hover:bg-gray-700 hover:text-white" -->

                        <a href="#" class="bg-gray-900 text-white px-3 py-2 rounded-md text-sm font-medium"

                            aria-current="page">Home</a>



                        <a href="/extractor"

                            class="text-gray-300 hover:bg-gray-700 hover:text-white px-3 py-2 rounded-md text-sm font-medium">Extractor</a>

                    </div>

                </div>

            </div>

        </div>



        <!-- Mobile menu, show/hide based on menu state. -->

        <div class="sm:hidden" id="mobile-menu">
```

```
                <div class="px-2 pt-2 pb-3 space-y-1">
                    <!-- Current: "bg-gray-900 text-white", Default: "text-gray-300 hover:bg-gray-700 hover:text-white" -->
                    <a href="/home" class="bg-gray-900 text-white block px-3 py-2 rounded-md text-base font-medium"
                        aria-current="page">Home</a>

                    <a href="/extractor"
                        class="text-gray-300 hover:bg-gray-700 hover:text-white block px-3 py-2 rounded-md text-base font-medium">Extractor</a>
                </div>
            </div>
    </nav>
    <div class="container mx-auto px-4 flex flex-col justify-center items-center">
        <img src="{{url_for('static', filename='keyword.svg')}}" alt="Keyword by FlatIcon" class="my-4" />
        <h1 class="text-2xl font-bold leading-7 text-gray-900 sm:text-3xl sm:truncate">Welcome to Keyword Extraction.
        </h1>
        <a href="/extractor" class="py-2 px-3 rounded-md bg-gray-700 mt-4 transition-colors hover:bg-gray-500 text-white font-bold">Click here to extract.</a>
    </div>
    <script src="" async defer></script>
</body>

</html>
```

Extraction.html

```
<!DOCTYPE html>
```

```
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->
<!--[if IE 7]>         <html class="no-js lt-ie9 lt-ie8"> <![endif]-->
<!--[if IE 8]>         <html class="no-js lt-ie9"> <![endif]-->
<!--[if gt IE 8]>      <html class="no-js"> <!--<![endif]-->
<html>

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Extractor</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css" rel="stylesheet">
</head>

<body>
    <!--[if lt IE 7]>
            <p class="browsehappy">You are using an <strong>outdated</strong> browser. Please <a href="#">upgrade your browser</a> to improve your experience.</p>
        <![endif]-->

    <nav class="bg-gray-800">
        <div class="max-w-7xl mx-auto px-2 sm:px-6 lg:px-8">
            <div class="relative flex items-center justify-between h-16">
                <div class="absolute inset-y-0 left-0 flex items-center sm:hidden">
                    <!-- Mobile menu button-->
                    <button type="button"
```

```
class="inline-flex items-center justify-center p-2 rounded-md text-gray-400 hover:text-white hover:bg-gray-700 focus:outline-none focus:ring-2 focus:ring-inset focus:ring-white"

aria-controls="mobile-menu" aria-expanded="false">

<span class="sr-only">Open main menu</span>

<!--

  Icon when menu is closed.



  Heroicon name: outline/menu



  Menu open: "hidden", Menu closed: "block"

-->

<svg class="block h-6 w-6" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24"

    stroke="currentColor" aria-hidden="true">

    <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"

        d="M4 6h16M4 12h16M4 18h16" />

</svg>

<!--

  Icon when menu is open.



  Heroicon name: outline/x



  Menu open: "block", Menu closed: "hidden"

-->

<svg class="hidden h-6 w-6" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24"
```

```
                            stroke="currentColor" aria-hidden="true">
                            <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
                                d="M6 18L18 6M6 6l12 12" />
                        </svg>
                    </button>
                </div>
                <div class="flex-1 flex items-center justify-center sm:items-stretch sm:justify-start">
                    <div class="flex-shrink-0 flex items-center">
                        <p class="text-white font-medium text-lg">Keyword Extraction</p>
                    </div>
                </div>
                <div class="absolute inset-y-0 right-0 flex items-center pr-2 sm:static sm:inset-auto sm:ml-6 sm:pr-0">
                    <div class="hidden sm:block sm:ml-6">
                        <div class="flex space-x-4">
                            <!-- Current: "bg-gray-900 text-white", Default: "text-gray-300 hover:bg-gray-700 hover:text-white" -->
                            <a href="/"
                                class="text-gray-300 hover:bg-gray-700 hover:text-white px-3 py-2 rounded-md text-sm font-medium"
                                aria-current="page">Home</a>

                            <a href="#"
                                class="bg-gray-900 text-white px-3 py-2 rounded-md text-sm font-medium">Extractor</a>
                        </div>
                    </div>
                </div>

            </div>



            <!-- Mobile menu, show/hide based on menu state. -->

            <div class="sm:hidden" id="mobile-menu">

                <div class="px-2 pt-2 pb-3 space-y-1">

                    <!-- Current: "bg-gray-900 text-white", Default: "text-gray-300 hover:bg-gray-700 hover:text-white" -->

                    <a href="/home" class="bg-gray-900 text-white block px-3 py-2 rounded-md text-base font-medium"

                        aria-current="page">Home</a>



                    <a href="/extractor"

                        class="text-gray-300 hover:bg-gray-700 hover:text-white block px-3 py-2 rounded-md text-base font-medium">Extractor</a>

                </div>

            </div>

    </nav>

    <div class="container mx-auto px-4 flex flex-col justify-center">

        <h1 class="text-2xl font-bold leading-7 text-gray-900 sm:text-3xl sm:truncate mt-4">Enter the text you want to extract...</h1>

        <form action="/keywords" method="POST">

            <label for="text" class="block mb-2 text-sm text-gray-600 dark:text-gray-400 mt-4">Text to Extract</label>

            <textarea name="text" class="resize-none border rounded-md w-3/4 h-64 px-3 py-2">Keyword extraction is tasked with the automatic identification of terms that best describe the subject of a document. Key phrases, key terms, key segments or just keywords are the terminology which is used for defining the terms that represent the most relevant information contained in the document. Although the terminology is different, function is the same: characterization of the topic discussed in a document. Keyword extraction task is important problem in Text Mining, Information Retrieval and Natural
```

```
Language Processing. Keyword assignment vs. extraction Keyword assignment methods can be roughly divided into: keyword assignment (keywords are chosen from controlled vocabulary or taxonomy) and keyword extraction (keywords are chosen from words that are explicitly mentioned in original text). Methods for automatic keyword extraction can be: supervised, semi-supervised and unsupervised. Unsupervised methods can be further divided into: simple statistics, linguistics, graph-based and other methods.</textarea>

            <label for="wordnum" class="block mb-2 text-sm text-gray-600 dark:text-gray-400 mt-4">Number of Keywords</label>

            <input type="number" min="0" name="wordnum" placeholder="5" required class="w-3/4 px-3 py-2 placeholder-gray-300 border border-gray-300 rounded-md focus:outline-none focus:ring focus:ring-indigo-100 focus:border-indigo-300 dark:bg-gray-700 dark:text-white dark:placeholder-gray-500 dark:border-gray-600 dark:focus:ring-gray-900 dark:focus:border-gray-500" />

            <button type="submit" class="w-1/4 font-bold mt-4 px-3 py-2 text-white bg-gray-700 rounded-md">Extract...</button>

        </form>

    </div>

</body>



</html>
```

Keywords.html

```
<!DOCTYPE html>

<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8 lt-ie7"> <![endif]-->

<!--[if IE 7]>         <html class="no-js lt-ie9 lt-ie8"> <![endif]-->

<!--[if IE 8]>         <html class="no-js lt-ie9"> <![endif]-->

<!--[if gt IE 8]>      <html class="no-js"> <!--<![endif]-->

<html>



<head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Keywords</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css" rel="stylesheet">
</head>

<body>
    <nav class="bg-gray-800">
        <div class="max-w-7xl mx-auto px-2 sm:px-6 lg:px-8">
            <div class="relative flex items-center justify-between h-16">
                <div class="absolute inset-y-0 left-0 flex items-center sm:hidden">
                    <!-- Mobile menu button-->
                    <button type="button"
                        class="inline-flex items-center justify-center p-2 rounded-md text-gray-400 hover:text-white hover:bg-gray-700 focus:outline-none focus:ring-2 focus:ring-inset focus:ring-white"
                        aria-controls="mobile-menu" aria-expanded="false">
                        <span class="sr-only">Open main menu</span>
                        <!--
                              Icon when menu is closed.

                              Heroicon name: outline/menu

                              Menu open: "hidden", Menu closed: "block"
                            -->
                        <svg class="block h-6 w-6" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24"
                            stroke="currentColor" aria-hidden="true">
```

```
              <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
                d="M4 6h16M4 12h16M4 18h16" />
            </svg>
            <!--
               Icon when menu is open.

               Heroicon name: outline/x

               Menu open: "block", Menu closed: "hidden"
              -->
            <svg class="hidden h-6 w-6" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24"
              stroke="currentColor" aria-hidden="true">
              <path stroke-linecap="round" stroke-linejoin="round" stroke-width="2"
                d="M6 18L18 6M6 6l12 12" />
            </svg>
          </button>
        </div>
        <div class="flex-1 flex items-center justify-center sm:items-stretch sm:justify-start">
          <div class="flex-shrink-0 flex items-center">
            <p class="text-white font-medium text-lg">Keyword Extraction</p>
          </div>
        </div>
        <div class="absolute inset-y-0 right-0 flex items-center pr-2 sm:static sm:inset-auto sm:ml-6 sm:pr-0">
          <div class="hidden sm:block sm:ml-6">
```

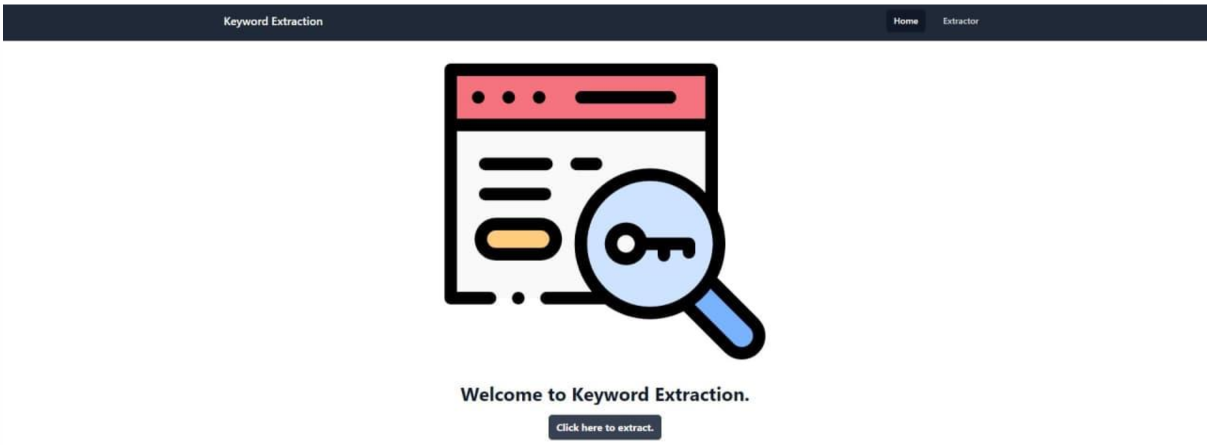
```
            <div class="flex space-x-4">
              <!-- Current: "bg-gray-900 text-white", Default: "text-gray-300 hover:bg-gray-700 hover:text-white" -->
              <a href="/"
                class="text-gray-300 hover:bg-gray-700 hover:text-white px-3 py-2 rounded-md text-sm font-medium"
                aria-current="page">Home</a>

              <a href="/extractor"
                class="text-gray-300 hover:bg-gray-700 hover:text-white px-3 py-2 rounded-md text-sm font-medium">Extractor</a>
            </div>
          </div>
        </div>
      </div>

      <!-- Mobile menu, show/hide based on menu state. -->
      <div class="sm:hidden" id="mobile-menu">
        <div class="px-2 pt-2 pb-3 space-y-1">
          <!-- Current: "bg-gray-900 text-white", Default: "text-gray-300 hover:bg-gray-700 hover:text-white" -->
          <a href="/home" class="bg-gray-900 text-white block px-3 py-2 rounded-md text-base font-medium"
            aria-current="page">Home</a>

          <a href="/extractor"
            class="text-gray-300 hover:bg-gray-700 hover:text-white block px-3 py-2 rounded-md text-base font-medium">Extractor</a>
            </div>
        </div>
    </nav>
    <div class="container mx-auto px-4">
        <h1 class="text-2xl font-bold leading-7 text-gray-900 sm:text-3xl sm:truncate mt-4">The extracted keywords
            are...</h1>
        <p id="keyword-list" class="my-4">{{ keyword }}</p>
    </div>
</body>

</html>
```

Test.py

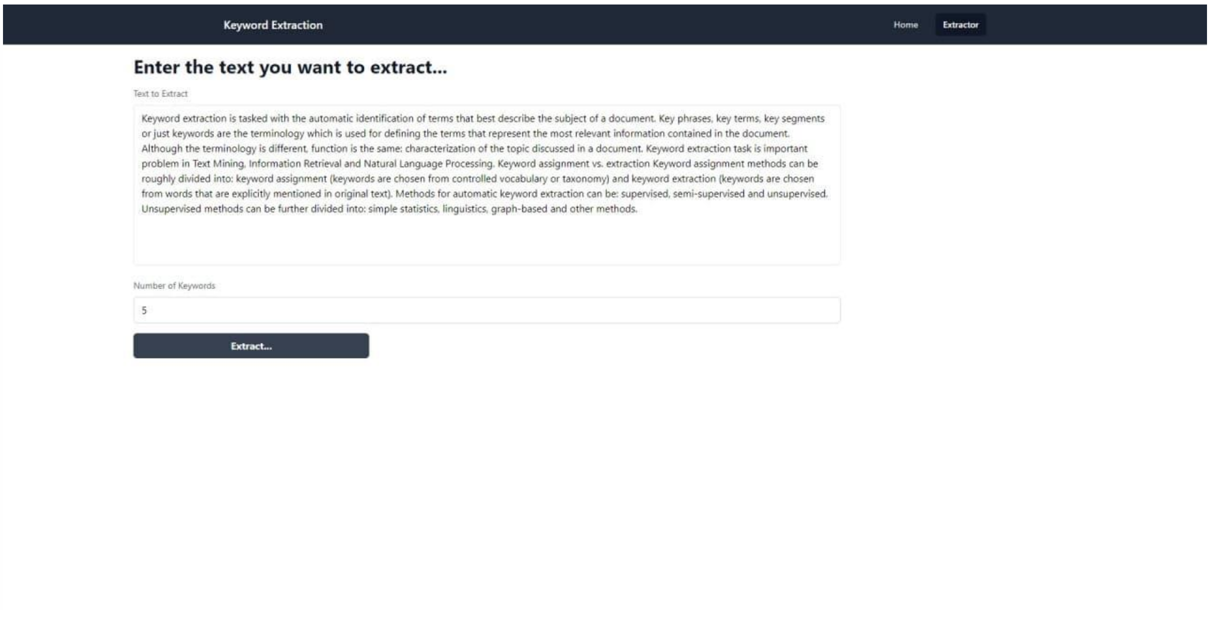
```
import requests


url = "https://textanalysis-keyword-extraction-v1.p.rapidapi.com/keyword-extractor-text"
payload = "text=Keyword%20extraction%20is%20tasked%20with%20the%20automatic%20identification%20of%20terms%20that%20best%20describe%20the%20subject%20of%20a%20document.%20Key%20phrases%2C%20key%20terms%2C%20key%20segments%20or%20just%20keywords%20are%20the%20terminology%20which%20is%20used%20for%20defining%20the%20terms%20that%20represent%20the%20most%20relevant%20information%20contained%20in%20the%20document.%20Although%20the%20terminology%20is%20different%2C%20function%20is%20the%20same%3A%20characterization%20of%20the%20topic%20discussed%20in%20a%20document.%20Keyword%20extraction%20task%20is%20important%20problem%20in%20Text%20Mining%2C%20Information%20Retrieval%20and%20Natural%20Language%20Processing.%20Keyword%20assignment%20vs.%20extraction%20Keyword%20assignment%20methods%20can%20be%20roughly%20divided%20into%3A%20keyword%20assignment%20(keywords%20are%20chosen%20from%20controlled%20vocabulary%20or%20taxonomy)%20and%20keyword%20extraction%20(keywords%20are%20chosen%20from%20words%20that%20are%20explicitly%20mentioned%20in%20original%20text).%20Methods%20for%20automatic%20keyword%20extraction%20can%20be%3A%20supervised%2C%20semi-supervised%20and%20unsupervised.%20Unsupervised%20methods%20can%20be%20f
```

```
urther%20divided%20into%3A%20simple%20statistics%2C%20linguistics%2C%20grap
h-based%20and%20other%20methods.&wordnum=100"

headers = {

    'content-type': "application/x-www-form-urlencoded",

    'x-rapidapi-key': "bdf00f69d0mshf71b715e665de84p187496jsn103deb88c72c",

    'x-rapidapi-host': "textanalysis-keyword-extraction-v1.p.rapidapi.com"

    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

b) UI output Screenshot.



Welcome to Keyword Extraction.



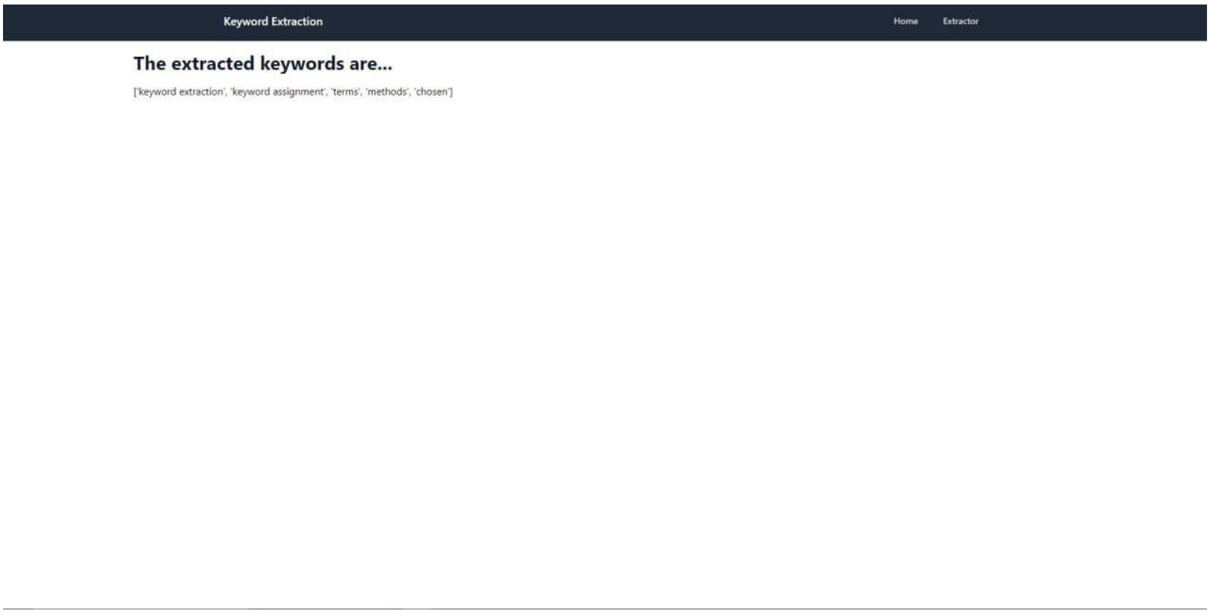
Enter the text you want to extract...

Text to Extract

Keyword extraction is tasked with the automatic identification of terms that best describe the subject of a document. Key phrases, key terms, key segments or just keywords are the terminology which is used for defining the terms that represent the most relevant information contained in the document. Although the terminology is different, function is the same: characterization of the topic discussed in a document. Keyword extraction task is important problem in Text Mining, Information Retrieval and Natural Language Processing. Keyword assignment vs. extraction Keyword assignment methods can be roughly divided into: keyword assignment (keywords are chosen from controlled vocabulary or taxonomy) and keyword extraction (keywords are chosen from words that are explicitly mentioned in original text). Methods for automatic keyword extraction can be: supervised, semi-supervised and unsupervised. Unsupervised methods can be further divided into: simple statistics, linguistics, graph-based and other methods.

Number of Keywords

5



The extracted keywords are...

['keyword extraction', 'keyword assignment', 'terms', 'methods', 'chosen']