CSD402 - Internet and Web Systems

Blog - Buddy
Project Report

Problem Definition

My application called ***'Blog-Buddy'*** assists budding bloggers by automatically generating tags for their articles. It will also generate an insightful word cloud for the articles to give the bloggers an idea about the word distribution in their article. This visual representation

of keyword metadata can help the writers to get some insights which could help them to improve their choice of words used in the blog draft.

Goals

1. Allow users to upload its blog draft.
2. Apply text analysis on the uploaded blog draft using a text- analysis algorithm.
3. Generate a word cloud of the most prominent words in the blog draft.
4. Suggests list of suitable tags for the blog article.

Motivation

Having recently started with blogging on Medium, I was facing a problem in coming up with appropriate tags for my article. Moreover, I found myself using some words too frequently unintentionally. And so I thought if I could automate the process of tagging, it could make the lives of bloggers way more convenient. Plus, generating a word cloud could help to minimize unnecessarily high usage of some words. This was the driving force behind my idea to come up with a tool that could assist bloggers in improving the quality of their articles. This is especially important because in today's time with thousands of content - creators putting up their content online, to ensure that your content stands out, you need to produce content of the best quality possible.

Tools and Technology Used

1. Flask
2. Python
3. HTML 5
4. CSS
5. Heroku - cloud platform (PaaS)

Details

I. Frontend

The frontend includes a webpage that supports upload of the text file containing the blog draft to be analyzed. Once the webpage is submitted,the next webpage presents results to the users in the form of - word cloud, suggested tags and text with the tags highlighted. The frontend is implemented using HTML 5 and CSS.

II. Backend

The server side of the website is implemented using the framework Flask. The Backend language used is Python. The algorithm at the backend makes use of the yake API to analyze text and produce suitable tags for it. [1], [2] YAKE! is a light-weight unsupervised automatic keyword extraction method which rests on text statistical features extracted from single documents to select the most important keywords of a text. It does not need to be trained on a particular set of documents, neither it depends on dictionaries, external-corpus, size of the text, language or domain.

Novelty

The project is first of its kind assisting tools for bloggers to improve the quality of their articles. Keyword Extraction is one of the most actively researched areas of NLP finding its use cases in various fields. There are various techniques available in text mining for keyword and keyphrase extraction. Keywords and keyphrases are very useful in analyzing large amounts of textual material quickly and efficiently searching over the internet besides being useful for many other purposes. Keywords and keyphrases are a set of representative words of a document that give high-level specification of the content for interested readers. They are used highly in the field of Computer Science especially in Information Retrieval and Natural Language Processing and can be used for index generation, query refinement, text summarization, etc. This project introduces a novel use

case of keyword extraction. Here, keyword extraction algorithm is used for **auto - tagging of articles**.

Project Analysis

I could achieve all the goals I had in mind at the start of the project. This includes developing a decent frontend and backend which provides features like upload of text file, analysis of the text using keyword extraction algorithm , and displaying the results in the form of word cloud and suggested tags. In fact, additionally, I was able to deploy my application using the Heroku cloud platform.

However, while working on this project, there could be a number of improvements that could be made to this tool to enhance its usability for users. Some of the possible improvements I would like to implement in future :

- Improve the keyword extraction algorithm such that words that are not even present explicitly in the text.
- Allow various types of file formats to be supported by the upload feature.
- Modify the text analysis algorithm such that it generates tags that describe the article the best as well as are least similar to each other.
- Currently, using n- gram analysis, the algorithm considers tags that are unigrams and bigrams. A possible improvement could be to allow the user to specify the range of length of keyphrases he/she desires.
- Improve the frontend of the website to make it more interactive and user friendly.

Moreover, there are a few more functionalities which I would like to my tool in future so that it could become a complete assist tool for bloggers:

- Image search for the tags to suggest most suitable images to accompany the blog article. The images should have Usage rights of Creative Commons License.

- Analyze the tone and sentiment of the article like formal/ informal, optimistic/ pessimistic, friendly/ accusatory, positive/ negative/ neutral, etc. This could provide additional insights to the author so that he/ she can improve the content.

Conclusion

Working on this project was a great learning experience. It helped me to get a greater understanding of frontend and backend frameworks. I also learned a lot about cloud platforms and the nitty gritty of hosting a site over the Internet.

I am really glad that I can actually use this tool to help me in writing better articles for my blog and I am hopeful that this tool could potentially help a lot of other budding bloggers like me in honing their writing skills. I am keen on working on this project even after the course ends so that I can possibly improve the existing functionality as well as add more useful features that would make the life of bloggers even more convenient.

Github Repository : https://github.com/manikamittal2898/Blog-Buddy.git

Website URL : https://blog-buddy.herokuapp.com/

References

[1] https://github.com/LIAAD/yake.git

[2] Campos, Ricardo et al. “YAKE! Collection-Independent Automatic Keyword Extractor.” *ECIR* (2018).