**Project 1: Bugzilla Search/Crawl Tool and MH Utility Tool**

**Editor: - Aniruddh Chauhan**

**8/30/2023**

* What is Bugzilla? Bugzilla is**a web-based general-purpose bug-tracking system and testing tool.**
* **I believe TMEIC got the free firmware and developed it on top of that.**
* **Currently, the project scope looks more like streamlining how the tool accesses the database of all the bugs ever reported.**
* **The database itself is hosted at a third-party company, if needed we could get them to dump all the data to us.**
* **From what I have understood, the previous interns were able to use an API to get the data from the database. I’m not sure what API they used since I haven’t received the codebase and documentation yet. They also got the data about the common bugs related to a specific project to plot some graphical charts that you could send to the respective project managers via Outlook.**

**Relevant Links**

* [3.3. Windows — Bugzilla 5.0.4 documentation (tmeic.com)](https://tools.tmeic.com/mh/docs/en/html/installing/windows.html)

**8/31/2023**

* We went for Lab tours under Niranjan Patel. We visited the Photovoltaic Test Lab at Cook Drive. It was truly amazing, and I learned a lot of amazing things. Each company has to get a UL Certification. TMEIC’s PV Test Lab here at Roanoke is a lab designated to test all TMEIC Drives to ensure they abide by the UL Certification. They also test the effectiveness of their drives.
* PV Grid Electric Drives usually get a lot of DC current from the solar panels; therefore, you need to convert that into AC to send it to the Grid. There is a whole level of IGBTS in either a 2 or 3-way topology which ‘chops’ the DC current and the noisy signals are filtered out. This is basically an Inverter.
* TMEIC’s Development team makes all the controller boards and the firmware for it in-house. They design and build their own custom PCBs and write their own software. These controller boards are throughout the company.
* The Lab in the basement has a test bench for every crane TMEIC has installed at a shipyard or port. This allows them to program, debug, and make software changes for the PLCs on board these massive cranes. They also maintain the old PLCs from before and interface them with the more modern systems. They are specialists in this field.

**09/01/2023**

* Notify the user the search is running and when the search is finished…success or not.
* The Project Components need to be put in a **drop-down menu** so that the user can select/deselect what they like rather than all the components being selected.
* Meet Sujit next Tuesday.

**09/05/2023**

* Obtained Project Code Base for the MH Utility Tool. The project is based on C#.
* Reading Documentation and Comments to understand the MH Project Utility Tool
* So far have a basic understanding of how it works on a High Level
* Need to figure out the scope of the Search/Crawl Tool for Bugzilla
* Bugzilla uses REST API: - **Representational State Transfer (REST)** is an architectural style that defines a set of constraints to be used for creating web services. **REST API is a way of accessing web services** in a simple and flexible way without having any processing.
* The **requests module allows you to send HTTP requests** using Python.
* The HTTP request returns a Response Object with all the response data (content, encoding, status, etc
* WebClient is a class in C# that **provides a simple way to make HTTP requests to web services or website**s. It's part of the System.Net namespace. You can use it to download data from a URL, upload data to a server, or perform various other HTTP operations. Here's a basic overview of how to use Web Client
* A screenshot of a white box

  Description automatically generated

**Relevant Links**

* [C# Access Strings (w3schools.com)](https://www.w3schools.com/cs/cs_strings_access.php)
* [WebClient Class (System.Net) | Microsoft Learn](https://learn.microsoft.com/en-us/dotnet/api/system.net.webclient?view=net-7.0)
* [How to create a drop down menu in WinForms and C# - Stack Overflow](https://stackoverflow.com/questions/27723668/how-to-create-a-drop-down-menu-in-winforms-and-c-sharp)
* [javascript - How can I extract certain attributes from a JSON file - Stack Overflow](https://stackoverflow.com/questions/55252338/how-can-i-extract-certain-attributes-from-a-json-file)
* [Working with JSON - Learn web development | MDN (mozilla.org)](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON)
* [Python Requests Tutorial - GeeksforGeeks](https://www.geeksforgeeks.org/python-requests-tutorial/)
* [REST API Introduction - GeeksforGeeks](https://www.geeksforgeeks.org/rest-api-introduction/)

**09/06/2023**

* String objects are immutable: they can't be changed after they've been created. All of the String methods and C# operators that appear to modify a string actually return the results in a new string object.
* Have a Dynamic Array that increases its size whenever a new object is selected and decreases the size whenever the object is deselected on the drop-down menu. Use a for loop to iterate through this Dynamic Array and send it to the Bugzilla website using the protocols that are already in place. Will need some string manipulation.

A computer screen shot of a program

Description automatically generated

1. **[STAThread]**: This attribute indicates that the application should run in a single-threaded apartment (STA) mode. In most Windows Forms applications, this attribute is necessary because the Windows Forms framework requires an STA thread to work properly.
2. **static void Main()**: This is the entry point of your application. When you run your program, execution begins here.
3. Application.EnableVisualStyles(): This method enables visual styles for your Windows Forms application. It ensures that your application's controls and user interface elements use the native Windows theme and appearance.
4. **Application.SetCompatibleTextRenderingDefault(false):** This method sets the default text rendering to be compatible with GDI (Graphics Device Interface). It's typically set to false because using GDI+ text rendering provides better quality text rendering.
5. **Application.Run(new MH\_tool())**: This line starts your Windows Forms application by creating an instance of your main form, which is named MH\_tool in this case. The Application.Run method starts the application message loop, which is responsible for handling user input and managing the graphical user interface.

**09/07/2023**

* Generating new SSH keys for setting up authentication for github can be tricky:- 1) Generate new Key $ ssh-keygen -t ed25519 -C "aniruddh.chauhan.2002@gmail.com" in Git Bash. 2) cd into .ssh/ directory and use notepad to view the id\_ed25519.pub, it has the ssh key that needs to be added to your Github account.
* Setting up an existing folder as a git repository, make a new repo on Github.com, get the ssh key authentication link:- 1) cd into the directory for that existing folder 2) git init 3) git add -A 4) git commit -m “initial commt” 5) git branch -m main 6) git remote add origin [git@github.com:sammy/my-new-project.git](mailto:git@github.com:sammy/my-new-project.git) 7) git push -u -f origin main
* make sure you have the identiy set using git global config
* For example, if your [repository was configured on Github](https://devconnected.com/how-to-setup-ssh-keys-on-github/), you would use the following command to change your remote.
* $ git remote set-url origin git@github.com:user/repository.git
* Winforms on C# sharp is pretty cool. It has the option to have the winform pre generated..basically boilree plate code. And then you can add whatever you want..for example IF YOU HAVE A WINFORM called Form. It will generate a Form.cs, Form.Designer.cs and Program.cs. 1) Form.cs is actually where you have your event handlers and basically the code that dictates what the app does. 2) Form.Designer.cs is where the winform GUI is actually generated 3) Program.cs is the main entry of the program.
* A screenshot of a computer program

  Description automatically generated

**Relevant links**

* [Generating a new SSH key and adding it to the ssh-agent - GitHub Docs](https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent)
* [Error: Permission denied (publickey) - GitHub Docs](https://docs.github.com/en/authentication/troubleshooting-ssh/error-permission-denied-publickey)
* [github - git: error: src refspec main does not match any - Stack Overflow](https://stackoverflow.com/questions/70179719/git-error-src-refspec-main-does-not-match-any)
* [How to Push an Existing Project to GitHu b | DigitalOcean](https://www.digitalocean.com/community/tutorials/how-to-push-an-existing-project-to-github)
* [Git - First-Time Git Setup (git-scm.com)](https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup)
* [How To Change Git Remote Origin – devconnected](https://devconnected.com/how-to-change-git-remote-origin/)

**09/08/2023**

* Talked to Matt about Bugzilla. I got an idea for an Advanced search query script for the advanced search page on Bugzilla for TMEIC. I could basically overlap C# with the database management using python. So python for the database management and C# sharp for the Desktop management EXAMPLE:

Creating a simple GUI application in C# that uses Python as the database requires a few steps. You can use the **subprocess** module in C# to interact with Python scripts that handle database operations. Here's a basic example of how to create such an application:

1. **Create the C# GUI Application**:

First, create a C# Windows Forms Application using Visual Studio or another C# development environment. Design your user interface as per your requirements.

1. **Add a Button for Database Interaction**:

In your C# GUI, add a button or any control that will trigger the database interaction with Python.

1. **Write a Python Script for Database Operations**:

Create a Python script that performs the database operations you need. This script should accept input from your C# application, execute the necessary database queries, and return the results.

* Made a script for filtering keywords. Chat gpt gave a good overview of it. I’m modify that to suit the API linkage that I already have
* Made an example drop-down menu. This menu is the one that I would try to integrate on the main source code.
* Could potentially present both ideas to Sujit and Tim. Their feedback on this would give me better clarity about the project since it’s really open-ended.
* Set a meeting with all the stakeholders to get clearer definition of the project and general consensus.

A computer screen shot of a code

Description automatically generatedImportant function for stripping common words.

**Relevant Links**

* [Read JSON file using Python - GeeksforGeeks](https://www.geeksforgeeks.org/read-json-file-using-python/)
* [How can I print specific values from a json file with python? - Stack Overflow](https://stackoverflow.com/questions/36970016/how-can-i-print-specific-values-from-a-json-file-with-python)
* [python - How to fix ''UnicodeDecodeError: 'charmap' codec can't decode byte 0x9d in position 29815: character maps to <undefined>''? - Stack Overflow](https://stackoverflow.com/questions/49562499/how-to-fix-unicodedecodeerror-charmap-codec-cant-decode-byte-0x9d-in-posit)
* [Create a new app with Visual Studio tutorial - Windows Forms .NET | Microsoft Learn](https://learn.microsoft.com/en-us/dotnet/desktop/winforms/get-started/create-app-visual-studio?view=netdesktop-7.0)

**09/11/2023**

* Meeting with Sujit, Tim, Mario to define scope of Project for the Search Crawl Tool: - Meeting Notes
* I have a database of technical bugs that come up across the company's software products and I am accessing this using an API in Python. I need to optimize how the information is presented to the User based on the number of times a particular bug has appeared. What do you think I should I next?
* For technical services, search by product…a set of solutions and rank them, first get them approved…come with a likely problem with a prior solution , mine the database for useful info -Tim
* What’s the most hurting points in the Bugzilla i.e., the most bug-some issues…e.g. Crane PLC, Maxview. Overall, group the bugs by components.

**Master Plan of Action: -**

1. **Get the JSON Output using the API.**
2. **Extract the component attributes e.g., Crane PLC, Maxview etc.**
3. **Run a search crawl on the data of bugs of these individual attributes. For e.g., Crane PLC has a recurring issue with Motor 9. Group all bugs related to Motor 9 and present data based on priority.**
4. **Use an intelligent tool i.e., a program to automate these processes.**

* **Short Term Goal: Extract component attributes and group relevant bugs together.**

A computer screen shot of a program code

Description automatically generated **Basic Component wise fetch**

**Relevant Links**

* [Python program for most frequent word in Strings List - GeeksforGeeks](https://www.geeksforgeeks.org/python-program-for-most-frequent-word-in-strings-list/)
* [Python String split() - GeeksforGeeks](https://www.geeksforgeeks.org/python-string-split/)
* [Python - max() function - GeeksforGeeks](https://www.geeksforgeeks.org/python-max-function/)
* [Defaultdict in Python - GeeksforGeeks](https://www.geeksforgeeks.org/defaultdict-in-python/)
* [how to remove a word in a string python - Search (bing.com)](https://www.bing.com/search?pglt=41&q=how+to+remove+a+word+in+a+string+python&cvid=1ebb798706ce46168d9d7b3408de90fa&aqs=edge.2.69i57j0l8j69i11004.21899j0j1&FORM=ANAB01&PC=U531)

9/12/2023 – Leave

**9/13/2023**

* Filtering out keywords efficiently and cleanly is turning out ot be alittle challenging. Looking for Python libraries that do that otherwise will have to expand my own library.
* Sort of hit a mental block. Not sure how to approach this. Gonna do some brainstorming
* I think I found a really relevant tool the nltk lib. It has stopwords and I can modify it if I can access the app data.
* I underestimated the filtering. It’s good amount of work refining it
* How am I actually going to group them. The json comes in
* A screenshot of a computer program

  Description automatically generated

**Relevant Links**

* [NLP | Filtering Insignificant Words - GeeksforGeeks](https://www.geeksforgeeks.org/nlp-filtering-insignificant-words/)
* [f-strings in Python - GeeksforGeeks](https://www.geeksforgeeks.org/formatted-string-literals-f-strings-python/)
* [Removing stop words with NLTK in Python - GeeksforGeeks](https://www.geeksforgeeks.org/removing-stop-words-nltk-python/)
* [Filter common words from documents (datasciencebytes.com)](https://www.datasciencebytes.com/bytes/2014/11/04/filter-common-words-from-documents/)
* [python - How to import and use stopwords list from NLTK? - Stack Overflow](https://stackoverflow.com/questions/70698947/how-to-import-and-use-stopwords-list-from-nltk)
* [Python NLTK | nltk.WhitespaceTokenizer - GeeksforGeeks](https://www.geeksforgeeks.org/python-nltk-nltk-whitespacetokenizer/)

9/14/2023

* Made a little GUI that supports user input to get the component wise data.
* Need to think of how to make the component wise thing to give the data once the user inputs a keyword.
* SHOULD I TRY SOMETHING CALLED SIMILARITY METRICS

1. **Collect Bug Data**: First, you need to have a database of bug data that you want to group. Ensure that each bug has relevant information such as a description, summary, or keywords.
2. **User Input**: Collect the user's search input, which could be a search query or keywords related to the bug they are interested in.
3. **Define a Similarity Metric**: To group similar bugs, you need a similarity metric or distance measure. Common measures include cosine similarity, Jaccard similarity, or edit distance, depending on the nature of your bug data.
4. **Calculate Similarity**: Calculate the similarity between the user's search input and each bug in your database using the chosen similarity metric. The bugs with higher similarity scores are more similar to the user's input.
5. **Threshold or Clustering Algorithm**: Decide on a similarity threshold or use clustering algorithms like k-means, hierarchical clustering, or DBSCAN to group bugs. For example, you might decide that bugs with a similarity score above a certain threshold should be considered similar.
6. **Grouping Bugs**: Cluster bugs based on calculated similarities. Each cluster will represent a group of similar bugs. You can use libraries like scikit-learn or NLTK for clustering algorithms.
7. **Present Results**: Present the grouped bugs to the user, showing them which bugs are similar to their query. You can display the bugs in each cluster and provide summaries or details.
8. **User Interaction**: Allow users to further refine their search or explore bugs within a cluster. This can include filtering by various attributes, sorting, or displaying additional bug details.
9. **Feedback and Refinement**: Collect user feedback on the results and continuously refine your clustering and similarity calculation techniques to improve the accuracy of bug grouping.
10. **Performance Optimization**: Depending on the size of your bug database, you may need to optimize the performance of the clustering and similarity calculation, such as using appropriate data structures or caching results.

* -ChatGPT. Provides some useful insights into this. Probably will change approach.
* Bug no, nested and nested
* Underscore is a PLC signal , any period within a word probably means that’s a function under PLC signals…make sub categories of after each keyword traversal…this streamlines it
* Trolley and gantry
* Nested lops is the way to go, is what I got from my meeting with Mario. I basically have to streamline the search variables. My initial approach in the tablet is basically what I gotta do. Find keywords and get a subcategory with those keywords, fin keywords in that category and streamline it even further.
* PLCs are cool…go to Global Academy to check out the course on PLCs…PLCs basically control relays, contactors and coils…they have even complex functions to.

**Relevant Links**

* [Python List index() - GeeksforGeeks](https://www.geeksforgeeks.org/python-list-index/)
* [Python GUI - tkinter - GeeksforGeeks](https://www.geeksforgeeks.org/python-gui-tkinter/)
* [Tkinter Progressbar Widget (pythontutorial.net)](https://www.pythontutorial.net/tkinter/tkinter-progressbar/)
* [python - How can I create a dropdown menu from a List in Tkinter? - Stack Overflow](https://stackoverflow.com/questions/45441885/how-can-i-create-a-dropdown-menu-from-a-list-in-tkinter)

**9/15/2023**

* Text similarity is a really useful natural language processing (NLP) tool. It allows you to find similar pieces of text and has many real-world use cases. I am going to use this for finding similarity with texts within the comments of the components.
* Gotta save the bug ids somewhere
* **Roadmap for Filtering keywords and grouping similar issues together.**
* A diagram of a computer program

  Description automatically generatedA diagram with many different colored boxes

  Description automatically generated with medium confidence

**Relevant Links**

* <https://lucid.app/lucidchart/e5952023-27fa-4c4e-a981-5be07e9542b5/edit?view_items=94.u2zemGyXF&invitationId=inv_ed8e9209-b73f-4bd9-a70c-ad729990a8c3>
* [An introduction to Machine Learning - GeeksforGeeks](https://www.geeksforgeeks.org/introduction-machine-learning/)

**9/18/2023**

* Not sure why all the words are not being filtered even though they are in the list of words to be filtered out.

### 1. Cosine similarity

This measures the similarity between two texts based on the angle between their word vectors. It is often used with [term frequency-inverse document frequency (TF-IDF)](https://spotintelligence.com/2022/11/28/tf-idf/) vectors, representing each word’s importance in a document.

Cosine similarity measures the similarity between two non-zero vectors of an inner product space. In the context of document similarity, it is often used to measure the similarity between two documents represented as vectors of word frequencies. The cosine similarity between two vectors is calculated as the cosine of the angle between them.

To compute the cosine similarity between two documents, first, a vector representation of each document is constructed, where each dimension of the vector corresponds to a word in the document, and the value of the dimension represents the frequency of that word in the document. The vectors are then normalized to have a unit length. The cosine similarity between the two documents is then calculated as the dot product of the two vectors divided by the product of their lengths.

The resulting cosine similarity value ranges from -1 to 1, where -1 indicates completely dissimilar documents, and 1 indicates identical documents. A value of 0 indicates that the two documents are orthogonal and have no similarity.

Cosine similarity is widely used in natural language processing and information retrieval, particularly in document clustering, classification, and recommendation systems.

### 2. Levenshtein distance

Levenshtein distance, or edit distance, measures the difference between two strings. It is the minimum number of single-character insertions, deletions, or substitutions required to transform one string into another.

For example, the Levenshtein distance between “kitten” and “sitting” is 3, since three single-character edits are required to transform “kitten” into “sitting”: substitute “s” for “k”, substitute “i” for “e”, and insert “g” at the end.

Levenshtein distance is used in various applications such as spell-checking, string matching, and DNA analysis.

### ****3. Jaccard index****

The Jaccard index, or the Jaccard similarity coefficient, measures the similarity between two sets. It is defined as the ratio of the size of the intersection of the sets to the size of the union of the sets. In other words, it is the proportion of common elements between two sets.

The Jaccard index is particularly useful when the presence or absence of elements in the sets is more important than their frequency or order. For example, it can be used to compare the similarity of two documents by considering the sets of words that appear in each document.

The Jaccard index is calculated as follows:

J(A,B) = |A ∩ B| / |A ∪ B|

where A and B are sets, and |A| and |B| represent the cardinality or size of the sets.

The resulting value of the Jaccard index ranges from 0 to 1, where 0 indicates no common elements between the sets, and 1 indicates that the sets are identical.

The Jaccard index is widely used in various applications such as information retrieval, data mining, and pattern recognition. It is particularly useful when dealing with sparse or high-dimensional data, where the presence or absence of features is more important than their actual values.

### ****4. Euclidean distance****

Euclidean distance is a measure of the distance between two points in a Euclidean space. It is calculated as the square root of the sum of the squares of the differences between the corresponding coordinates of the two points.

For example, the Euclidean distance between two points (x1, y1) and (x2, y2) in a two-dimensional space is given by:

euclidean distance = sqrt((x2 - x1)^2 + (y2 - y1)^2)

The Euclidean distance can be extended to spaces of any dimension. It is commonly used in machine learning and data analysis to measure the similarity between two vectors in a high-dimensional space.

In the context of document similarity, the Euclidean distance can be used to compare the frequency of words in two documents represented as vectors of word frequencies. In this case, the Euclidean distance between the two vectors is calculated as the square root of the sum of the squared differences between the corresponding frequency values in the two vectors.

The resulting value of Euclidean distance ranges from 0 to infinity, where 0 indicates identical vectors and larger values indicate greater dissimilarity between the vectors.

Euclidean distance is widely used in various applications such as clustering, classification, and anomaly detection. It is particularly useful when dealing with continuous variables or data that can be represented as vectors in a high-dimensional space.

### ****5. Hamming distance****

Hamming distance measures the difference between two strings of equal length. It is defined as the number of positions at which the corresponding symbols differ. In other words, it is the minimum number of single-character substitutions required to transform one string into another of equal length.

For example, the Hamming distance between “101010” and “111011” is 2, since two positions differ between the two strings: the second and fifth.

Hamming distance is used in various applications such as error-correcting codes, coding theory, and cryptography. It can also be used to compare the similarity of binary strings, such as DNA sequences.

In computer science, Hamming distance is often used as a metric to measure the quality of codes. For example, in error-correcting codes, the minimum Hamming distance between codewords determines the number of errors that can be corrected by the code. Codes with a larger minimum Hamming distance are more robust to errors.

The Hamming distance can be calculated using a simple algorithm that compares the symbols at each position in the two strings and counts the number of positions where they differ.

### 6. Word embeddings

[Word embeddings](https://spotintelligence.com/2022/11/30/word-embedding/) are distributed representations of words in a natural language. They represent words as vectors of real numbers, where each vector dimension represents a different feature or aspect of the word’s meaning. Word embeddings are often fundamental in many natural language processing tasks, such as machine translation, text classification, and information retrieval.

Word embeddings are typically learned from large corpora of text data using neural network models, such as the famous Word2Vec model or GloVe. These models map words to a high-dimensional space where semantically similar words are mapped to nearby points. The learned embeddings capture both the syntactic and semantic relationships between words and can capture complex analogies and relationships between words.

Word embeddings have several advantages over traditional methods for representing words in natural language processing, such as [one-hot encoding](https://spotintelligence.com/2023/01/12/one-hot-encoding/) or [continuous bag-of-words](https://spotintelligence.com/2023/07/27/continuous-bag-of-words/) representations. For example:

* They are dense, meaning they are more space-efficient than sparse representations like one-hot encoding.
* They can capture semantic relationships between words that cannot be easily captured by traditional methods.
* They can infer relationships between words or generate new representations of words not seen in the training data.

Many pre-trained word embeddings are available, which can be used for various NLP tasks. Additionally, custom word embeddings can be trained on specific domains or datasets to improve performance on specific tasks.

### 7. Pre-trained language models

Pre-trained language models are powerful tools for text similarity tasks, as they can learn high-quality representations of text that capture both semantic and syntactic information. Here are some of the most widely used pre-trained language models for text similarity tasks:

1. **BERT (Bidirectional Encoder Representations from Transformers)**: BERT is a transformer-based pre-trained language model widely used for various natural language processing tasks, including text similarity. It has been shown to outperform previous state-of-the-art methods on several benchmark datasets.
2. **RoBERTa (Robustly Optimized BERT Pretraining Approach)**: RoBERTa is a variant of BERT that is pre-trained using additional data and training strategies. It has achieved state-of-the-art performance on several text similarity benchmarks.
3. **DistilBERT**: DistilBERT is a smaller and faster version of BERT trained using a knowledge distillation technique. It has achieved competitive performance on several text similarity benchmarks much faster than BERT.
4. **USE (Universal Sentence Encoder)**: USE is a pre-trained model developed by Google that can encode sentences into fixed-length vectors. It can be used for text similarity tasks by computing the cosine similarity between the sentence embeddings.
5. **ALBERT (A Lite BERT)**: ALBERT is a variant of BERT that reduces the number of parameters and improves training efficiency while maintaining comparable performance.

These pre-trained language models can be fine-tuned on specific text similarity tasks using transfer learning, which involves training the model on a smaller dataset of labelled examples. [Fine-tuning](https://spotintelligence.com/2023/04/21/fine-tuning-gpt-3/) can further improve the performance of these models on specific tasks.

**Relevant Links**

* [Top 7 Ways To Implement Text Similarity In Python (spotintelligence.com)](https://spotintelligence.com/2022/12/19/text-similarity-python/)
* [scikit-learn: machine learning in Python — scikit-learn 1.3.0 documentation](https://scikit-learn.org/stable/index.html)
* [TF-IDF Explained And Easy Examples To Get Started (spotintelligence.com)](https://spotintelligence.com/2022/11/28/tf-idf/)