**Project for Propulsion Academy**

**Project Title** : Relevapp : A tool to tag documents and find similar words and sections in other documents

**Student Contact** : [severin.kappeler@gmail.com](mailto:severin.kappeler@gmail.com); patrickrs@gmail.com

**Propulsion Academy Contact** : nitink@propulsionacademy.com

**Project Type** : 3.5 Weeks Bootcamp Project

**Project Date** : 30.03.2020 - 24.04.2020

**Important** : This project will have two parts one will work over documents (pdfs, docs, etc.) another over weblinks (not so much tagging functionality there). For project template of part 2 look towards the end of this document.

**Joint Project between Data Science and Full-Stack**

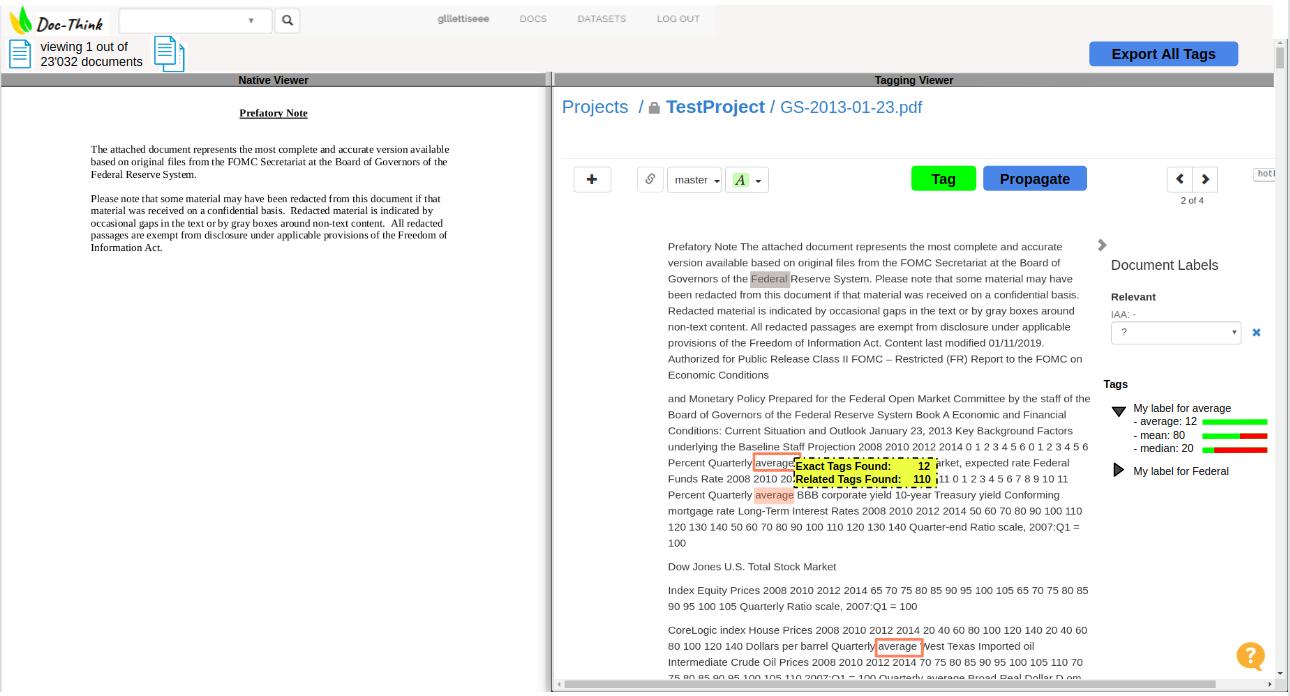
**Project Description (Part 1)**:

It is estimated that 80-90% of the data in corporations is unstructured in the form of electronic documents (PDFs, Wordfiles, Emails, weblinks, etc.), with rising trends. In order to best process/analyze the information, the data needs to be structured and labelled. The structuring and finding of similar information across multiple documents is still to a great extent done manually in most corporations.

This project will focus on developing a tool enabling a user to efficiently tag and label information present in documents. Furthermore, similar words and contexts to these tags can then be searched within the entire universe of uploaded documents. *Similar words* are defined as words which show a resembling context structure, not only dictionary synonyms. This allows the user to quickly propagate the tagged information to a big set of documents and access the tagged data in a structured manner.

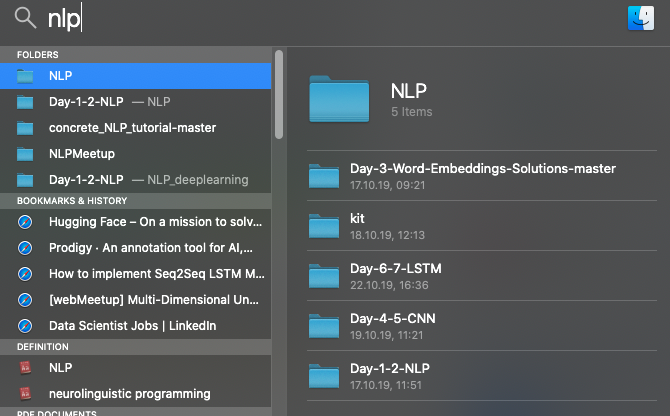
This functionality could be useful to, for example, lawyers reviewing contracts, fraud analysts reviewing emails, or academics researching papers.

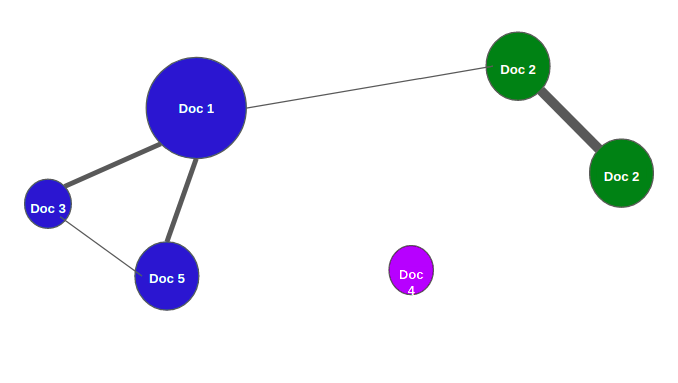
### **Design Mock**



User Interface/Feature Description

1. The user is able to drag-drop, upload from folder or upload documents through a weblink to the platform.
2. The main interface consists of a Native Viewer showing the original document and a Tagging Viewer showing the raw text of the file.
3. The user is able to select parts of the text and can store and label it by hitting the "Tag" button.
4. The tagged text, the specified label, the location in the text and the document ID gets stored in database in the backend.
5. Hitting the "Propagate" displays a small dropdown letting the user specify to which population the tagged terms need to be propagated to. Then this scope is searched for the tagged texts (exact matches) as well as every term that is, concerning the context of the term, related/similar to the tagged term. These are then also saved in the database. The context similarity will be obtained using machine learning models and so when a user selects find similar tags, the machine learning model (supplied by data science) should be executed to find similar terms. There should be a possibility to quickly scan all documents and tags associated with those documents.





1. If the user finds that a section identified as related/similar is irrelevant, he can untag that section and the system will learn based on this feedback.
2. On the right hand-side the user can see all tag labels present in this document. By expanding the arrow, the originally tagged term as well as all the similar terms are displayed together with the count of occurrences. Moreover, a bar is representing the similarity of the term to the original term.
3. Hovering over the similar term allows jumping directly to the document where it appeared.
4. Hovering over a term shows the count of exact matches in the universe and the similar terms.
5. The "Export All Terms" button allows the user to export the entire tag table from the backend into a desired format (Excel, csv)
6. Another interface should enable the user to compare all terms from the same tag label and see the context of this term. This to verify whether the model picked a truly similar term.

Features probably needed:

1. A simple user interface, that works on cloud
2. A system that somehow directly lets a user push a weblink (the weblink may as well contain a video) to this platform
3. user should be able to view the file
4. A search functionality that lets the user search inside documents
5. A tagging functionality
6. A foder/document management system
7. A small note taking section (if possible)
8. An AI/Machine Learning system that lets user find similar terms in other documents from tagged keywords

**Project Goals** : Get a functional tool for tagging and structuring PDF data efficiently.

### **Milestones**

#### **Milestones 1 :**

*Fullstack:* Design user interface including upload, tagging and exporting functionality with fully connected backend.

develop machine learning system to identify similar terms (and later sections) within document.

DataScience:

1. ***31/03****, Tue: Decide on a dataset, download and process it (see list on the bottom)*
2. ***01/04****, Wed: Develop word2vec model for word tagging. This will serve as a baseline against which future models will be compared.*
3. ***03/04****, Fri: Develop BERT model for word tagging and compare to baseline*
4. ***06/04****, Mon: Eventually refine model with customized word embeddings and (RNN ?).*

#### **Milestones 2 :**

*FullStack*: Implement Propagate functionality as well as similarity scores.

*DataScience:* Integrate Milestone 1 system into the software.

Details:

1. ***06/04, Mo****: Create functional workflow of Milestone 1 (hit button & run)*
2. ***07-08/04, Tue/We****: Implement feedback procedure when user rejects/accepts a suggested related term.*
3. ***09/04, Thu****: Adjust workflow to the user input format as handed over by FS. Our workflow needs to be able to handel all different kind of userinputs such as:*
   1. *user tags*
   2. *Handle click on "Propagate" (apply current model to find similar terms)*
   3. *Handle click on "Rerun model" (adjust model to the feedback from the user wheter a suggestion has been accepted or rejected)*
4. ***10/04, Fr****:Adjust workflow to output format required by FS to hand back to the tool. This includes:*
   1. *newly found related tags (in structured format)*
   2. *similarity of related tag to original one*

#### **Milestones 3 :**

*FullStack + DataScience:* enable classification functionality for Document labels. The document label needs to be dependent on a text section.

*Details Data Science:*

1. *Implement data preparation for Machine Learning training*
   1. *From Milestone 2:*

*Features: Document(ID), text, tags, context of tags, similarities*

* 1. *From user input:*

*Labels: desired categories*

* 1. *Alternative: unsupervised clustering of documents + possibility for User to ‘label’ them?*

1. *Implement Machine Learning (NN, RF…?)*
   1. *Try different options*
   2. *Optimize*
2. *(Save / store / download model)*
3. *(Use model for novel documents)*

#### **Bonus:**

*DataScience:* Find system to cluster text into chunks of context

### 

### **Possible data sources:**

**Legal cases reports:** <https://archive.ics.uci.edu/ml/datasets/Legal+Case+Reports>

**Value proposition**: Lawyer consulting previous decisions in order to get acquainted with case-law. Swiss case: <https://www.bger.ch/ext/eurospider/live/de/php/clir/http/index.php?type=start&lang=de>

**Literary works**: <http://www.gutenberg.org/wiki/Gutenberg:Offline_Catalogs>

**Value proposition**: academics researching aspects of literary style.

[Ten Thousand German News Articles Dataset](https://tblock.github.io/10kGNAD/): 10273 german language news articles categorized into nine classes for topic classification. <https://github.com/tblock/10kGNAD>

**Value proposition:**

**References** :

<https://link.medium.com/kuybn5E0E4>

<https://www.tagtog.net/> (Baseline)

<https://turicode.com>

<https://parashift.io>

[**https://prodi.gy/**](https://prodi.gy/)

**Project Description (Part 2)** : This project is a form of mini-google. A tool where weblinks could be easily bookmarked (or copied too) and searched at any given time directly (paid version with offline reading). This search has to happen over document text. Here is a list of key features needed

1. A simple user interface, that works on cloud
2. A system that somehow directly lets a user push a weblink (the weblink may as well contain a video) to this platform
3. User should be able to view the file
4. A search functionality that lets the user search inside those weblinks
5. A tagging functionality
6. A small note taking section (if possible) and a article recommendation engine (may be)

Example - google.com, <https://news.ycombinator.com/>

### **Questions**:

1. What about different languages? Even within same doc?