Eigen Technologies task

Giussepi Lopez

<2018-09-21 Fri>

1 Assumptions

- We are extracting common words per file and not overall common words, because a common word for a file may not be common or may not be present in other file.
- By default we are extracting the 10 most common words per file.
- This app currently support files with extension: txt, docx and no entension at all.
- When trying to upload a PDF file a NotImplementedError will be raised. This means that we are considering that format but we have not implemented its support yet.
- When trying to upload any other file format a ValueError will be raised.

2 Installation

2.1 Open a shell and open the folder Giussepi Lopez

```
cd Giussepi_Lopez
```

2.2 Create a virtual environment

```
virtualenv -p python3 env
```

2.3 Activate virtual environment

```
source env/bin/activate
```

2.4 Install requirements

```
pip install -r requirements.txt
```

2.5 Download required nltk data

```
python -m nltk.downloader -d env/share/nltk_data stopwords punkt gutenberg genesis
python -m nltk.downloader -d env/share/nltk_data inaugural nps_chat webtext treebank
```

2.6 Go to eigentest project

```
cd eigentest
```

2.7 Create the database

```
./manage.py migrate
```

2.8 Create a django admin super user

```
./manage.py createsuperuser
```

2.9 Run Django project

```
./manage.py runserver 0.0.0.0:8082
```

2.10 Open the admin and add text files

1. Open your browser and go to `http://127.0.0.1:8082/admin/`
2. Login with the super user credentials created
3. Click on "Documents" and add some text files. Some sample files are on the folder "test docs".

2.11 Go to the homepage and navigate the results

Open a new tab on your browser and go to `http://127.0.0.1:8082/`

3 Configuration

3.1 NUMBER OF COMMON WORDS PER FILE

1. Open your configuration file ('Giussepi_Lopez/eigentest/eigentest/settings.py')

2. Go to the last line and change the number of common words to be extracted per file:

   ```
   COMMON_WORDS_NUMBER = <new number>
   ```

3. If this change is done after having added some files through the admin run the following management command to update the hashtags and their relations:

   ```
   ./manage.py update_hashtags_relations
   ```

   To do this fist stop the Django development, to do so just press CONTROL - C. Then you can execute the management command (as shown in the above line) and run again the Django development server:

   ```
   ./manage.py runserver 0.0.0.0:8082
   ```

4 Extending to other document types and sources

4.1 Add support for more document types

Just add code to handle the new document type properly on the following methods:

- apps/documents_analizer/models.Document.get_full_text_from_source
- apps/documents_analizer/models.Document.get_lines_from_source

4.2 Add new text source

For instance, if you want to be able to read from a URLs then add an attribute to store the url in the class apps/documents_analizer/models.Document:

```
url = models.URLField(blank=True)
```

You also must to make it non required (blank=True) and also modify the doc_file attribute to be non required too; furthermore, you must add a validation to have at least one of those attributes.

The rest is modifying the two methods mentioned in the previous point to consider the new field, and deciding if storing the url content or making an http request all the time (the former is recommended).