Language Teaching in Drama Plays

What is this? A tool for searching grammar topics in Shakespeare drama plays, especially designed to be used by teachers.

It is the result of the project for the course Text Technology from the Master's Program in Computational Linguistics (University of Stuttgart, Germany).

The course was conducted by Prof. Kerstin Jung. The project was developed by Felipe Nobrega.

Table of Contents

Goal

Provide easy access to grammar topics that are present on William Shakespeare drama plays and may be useful for language teachers. Therefore, the project can be used to facilitate the usage of literature in the classroom and shorten class preparation time.

How?

- Website that allows queries for specific grammar topics within the corpus of drama plays.
- Provide a list with results of the query and the context of the drama play where it is found.

This project is based in three steps according to the project requirements of the course. Below we can find an overview of the project along with the required steps.

Step	Description
Collect	1. Collect data from Shakespeare plays. DraCor Corpora was used. The files are found in `.xml`.

Step	Description
Prepare	1. Perform data extraction from `.xml` using XPath in XML Copy Editor. 2. Parse the extracted data in simple text to be processed by PostgreSQL in order to populate tables. 3. Create a database to store information about the plays and the grammar topics. 4. Extract main text from play's `.xml` files using Xpath. Add the root <text> in each file. 5. Convert the `.xml` into `.html` in order to display the context in a web page, using `<span>` to identify the lines.
Access	1. Create a Python app to connect to PostgreSQL to get information and perform queries within the plays database. 2. Display the results in a web page - a table with the line, section, desired word and context from the play.

Log

Version 1.1

- Working version with famous Shakespeare plays: Romeo and Juliet, Hamlet, Macbeth, King Lear and Othello.
- Database for modal verbs is completed for all the plays.
- Webpages are working. Strong improvements in the user interface.

Version 1.0

- Working version with Romeo and Juliet play only.
- Database is set with modal verbs tables.
- All web pages are working as intended. However, pages look raw.

Tools

The following tools were used in order to develop the project:

1. XMLCopy Editor
2. PostgreSQL
3. Python
4. Visual Studio Code

Python libraries:

- ElementTree: for XML handling
- flask: framework for web development
- psycopg2: PostgreSQL database adapter

Configuration

Requirements:

- All files contained in this project. Please preserve the file structure.
- SQL Server. I use PostgreSQL running a local server in my device.
- Any program that can open text files. You need it to open the `.py` script and modify the connection details to your desired database. I use Visual Code Studio, but Notepad, Brackets etc work fine.
- Python installed in order to run `.py` scripts.
- Web browser.

In order to get the project working, first you need to set up the SQL database:

1. In root folder, locate `database_lang_drama_plays_back.sql`.
2. In your preferred SQL database environment, load this file in order to create the schema containing all tables that will be used.
3. Make sure your database is running, otherwise the script will not be able to connect.

Now you need to run the webapp script to create a connection to your database. However, you need to modify the script before:

4. Open the `webapp.py` code in a program that can read and modify the script.
5. Locate the `def get_db_connection():` function (line 6).
6. Modify the `dbname`, `user`, `password`, `host` and `port` fields to match your database connection information.
7. Save the file.

Run the script in a Python terminal to get it running.

8. Open a Python terminal (or any program that supports it, such as Visual Studio Code).
9. Run `webapp.py`.
10. If the connection has succeeded, the terminal will display `Running on http://127.0.0.1:5000` and some other information.
11. Copy the `http` link being displayed and paste it in your web browser. Alternatively, you can `ctrl + click` in the link if your terminal supports it.

Navigate through the website.

12. After you access the link provided by the terminal, you will see the index page.
13. Further instructions to use the interface are given in the webpage. However, it is very straightforward, as follows:
14. Select the desired play.
15. Select the desired modal verb.
16. Click on `search` button.
17. A list will be shown with all the modal verbs and their respective sections and lines.
18. If you want to locate where the modal verb is, click in `View in Context` button.

Roadmap

In future versions we plan to implement:

- Better **coverage of grammar topics**, such as prepositions, conjunctions, verb tenses, for current plays (Romeo and Juliet, Hamlet, King Lear, Macbeth and Othello).
- Display the surrounding words from the play in the *context* column in the results page.
- Addition of all Shakespeare plays in the project.
- Include other famous drama plays from other authors.

Structure

Below we provide a brief description of the file structure of the project and import files.

`Root folder`

- Script files for tasks suck as parsing and converting the data. The `webapp.py` is the main file to connect to the database and fetch the results in webpages.

- Database back with the schemas and tables for plays and grammar topics - Text-based files related to the description and report of the project.

`images`

- Contains images used in the report.

`modal_verbs`

- `.xml` files with the extracted modal verbs for each play.

- `.txt` files with parsed text to populate the database tables.

`plays`

- `.xml` files downloaded from DraCor corpora.

- Other `.xml` files with only the main text from the play (later used in the `html` conversion script)

`text`

- The `.html` files containing the whole text from the plays.

`template`

- `.html` files related to the webpages Index and Results.

Contact

Felipe Nobrega. felipeaugnobrega@gmail.com / github: github.com/felipenob