**🔍**

D cLens

Search Engine for docx files

Rajkumar Selvam (2017HT12630): Information Retrieval Assignment

**Environment**

* Python 3.x
* Windows OS

**Usage Instructions (Executable)**

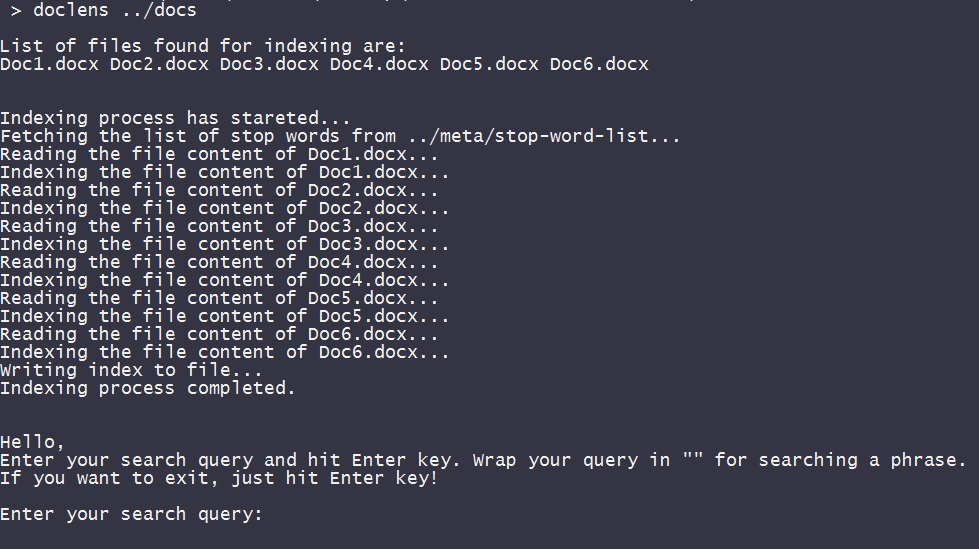
1. Open the Command Prompt, navigate to **doclens/bin** directory and execute the executable using the following command. You can also double-click the doclens.exe file directly to run the program.

**doclens.exe**

1. By default, the program uses document collection provided for assignment. To provide new document collection, provide the path of the directory which contains documents as command line argument

**doclens.exe /doc-collection**

1. Once the initial loading and indexing is completed, the program will prompt to enter your search query.



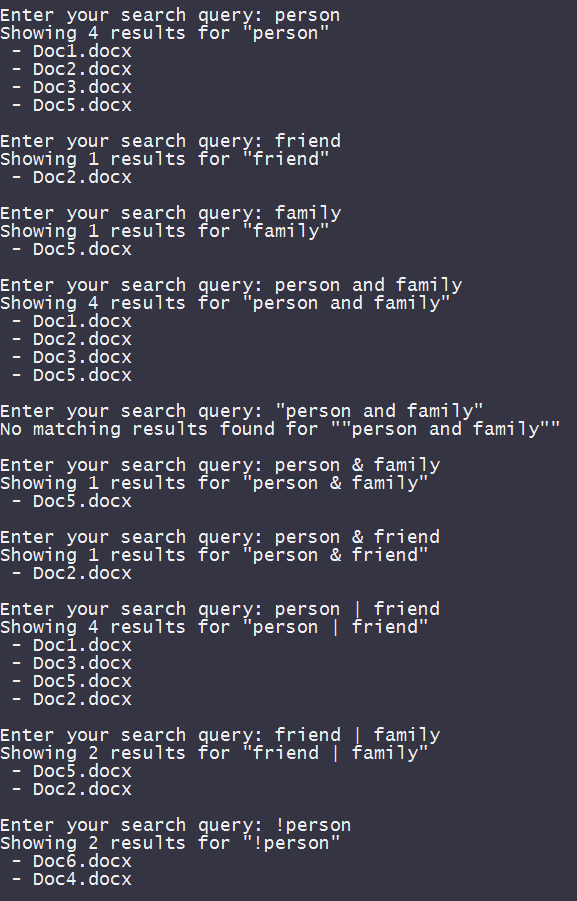
1. Enter your search query and hit Enter key. Top 10 results will be displayed as shown in the screenshot below. Wrap your query in "" (double-quotes) for searching a phrase. If you want to exit, hit Enter key without providing any search query.
2. Supported query types

* One Word Queries
* Free Text Queries
* Phrase Queries
* Boolean Conjunctive Queries

Supported Boolean Operators

* & for Logical AND
* | for Logical OR
* ! for Logical NOT

1. Below screenshot shows all possible queries



**Usage Instructions (Source Code)**

1. Open the Command Prompt, navigate to **doclens/src** directory and execute the source code using the following command.

**python main.py**

1. By default, the program uses document collection provided for assignment. To provide new document collection, provide the path of the directory which contains documents as command line argument

**python main.py /doc-collection**

1. Rest of the steps remains as same as in **Usage Instructions (Executable)**

**Generated Index**

* Once the initial loading and indexing is completed, the index will be stored in **doclens/meta/index** file.
* Total 628 dictionary terms are generated in the indexing process

**How application works**

* First, the application loads and parses all the available docx files in the given document collection (directory).
* Once all the document is parsed, indexing process is initiated.
* Stop words from **doclens\meta\stop-word-list** file is used for avoiding all the stop words from the document collection from being indexed.
* After filtering the stop words, **stemming** is done on all the tokens.
* After stemming process, document frequency and term position are calculated and updated in the **posting list** for each dictionary term.
* Query string also goes through the similar process.
* Logical operators **& | !** are specially handled for query string.

**Assumptions**

* Document contains only plain text (No image, table, etc.,)
* Text is only in **English** Language

**Limitations**

* Only **.docx** files are supported
* If document contains multimedia, they are ignored
* Search results are not sorted in alphabetical order
* Grouping of Boolean query is not possible currently.

Example:

**! (A & B)** is not possible

* Supports only Windows platform. For other platforms, directly execute the source code using python

**Libraries/References Used**

* Porter Stemmer - For stemming the index tokens/terms
* List of Stop Words: <http://xpo6.com/download-stop-word-list/>

**Build Instructions**

For compiling and creating an executable build, follow the below steps:

* Install **pyinstaller** using the following command in Command Prompt

**pip install pyinstaller**

* Clone the repository from <https://github.com/raajkumar-s/doclens.git>
* cd to **doclens** directory
* Execute the **build.bat** file to create the executable
* If build is successful, the executable will be available in **doclens/bin**