1. **Context Term Detection Tool from Abstract/Full-Length Articles**

The context term detection tool detects important context terms such as protein, site, disease, species, tissue and cell line from both abstracts and full-length articles **(Sections such as Title, Abstract, Methods, Results, Discussion, Conclusion)** based on the pubtator annotations and dictionaries. Protein, disease, species, cellline are detected using pubtator3 whereas tissues are detected using a comprehensive tissue dictionary named “Brenda”. We used an in-house sitedetector tool to detect sites from the text. For dictionary-based detection, we use OGER. Pubtator3 API is used first to collect all the biocxml files for a list of pmids from where texts are parsed for dictionary-based context term detection, site detection and final output creation. The outputs are created as an Excel spreadsheet and an HTML file highlighting the context terms with multiple colors. The description of the output files is given below:

**Output File for Abstracts:**

**i) Excel Sheet**

The output file is generated after running a pmid.txt file in a pipeline to detect context terms from the abstracts and is available in the “Outputs\_Abstract” directory named “1\_new\_dis\_spec\_tis\_cel\_output.xlsx” [N.B: Please ignore the name of the file as it was given while the development was done]. Some intermediate additional files generated while going through the pipeline can also be found under the same directory and ignored.

The column descriptions are given below:

pmid: This column has all the pmids from where the entities are detected.

entity: The term that matches any of the utilized dictionaries.

entity\_type: The specific type of the entity whether it is a protein or site or disease or species or cell line or tissue.

start\_offset: start offset of the entity term in the sentence from where it is detected.

end\_offset: end offset of the entity term in the sentence from where it is detected.

source: From which dictionary the entity was detected. For example, if any tissue is detected from Brenda, it will be mentioned here.

id: Each of the entities detected has an id (e.g, MESH id for diseases, Cellosaurus id form celllines, gene id for proteins, etc.).

doid\_for\_diseases: This is only applicable for the diseases where we mapped the MESH IDs with the disease ids by mesh to doid mapping technique.

uniprot\_ac: This column provides the corresponding UniProt accession id of the genes detected by pubtator by a gene to uniprot mapping technique.

sent\_index: The sentence number of the corresponding sentence is provided here. Note here that, this index number is based on each section. Because various modules in this tool use different techniques to tokenize and split sentences, a -1 index could be found. You will find the sentence of the corresponding index at the top of all the sentences found from a section in the HTML file. This could match partially with some other sentences. This will be tried to solve in the next version.

sentence: The sentence from where the entity was detected.

**ii) HTML File**

The detected terms are also highlighted in the text with multiple colors. Legends are available at the beginning of the HTML file after the heading to let the users know about the terms and the corresponding color code. The terms are detected in each sentence. The titles of each abstract is given as the pmid number. For example, **PMID: 18454162** means the sentences in this abstract are coming from the pmid 18454162. The same sent\_index from the Excel sheet will be found at the beginning of each sentence. The IDs will NOT be available here for each entity as this is only for visualization purposes. The HTML file can be found in the “Output\_Abstract” directory named “highlighted\_entities.html”.

**Output File for Full-Length Articles:**

**i) Excel Sheet**

The output file is generated after running a pmid.txt file in a pipeline to detect context terms from the full-length articles and is available in the “Outputs\_FL” directory named “14\_fl\_dis\_spec\_tis\_cel\_output.xlsx” [N.B: Please ignore the name of the file as it was given while the development was done]. Some intermediate additional files generated while going through the pipeline can be also found under the same directory and can be ignored. However, two additional items are available of interest in the “Outputs\_FL” directory.

i) missing\_full\_length\_pmids.txt: This file has a collection of pmids that do not have open access to full-length articles.

ii) pmids\_pmcids.xlsx: mapping of the pmid to pmcid (Ignore this file as it is available in the output sheet)

The column descriptions are given below:

pmid: This column has all the pmids from where the entities are detected.

pmcid: This column has the corresponding pmcid of the pmids.

entity: The term that matches any of the utilized dictionaries.

entity\_type: The specific type of the entity whether it is a protein or site or disease or species or cell line or tissue.

start\_offset: start offset of the entity term in the sentence from where it is detected.

end\_offset: end offset of the entity term in the sentence from where it is detected.

[N.B: You can ignore these two columns as they are sometimes incorrect because of parsing and integrating different modules together]

source: From which dictionary the entity was detected. For example, if any tissue is detected from Brenda, it will be mentioned here.

id: Each of the entities detected has an id (e.g, MESH id for diseases, Cellosaurus id for celllines, gene id for proteins, etc.).

doid\_for\_diseases: This is only applicable for the diseases where we mapped the MESH IDs with the disease ids by mesh to doid mapping technique.

uniprot\_ac: This column provides the corresponding UniProt accession id of the genes detected by pubtator by a gene to uniprot mapping technique.

sent\_index: The sentence number of the corresponding sentence is provided here. Note here that, this index number is based on each section. Because various modules in this tool use different techniques to tokenize and split sentences, a -1 index could be found. You will find the sentence of the corresponding index at the top of all the sentences found from a section in the HTML file. This could partially match some other sentences. This will be tried to solve in the next version.

section: This column presents the section where the entity and the sentence were found.

sentence: The sentence from where the entity was detected.

**ii) HTML File**

The detected terms are also highlighted in the text with multiple colors. Legends are available at the beginning of the HTML file after the heading to let the users know about the terms and the corresponding color code. The terms are detected in each section. The sections are given as PMCID\_SECTIONNAME. For example, **PMC2391110\_ABSTRACT** means the sentences in this section are coming from the pmcid PMC2391110 and the section name is ABSTRACT. It could be TITLE, RESULTS, etc, as well. The same sent\_index from the Excel sheet will be found at the beginning of each sentence. The IDs will NOT be available here for each entity, as this is only for visualization purposes. The HTML file can be found in the “Output\_FL” directory named “highlighted\_entities\_FL.html”.

1. **Glycan Structure Terms (GSTs) Detection from Abstract/Full-Length Articles**

The GST tool detects important glycan structure terms from abstracts and full-length articles **(Sections such as Title, Abstract, Methods, Results, Discussion, Conclusion)** based on the ML-based BERT model. The outputs are created as an Excel spreadsheet and an HTML file highlighting the GSTs with blue color. Currently, the ML model has been trained with 400 abstracts. However, there might be missing GSTs or some incorrect predictions that can be addressed by developing a more robust model. The description of the output files is given below:

**Output File for Abstracts:**

**i) Excel Sheet**

The output file is generated after running a pmid.txt file in a pipeline to detect GSTs from the abstracts and is available in the “Outputs” directory named “final\_gst.xlsx” [N.B: Please ignore the name of the file as it was given while the development was done]. Some intermediate additional files generated while going through the pipeline can also be found under the same directory and can be ignored.

The column descriptions are given below:

pmid: This column has all the pmids from where the GSTs are detected.

sent\_index: The sentence number of the corresponding sentence is provided here.

sent\_text: The sentence from where the GSTs were detected.

char\_start: start offset of the sentence.

char\_end: end offset of the sentence.

[N.B.: char\_start and char\_end can be ignored as they came from the sentences as a part of a script.]

BERT\_Output: If any GST term is detected by the ML model, it will be found here. If there are several GST terms, those will be separated by “ | ”.

**ii) HTML File**

The detected GST terms are also highlighted in the text with a blue color in each sentence. The titles of each abstract are given as the pmid number. For example, **PMID: 18454162** means the sentences in this abstract are coming from the pmid 18454162. It is colored in red. The same sent\_index from the Excel sheet will be found at the beginning of each sentence. The HTML file can be found in the “Outputs” directory named “Highlighted\_ML\_Predictions.html”.

**Output File for Full-Lengths:**

**i) Excel Sheet**

The output file is generated after running a pmid.txt file in a pipeline to detect GSTs from the abstracts and is available in the “Outputs” directory named “final\_gst.xlsx” [N.B: Please ignore the name of the file as it was given while the development was done]. Some intermediate additional files generated while going through the pipeline can also be found under the same directory and can be ignored.

The column descriptions are given below:

pmcid: This column has all the pmcids from where the GSTs are detected.

section: This column presents the section where the GSTs and the sentence were found.

paragraph: Each section could have multiple paragraphs. The paragraph number is parsed sequentially from each section and numbers are provided accordingly.

sent\_index: The sentence number of the corresponding sentence from the paragraph following the section following the pmcid is provided here.

sent\_text: The sentence from where the GSTs were detected.

char\_start: start offset of the sentence.

char\_end: end offset of the sentence.

[N.B.: char\_start and char\_end can be ignored as they came from the sentences as a part of a script.]

BERT\_Output: If any GST term is detected by the ML model, it will be found here. If there are several GST terms, those will be separated by “ | ”.

**ii) HTML File**

The detected GST terms are also highlighted in the text with a blue color in each sentence. The titles of a set of sentences are coming from the pmcid, section and paragraph. For example, **PMC2391110\_DISCUSS\_Paragraph1** means the sentences in this article are coming from the pmcid 2391110’s discussion section’s paragraph no. 1. It is colored in red. The same sent\_index from the Excel sheet will be found at the beginning of each sentence for each set of sentences. With the set of sentences, we meant the paragraphs. However, if there is only one paragraph, in that case, there will be no paragraph number. Rather, the PMCID followed by the Section name will be found. The HTML file can be found in the “Outputs” directory named “Highlighted\_ML\_Predictions\_Full.html”.

1. **One docker for Context Term + GST:**

**Instruction for pulling and running the docker container:**

*Software requirements:* Docker (v18.09.6 or newer)

To pull and run the docker image, follow the steps:

***Step 1:*** Run the bash script run\_docker.sh by “./run\_docker.sh”. Make sure the bash script is executable by running the command “chmod +x run\_docker.sh”

***Step 2:*** When the bash script runs, a prompt will appear to know what type of article you want to process. There are five options: abstract/full-length/both/gst\_abstract/gst\_full. You have to make sure that you are typing the exact name without any spelling mistakes. After that, another prompt will appear to know whether you want a new pmid file or not. If yes, type “yes”, hit enter/return and then provide the path of the pmid file (e.g., /home/shovan/context\_term\_docker/pmid.txt). A sample pmid file (pmid.txt) is provided for testing. This will create the necessary directories, run the docker and copy essential file from the dockerhub to the local repository for the new/existing pmid files. If not, type “no” which will pull the files including the mentioned folders for the previous set of abstracts that were run. We recommend giving the new pmid file for testing. Note that, for GST, you have to write either “abstract” or “full” followed by gst. The rest three options are for the context term tool. For the context term, if you select abstract, only the abstract-related directories and scripts will be available. This will be the same for the full-length. However, if you select both, then all the files for the abstract-based pipeline and full-length-based pipeline will be pulled to the local repository. Since the docker uses several APIs to collect entities from external sources such as Pubtator or Uniprot, it will take some time to finish the running of the whole docker. For example, if you select both and give 5 pmids, based on the size of the full-length articles and some gene mentions, it will take up to 30-45 minutes to complete the process. Therefore, it is recommended to run keeping the time constraint in mind. For the GST, it will not take time like the context term tool, however, it will still require around 15 minutes for 5 pmids.

**Docker container Files:**

The description of the docker container items is given below. Please note that we are mentioning either the folder/item for the abstract version or the full-length version in the brackets.If the choice is to get both tool outputs, then all the items will be found in the local repository.

abstract\_xml (Abstract + GST\_Abstract): Extracted xml files from pubtator3 for the given pmids.

abstract\_text (Abstract + GST\_Abstract): Folder to store the text files for the given pmids.

Outputs\_Abstract (Abstract): The output spreadsheet of the tool, HTML file, intermediate outputs, etc are available in this directory.

Scripts\_Abstract (Abstract): All the scripts used in the development of the pipeline are found here.

bioc\_xml\_fl (FL + GST\_Full): Extracted full-length xml files from pubtator3 for the given pmids

output\_text\_section\_FL (FL + GST\_Full): Folder to store the text files (for each section) for the given pmids. These files are generated from the full\_length xml files.

output\_xml\_section\_FL (FL + GST\_Full): This folder contains the section-wise xml files by breaking the original xml files.

Outputs\_FL (FL): The output spreadsheet of the tool, HTML file, intermediate outputs, missing OA article lists, a pmid\_to\_pmcid mapping file, etc are available in this directory.

Scripts\_FL (FL): All the scripts used in the development of the pipeline are found here.

Dictionary (Abstract + FL): Holds one obo file (Brenda: for tissue)

mesh\_to\_doid (Abstract + FL): Mapping files and scripts to map mesh ids to doids for diseases

OGER (Abstract + FL): A tool to detect entities based on dictionary terms (Specially used to detect tissues).

obo\_to\_tsv\_tissue.py (Abstract + FL): This script can be used to update the OGER dictionary if a new/updated obo file is available for Brenda.

Outputs (GST\_Abstract/GST\_Full): The output spreadsheet of the GST tool, HTML file, intermediate outputs, etc are available in this directory.

Scripts (GST\_Abstract/GST\_Full): All the scripts used in the development of the GST pipeline are found here.

Saved\_Model (GST\_Abstract): Model weights, tokenizer, dictionary, etc. of the ML model are available here.

requirements.txt (Abstract + FL + GST\_Abstract + GST\_Full): All the library requirements can be found here.