

CMPE 492

TEMPHI

Text Mined Pathogen Host Interactions

Sinan Kerem Gündüz

Yavuz Samet Topçuoğlu

Advisor:

Arzucan Özgür

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1. Broad Impact	1
1.2. Ethical Considerations	2
2. PROJECT DEFINITION AND PLANNING	4
2.1. Project Definition	4
2.2. Project Planning	5
2.2.1. Project Time and Resource Estimation	5
2.2.2. Success Criteria	6
2.2.3. Risk Analysis	7
2.2.4. Team Work	8
3. RELATED WORK	10
4. METHODOLOGY	13
5. REQUIREMENTS SPECIFICATION	16
5.1. Functional Requirements	16
5.1.1. User Requirements	16
5.1.2. System Requirements	16
5.2. Non-functional Requirements	17
5.2.1. Performance	17
5.2.2. Security and Privacy	17
5.2.3. Localization and Accessibility	17
6. DESIGN	18
6.1. Information Structure	18
6.2. Information Flow	19
6.3. User Interface Design	20
7. IMPLEMENTATION AND TESTING	22
7.1. Implementation	22
7.1.1. Creating the PHISTO Dataset	22
7.1.2. PubMed Extraction	22
7.1.3. PubTator Conversion	23

7.1.4.	Named Entity Recognition	23
7.1.5.	UniProt ID Fetching	24
7.1.6.	Relation Extraction	24
7.1.7.	Backend	26
7.1.8.	Data Generation	26
7.1.9.	Frontend	29
7.2.	Deployment	30
8.	RESULTS	32
8.1.	Model Comparison	32
8.2.	Computational Efficiency	34
8.2.1.	Training Efficiency	34
8.2.2.	Testing Efficiency	35
8.2.3.	Significance and Impact	35
8.3.	Error Analysis	36
8.3.1.	Annotated Data Quality	36
8.3.2.	Named Entity Recognition (NER) Model Concerns	36
8.3.3.	Multiple ID Existence for Proteins	36
8.3.4.	Error Analysis Insights and Future Directions	37
9.	CONCLUSION	39
	REFERENCES	41

1. INTRODUCTION

1.1. Broad Impact

The development of an advanced and automated text mining pipeline to extract human-pathogen protein interactions from PubMed articles will have a huge impact on various domains and initiatives. This comprehensive database will serve as a valuable resource for researchers, scientists, educators, and students in molecular biology, biochemistry, infectious diseases, medicine, and molecular genetics. By providing a wealth of up-to-date information on protein-protein interactions, the database will accelerate the pace of discovery.

Industrial applications may include pharmaceutical companies leveraging the database for drug discovery and biotechnology firms using the information to develop new diagnostic tools. It facilitates the identification of potential therapeutic targets and accelerates research. It could indirectly benefit these industries. As an example, in the context of the COVID-19 pandemic, the project could have helped gather data from numerous scientific papers published within a short period, expediting the development of treatments and vaccines.

There is potential for educational applications, even though the primary focus of this project is on scientific research. The database could be integrated into university courses or training programs in medicine and other relevant fields. This resource could enrich the learning experience for students and professionals alike, enabling them to better understand the complexities of human-pathogen interactions.

The database's impact on global health could be significant, particularly in understanding, preventing, and controlling infectious diseases. It provides insights into the relationships between human and pathogen proteins and therefore it can inform the development of novel treatments and preventive measures against various diseases. Identifying common factors across different pathogens and their interactions with human proteins

could lead to more efficient diagnoses and targeted interventions.

All in all, this project has the potential to contribute significantly to the advancement of scientific research in various domains, enhance industrial applications and global health efforts. By developing a comprehensive and continually updated human-pathogen protein interaction database, the project will facilitate interdisciplinary collaboration, accelerate the discovery process, and ultimately improve public health outcomes.

1.2. Ethical Considerations

The development of a comprehensive database for human-pathogen protein interactions raises several ethical considerations that must be addressed to ensure the responsible use of this resource. These concerns primarily revolve around data privacy, data quality, potential biases, and the equitable distribution of benefits arising from the project.

Data Privacy: While the data used in this project is primarily derived from publicly available sources such as PubMed, it is crucial to ensure that no sensitive information is unintentionally included in the database. Appropriate measures must be taken to anonymize any potentially identifying information before incorporating it into the database.

Data Quality: The accuracy and reliability of the extracted protein interactions are paramount for ensuring the ethical use of the database. To maintain high data quality, the project team should regularly evaluate and validate the performance of the named entity recognition and relation extraction tools. Additionally, incorporating a user feedback mechanism can help identify and correct potential errors in the data.

Potential Biases: As the project relies on machine learning models, there may be inherent biases present in the training data or algorithms. To minimize the impact of these biases, the project team should carefully examine the data sources and algorithms used to ensure that a diverse and representative set of articles is processed, and that the models employed are as unbiased as possible.

Equitable Distribution of Benefits: Ensuring that the benefits of the database are accessible to a broad range of users, including researchers, scientists, educators, and students from different countries and socioeconomic backgrounds, is essential. Efforts should be made to provide open access to the database and create user-friendly interfaces that cater to diverse user needs.

By addressing these ethical considerations, the project team can create a responsible and valuable resource that accelerates scientific research, fosters interdisciplinary collaboration, and ultimately improves public health outcomes.

2. PROJECT DEFINITION AND PLANNING

2.1. Project Definition

The objective of this project is to develop an advanced and automated pipeline that systematically fetches abstracts and titles from PubMed articles related to specific MeSH terms, employs state-of-the-art text mining techniques to extract human and pathogen protein-protein interactions, and compiles the extracted relations in a comprehensive, easily accessible database. The pipeline will be designed to consistently update the database with new articles, ensuring the information remains current, relevant, and valuable to the scientific community.

With a target to process the majority of PubMed articles relevant to the chosen MeSH terms, this project aims to surpass existing databases' limitations by offering a more extensive resource. The project is expected to be completed by the end of the term, but it will be designed for continuous improvements and upgrades. Collaborating with a Master's student specializing in Named Entity Recognition, the pipeline will make use of his expertise to achieve the desired output. The front-end website will be accessible to all interested users, including academic researchers and medical professionals.

To evaluate the pipeline, approximately 60,000 labeled data points from the PHISTO database, derived from around 2,350 articles, will be used as a test dataset. User feedback mechanisms will be added to enhance the models' confidence in labeling relations and extracting relevant information, allowing users to provide insights on abstracts and their corresponding relations or proteins.

The project's scope may expand in the future to include additional types of proteins or genes, as well as other molecular interactions. The primary source of articles will continue to be PubMed, ensuring reliability and ease of use. The database will not only accelerate academic research processes but also serve as a valuable resource for doctors and other professionals in related fields.

By developing a comprehensive and constantly updated protein - protein interaction database, this project aims to foster interdisciplinary collaboration and contribute significantly to the advancement of scientific research in molecular biology, biochemistry, and infectious diseases. The pipeline will be designed with a focus on scalability and adaptability, allowing for future expansions, improvements, and integration with additional data sources and molecular interactions.

2.2. Project Planning

2.2.1. Project Time and Resource Estimation

Team Composition: Our project team consists of two members, with roles and responsibilities evenly distributed. Both team members contribute to coding and research. In addition, there are two external collaborators who provide valuable resources in the form of Named Entity Recognition and Relation Extraction tools.

Project Milestones: We may say there are four main milestones. The first one is completing the integration of the named entity recognition tool into our pipeline. The following milestone is the integration of the relation extraction tool into the project. The third one is the implementation of frontend web design and data generation. And last but not least, the backend and database management of our project is also completed.

Resource Requirements: The primary resource requirement for this project is data storage, as the goal is to generate an extensive database by processing publicly available articles from PubMed. No excessive computational power is anticipated to be needed. However, it would be better to have a GPU-supporting machine for the relation extraction part.

Time Allocation: The two team members have been working on a regular basis and holding weekly meetings with the external collaborators and their advisor. Tasks have been fairly divided between the team members, ensuring that they are completed in a timely manner and do not cause any bottlenecks in the project's progress.

Contingency Planning: The project is dependent on external tools for entity recognition and relation extraction. While no major contingencies are expected, there is a possibility that these tools may not perform up to expectations. In such cases, the team will need to consider alternative solutions to ensure the project's success.

2.2.2. Success Criteria

Performance Metrics: The project's success will be measured based on the accuracy of the combined Named Entity Recognition and Relation Extraction tools when tested against the generated database. As there are no similar systems available for comparison, the team aims to achieve a high level of confidence in their test results. Additionally, the response time for the frontend web interface should be within seconds, ensuring efficient information retrieval and database handling for users.

Benchmark Comparisons: Due to the absence of comparable projects or systems that utilize machine learning models for pathogen-host interactions, we cannot make direct performance comparisons. However, we aim to surpass existing search tools in terms of data coverage by leveraging our machine learning-based approach and the pipeline.

User Feedback: The ratio of positive to negative user feedback will be considered as a success criterion. This will help the team understand the effectiveness of the project and identify areas for improvement.

Scalability and Maintainability: The project's scalability will be demonstrated by creating automated jobs that regularly scan the PubMed database for new articles and process them through the pipeline. This approach will ensure that the database stays up-to-date and continues to grow over time.

Impact on Research and Industry: While it is currently not possible to measure the project's impact on academic research and industry, the team aims to create a tool that can potentially benefit multiple fields, including biology, biochemistry, infectious diseases, medicine, and molecular genetics.

By clearly defining the success criteria, the team can focus on achieving their goals and addressing potential challenges throughout the project. This section will also help stakeholders understand what to expect from the project and gauge its overall effectiveness upon completion.

2.2.3. Risk Analysis

Model Performance: The performance of the Named Entity Recognition and Relation Extraction models is currently uncertain, as the pipeline has not been completed and tested. Suboptimal performance could affect the quality of the project's output. However, it is important to note that the overall project milestones can still be completed regardless of the models' accuracy scores. This includes designing database schemas, the frontend, and the pipeline. If necessary, improvements to the models' accuracy could be addressed in future iterations of the project.

Data Quality: Data quality is not a significant concern for the input data, as PubMed's API enables direct access to article titles and abstracts. As long as the pipeline operates correctly and accurately, incorporating new data (i.e., abstracts and titles) should not pose any issues.

Integration Challenges: The main integration challenge involves aligning the input and output formats of the Named Entity Recognition and Relation Extraction models. For instance, the entity recognition tool outputs entities in a format that cannot be directly used by the relation extraction tool. To address this, the output of the entity recognition tool must be reformatted before being passed to the next model. Additionally, there could be some challenges in displaying the data on the frontend and fetching it from the database via the backend. However, these challenges are expected to be solvable in a reasonable amount of time.

Scalability and Maintainability: The scalability of the project might be a concern, as the structure of the database has not been clearly defined. With a large amount of data, there could be issues related to database requests, such as scanning the entire database

with keywords like protein names. Maintainability requires consistent server maintenance to ensure 24/7 availability of web services. Automatically updating the database with new articles may also present challenges.

Ethical and Legal Concerns: As also mentioned above in the Ethical Consideration section related to copyright issues. However, in the first iteration of the project, only abstracts and titles will be used, which are generally considered publicly accessible. As a result, copyright concerns are expected to be minimal in the initial phase of the project.

2.2.4. Team Work

Our team, consisting of two members, Samet and Kerem, has worked effectively together throughout the project. We divided tasks and responsibilities to capitalize on each member’s skills and expertise. Kerem was responsible for generating the test dataset from the PHISTO website, while Samet focused on retrieving titles and abstracts related to PubMed IDs and converting them to the PubTator format. Other tasks, such as research and investigation, were shared by both members.

To maintain effective communication and collaboration, our team regularly held online meetings to discuss project issues and progress. We used Git and GitHub for code management, which allowed us to efficiently work together on code development.

Our team experienced minimal conflicts during the project, and any disagreements were resolved through open communication and collaborative problem-solving.

One of the key strengths of our team was our strong communication skills, which facilitated smooth collaboration and effective task division. However, our team’s lack of experience in natural language processing (NLP) posed a challenge. To overcome this, we ensured that each team member shared their knowledge and research findings with the other. For example, Kerem researched online databases and shared his findings with Samet, while Samet learned about the Named Entity Recognition tool and explained it in detail to Kerem.

Our team's ability to communicate effectively, share knowledge, and divide tasks appropriately has contributed to the successful execution of our project. Despite our limited experience in NLP, we have worked together to build a strong foundation for our project's ongoing development and improvement.

Following the first milestone, we continued with utilising the Named Entity Recognition, and at that point, we get help from Berke Kavak. Thanks to him, we did not focus too much on the NER part of the project.

The next step in the pipeline, which is to extract relations are handled by Samet as well as data generation to display on the frontend. Kerem gathered the tools and developed the backend and the frontend of the project.

In conclusion, the work is divided even between the members, and at every step of the project both members helped out each other.

3. RELATED WORK

The study of pathogen-host interactions (PHIs) has become increasingly important for understanding infection mechanisms and for the development of new strategies to prevent or treat infectious diseases. In recent years, several databases have been developed to facilitate access to PHI data and provide tools for the analysis of these interactions. In this section, we will review five prominent databases: PHISTO, APID, DIP, MINT, and IntAct, discussing their primary features, data sources, curation methods, and other notable characteristics.

PHISTO is a bioinformatics web platform that focuses on providing up-to-date PHI data for all pathogen types and experimentally verified protein interactions with hosts [1]. The database covers a wide range of host organisms and includes interactions verified through various experimental methods. PHISTO also offers tools for the visualization and analysis of PHI networks through its user-friendly interface, allowing users to explore the data in a more intuitive manner. For our project, PHISTO’s dataset will serve as the test dataset to evaluate the performance of our machine learning models.

APID (Agile Protein Interactomes DataServer) is a resource that offers a comprehensive collection of protein interactomes for over 400 organisms, integrating known experimentally validated protein-protein physical interactions (PPIs) [2]. The interactomes are constructed using a methodological approach that reports quality levels and coverage over the proteomes for each organism included. APID includes a new data visualization web-tool that allows users to construct and explore sub-interactomes using query lists of proteins of interest and interactive selection of interaction properties and functional annotations.

DIP (Database of Interacting Proteins) catalogs experimentally determined protein-protein interactions, combining information from various sources to create a consistent set of PPIs [3]. The database is useful for understanding protein function and relationships, studying the properties of networks of interacting proteins, and benchmarking predictions

of protein-protein interactions. DIP also contributes to studying the evolution of protein-protein interactions and offers researchers an extensive dataset to investigate protein interaction networks in biological processes.

MINT (Molecular INTeraction Database) is a public repository for experimentally verified protein-protein interactions curated from scientific literature by expert curators [4]. MINT was designed to store data on functional interactions between proteins, including binary complexes and enzymatic modifications of one of the partners. The database emphasizes the importance of well-organized and easily accessible databases to allow for easy retrieval and analysis of large interaction datasets.

IntAct is an open-source, open data molecular interaction database containing evidence for molecular interactions curated from literature or direct data depositions [5]. The IntAct team has developed a web-based curation tool that supports both IMEx- and MIMIx-level curation. Recently, MINT and IntAct databases merged their efforts to optimize resource usage and maximize curation output, which demonstrates the importance of collaboration among databases to ensure efficient use of resources and improve the quality of the curated data.

While these databases provide valuable information on molecular interactions, our project specifically focuses on pathogen-host interactions. By leveraging recent advancements in natural language processing and machine learning, our project incorporates machine learning models, such as Named Entity Recognition and Relation Extraction, to generate a comprehensive database of pathogen-host interactions derived from a wide range of sources, including PubMed articles. This approach aims to enhance the extraction and analysis of PHI data, offering a more extensive understanding of pathogen-host interactions and setting our project apart from existing resources. Our method also allows for continuous updates, ensuring that the database remains current and relevant for ongoing research in the field of pathogen-host interactions.

AIONER [6] is a novel approach to biomedical named entity recognition (BioNER) that addresses the challenges of data scarcity and limited generalizability. By using

external data from existing annotated resources, AIONER enhances the accuracy and stability of BioNER models. The AIONER tool, built on deep learning and the all-in-one (AIO) schema, demonstrates its effectiveness and robustness in 14 BioNER benchmark tasks. It outperforms other state-of-the-art approaches and showcases its practical utility in recognizing entity types not seen in training data, as well as processing biomedical text at a large scale.

BioRED [7] is a unique biomedical relation extraction dataset that goes beyond existing benchmarking datasets. It offers multiple entity types (e.g., gene/protein, disease, chemical) and relation pairs (e.g., gene-disease, chemical-chemical) at the document level, enabling the exploration of complex relationships in biomedicine. The dataset also includes annotations distinguishing between novel findings and previously known background knowledge. BioRED facilitates the benchmarking of existing methods, such as BERT-based models, on both named entity recognition (NER) and relation extraction (RE) tasks. The results highlight the need for improvement in RE, particularly for extracting novel relations. The availability of the BioRED dataset and annotation guidelines further promotes the development of more accurate and robust RE systems in biomedicine.

4. METHODOLOGY

Our system is designed as a comprehensive tool to facilitate the study of protein-protein interactions, specifically focusing on the interactions between pathogens and their human hosts. The architecture of the system is built upon three primary components: the backend, frontend, and database. The backend is responsible for processing user requests and running the machine learning models, while the frontend offers an intuitive user interface for inputting queries and displaying the results. The database plays a crucial role in storing and retrieving the protein interaction relations extracted from the literature.

Data Collection and Preprocessing: To collect relevant information, we utilize the PubMed API to fetch article titles and abstracts from the biomedical literature. Once the data is gathered, a preprocessing step is performed, transforming the collected abstracts into the PubTator format. This format serves as the input for the first machine learning model (Named Entity Recognition).

Named Entity Recognition (NER): The NER model, developed by a master’s student from our university, Berke Kavak, is a pre-trained model specifically designed to identify and recognize human and pathogen proteins within the text. The output from this model comprises line-by-line entries of detected entities, such as proteins and genes.

Relation Extraction: Building upon the output of the NER model, we generate inputs for the second machine learning model (Relation Extraction). This model accepts two parameters: sentence, and its relation label. By identifying all possible combinations of entities within the same sentence, we feed this data into the Relation Extraction model. The model then determines if a relation exists between the given entities within the context of the sentence.

Data Normalization: In order to ensure consistency and accuracy in the results, we perform a normalization process on protein names. This step accounts for the various

naming conventions used to describe specific proteins and allows for a more reliable representation of the extracted information.

Database Management: Our database technology of choice is SQLite, which offers a simple yet effective solution for managing the data schema. The schema consists of 1 table at the moment which is the relations table. The columns of the relation table is also provided in the "information structure" section but it mainly stores the relation between 2 proteins and respectively their ids, sentence of the presence of the proteins and the pubmed_id of the corresponding article. Since the database is not very complex, SQLite was enough at this step.

Frontend and User Interface: The frontend initially will be developed using HTML, CSS, and JavaScript to mainly manage the api requests. However, if the code becomes too complex to manage, we may transition to a React-based framework. The user interface consists of a main page where users can choose to search by keyword or full text (e.g., abstract) and view the results in a user-friendly manner.

Data Update and Scheduling: In order to keep the system up-to-date with the latest research, we are planning to implement a scheduled job that runs on a weekly basis at our backend service as future work. This job will fetch new articles from PubMed using their API, run the entire pipeline on the newly fetched data, and store the extracted relations in the database. This ensures that our system consistently provides the most recent and relevant information.

Performance Evaluation: We used RelX and PPI datasets in order to test the accuracy and reliability of our system. The evaluation metrics include F1-score and accuracy scores, the focus will be mainly on minimizing false positives in the results. We are less concerned about false negatives at first, as the primary goal is to deliver high-confidence results to users. The RelX and PPI datasets will be examined further in the report.

Overall, the methodology ensures that our system is capable of extracting meaning-

ful and accurate relations between pathogens and human proteins from the article texts and offering valuable insights

5. REQUIREMENTS SPECIFICATION

5.1. Functional Requirements

5.1.1. User Requirements

- (i) Users will be able to access information from automatically extracted PubMed articles related to predetermined MeSH terms.
- (ii) Users will be able to search for proteins by UniProt ID, with interactions displayed in a table on a webpage.
- (iii) Users will be able to input free text to view the relations mentioned in the text.
- (iv) Users shall be able to filter search results based on predefined criteria, such as protein type or relation type.
- (v) Users shall have access to a help section with a comprehensive user guide and frequently asked questions.
- (vi) Users shall be able to provide feedback and report issues through a dedicated feedback form or contact information.

5.1.2. System Requirements

- (i) The system shall provide regular updates to incorporate new articles into the database.
- (ii) The system shall convert title and abstract tuples into PubTator documents.
- (iii) The system shall process PubTator documents using Named Entity Recognition and Relation Extraction tools.
- (iv) The system shall update the database with extracted relations.

5.2. Non-functional Requirements

5.2.1. Performance

- (i) Support for up to 50 concurrent users.
- (ii) Response times of less than 60 seconds for most queries.

5.2.2. Security and Privacy

- (i) No user authentication, login, or registration mechanisms required.
- (ii) Publicly accessible website.

5.2.3. Localization and Accessibility

- (i) Development in English.
- (ii) Responsive design to support both desktop and mobile devices.

6. DESIGN

6.1. Information Structure

The primary data source is PubMed, a reputable database of life sciences literature. The pipeline fetches new articles from PubMed and processes them to continuously update its database with the most recent protein-protein interaction data, ensuring the information remains current and valuable for the scientific community. The extracted data is stored in an SQLite database due to its current size and simplicity. It contains a single table, 'Relations', with fields for the unique ID, PubMed ID, protein names and IDs, and the sentence from which the relationship was extracted. This database design allows for efficient data retrieval and ensures ease of use. At present, the project allows users to run the pipeline and search by UniProt IDs. Users can fetch a list of available protein IDs and then select a single protein ID or a pair to retrieve relevant protein-protein interaction data. While the current search functionality is minimalistic, it effectively caters to users' needs for specific protein-protein interactions.

Columns of Relation Table:

- **id:** database id: int
- **pubmed_id:** Pubmed Id: int
- **protein_1:** Protein 1 name: str
- **protein_2:** Protein 2 name: str
- **protein_1_id:** Protein 1 id: str
- **protein_2_id:** Protein 2 id: str
- **sentence:** Sentence that contains the relation: str

6.2. Information Flow

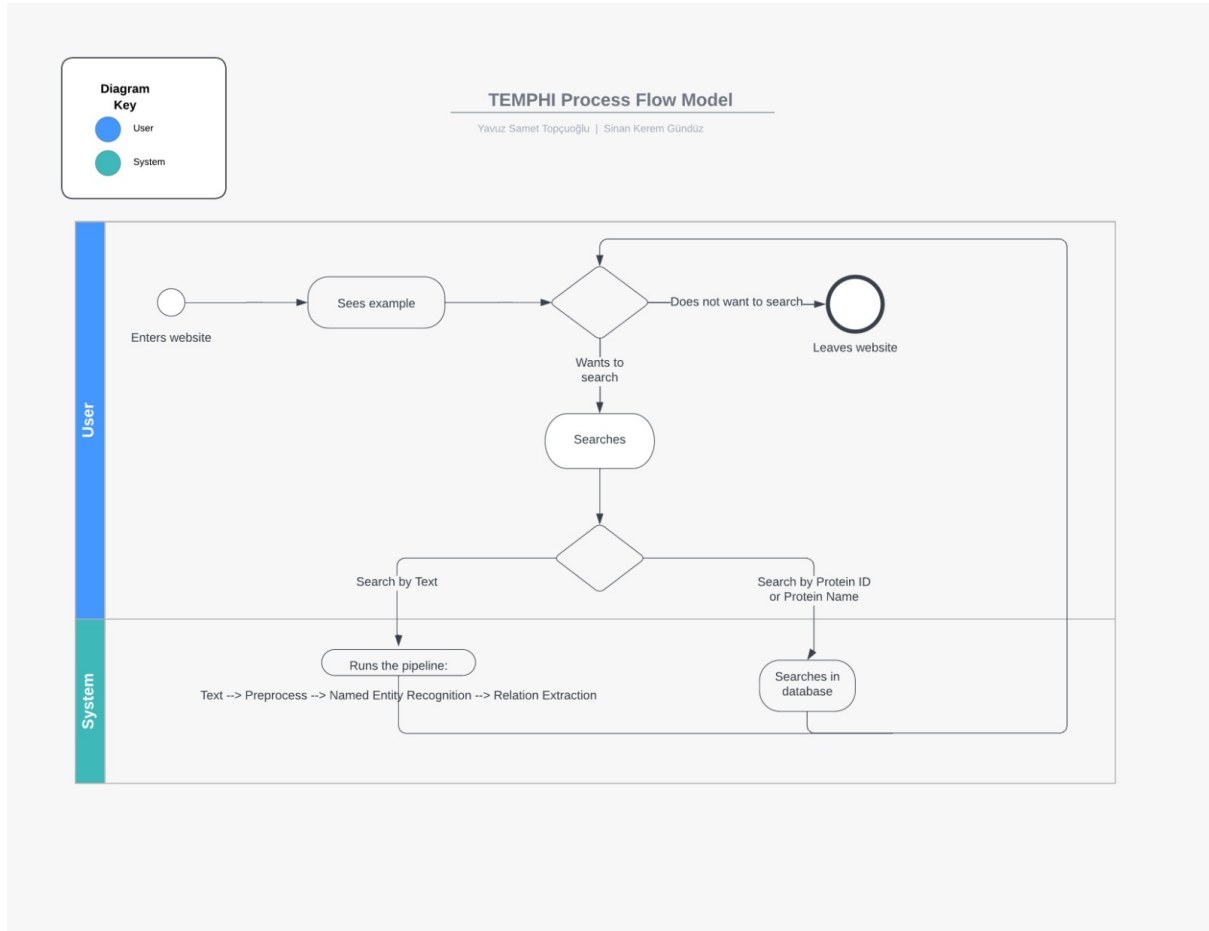


Figure 6.1. Business Process Modelling Notation.

The pipeline begins by fetching titles and abstracts from PubMed or from the user and converting them into the PubTator format. It then uses a Named Entity Recognition (NER) model to extract protein entities from the texts, obtains the UniProt IDs of these proteins, and divides the articles into sentences. For each sentence, it creates pairs from the protein entities and employs a relation extraction tool to identify any existing relationships between the proteins. These relations are stored along with the associated sentence, the names and IDs of the proteins, the PubMed ID, and a unique database ID. The relations are also returned as output for immediate use.

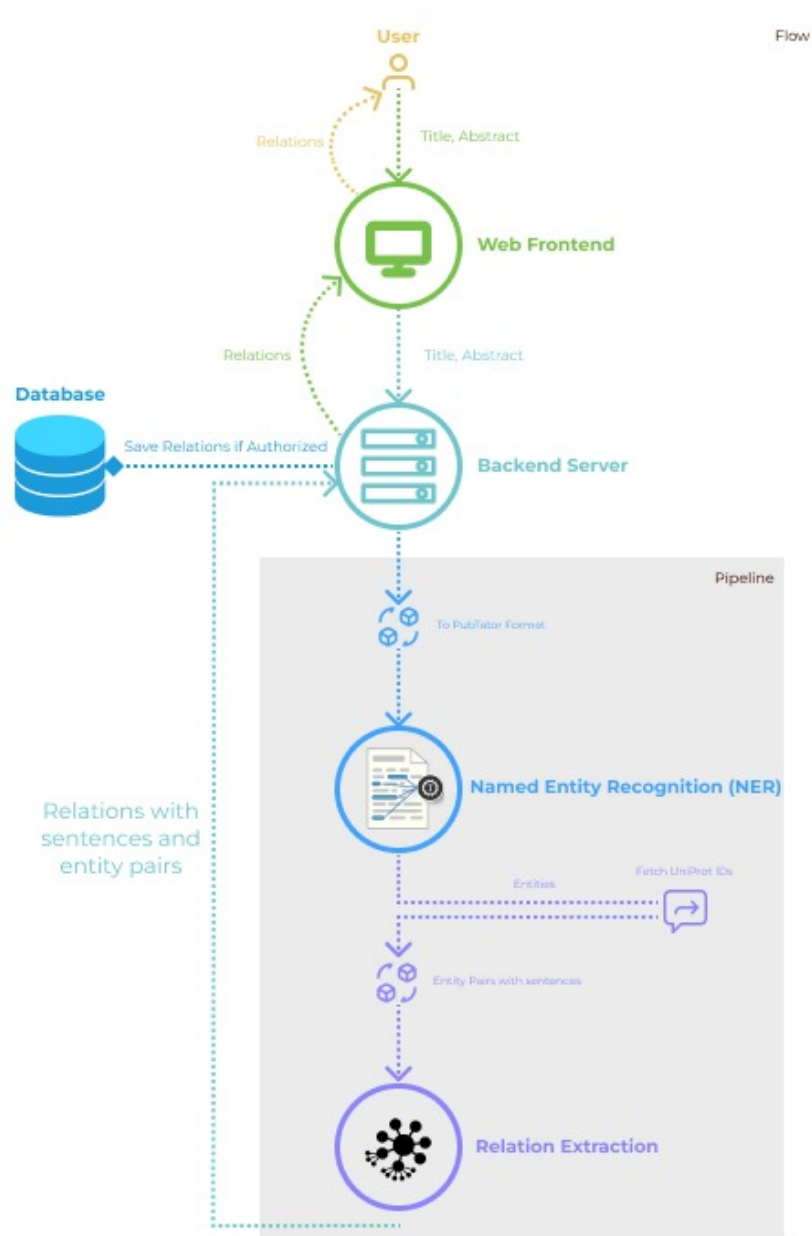



Figure 6.2. Pipeline.

6.3. User Interface Design


Bogazici University
Text Mined Protein - Protein Interactions



Find by Text
Find by Protein ID

🔍

Please search by a title and an abstract or search by protein Ids.

Figure 6.3. Users can use the pipeline by just sending the title and abstract.


Bogazici University
Text Mined Protein - Protein Interactions



Find by Text
Find by Protein ID

🔍

Please search by a title and an abstract or search by protein Ids.

Figure 6.4. Users can search the database with protein IDs. [8] [9]

Find by Text
Find by Protein ID

🔍

Relation between **FAK** and **RhoA**

PubMed ID: 37269754

- Mechanistic insights into the anti-tumor and anti-metastatic effects of Patriniia villosa aqueous extract in colon cancer via modulation of TGF- β R1-smad2/3-E-cadherin and **FAK-RhoA**-cofilin pathways

Relation between **FAK** and **cofilin**

Relation between **FAK** and **JAK2**

Relation between **FAK** and **STAT3**

PubMed ID: 37185776

- FAK**-mediated JAK2/**STAT3** and MEK/ERK signaling was attenuated after EBA challenge in vitro and in vivo

Figure 6.5. Users can see the results without being redirected. In other words on the same page.

7. IMPLEMENTATION AND TESTING

7.1. Implementation

7.1.1. Creating the PHISTO Dataset

To generate a dataset for testing our pipeline, we needed a human-labeled dataset, and we chose PHISTO as our source. However, PHISTO did not provide an API for fetching all database relations, nor could we access the database directly. Therefore, we resorted to web scraping to extract the necessary information, utilizing Selenium, Webdriver, and BeautifulSoup libraries.

Selenium and Webdriver were employed to simulate the search operation on the PHISTO website by locating the search input and search button using CSS_Selector and ID selectors. The search results were then fetched in batches of 15 entries per page. These entries, along with their 14 columns, were returned in HTML format, and BeautifulSoup was used to manage the HTML elements (such as rows represented by ‘tr’ and column values by ‘td’) and extract the relevant data.

To streamline the saving process, we stored 4,000 out of the 62,000 entries in separate Excel files at a time, using Pandas to write the data. Finally, we combined 15 Excel files to create a comprehensive dataset containing 62,000 rows and 14 columns each.

7.1.2. PubMed Extraction

Following the PHISTO data extraction, we proceeded to scrape the titles and abstracts of PubMed documents (whose IDs were stored in our dataset) from the National Library of Medicine’s website. We made simple HTTP requests to the following address: `https://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=pubmed&id={id}&retmode=xml`. The responses were parsed using Element Tree, and the retrieved data was stored in a CSV file with the help of Pandas.

7.1.3. PubTator Conversion

With the CSV file of titles and abstracts in hand, our next task was to convert this data into PubTator format, which serves as the input for the Named Entity Recognition Model used in subsequent steps of our pipeline. A PubTator-formatted document is structured as follows:

```
{Document ID}|t|{Title}  
{Document ID}|a|{Abstract}
```

7.1.4. Named Entity Recognition

Named Entity Recognition (NER) plays a crucial role in our pipeline by identifying and extracting relevant entities from the text, to be more specific: protein names in our use case. For this task, we utilized the code given by AIONER (All-in-one scheme-based biomedical named entity recognition using deep learning) [6], a pre-trained model specifically designed for NER in the biomedical domain.

To incorporate NER into our pipeline, we performed PubTator conversion as depicted above to documents containing titles and abstracts. AIONER provided ready-to-use functionality for NER, allowing us to seamlessly apply the model to our documents. By leveraging AIONER’s capabilities, we were able to extract protein names and other relevant entities from the text, facilitating subsequent steps in our pipeline.

It is important to note that we utilized the NER model as an off-the-shelf solution without making significant modifications or implementing it from scratch. By leveraging the expertise and efforts of the AIONER development team and Berke Kavak who provided the code to us, we were able to benefit from state-of-the-art NER capabilities in the biomedical domain.

The integration of NER into our pipeline has significantly enhanced our ability to identify and extract relevant entities, contributing to the overall effectiveness and

accuracy of our system.

7.1.5. UniProt ID Fetching

In order to retrieve the UniProt IDs corresponding to the extracted protein names, we utilized the UniProt REST API. This API provides a straightforward and efficient way to query the UniProt Knowledgebase and obtain the necessary information.

Using a custom function, we implemented the process of mapping protein names to UniProt IDs. The function takes a protein name as input and constructs a search query using the UniProt REST API. Specifically, we searched for reviewed proteins with matching protein names in the UniProt database. This is how we constructed the query: `"https://rest.uniprot.org/uniprotkb/search?query=(reviewed:true)\%20AND\%20(protein_name:{0})&fields=accession".format(protein_name)"`.

The API response, in JSON format, contains the primary accession IDs associated with the queried protein names. We extracted the UniProt IDs from the response and returned them as the output to be saved in the database.

It is worth noting that the UniProt ID fetching functionality was implemented as a standalone module in our pipeline, enabling seamless integration with other components. By incorporating this step, we established a direct link between the extracted protein names and their corresponding UniProt IDs.

Note that the problems with this ID matching will be explained in error analysis.

7.1.6. Relation Extraction

In the initial stages of our project, we encountered challenges with relation extraction due to the lack of labeled datasets. We experimented with pre-trained models without fine-tuning, which resulted in unsatisfactory outcomes. The models exhibited random weight initialization, lacked stability and consistency, and produced inconsistent

predictions even with the same input data.

Realizing the crucial need for a labeled dataset, here are the two solutions we came up with:

(i) **RelX**

We embarked on the task of generating our own dataset, which we named RelX. To construct this dataset, we leveraged the PHISTO dataset described earlier in this report. We scraped the corresponding PubMed documents and performed Named Entity Recognition (NER) on the extracted sentences.

This process yielded a large dataset comprising sentences from the PubMed documents, along with entity pairs within those sentences. For each entity pair in the dataset, we cross-referenced the PHISTO dataset. If the entity pair existed in the PHISTO dataset, we labeled it with 1 to indicate the presence of a relation. Conversely, if the entity pair was not found in the PHISTO dataset, we labeled it with 0.

(ii) **PPI**

While the RelX dataset provided a foundation for training and testing our Relation Extraction model, the results were not initially convincing. In our pursuit of high-quality labeled protein-protein interaction datasets, we received invaluable assistance from our advisor, Arzucan Özgür. Through her guidance, we were introduced to the PPI dataset.

PPI is combined a variety of RE datasets in the general domain have been constructed to promote the development of RE systems [7].

With the PPI dataset in hand, we proceeded to fine-tune [10] the pre-trained Relation Extraction model. Comparing the performance of the fine-tuned model with the RelX dataset, we obtained detailed results, which will be further discussed in the Results chapter.

Recognizing the superiority of the PPI dataset for model fine-tuning, we employed it to fine-tune various pre-trained models, including bert-base-uncased [11], dmis-lab/biobert-base-cased-v1.1 [12], microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract [13],

and bionlp/bluebert pubmed uncased L-12 H-768 A-12 [14]. These models were then rigorously compared and evaluated, shedding light on their respective performances.

7.1.7. Backend

The backbone of the TEMPHI project, the back-end server, has been developed using Python and FastAPI. Our server takes up multiple responsibilities, including request handling, direct database management, and executing operations as per API calls. Our server interfaces directly with the SQLite database since it is locally hosted, thereby making fetch and save operations convenient. we have configured the back-end server with several APIs:

- (i) Fetch all relations with /relations GET endpoint.
- (ii) Save relation with /relation POST endpoint. with param: relation.
- (iii) Update relation with /relation PUT endpoint. with param: relation.
- (iv) Fetch relations by protein-id or ids with /relations-by-protein-ids GET endpoint with params: protein-1-id (optional) protein-2-id (optional)
- (v) Fetch all protein-ids in the database with /protein-ids GET endpoint.
- (vi) Run the pipeline with / POST endpoint. with params: title, abstract, pubmed-id
- (vii) Run the pipeline and save resulting relations to the database with /add-article POST endpoint. with params: title, abstract, pubmed-id

In addition to these APIs, our back-end server manages the pipeline operations like PubTator, Named Entity Recognition (NER), conversion to Relation Extraction format, and Relation Extraction.

7.1.8. Data Generation

One of the primary objectives of our project is to provide users with a vast amount of information that they can utilize in their research. To achieve this, we implemented a data generation process that involved scraping documents from PubMed. By doing so, we aimed to compile a comprehensive collection of relevant information on various topics.

We formulated several different queries to capture a wide range of research areas. Here are the queries that we have used to retrieve documents from different topics:

- cancer protein markers
- protein targets for cancer therapy
- coronavirus spike protein
- protein-protein interactions in diabetes
- neurodegenerative diseases protein aggregation
- protein biomarkers for cardiovascular diseases
- protein signaling in immune response
- proteins involved in Alzheimer's disease
- protein expression in autoimmune disorders
- protein targets for antiviral drugs
- protein interactions in Parkinson's disease
- proteomics of infectious diseases
- protein-protein networks in HIV infection
- proteins implicated in asthma pathogenesis
- protein-protein interactions in liver disease
- proteomic profiling of kidney cancer
- protein markers for autoimmune thyroid diseases
- protein regulation in inflammatory bowel disease
- proteins involved in lung cancer progression
- protein interactions in multiple sclerosis
- protein-protein interactions in heart disease
- proteomics of breast cancer
- protein markers for liver cirrhosis
- protein interactions in viral pathogenesis
- protein-protein networks in autoimmune disorders
- proteins involved in wound healing
- protein targets for antimicrobial therapy
- protein regulation in neurodevelopmental disorders
- proteomic profiling of pancreatic cancer

- protein markers for kidney disease
- protein interactions in rheumatoid arthritis
- proteins implicated in stroke pathophysiology
- protein-protein networks in metabolic syndrome
- proteomics of inflammatory lung diseases
- protein markers for ovarian cancer
- protein regulation in neurodegenerative diseases
- proteins involved in autoimmune encephalitis
- protein interactions in gastrointestinal disorders
- proteomic profiling of melanoma
- protein targets for regenerative medicine
- protein markers for prostate cancer
- protein interactions in inflammatory skin diseases
- proteins implicated in liver fibrosis
- protein-protein networks in kidney disease
- proteomics of cardiovascular diseases

For each query, we retrieved a maximum of 500 documents from PubMed. These documents were then fed through our pipeline, which encompassed processes such as Named Entity Recognition (NER) and Relation Extraction. The pipeline extracted valuable insights from the documents, such as protein-protein interactions, protein markers, and protein regulation information.

The results obtained from the pipeline were stored in a CSV file, enabling easy access and management of the data. Furthermore, to facilitate the seamless presentation of this information, we stored the generated data in a database. This database serves as the backbone for the frontend of our application, enabling users to search, browse, and access the wealth of information we have compiled.

By generating and organizing this vast amount of data, we aim to empower researchers and scientists in their endeavors by providing a valuable resource that can aid in the exploration of protein-related topics across a multitude of research areas. The data

is readily accessible on the Temphi website (temphi.site), where users can benefit from the comprehensive information we have gathered through our data generation process.

7.1.9. Frontend

The implementation of the TEMPHI project's frontend primarily revolves around the homepage, designed as a single-page application for ease of use and navigability. The homepage utilizes React, a JavaScript library renowned for its efficiency and scalability in building user interfaces, and Bootstrap, a widely-used toolkit for developing with HTML, CSS, and JS.

As a parent component, the HomePage itself renders four other components: Header, SearchForm, ResultForm, and SearchProteinForm. The Header component is dedicated to rendering the website's header, providing a consistent top-level UI element.

The SearchForm component is an interactive element where users can input titles and abstracts for processing. It also triggers the API call to run the pipeline, making the component a crucial part of the frontend-backend interaction.

To display the results, the ResultForm component comes into play. It's responsible for rendering the relations returned from the backend, transforming raw data into user-friendly information.

Further enhancing the user experience, the SearchProteinForm component enables users to fetch protein-IDs from the backend, view them in select components, and initiate a backend search when a protein-ID is selected.

All frontend and backend interactions, including those for pipeline, fetching protein-IDs, and search by protein-IDs operations, are facilitated through the Axios library, known for its promise-based HTTP client for making asynchronous requests.

The frontend is designed with effective error handling mechanisms. In case of an

unsuccessful API call, it displays a message "No relations found," ensuring a smooth user experience by avoiding abrupt disruptions or breakages.

State management is an essential aspect of our frontend implementation. By utilizing React's `useState` and `useEffect` hooks, the dynamic management of states for relations, title, abstract, and search forms for protein-IDs is achieved. As the `HomePage` component holds the state for the results as relations, it enables a seamless and responsive flow of information across the components.

This detailed yet focused approach to the frontend implementation ensures that the TEMPHI project not only functions optimally but also delivers an effective and intuitive user experience.

7.2. Deployment

The deployment of our TEMPHI project was accomplished using the robust and flexible Google Cloud Platform's (GCP) Virtual Machine (VM) instances. This strategic decision facilitated the efficient and seamless operation of our project components across two main categories: backend and frontend.

Our backend, housing the critical components such as the models, pipeline, and database operations, was deployed on a GCP VM instance. Leveraging the capabilities of Python and FastAPI, the application was served using Uvicorn, an ASGI server, ensuring a high-performance runtime environment. Despite our application not being containerized with Docker, we managed a consistent operation across our development and production environments. The backend application directly accesses the SQLite database, which is file-based and located on the same VM for convenience.

To support a smooth and responsive user experience, the frontend was deployed separately on another GCP VM instance. Constructed with React, our frontend effectively communicates with the backend through API calls. This setup allows for efficient data fetching and transmission of user input to the backend for processing. To further enhance

accessibility, we have linked our frontend deployment to the domain "temphi.site".

As the project scales, we may consider migrating to a more robust database management system such as PostgreSQL or MySQL and deploying it either on a separate server or a managed service.

Regarding application monitoring post-deployment, we have utilized the built-in tools of GCP. These tools help monitor the application's performance and maintain logs, thereby assisting us in diagnosing and troubleshooting potential issues effectively.

Through a strategic deployment on GCP's VM instances, our TEMPHI project ensures a scalable and efficient system, offering a seamless and responsive experience to end-users.

8. RESULTS

8.1. Model Comparison

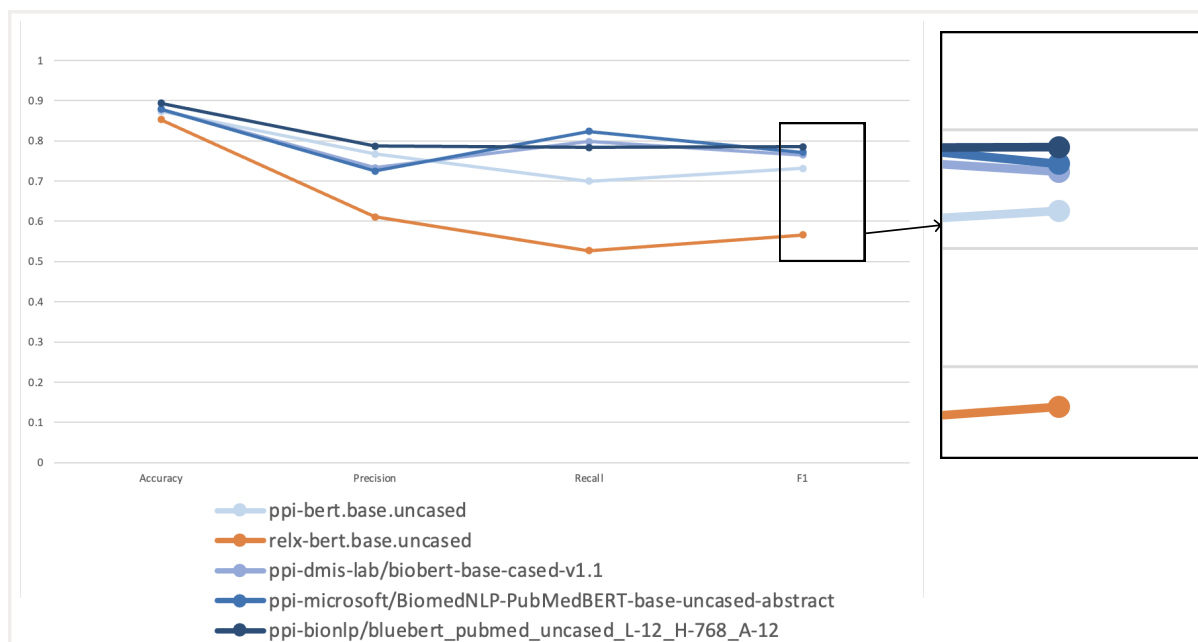


Figure 8.1. Model comparison on four different metrics.

(i) RelX Dataset - bert-base-uncased:

- Test accuracy: 0.853
- Test precision: 0.611
- Test recall: 0.527
- Test F1 score: 0.566

This model achieved moderate accuracy and precision, indicating a reasonable ability to classify relations. However, the relatively low recall suggests that it may have missed some true positive instances, leading to a lower F1 score.

(ii) PPI Dataset - bert-base-uncased:

- Test accuracy: 0.873
- Test precision: 0.767
- Test recall: 0.699
- Test F1 score: 0.732

This model demonstrated higher accuracy, precision, recall, and F1 score compared

to the previous model. It achieved a good balance between identifying positive instances and minimizing false positives.

(iii) PPI Dataset - dmis-lab/biobert-base-cased-v1.1:

- Test accuracy: 0.878
- Test precision: 0.733
- Test recall: 0.799
- Test F1 score: 0.765

This model showed a similar level of accuracy and precision compared to the previous model but demonstrated a higher recall, suggesting its effectiveness in capturing more true positive instances. This resulted in a higher F1 score.

(iv) PPI Dataset - microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract:

- Test accuracy: 0.879
- Test precision: 0.725
- Test recall: 0.824
- Test F1 score: 0.771

This model exhibited a comparable accuracy and precision to the previous models but achieved a higher recall, indicating its effectiveness in identifying a larger number of true positive instances. This led to an improved F1 score.

(v) PPI Dataset - bionlp/bluebert_pubmed_uncased_L-12_H-768_A-12:

- Test accuracy: 0.894
- Test precision: 0.787
- Test recall: 0.784
- Test F1 score: 0.785

This model demonstrated the highest accuracy, precision, and recall among all the models. It achieved a good balance between identifying positive instances and minimizing false positives, resulting in a high F1 score.

The significant performance disparity observed between the models fine-tuned with the PPI dataset and the model fine-tuned with the RelX dataset underscores the crucial role of the training dataset in relation extraction. The PPI dataset, specifically curated

and labeled by the experts for protein-protein interactions, provided more relevant and specific examples for the models to learn from, resulting in better performance metrics across the board. The inclusion of domain-specific information in the training data allowed the models to capture the intricacies and nuances of the relation extraction task in the context of pathogen-host interactions. Conversely, the RelX dataset, which is generated using the technique described earlier, might have provided less accurate training instances for this particular task. These results highlight the importance of using high-quality, domain-specific training datasets to achieve optimal performance in relation extraction tasks.

In summary, the results suggest that the models trained on the PPI dataset, particularly using the dmis-lab/biobert-base-cased-v1.1 and bionlp/bluebert_pubmed_uncased_L-12_H-768_A-12 pre-trained models, outperformed the model trained on the RelX dataset using the bert-base-uncased pre-trained model. These models demonstrated better accuracy, precision, recall, and F1 scores, indicating their effectiveness in relation extraction for protein-protein interactions.

8.2. Computational Efficiency

In this section, we discuss the computational efficiency of our relation extraction models using both CPU and CUDA (Compute Unified Device Architecture). The choice of hardware has a significant impact on the training and testing times, which directly affects the overall efficiency of the project.

8.2.1. Training Efficiency

During the training phase, we compared the performance in terms of speed of our models using CPU and CUDA. Training the models with CPU took almost a day, while CUDA significantly reduced the training time to minutes to hours. This remarkable difference in training times can be attributed to the parallel computing power of GPUs, as CUDA harnesses the potential of NVIDIA GPUs for highly parallel computations.

8.2.2. Testing Efficiency

Similarly, we evaluated the testing efficiency of our models using both CPU and CUDA. When testing with CPU, the process took 2 hours and 12 minutes to complete. However, utilizing CUDA for testing significantly accelerated the process, completing it within just 2 minutes. This drastic reduction in testing time demonstrates the computational superiority of GPUs and the efficiency they bring to the testing phase.

8.2.3. Significance and Impact

The difference in computational efficiency between CPU and CUDA is crucial for our project. The extended training time required by CPU can be impractical, leading to delayed iterations and slower experimentation. In contrast, the accelerated training times achieved with CUDA allow for faster model development and optimization. The reduced testing time further facilitates rapid analysis and evaluation of the models' performance on unseen data.

The use of CUDA, leveraging the parallel processing power of GPUs, aligns with our goal of creating an automated tool for relation extraction. The significant reduction in training and testing times enables us to process large datasets efficiently and provide near real-time results, enhancing the overall productivity and feasibility of our project.

In conclusion, the integration of CUDA in our project has significantly improved the computational efficiency of our relation extraction models. The remarkable reduction in training and testing times, from days to minutes or hours, allows for faster iterations, quicker experimentation, and timely analysis. By harnessing the power of GPUs, we have successfully optimized our project for efficient processing of pathogen-host interaction data.

8.3. Error Analysis

In this section, we conduct an in-depth analysis of the errors made by our pipeline. Understanding the nature and sources of these errors is crucial for improving the accuracy and reliability of our system.

8.3.1. Annotated Data Quality

The quality of annotated data plays a pivotal role in the performance of relation extraction models. While we made extensive efforts to create and find a high-quality annotated dataset, challenges were encountered during the annotation and searching process. The availability of reliable and comprehensive annotated data for pathogen-host interactions is limited, making it difficult to obtain a large-scale dataset with diverse relations and entities. These limitations can affect the generalization and coverage of our models, particularly for rare or emerging relations.

8.3.2. Named Entity Recognition (NER) Model Concerns

One of the key components of relation extraction is accurate entity recognition. We employed AIONER, a state-of-the-art NER model, to identify and extract protein entities from the text. However, the NER model is not immune to errors and may occasionally provide false information based on the context. This can result in incorrect entity identification, which subsequently impacts the performance of our relation extraction models. We acknowledge the challenges posed by entity recognition and plan to investigate techniques to improve its accuracy and robustness.

8.3.3. Multiple ID Existence for Proteins

Another challenge we encountered is the presence of multiple identifiers for almost every protein. The UniProt database, which serves as a valuable resource for protein information, often assigns multiple identifiers to a single protein due to various factors such as different isoforms, gene variants, or evolutionary variations. This situation compli-

cates the retrieval of the correct protein identifier based on its name or other attributes. The ambiguity in protein identification introduces challenges in accurately establishing relations between proteins and pathogens. We are actively exploring methods to address this issue, such as leveraging additional sources of protein information or implementing advanced disambiguation techniques.

8.3.4. Error Analysis Insights and Future Directions

Through our error analysis, we have gained valuable insights into the limitations and challenges of our relation extraction system. The quality and availability of annotated data, concerns with the NER model, and the complexities of protein identification have emerged as crucial areas for improvement.

To mitigate these challenges, we plan to focus on the following future directions:

- **Data Quality Enhancement:** We will continue refining and expanding our annotated dataset to address the limitations of data availability. This may involve collaborating with domain experts, leveraging existing curated databases, or exploring crowdsourcing options to ensure a more comprehensive and accurate dataset.
- **NER Model Improvement:** We will explore techniques to enhance the performance of the NER model, such as domain-specific fine-tuning, incorporating domain-specific dictionaries, or integrating external knowledge sources. This will help mitigate the risks associated with false information provided by the NER model and improve entity recognition accuracy.
- **RE Model Improvement:** To further improve our relation extraction (RE) model, we will explore avenues such as seeking better baseline models and acquiring high-quality data. We aim to identify and evaluate alternative baseline models that exhibit superior performance in relation extraction tasks. By experimenting with different models and architectures, we can identify the ones that are better suited for our specific domain of pathogen-host interactions. Furthermore, we recognize the importance of high-quality data in training and evaluating our models. We will invest efforts in obtaining annotated datasets with higher accuracy and reliability,

enabling us to train our models on more comprehensive and representative data. By focusing on these aspects, we strive to enhance the effectiveness and robustness of our RE model in extracting accurate and meaningful relations between pathogens and hosts.

- **Protein Identifier Disambiguation:** We aim to develop algorithms and strategies to resolve the challenges arising from multiple protein identifiers. This may involve integrating advanced disambiguation techniques, leveraging ontologies or semantic networks, or considering context-based disambiguation methods to identify the correct protein identifier.

9. CONCLUSION

Last but not least, we reflect on the achievements and contributions of our project, highlighting key findings, implications, and avenues for future research in the field of pathogen-host interaction analysis.

Throughout this project, we successfully developed a tool, TEMPHI (Text Mined Pathogen Host Interactions), for extracting relations from academic biological papers. By leveraging state-of-the-art techniques in Named Entity Recognition (NER) and Relation Extraction (RE), we were able to generate a dataset of entity pairs and provide a user-friendly interface for exploring and searching protein relations.

Our results demonstrate the effectiveness of fine-tuning pre-trained models on domain-specific datasets. The models fine-tuned on the PPI dataset exhibited higher accuracy, precision, recall, and F1-scores compared to the model trained on the RelX dataset. This highlights the importance of training models on relevant and high-quality data to achieve optimal performance.

Furthermore, our error analysis shed light on the challenges associated with annotated data quality, the limitations of the NER model, and the complexities of protein identification. These insights provide valuable directions for future improvements. Enhancing annotated data quality, refining the NER model, and addressing the challenges of protein identifier disambiguation are crucial areas for further research and development.

As we conclude this project, we recognize that there is still room for improvement and further exploration. We aim to seek better baseline models, investigate advanced techniques, and leverage high-quality datasets to enhance the performance and robustness of our system. Additionally, collaborating with domain experts and incorporating user feedback will provide valuable insights for refining our tool and ensuring its usability in academic research.

In conclusion, TEMPHI represents a significant step forward in the automated extraction of protein-protein interactions from academic literature. By leveraging cutting-edge techniques in natural language processing and machine learning, we have developed a tool that enables researchers to access valuable information and explore relationships between proteins. This project not only contributes to the field of protein-protein interaction analysis but also provides a foundation for future advancements and discoveries in the domain.

We extend our gratitude to all individuals who have supported and contributed to the successful completion of this project, including the research advisor Arzucan Özgür and helping Master's Student Berke Kavak. Their invaluable guidance, insights, and assistance have played a vital role in shaping the outcomes and achievements of this endeavor.

With the completion of this project, we look forward to continued exploration, innovation, and collaboration in the field of text mining and bioinformatics, ultimately advancing our understanding of protein-protein interactions and their implications in various domains.

You can reach the code of this project from:
<https://github.com/yavuuzsameet/TEMPHI>

REFERENCES

1. Durmuş Tekir, S., T. Çakır, E. Ardiç, A. S. Sayılırbaş, G. Konuk, M. Konuk, H. Sarıyer, A. Uğurlu, I. Karadeniz, A. Özgür, F. E. Sevilgen and K. Ülgen, “PHISTO: pathogen–host interaction search tool”, *Bioinformatics*, Vol. 29, No. 10, pp. 1357–1358, May, 2013.
2. Prieto, C. and J. De Las Rivas, “APID: Agile Protein Interaction DataAnalyzer”, *Nucleic Acids Res.*, Vol. 34, pp. 298–302, Jul, 2006.
3. Xenarios, I., D. W. Rice, L. Salwinski, M. K. Baron, E. M. Marcotte and E. D., “DIP: the database of interacting proteins”, *Nucleic Acids Res.*, Vol. 28, No. 1, pp. 289–291, Jan, 2000.
4. Licata, L., L. Briganti, D. Peluso, L. Perfetto, M. Iannuccelli, E. Galeota, F. Sacco, A. Palma, A. P. Nardoza, E. Santonico, L. Castagnoli and G. Cesareni, “MINT, the molecular interaction database: 2012 update”, *Nucleic Acids Res.*, Vol. 40, pp. 857–861, Jan, 2012.
5. Hermjakob, H., L. Montecchi-Palazzi, C. Lewington, S. Mudali, S. Kerrien, S. Orchard, M. Vingron, B. Roechert, P. Roepstorff, A. Valencia, H. Margalit, J. Armstrong, A. Bairoch, G. Cesareni, D. Sherman and R. Apweiler, “IntAct: an open source molecular interaction database”, *Nucleic Acids Res.*, Vol. 32, pp. 452–455, Jan, 2004.
6. Luo, L., C.-H. Wei, P.-T. Lai, R. Leaman, Q. Chen and Z. Lu, “AIONER: all-in-one scheme-based biomedical named entity recognition using deep learning”, *Bioinformatics*, Vol. 39, No. 5, may 2023, <https://doi.org/10.1093/bioinformatics/btad310>.
7. Luo, L., P.-T. Lai, C.-H. Wei, C. N. Arighi and Z. Lu, “BioRED: a rich biomedical relation extraction dataset”, *Briefings in Bioinformatics*, Vol. 23, No. 5, 07 2022, <https://doi.org/10.1093/bib/bbac282>, bbac282.

8. Köksal, A., H. Dönmez, R. Özçelik, E. Ozkirimli and A. Özgür, “Vapur: A Search Engine to Find Related Protein-Compound Pairs in COVID-19 Literature”, *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, 2020.
9. Köksal, A., “Vapur”, <https://tabilab.cmpe.boun.edu.tr/vapur/>.
10. “GitHub - hushee69/biobert-relation-extraction: Relation Extraction using BERT and BioBERT - using BERT, we achieved new state of the art results — github.com”, <https://github.com/hushee69/biobert-relation-extraction>.
11. Devlin, J., M. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, *CoRR*, Vol. abs/1810.04805, 2018, <http://arxiv.org/abs/1810.04805>.
12. Lee, J., W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So and J. Kang, “BioBERT: a pre-trained biomedical language representation model for biomedical text mining”, *Bioinformatics*, Vol. 36, No. 4, pp. 1234–1240, 2020.
13. Gu, Y., R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao and H. Poon, “Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing”, , 2020.
14. Peng, Y., S. Yan and Z. Lu, “Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets”, *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pp. 58–65, 2019.