Symptom Extraction and Linking from COVID-19 Related Vaccine Adverse Event Reports

Matthew Hullstrung

1. Introduction

This project focuses on the automated extraction and linking of symptoms from Vaccine Adverse Event Reports (VAERS). VAERS serves as a crucial early warning system co-managed by the Centers for Disease Control and Prevention (CDC) and the U.S. Food and Drug Administration (FDA), monitoring potential safety concerns following vaccine administration. I will be specifically focusing on COVID-19 related reports. However, the issue with this system is that it is a passive reporting system, as in, not automatic. My primary goal is to develop a system that can automatically identify symptoms within the narrative text of VAERS reports and link them to standard symptom terms using named entity recognition packages and entity linking methods.

The task is challenging due to the varying nature of VAERS reports and the absence of definitive ground truth annotations. This project addresses these challenges through innovative natural language processing techniques to enhance vaccine safety monitoring.

Data collection involved navigating challenges in the VAERS dataset, including unconventional formatting, while data preprocessing encompassed selecting a representative dataset, conducting statistical analyses, and creating a relevant vocabulary. Symptom-Related Entity Extraction was accomplished using the Stanza package and biomedical/clinical NER models. The subsequent step of Entity Linking was implemented through rule-based methods, featuring fuzzy matching techniques with the fuzzywuzzy package.

This introduction sets the stage for a detailed exploration of the methods employed in Named Entity Recognition (NER) and Entity Linking, providing insight into how these techniques address the unique challenges posed by the VAERS dataset. The following sections will delve into the intricacies of the project, presenting findings, lessons learned, and avenues for future improvements.

2. Problem Formulation

Formally, this project is a sequence labeling task. Given VAERS reports as input, the objective is to extract and link symptom-related entities within the text. The task can be defined as follows:

Step 1: Extracting Symptom-Related Entities

Input: Descriptions of COVID-19 related vaccine adverse events (SYMPTOM TEXT in VAERS DATA table).

Output: A list of symptom-related entities

Task Type: Named Entity Recognition

Step 2: Link Entities to Standard Symptoms

Task: For each extracted symptom entity, map it to the term in a standard symptom list

Input: A symptom detected in STEP 1, Vocabulary 1 (a list of all standard symptoms), Vocabulary 2 (a list of the top 100 most common standard symptoms)

Output: Its standard symptom

Task type: Entity Linking

3. Methods

In this project, I will employ a combination of methods and tools to achieve the defined task:

Data Collection: VAERS dataset, focusing on the COVID-19 vaccine reports from 2022. Importing the data presented a hurdle due to unconventional formatting in the .csv data file. By excluding unnecessary fields like hospital visit dates, this challenge was addressed, resulting in a streamlined data import process. This not only resolved the formatting issue but also optimized memory usage for program execution.

Data Preprocessing: Selection of a representative dataset, statistical analysis, and vocabulary creation. We consider only COVID-19 vaccine data from 2022, sampling 10,000 reports. I filter out samples with missing reports and samples with reports outside of character length range [50,3000]. Statistical analysis consists of getting distributions of the counts of different symptoms related to the COVID-19 vaccine as well as getting the distribution of the length of the different reports. This analysis will help us better understand the data we will work with.

Symptom-Related Entity Extraction: Utilizing the Stanza package and biomedical/clinical NER models. To accomplish NER, we use a pipeline with the MIMIC III syntactic analysis model with the i2b2 clinical NER processor. By considering items labeled PROBLEM as symptoms, we successfully accomplish our task of symptom-related entity extraction.

Entity Linking: Implementation of rule-based methods. I utilize fuzzy matching using the fuzzywuzzy package on extracted symptoms. To be matched, entities must have a fuzzy matching score above a threshold of 90. If this score is not met, try matching the synonyms of this entity with the same threshold. If there are multiple potential matches for some entity, choose the one with the highest score. If no symptoms are linked, consider this a case of “No adverse event.” I initially used exact matching, but fuzzy matching captured many more entities. However, in the cases of symptoms such as “fever” having to be matched to “pyrexia,” matching on the symptoms was the best solution I could find while still using rule-based methods. In further implementations, I would consider switching to similarity-based methods to counteract these issues more robustly.

Evaluation: Systematic evaluation, including manual validation and metrics such as precision, recall. Consider VAERS columns SYMPTOM1 through SYMPTOM5 as ground truth (though in reality they are far but the ground truth, more discussion later).

To perform automatic evaluation, I consider the symptoms in columns SYMPTOM1 through SYMPTOM5 as ground truth. However, as we will later learn, these are not the best at doing so. Further implementation could possibly include ChatGPT results as ground truth for evaluation. With these ground truth annotations, we can then calculate precision and recall. For any single report, a true positive is a linked symptom that's also a ground truth symptom, a false positive is a linked symptom found that's not in the ground truth symptom list (this is slightly flawed), and a false negative is a ground truth symptom not found in the extracted symptom list. False negatives are not necessary because I am not interested in calculating accuracy. We aggregate these values for each report, and calculate the aggregated precision and recall as the following:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

I checked if a linked symptom was equal to a ground truth symptom using fuzzy matching to account for slight misspellings.

To perform manual evaluation, I checked the first 25 reports by hand. I do not calculate precision or recall by hand, but I did pay close attention to what would be considered false positives or false negatives to explain anomalies in the automatic evaluation.

4. Dataset and Experiments

The project uses the VAERS dataset, focusing on COVID-19 vaccine-related reports from 2022. Data preprocessing includes subset selection (10,000 samples) and standard symptom list creation. I further preprocessed the data by filtering out samples with missing symptom texts and only including samples with report lengths between 50 and 3,000 characters to help reduce number of outliers and computation time. Experiments formally involve Named Entity Recognition (NER) and Entity Linking (EL) tasks, evaluated automatically by calculating precision and recall and manually through my own personal evaluation on a small subset of the samples.

To analyze the statistics of the dataset, I plotted distributions of the standard symptom counts and report lengths. Each of these plots can be seen as *Figure 1* and *Figure 2* below (respectively):

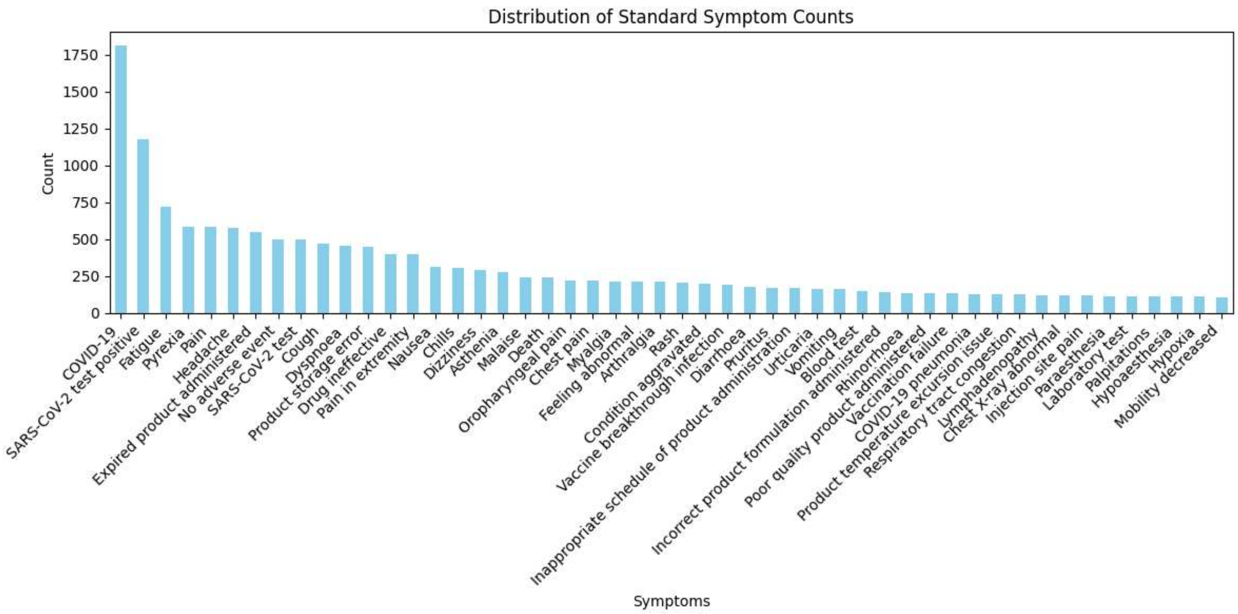

Figure 1

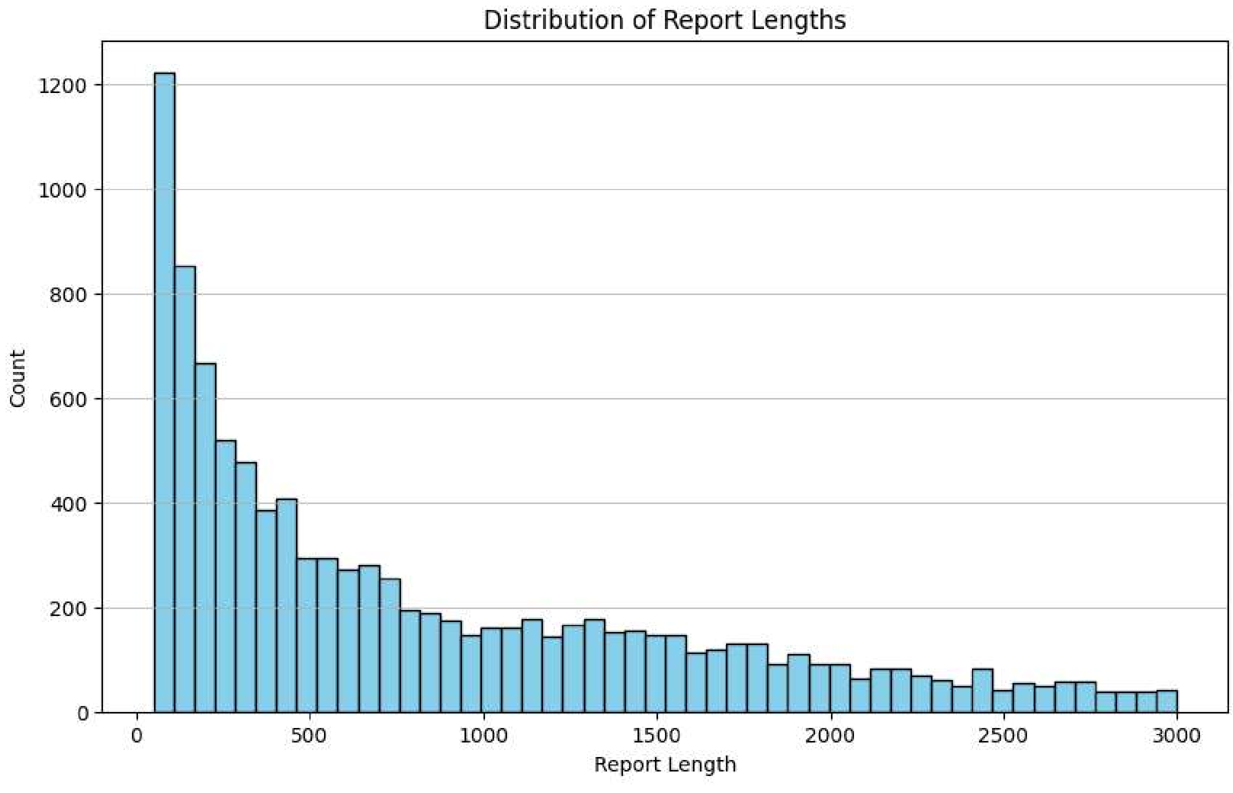

Figure 2

As shown in *Figure 1*, the top 5 standard symptoms from the reports were COVID-19, a positive SARS-CoV-2 test, fatigue, and an expired product administered. The top 50 symptoms are in the attached *most_common_symptoms.csv* file.

To help justify the distribution given in *Figure 2*, I also performed some numerical statistics of the report lengths, including mean, standard deviation, minimum, maximum, and the number of missing reports.

There are no missing reports, meaning the count remains at 10,000. The mean of the report lengths is 825.47 characters with a standard deviation of 750.28 characters. The minimum report length is 50 characters while the maximum report length is 2,999 characters.

Named Entity Recognition

I have accomplished Named-Entity Recognition using the Stanza package, with the pipeline consisting of the MIMIC III syntactic analysis model using the i2b2 clinical processor for NER.

An example input and output of a single sample's Named-Entity Recognition is shown below:

Input: Pt was admitted to hospital with a positive COVID test on 1/21 and transferred to another hospital on 1/25. Prior to arrival the pt had 6 days of worsening cough, fever, extreme fatigue, and worsening shortness of breath.

Output Symptoms (PROBLEM): ['a positive COVID test', 'worsening cough', 'fever', 'extreme fatigue', 'worsening shortness of breath']

I manually checked over 20 samples, and as seen in this single example, our pipeline does a good job at extracting symptom-related entities. This task has been successfully accomplished. However, as we will later learn, the Stanza NER model we created tends to miss symptoms such as 'COVID-19' and 'Expired product administered.' These symptoms are unfortunately the most common symptoms, so it takes a severe hit on our evaluation. Further implementations should address this issue through usage of a different model or maybe even an LLM to help find missing symptoms such as these.

Entity Linking

Entity Linking (EL) was the next step on the project timeline, which was evaluated automatically by calculating precision and recall and manually through my own evaluation on a small subset of the samples.

I began entity linking using exact matching, but against my manual evaluation, it proved to miss linking many extracted symptoms. This is when I realized I will need to implement fuzzy matching using the fuzzywuzzy package. I implemented fuzzy matching with a matching threshold of 90. In other words, we consider fuzzy matching scores below 90 to have no match in the standard symptom list. However, this also had trouble linking some symptoms. Most notably, the standard symptom for 'fevers' is known as 'pyrexia.' I had to modify my entity linking algorithm to look for alternate definitions such as these. The best way I found to account for this was checking if the synonyms of an extracted symptom have a higher match score than the symptom itself. This entirely solved the issue with matching 'fevers' to 'pyrexia' along with many other extracted symptoms with this same issue.

Results

After symptom extraction performed by an NER model using the Stanza package and entity linking using the fuzzywuzzy package, we can now evaluate our results using the methods described above in Section 3. I split this into two evaluations: automatic and manual.

The results were unamusing, with a precision of 34.84% and a recall of 37.54% for symptoms linked to all standard symptoms and a precision of 42.05% and recall of 45.68% for symptoms linked to the top 100 most common symptoms. To further explain these results and why they may be so low, I also performed a manual evaluation.

My manual evaluation consisted of looking at 25 samples (the first 25 samples in the *evaluation_data.csv* file). I did not manually calculate precision or recall as this would result in basically the same results as my automatic evaluation. Instead, I did a more analytical breakdown of each report, looking to explain the automatic evaluation rather than creating more evaluation. After looking at these samples, the biggest missed symptoms were the symptoms relating to storage and expiration date of the vaccine. These symptoms were missed by the NER model because these are not real symptoms of the vaccine, making evaluation unfair. This meant that the number of false negatives (ground truth symptoms not extracted) was inflated by symptoms that were not truly symptoms. Also, the false positives (expected linked symptoms not in the ground truth) almost always happened to be true positives after manually analyzing the original symptom text. This meant our precision should have been higher than it was in our automatic evaluation. An example of this phenomenon is seen in the following sample:

SYMPTOM TEXT:

"she's having major back pain since sunday, after she got the 2nd covid shot."; This is a spontaneous report received from contactable reporter(s) (Other HCP, Consumer or other non HCP and Physician) for a Pfizer sponsored program (141003). A 71-year-old female patient received bnt162b2 (BNT162B2) (Batch/Lot number: unknown) as dose 2, single for covid-19 immunisation. The patient's relevant medical history was not reported. Concomitant medication(s) included: XELJANZ XR taken for rheumatoid arthritis, start date: Sep2020. Vaccination history included: Bnt162b2, for COVID-19 immunization. The following information was reported: BACK PAIN (non-serious), outcome "unknown", described as ""she's having major back pain since sunday, after she got the 2nd covid shot."' Additional information: It was reported that, Case is from Marketing program but suspect product(s) are not part of the case Suspect product Xeljanz XR updated to Pfizer covid 19 Vaccine as per the events reported and Xeljanz XR considered as Concomitant product coded via CDD. spouse reported md discontinued xeljanz "about 2 weeks ago, "because it just wasn't working, her inflammation seemed to be getting worse". No follow-up attempts are possible; information about lot/batch number cannot be obtained. No further information is expected.

Ground Truth:

['Back pain']

Expected Linked Symptoms:

{'Rheumatoid arthritis', 'Back pain', 'Inflammation'}

All expected linked symptoms are true, even though the ground truth only contains 'Back pain.' This sample has contributed to incorrectly classified false positives in our automatic evaluation, erroneously decreasing precision. This happens quite frequently and tells us a lot about the value of proper ground truth annotations and what happens to our evaluation if they are not properly set. Changing the ground truth annotations to extract symptoms from the symptom text using an LLM such as ChatGPT may prove to be useful and accurate, though I did not perform any testing on this matter.

5. Project Management

Matthew Hullstrung: Project Lead and Data Scientist

Responsibilities: Coordination, data collection, model implementation, and project oversight.

As the sole member of this project, I am responsible for all aspects of its execution. Here was my abbreviated project timeline:

Weeks 1-2: Data collection and preprocessing, including dataset selection.

Week 3: Implementation of entity extraction (NER).

Week 4: Midterm Report.

Weeks 5-6: Development of entity linking methods. Manual evaluation and model validation.

Week 7: Final evaluation, results analysis, and system refinement.

End of Semester: Project completion and submission.

6. Conclusion

In addressing the sequence labeling task of extracting and linking symptom-related entities from COVID-19 vaccine reports in the VAERS dataset from 2022, this project employed a diverse set of methods and tools. The project began with data collection from the VAERS dataset, diligently addressing formatting challenges for streamlined import. Data preprocessing involved statistical analysis and vocabulary creation on a representative dataset of 10,000 reports, filtering out irrelevant samples and ensuring report lengths within a specified range.

Symptom-related entity extraction was achieved using the Stanza package and biomedical/clinical Named Entity Recognition (NER) models, focusing on items labeled as

"PROBLEM." Entity Linking (EL) followed, employing rule-based methods and fuzzy matching techniques with the fuzzywuzzy package. Evaluation methods included systematic evaluation, manual validation, and metrics such as precision and recall, using VAERS columns SYMPTOM1 through SYMPTOM5 as ground truth, acknowledging limitations in this definition.

Moving beyond methodology, the dataset and experiments involved the VAERS dataset, focusing on COVID-19 vaccine-related reports from 2022 as stated. Subset selection, standard symptom list creation, and further preprocessing were conducted. Statistical analysis included plotting distributions of standard symptom counts and report lengths. Named Entity Recognition using the Stanza package successfully identified symptom-related entities, but challenges were identified in missing common symptoms like 'COVID-19' and 'Expired product administered.'

Entity Linking, evaluated automatically and manually, initially used exact matching but transitioned to fuzzy matching with a threshold of 90. Challenges in linking symptoms with alternate definitions were addressed by considering synonyms. Results after symptom extraction and entity linking showed unamusing precision and recall percentages, prompting a manual evaluation of 25 samples. The manual evaluation revealed challenges related to missed symptoms during extraction, particularly those related to vaccine storage and expiration dates. The evaluation also proved inaccuracies in the calculation of false positives and true positives, impacting precision. Further implementations should consider the use of LLMs such as ChatGPT to conduct better evaluation without relying on VAERS' attempt at providing a ground truth.

In terms of project management, I served as the Project Lead and Data Scientist, overseeing all data collection, model implementation, and project execution. The project timeline spanned data collection and preprocessing in the initial weeks, followed by the implementation of entity extraction (NER) in week 3, a midterm report in week 4, development of entity linking methods and manual evaluation in weeks 5-6, final evaluation, results analysis, and system refinement in week 7, and concluded with project completion and submission at the end of the semester.

7. Key references

1. Qi P, Zhang Y, Zhang Y, Bolton J, Manning CD. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations 2020 Jul (pp. 101-108).
2. Yuhao Zhang, Yuhui Zhang, Peng Qi, Christopher D Manning, Curtis P Langlotz, Biomedical and clinical English model packages for the Stanza Python NLP library, Journal of the American Medical Informatics Association, Volume 28, Issue 9, September 2021, Pages 1892–1899, https://doi.org/10.1093/jamia/ocab090
3. Chen Q, Peng Y, Lu Z. BioSentVec: creating sentence embeddings for biomedical texts. The 7th IEEE International Conference on Healthcare Informatics. 2019.
4. Gu Y, Tinn R, Cheng H, Lucas M, Usuyama N, Liu X, Naumann T, Gao J, Poon H. Domain specific language model pretraining for biomedical natural language processing. ACM Transactions on Computing for Healthcare (HEALTH). 2021 Oct 15;3(1):1-23.