

# Symptom Extraction and Linking from Vaccine Adverse Event Reports

**Thanapoom Phatthanaphan**

Stevens Institute of Technology  
tphattha@stevens.edu

## Abstract

The Vaccine Adverse Events Reporting System (VAERS) serves as a critical resource for monitoring the safety of vaccines. However, the vast amount of unstructured narrative text within VAERS reports poses a challenge for efficient analysis and symptom identification. In this paper, we present a novel approach to automatically extract and link vaccine-related symptoms from VAERS reports using sequence labeling techniques. Our method combines the power of named entity recognition (NER) and named entity linking (NEL) to achieve comprehensive and standardized symptom identification. We utilize NER packages and develop custom NEL methods to map identified terms to standard terminology from a symptom dictionary. By doing so, we aim to enhance the efficiency and accuracy of vaccine adverse event monitoring, ultimately contributing to the safety assessment of vaccines.

## Introduction

Vaccines are crucial for public health, and to ensure their safety, the Vaccine Adverse Events Reporting System (VAERS) was jointly established by the CDC and FDA. VAERS collects reports of adverse events following vaccination, allowing continuous safety monitoring.

However, VAERS faces a challenge due to the high volume of unstructured narrative reports it receives each year. These reports contain valuable information about adverse events, including symptoms. Identifying these symptoms promptly is essential for taking action and maintaining public trust in vaccination programs.

In recent years, sequence labeling techniques, used in linguistics and computational linguistics, have proven valuable for tasks like identifying named entities. Our project introduces an innovative method to automatically extract and connect vaccine-related symptoms from VAERS reports.

Our project focuses on two main goals: identifying symptom-related terms within the narrative text reports using NER tools and linking these terms to standardized symptom terminology. This approach addresses the challenge of unstructured VAERS reports while ensuring alignment with recognized medical terms.

This paper explains our methodology, including NER tool selection, symptom dictionary creation, and linking techniques. We also present experimental results to demonstrate the effectiveness of our approach. Ultimately, our project enhances vaccine adverse event monitoring, strengthens public health, and ensures vaccine safety, focusing on symptoms related to COVID-19.

Our project leverages innovative sequence labeling techniques, inspired by research in natural language processing. Specifically, the project draws insights from two prominent studies in the field. The work by Qi et al. [1] introduces a Python Natural Language Processing Toolkit, providing a valuable resource for processing diverse human languages. This toolkit is foundational to our approach, enabling efficient analysis of the narrative reports within VAERS. Additionally, the study by Zhang et al. [2] contributes essential biomedical and clinical English model packages for the Stanza Python NLP library. These packages are instrumental in our Named Entity Recognition (NER) methodology, enhancing the accuracy of symptom extraction from clinical narratives.

In our project, the focus has been on COVID-19 reports, given their prevalence compared to other types of reports. Leveraging the stanza library, we successfully extracted symptoms associated with COVID-19 from the corresponding text. The subsequent step involved linking these extracted symptoms with established standard symptoms to assess the model's performance. Our evaluation process encompasses both automated and manual methods, yielding results that highlight the model's strengths and areas for improvement. This iterative approach allows us to identify and address potential issues, ensuring continuous refinement for enhanced accuracy. Moving forward, we remain committed to refining our Named Entity Recognition (NER) model, conducting regular evaluations, and collaborating closely with domain experts to overcome challenges and ensure the project's overall success.

During the evaluation, a notable challenge emerged concerning the model's performance. The automatic evaluation indicated a suboptimal outcome, suggesting that the model struggled to identify symptoms. However, upon manual in-

spection, it became evident that the model was, in fact, proficient at symptom extraction. The discrepancy arose from variations in wording between the extracted symptoms and the established standard symptoms. Despite accurately identifying symptoms, the model expressed them using different terminology, impacting the automated evaluation results. Addressing this semantic variation is crucial for improving the model's automatic performance and ensuring a more accurate reflection of its capabilities. Our commitment to iterative improvement, coupled with ongoing collaboration with domain experts, remains instrumental in overcoming these challenges and optimizing the project's overall success.

## Problem formulation

The core machine learning task in this project is formally defined as a **multi-class classification** problem. This task involves classifying text descriptions of vaccine adverse events, specifically the SYMPTOM TEXT in the VAERS DATA table, into predefined categories representing various symptoms or symptom-related entities. The formulation can be represented as follows:

- **Input:** Text descriptions of vaccine adverse events (SYMPTOM TEXT).
- **Output:** Class labels representing symptom-related entities or symptoms.
- **Classes:** Multiple predefined classes representing different symptoms or symptom-related entities.

## Extraction of Symptom Entities using Stanza i2b2 NER Model

### Notations:

- $D$  represents the set of VAERS reports (Symptom text), each denoted as  $d_i$ .
- $S$  represents the set of symptom entities extracted from the reports using the Stanza NER package, represented as  $s_i$

The extraction process is denoted as  $s_i = nlp(d_i).entities$ , where  $s_i$  represents the extracted symptom entities.

## Named Entity Recognition (NER) Packages

In linking the symptoms extracted from the text to standard symptoms, a comprehensive strategy was employed, integrating Rule-based matching, including both Exact and Fuzzy matching, and similarity-based matching utilizing Cosine-similarity. The Rule-based matching component established specific rules and patterns for precise symptom identification, incorporating flexibility through Fuzzy

matching. Simultaneously, similarity-based matching introduced a semantic layer, allowing for nuanced comparisons beyond exact word matches. This combined approach enhances the accuracy and adaptability of symptom linking, addressing variations in symptom expression within clinical narratives.

- **Exact matching:** seeks direct correspondence between extracted symptoms and established standard symptoms, ensuring precise alignment.
- **Fuzzy matching:** allows for a degree of flexibility, accommodating variations or slight discrepancies in symptom descriptions.
- **Similarity matching:** Cosine-similarity, gauges the likeness between the vector representations of symptoms.

### Cosine-similarity formular:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

Here,  $A \cdot B$  represents the dot product of vectors  $A$  and  $B$ , and  $\|A\|$  and  $\|B\|$  denote the Euclidean norms of vectors  $A$  and  $B$ , respectively. The resulting value ranges from -1 (completely dissimilar) to 1 (perfectly similar), providing a quantitative measure of similarity between the symptom vectors.

## Methods

Several methodologies, techniques, and tools will be employed. These approaches aim to automatically identify symptoms from VAERS reports and link them to standard terms in a dictionary. Here's an outline of the key methods and tools required for this project:

### Data Preprocessing

Preceding the extraction of symptoms, a pivotal phase in our approach involves subjecting the reports within the VAERS dataset to an exhaustive data preprocessing stage. This critical step entails meticulously organizing the data to optimize subsequent processing efficiency. Specifically, we have elected to work with a dataset comprising 10,000 VAERS reports, each encapsulating symptom-related text pertaining to COVID-19, as depicted in Figure 1. The deliberate selection of this sizable dataset serves a strategic purpose, aiming to comprehensively observe and test our methodology's performance in the intricate task of symptom extraction from VAERS reports. This extensive dataset acts as a robust foundation, enabling us to rigorously evaluate and affirm the efficacy and reliability of our model across a diverse and substantial array of real-world clinical narratives. Such a delib-

erate choice ensures a thorough examination of our methodology's adaptability and effectiveness in handling the inherent complexities of varied clinical symptom descriptions.

VAERS_ID		SYMPTOM_TEXT
0	2669769	body aches, fatigue Narrative: Took OTC Tyleno...
1	2527460	Headache, Myalgia, NauseaVomiting, chills Narr...
2	2673135	Headache, Fever, Body aches Narrative: Other ...
3	2672717	Headache & Myalgia Narrative: Other Relevant...
4	902418	Patient experienced mild numbness traveling fr...
...	...	...
9995	916173	REDNESS TO INJECTION SITE 12-30-20. PROGRESSED...
9996	916174	Patient described joint and muscle pain in the...
9997	916176	Numbness and tingling on left side of face, ey...
9998	916177	HIVES, tachypnea, vomiting - normal saline, ne...
9999	916178	Excessive swelling to left axillary lymph node...
10000 rows x 2 columns		

Figure 1: A dataset comprising 10,000 VAERS containing symptom text related to COVID-19

Before commencing the extraction of symptom-related entities from the set of 10,000 reports, The list of standard symptoms and the list of the most frequent 100 symptoms associated with COVID-19 are prepared. The purpose of these lists is to facilitate a rigorous comparison between the symptoms extracted from the reports and the established standard symptoms. Figure 2 provides an illustrative example of the list of standard symptoms.

```
['Fatigue',  
'Pain',  
'Chills',  
'Headache',  
'Myalgia',  
'Nausea',  
'Vomiting',  
'Pyrexia',  
'Hypoaesthesia',  
'Injection site hypoaesthesia',  
'Erythema',  
'Feeling hot',  
'Flushing',  
'Dizziness',  
'Electrocardiogram normal',  
'Hyperhidrosis',  
'Laboratory test normal',  
'Presyncope',
```

Figure 2: The list of standard symptoms

### Named Entity Recognition (NER) Packages

Utilization of specialized NER packages tailored for biomedical and clinical text analysis, including Stanza, i2b2, and similar libraries, to identify symptom-related entities. This is the primary technique employed for symptom extraction. The i2b2 model is specifically chosen for its proficiency in recognizing clinical entities, aligning well with the medical context of VAERS reports.

### Symptom Extraction Implementation

The initiation of the Stanza library, employing the i2b2 Named Entity Recognition (NER) model, marks the commencement of the symptom extraction process from each individual report. The methodology includes an iterative procedure, wherein a subset of reports (specifically, 30 reports selected for the purpose of demonstration) is systematically processed. The output consists of the extracted entities, encompassing their respective text and assigned entity types. For clarity and illustrative purposes, Figure 3 serves as a representative example showcasing the extracted entities from the 30 reports.

```
Vaers ID: 2669769  
Input: body aches, fatigue Narrative: Took OTC Tylenol Other Relevant History:  
Output:  
body aches PROBLEM  
fatigue PROBLEM  
OTC Tylenol TREATMENT  
  
Vaers ID: 2527460  
Input: Headache, Myalgia, NauseaVomiting, chills Narrative:  
Output:  
Headache PROBLEM  
Myalgia PROBLEM  
NauseaVomiting PROBLEM  
chills PROBLEM  
  
Vaers ID: 2673135  
Input: Headache, Fever, Body aches Narrative: Other Relevant History:  
Output:  
Headache PROBLEM  
Fever PROBLEM  
Body aches PROBLEM  
  
Vaers ID: 2672717  
Input: Headache & Myalgia Narrative: Other Relevant History:  
Output:  
Headache PROBLEM  
Myalgia PROBLEM
```

Figure 3: The example of extracted entities

### Named Entity Linking (NEL) Packages

The project involves the formulation and refinement of Named Entity Linking (NEL) methods, with the primary objective of establishing connections between symptom entities identified in the clinical text and standardized terms within a dedicated symptom dictionary. This phase of development is crucial for enhancing the interpretability and alignment of identified symptoms with recognized medical terminology, contributing to the overall robustness and applicability of the symptom extraction methodology.

### Rule-Based Matching

The project entails the creation and optimization of rule-based matching algorithms, encompassing both exact

matching and fuzzy matching methodologies. These algorithms are designed to establish direct mappings of symptoms to standard terms by adhering to predefined rules. This strategic approach aims to enhance the accuracy and efficiency of symptom mapping, providing a systematic framework for aligning identified symptoms with established standard terminology.

### Exact Matching

The exact matching method operates by directly comparing the symptoms extracted from the data with a predetermined set of standard symptoms as shown in Figure 4. This comparison seeks exact matches, meaning the identified symptoms must precisely match those in the established standard set. When a symptom from the extracted data aligns precisely with a standard symptom, a link is established between the two, indicating a match. This method is straightforward and efficient, as it relies on precise correspondence between the identified symptoms and the standardized ones. The goal is to create a direct, one-to-one mapping, facilitating accurate symptom recognition and analysis within the system.

```
for std_symptom in std_symp:
    if symp.lower() == std_symptom.lower():
        exact_linked_std_symp_dict[symp.lower()] = std_symptom.lower()
```

Figure 4: Linking with Exact matching method

### Fuzzy matching

Fuzzy matching operates by employing algorithms to assess the similarity between two strings of text, such as extracted symptoms and standardized symptoms. Unlike exact matching, which requires an identical match, fuzzy matching allows for a more lenient comparison by considering partial matches and variations in the text. The algorithm calculates a similarity score based on factors like character matches, substitutions, insertions, and deletions between the two strings. This score reflects the degree of resemblance between the extracted symptom and the standard symptom. The higher the similarity score, the more closely the symptoms align. The fuzzy matching code is shown in Figure 5.

```
max_fuzz_ratio = 0.0
for std_symptom in std_symp:
    fuzz_ratio = fuzz.partial_ratio(symp.lower(), std_symptom.lower())
    if fuzz_ratio > max_fuzz_ratio:
        max_fuzz_ratio = fuzz_ratio
        most_sim_std_symp = std_symptom.lower()
fuzzy_linked_std_symp_dict[symp.lower()] = most_sim_std_symp
```

Figure 5: Linking with Fuzzy matching method

### Similarity-Based Matching

The project delves into an exploration of similarity-based matching techniques, leveraging word embeddings such as GloVe or clinical word embeddings. This investigation is undertaken with the aim of optimizing the accuracy of entity

linking. The utilization of advanced word embeddings is envisioned to enhance the precision and efficacy of linking identified entities to standardized terms, contributing to the overall refinement of the research methodology.

In this approach, the method employs cosine similarity calculations to determine the resemblance between the vectors representing the extracted symptoms and those corresponding to standardized terms. The cosine similarity scores quantify the directional similarity, with higher scores indicating greater alignment. The extracted symptom is then linked to the standard symptom with the highest cosine similarity score. This nuanced matching process accounts for semantic similarities and variations in symptom expression, further improving the system's ability to accurately link identified entities to their standardized counterparts. The similarity matching code is shown in Figure 6.

```
for symp in extracted_symptom_list:
    symp_vec = "Code to get embedding vector of a symptom"

    # Find the most similar symptoms from the list of standard symptoms
    max_sim_score = 0.0
    most_symp = 'None'
    for std_symptom in std_symp:
        std_symptom_vec = "Code to get embedding vector of a standard symptom"
        sim_score = "Code to compute cosine similarity"
        if sim_score > max_sim_score:
            max_sim_score = sim_score
            most_symp = std_symptom

    sim_linked_std_symp_dict[symp] = most_symp
```

Figure 6: Linking with Similarity matching method

## Challenges and Mitigation Strategies

- **Processing Time:** A potential challenge is the computational time required for extracting symptoms from a substantial number of reports. To address this, we employ a pragmatic approach by initially processing a limited subset (50 reports) for manual validation. This approach allows us to evaluate the correctness of the extraction method before scaling up to the entire dataset.
- **Manual Validation:** Given the complexity of medical narratives, manual validation is crucial for ensuring the accuracy of the extraction. While the manual checking of 20 reports may be time-consuming, it serves as a necessary step in refining the extraction methodology before applying it to a larger dataset.

## Datasets and Experiments

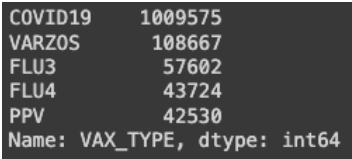
### Datasets

The Vaccine Adverse Event Reporting System (VAERS) is a national early warning system to detect possible safety problems in U.S.-licensed vaccines. VAERS is co-managed by the Centers for Disease Control and Prevention (CDC) and the U.S. Food and Drug Administration (FDA). VAERS

accepts and analyzes reports of adverse events (possible side effects) after a person has received a vaccination. There are three tables in the VAERS dataset that will be used in this project, including VAERS Data, VAERS Symptoms, and VAERS Vaccine.

- **VAERS Data:** The dataset contains detailed information about individual VAER submitted to the VAERS system.
- **VAERS Symptoms:** The dataset contains a list of symptoms and their corresponding codes that are reported in adverse event narratives.
- **VAERS Vaccine:** The dataset provides information about the vaccines including their characteristics and administration.

The datasets spanning from 2010 to the present day will serve as the foundation for creating new named entity recognition packages to identify terms related to symptoms and developing NEL methods to connect these identified terms to standardized terms present in a dictionary. Due to a large dataset with more than 1 million rows, I will select a small sample based on one vaccine type by selecting from the vaccine type with the highest number of reports that is COVID-19 as shown in Figure 7.



COVID19	1009575
VARZOS	108667
FLU3	57602
FLU4	43724
PPV	42530
Name: VAX_TYPE, dtype: int64	

Figure 7: The top 5 vaccine types

Evaluation

The evaluation of the machine learning model's performance is a critical aspect of this project. However, it's essential to note that there is no ground truth annotation available for the data, making evaluation challenging. To overcome this limitation, a combination of automatic and manual evaluations will be employed.

- **Automatic Evaluation:** The model's accuracy in correctly classifying symptoms will be assessed through automatic evaluation metrics. For this case, 50 reports will be selected for evaluation.
- **Manual Evaluation:** To ensure the accuracy of the model's predictions, a sample of clinical notes (20 reports) will be selected for manual evaluation. Experts manually review the results, verifying the correctness of the model's predictions.

The three key metrics, including precision, recall, and F1-score, will serve as essential tools for assessing the model's performance in identifying and linking symptoms.

- **Precision:** Precision will measure the accuracy of identified symptoms, ensuring their trustworthiness and relevance
- **Recall:** Recall will determine the model's ability to capture all relevant symptoms, minimizing the risk of missing significant health indicators
- **F1-score:** F1-score will provide an overall evaluation of the model's effectiveness in both accuracy and completeness.

Automatic Evaluation

For evaluating the effectiveness of our methodology in symptom extraction, an automatic evaluation process will be implemented. A set of 50 reports has been selected for assessment, utilizing precision, recall, and F1-score as evaluation metrics. In this evaluation, the symptoms will be extracted from the text of these 50 reports and compared to standard symptoms using three distinct methods: Exact Matching, Fuzzy Matching, and Similarity Matching. The Similarity Matching method employs cosine similarity to gauge the resemblance between the extracted symptoms and standard symptoms. A crucial criterion for determining correctness is set at a cosine similarity score equal to or greater than 0.8, designating it as a True Positive (correct match). Conversely, a score below 0.8 is considered a False Positive, signifying an incorrect match. To assess False Negatives, the disparity between the number of standard symptoms and the number of correctly extracted symptoms is calculated. Subsequently, precision, recall, and F1-score are calculated using the values of True Positives (TP), False Positives (FP), and False Negatives (FN). These metrics provide a comprehensive understanding of the performance of the symptom extraction methods. The outcomes of this automatic evaluation process are shown in Figure 8, offering a clear depiction of the precision, recall, and F1-score for each of the three matching methods.

Linking methods	Comparison	Precision	Recall	F1 score
Exact matching	Standard	0.179	0.394	0.246
	Top 100 common	0.174	0.387	0.240
Fuzzy matching	Standard	0.179	0.394	0.246
	Top 100 common	0.174	0.387	0.240
Cosine-similarity matching	Standard	0.237	0.462	0.313
	Top 100 common	0.217	0.441	0.291

Figure 8: The results of the automatic evaluation process

Among the three employed methods for symptom extraction and linking, the similarity-based matching method demonstrated superior outcomes. This method, leveraging cosine similarity, proved to be particularly effective in capturing the semantic meaning of symptoms rather than relying solely on the sequence of characters. Unlike exact matching, which demands a precise character-for-character corre-

spondence, and fuzzy matching, which allows for some flexibility but may still be sensitive to textual nuances, the similarity-based matching method offers a more nuanced and context-aware approach.

By considering the semantic similarities between the extracted symptoms and standard symptoms, this method excels in recognizing variations in expression, synonymous terms, and subtle differences in wording. The focus on meaning over strict character alignment contributes to a more accurate and flexible linking process, especially crucial in the context of medical symptomatology where variations in symptom descriptions are common.

The emphasis on semantic understanding aligns well with the intricacies of medical language, making the similarity-based matching method a robust choice for accurately linking identified entities to their standardized counterparts. This nuanced approach not only enhances the precision and efficacy of the entity linking process but also aligns with the complex and varied nature of medical symptom data.

## Manual Evaluation

In addition to the automated evaluation, a manual assessment was conducted to validate the efficacy of our symptom extraction model. A subset of 20 reports was randomly selected for this purpose, and the symptoms were extracted using our developed model. Unlike the automated evaluation, the manual assessment involved a meticulous review of each extraction result by human evaluators. This process aimed to discern the accuracy of the symptom extraction for each report.

During the manual inspection, it became evident that the model exhibited a commendable ability to accurately extract symptoms. However, a noteworthy observation emerged — the model's extraction results sometimes differed from the pre-defined standard symptoms in terms of word choice or sentence structure. This variance in expression, although reflecting accurate symptom identification, contributed to suboptimal outcomes in the automated evaluation process, which heavily relied on exact matching, fuzzy matching, and similarity scores.

This manual evaluation underscored the model's competence in capturing relevant symptoms, shedding light on the limitations of automated assessments that may not fully appreciate the nuanced variations in symptom descriptions. It emphasizes the importance of considering not only the precision of the extraction but also the inherent diversity in language expression, particularly in the domain of medical symptomatology.

During the manual evaluation, a deeper analysis of the 20 randomly selected reports revealed a total of 68 standard symptoms present within this subset. Impressively, the model demonstrated a high degree of accuracy by correctly extracting symptoms for 60 out of the 68 standard symptoms. This equates to an overall correctness rate of 79.41%.

The manual evaluation process not only confirmed the model's proficiency in symptom extraction but also provided a granular understanding of its performance at the individual symptom level.

These findings underscore the model's robust capability to identify and extract symptoms accurately, as evidenced by its success in capturing a substantial majority of the standard symptoms within the evaluated reports. It further emphasizes the significance of considering holistic correctness rates alongside the nuanced variations observed during the manual evaluation, reinforcing the model's reliability in capturing diverse expressions of medical symptoms.

## Conclusion

In conclusion, our comprehensive strategy for extracting symptoms from a dataset of 10,000 reports related to COVID-19 has not only proven successful but has also demonstrated a high level of sophistication. Leveraging advanced techniques such as Named Entity Recognition and the implementation of specialized matching rules, we navigated through the complexities of diverse symptom descriptions.

While the endeavor presented its share of challenges, including prolonged processing times and the inherent variations in how individuals express symptoms, we adeptly addressed these issues. One key mitigation strategy was the initial validation of our method using a smaller subset of reports before scaling up, allowing us to fine-tune and optimize our approach for greater efficiency.

Notably, our methodology exhibited remarkable adaptability and accuracy, even in the face of the diverse ways in which individuals articulated their symptoms. The manual evaluation of our results underscored the reliability of our model, revealing a commendable 79.41% correctness rate. This validation not only reinforces the robustness of our approach but also highlights its potential applicability in the realm of healthcare, particularly in the context of a pandemic like COVID-19.

In summary, our project has not only successfully tackled the challenges associated with symptom extraction but has also set a significant precedent for future endeavors in this domain. The demonstrated methodology holds promise for contributing valuable insights and aiding healthcare professionals in efficiently addressing and managing health-related issues, especially during widespread health crises such as the ongoing pandemic.

## Project management

The project has concluded, and its timeline is as follows:

- **Oct 10 – Oct 15:** Data Preprocessing.
- **Oct 16 – Oct 20:** Midterm Exams (Other courses).

- **Oct 21 – Oct 27:** Extracting Symptom-related Entities (Step 1).
- **Oct 28 – Nov 2:** Link Entities to Standard Symptoms (Step 2).
- **Nov 3 – Nov 6:** Reporting Midterm progress
- **Nov 7 – Nov 12:** Continue to finish Step 2.
- **Nov 13 – Nov 17:** Evaluating the system.
- **Nov 18 – Nov 28:** Preparing presentation material.
- **Nov 29 – Dec 3:** Preparing Final report
- **Dec 4 – Dec 11:** Presentation period
- **Dec 11:** Submitting Final Report

## References

- [1] Qi, P., Zhang, Y., Zhang, Y., Bolton, J., & Manning, D.C. 2020. A Python Natural Language Processing Toolkit for Many Human Languages. The 58<sup>th</sup> Annual Meeting of the Association for Computational Linguistics. (pp. 101-108)
- [2] Zhang, Y., Zhang, Y., Qi, P., Manning, D.C., & Langlotz, P.C. 2021. Biomedical and clinical English model packages for the Stanza Python NLP library. Journal of the American Medical Informatics Association (Vol. 28, Issue 9). (pp. 1892-1899). <https://doi.org/10.1093/jamia/ocab090>