CS584: BIONLP

Ramon L. Correa
[1]**Emory University , Atlanta, Georgia , United States**

Abstract

Monitoring the presense of COVID symptoms has become difficult due to lack of adequate testing and symptom variance. We propose a rule based model for monitoring social media post regarding COVID to observe the incidence of certain COVID related symptoms. We obtain acceptable performance wit ha precision of .72.

Introduction

In late 2019 early 2020 COVID-19 changed how we monitor diseases . The main problem in monitoring COVID was observing the presence of the diseases symptoms. Unlike other diseases patients symptoms would vary greatly on a case by case basis. To better assist in the monitoring of diseases progression amongst communities some would propose monitoring the incidence of several disease symptom's on social media. The main idea being that individuals in social networks would mention their symptoms before even testing positive to the disease. Thus allowing medical professionals to have an awareness for greater disease incidence in certain regions. This would be helpful for regions where social media connectivity is quite high but access to corona virus testing is quite limited. In this work we shall describe how we process social media post and built a rule based system to identify COVID symptoms.

Methods

Text processing

Each input is a multi-line post provided by a reddit user. Meaning this text will be less structured than those seen in other nlp applications. For this reason we attempted to normalize the contents of the file by applying the following steps. We first force all the text to be lowercase. This would allow us to avoid rare cases where certain key symptoms might have uppercase letters or random case pattern such as "heaDache". We then tokenized each of the terms and merged the tokens onto a single string. This was done to remove line endings and also avoid the case of duplicate white space.

Building the lexicon

As part of the assignment we where provided with a series of example post to annotate for symptoms. This allowed us to identify potentially new terms onto our lexicon providing better performance. The lexicon was built as follows. We first loaded the covid twitter lexicon and created a dictionary mapping symptom expression to their corresponding CUI code. We then load the manually generated annotation file to add new terms onto our lexicon.

Identifying Symptoms Terms

Once processing was completed we applied a simple searching strategy applying the regex bellow onto a single sentence.

```
(\b\|\W)({k})(\b\|\W)
```

K is each term in the lexicon we had built in the previous step. The regex is meant to capture individual words and avoid having terms like "ache" match with "headache". It's quite possible the same term may occur in the same sentence requiring multiple searches. The python regex library provides a utility for this through the finditer method which returns a list of matches. The series of matches are then appended to a list of all matches found in the text.

Identifying negations

Negations were built in a similar fashion to the lexicon with the exception that instead of a dictionary we produce a list of negation terms. The regex for these terms is more complicated as we need to have the ability to look ahead 3 terms. The necessary regex was built using multiple regex groups. The first group is in section 1 which matches the negation word. The other component is in section 2 bellow which matches the adjacent terms which will allow us to identify possible negations.

```
1. (\\b{k}\\b)
2. (\s?\W\s?)?(\w*\b)?(\s?\W\s?)?(\w*\b)?(\s?\W\s?)?(\w*\b)
```

Once a negation is identified the series of groups is generated on to a single list resembling [(negation), (term a), (term b), (term c)]and so forth. Should there be a punctuation mark we prune the list containing the punctuation and terms to the right of it. Once We have the list of all possible terms we iterate through the list of symptom matches and check if their term match occurs within the term match range of the negation. Should that be true the term is marked as negated.

1 Results

Once the terms are identified and converted to CUIs we stored the data as an xlsx file. We then run the analysis scripts provided by the course instructor

2 Results

Table 1: Final Performance obtained when comparing our systems classification to that of the gold standard

Metric	Count
Precision	0.7225
Recall	0.6509
F1 Score	0.6848

Table 2: Initial model performance obtained when comparing our systems classification to that of the gold standard

Metric	Count
Precision	0.7158
Recall	0.6415
F1 Score	0.6766

discussion

One glaring observation was the imbalance seen between the precision and recall scores. Upon further inspection we noted that several of the terms being misclassified were ”general” terms that may perhaps have multiple mappings. As an example in our system the word ”congestion” used to be applied to the symptom ” chest pain” according to the original lexicon but in the gold standard it appears to apply to nasal congestion CUIs. Once we modified the dictionary we saw an increase in performance. Prompting the discussion on mapping colloquial terminology to symptoms. There may be other ambiguous terminology who’s colloquial use may not be the same as used in the lexicon leading to errors. For our submission we only utilized the congestion modification as it’s most likely other reddit users will use congestion in the same way. We aim to explore the use of colloquial terminology in future projects.