Wearable Cognitive Assistant Systems for Emergency Response

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

Michael Tang
Spring, 2020

Technical Project Team Members
Sarah Preum
M. Arif Rahman
Homa Alemzadeh

On my honor as a University Student, I have neither given nor received
unauthorized aid on this assignment as defined by the Honor Guidelines
for Thesis-Related Assignments

Wearable Cognitive Assistant Systems for Emergency Response

Overview

This paper details the work and progress I have done over the Fall 2019 semester with regards to the negation detection for the EMS pipeline. I will first discuss the challenges being solved and previous attempts. Then, I will talk about the current approach, dataset, evaluation metrics, and evaluation techniques. Finally, I will go over the current results, any challenges I encountered during this semester, and work I will be doing next semester for negation detection.

Challenges

Detecting negations in a medical context is a very unique challenge for two reasons. Negations in a medical context are different from those in normal, conversational English because of where they are placed in relation to the verb. In a medical context, the verb itself is often a negative verb, whereas in conversational English, the verb is usually accompanied by a negative word. For example, the phrase "patient denies taking medication" has the negative verb "denies," while in the phrase "patient does not take medication," the word "not" negates the verb "take." In many cases, sentences in a medical context take on the form with the negative verb. This makes negation detection in a medical context difficult, as searching for negative words in a sentence such as "no" or "not" will not yield all the negated concepts. In addition, only the concepts that were already determined to be of interest should be included in the list of detected negations. For example, in the sentence "the patient said they do not feel good, and denies chest pain," only the concept "chest pain" should be returned as negated. In order for a program to be good at negation detection in a medical context, it needs to address both of these challenges.

Previous attempts

After determining the challenges related to negation detection in a medical context, I quickly realized that I would not be able to use several pre-existing natural language processing tools effectively. For example, the Stanford CoreNLP and spaCy libraries would not be able to detect many negations, as their English models were trained on general conversational English rather than English in a medical context. In addition, for each of their detected list of negations, a substring check would have to be applied to each list, in order to figure out which negations were actually relevant to the pipeline. However, I did find a model used for spaCy trained specifically on biomedical text, called scispaCy. I tried using this model with spaCy on a transcript, and got the results shown in Table 1.

Table 1. *Result for one of the transcripts from the scispaCy model.*

Transcript	**Ground truth negated concepts**	**Detected negations**
9.txt-> I went to a call for a 60y/o choking (didn't know he was choking), and I couldn't feel a pulse. No carotid. No radial. No nothing. I looked at him when I entered the room, and was thought "yup.... he's dead" then confirmed it when I couldn't find a pulse. I moved his arm off his chest as to start compressions, and noticed it was rigid. Then the unthinkable happened.... he put his arm back to his chest! I redirected my focus and started bagging his stoma, and my partner snapped on an Sp02 monitor to find a waveform and a readout. couple breaths in he started getting some colour back and somewhere over the next while we pulled out an obstruction. Full recovery No CPR	radial carotid pulse	know [went, (, did, n't, choking, ,, and, feel] feel [I, could, n't, pulse, .] carotid [No,] radial [carotid, No,] nothing [radial, No, .] find [when, I, could, n't, pulse] CPR [No]

In Table 1, the original transcript is shown in the first column, the list of negated ground truth negations are in the middle column, and the list of detected negated columns are in the third column. As you can see, scispaCy detected the concepts “radial” and “carotid,” but failed to detect “pulse” because the word “pulse” was not being negated, rather “feel” was being negated. In addition, scispaCy detected several useless negations in the context of our pipeline, such as “know” in the phrase “didn’t know he was...”), “nothing” in the phrase “No nothing”, and “find” in the phrase “couldn’t find.”

The results from scispaCy highlight a problem for our use case not only with natural language processing libraries such as Stanford CoreNLP and spaCy, but also for machine learning based approaches in general. In order to create a machine learning model for our use case that can accurately and precisely detect negations in a medical context, there would have to be an extremely large amount of correctly annotated EMS transcripts to be used as training and testing data. Although the number of transcripts and annotations done by Haydon and Eimara is large, it is not enough to create a machine learning model that would be accurate, precise, and generic enough.

Current approach

My current approach uses a user-extendable program called pyConTextNLP. This program uses a list of around 400 known regular expressions for different English modifiers commonly used in a medical context to determine whether a given sentence contains a modifier describing a concept or not.

```
Comments: 3/22/2013
Direction: forward
Lex: mimicking
Regex: ''
Type: PROBABLE_NEGATED_EXISTENCE
---
Comments: ''
Direction: forward
Lex: must be ruled out for
Regex: ''
Type: DEFINITE_NEGATED_EXISTENCE
---
Comments: ''
Direction: forward
Lex: negative examination for
Regex: negative (examination|study|exam|evaluation) for
Type: DEFINITE_NEGATED_EXISTENCE
```

Figure 1. Examples of negation regular expressions used in pyConTextNLP.

Some of these regular expressions are shown in Figure 1. In Figure 1, each block separated by three dashes represents a YAML document, describing the type of and the regular expression for the modifier. pyConTextNLP uses regular expressions to detect modifiers in a text, and my program uses the type of the modifier to determine whether it is negative or not.

```
Comments: ''
Direction: ''
Lex: cool skin
Regex: (diphoretic|cool skin)
Type: COOL_SKIN
---
Comments: ''
Direction: ''
Lex: cough
Regex: (dbc|cough|coughing|barky cough)
Type: COUGH
---
Comments: ''
Direction: ''
Lex: decreased mental status
Regex: (appears lethargic|anox4|AY&Ox4|aox2|found altered|alert to verbal|caox2|avpu|ayox4|slumped
  over|not alert|caox3|caox1|gcs|cx|ams|caox4|AMS|AOX4|aox4|Aox2|aox1|A&O x 1|A&Ox4|GCS|acting
  appropriately|responsive to verbal|post ictal|A&Ox2 basline|ANOx4|lethargic|slow
  responsiveness|alert|semiconscious|unresponsive|aox3|responsive to pain|acting out|CAox4)
Type: DECREASED_MENTAL_STATUS
```

Figure 2. Examples of user-defined concepts used by pyConTextNLP.

If a given sentence contains a modifier, pyConTextNLP looks at the rest of the sentence to determine all the subjects that are related to that modifier. I am able to provide pyConTextNLP with a list of all the medical subjects of interest, such as “pulse rate,” “headache,” or “chest pain.” Some of these subjects are shown in Figure 2. The program uses this list to determine whether a negation pertains to a medical concept or not, thereby eliminating unwanted non-medical negations. In order to create this list of relevant concepts, I used all the concepts from the required concepts list, located in the Google Sheet titled “ConceptExtracEval_ODEMSA.xlsx” containing all the annotations done by Haydon and Eimara.

cool skin	cool skin	
cool skin	diphoretic	
cough	barky cough	
cough	cough	
cough	coughing	
decreased mental status	A&Ox2 basline	
decreased mental status	A&Ox4	
decreased mental status	acting appropriately	

Figure 3. Example from the EMS ontology provided by Sarah.

To create regular expressions for each of these concepts, I used the EMS ontology used for protocol modeling in the pipeline, provided by Sarah. The ontology contains a list of relevant concepts, along with the various different forms it could appear as in text. An excerpt from this ontology is shown in Figure 3. I wrote a program that is able to read in these different forms for each concept, and convert them into regex statements for each concept as shown in Figure 2. So in the future, if the ontology is ever updated, it will be very easy for me to export the ontology to regular expressions for pyConTextNLP to use.

Dataset

In the beginning of the semester, when I was still trying out different tools to detect negations, I needed a simple way to ensure that whatever code I was writing was working properly. To accomplish this, I was generating simple sentences containing some negated concepts, and seeing if my code would find those concepts and output them. However, with Sarah's guidance, I was able to develop a more robust and automated dataset and testing plan. The dataset is composed of a list of EMS transcripts, with concepts in the transcripts annotated by people with EMS training, such as Haydon and Eimara.

Narratives	Being Done by	ALS/BLS	Concepts
Dispatched for seizures Aos to find 59 yof laying in the recovery position on a bed next to the front door of a single family row house. Chief complaint: chest pain Hx: Pt stated that she was admitted to the ICU 2 weeks ago for a massive GI	Michael	BLS treatment	(chest pain,True,chest pain) (breath,True,breathing rate) (respiratory rate,True,breathing rate) (seizure,True,seizures) (fever,True,feverish) (abdominal pain, true, abdominal
D - Dispatched P1 to 6033 LAMAR DR for sick person. Notes state that patient recently had an appendectomy and now her abdomen hurts. A - Arrived on scene to find a female patient sitting at the edge of her bed. Patient is alert and	Michael	BLS treatment	(chest pain,False,chest pain, denies) (shortness of breath, False,shortness of breath, no) (abdominal pain, true, abdominal pain)
D: to a street intersection for a MVA ATF: 3 vehicals with moderate to heavey damage RFD getting extricaton tools out RPD (417) on scene this report is in ref to an occupent of the mini van that had 2 front seat occupants report on	Michael	BLS treatment	(shortness of breath,False,sob, no) (trauma,False,trauma, no)
D: Medic 568 dispatched priority 1 for seizures at 622 N 33rd st Apt A with Engine 11. Unit responding lights and sirens without incident. A: Arrived on scene at the above address to find a 5 y/o AA F laying on her R side on the couch. C:	Michael	ALS - Seizure	(seizure,True,seizure)

Figure 4. Part of the dataset annotated by Haydon.

Figure 4 shows part of the list of transcripts that were annotated by Haydon used as part of the dataset. Each of the transcripts has a corresponding concepts column, containing all relevant concepts described within the transcript. A concept entry in the dataset can take on one of two forms: if the concept being described is not negated in the transcript, then it has the form:

(concept, True, words in transcript that describe concept)

Otherwise, if the concept is being negated, then it has the form:

(concept, False, words in transcript that describe concept,

words in transcript that negate concept)

For example, if we had the sentence "patient denies cx pain," then the corresponding concept entry for "chest pain" would be:

(chest pain, False, cx pain, denies)

In order to make sure this form of concept entries was standardized throughout the transcripts Haydon annotated, Arif, and Sarah, and I went through all the transcripts and added the words in transcripts that negated concepts to concepts that were negated. In addition, we removed any incorrect concepts from the concepts column, and also moved any inferred concepts (concepts that were not directly mentioned in the transcript) into a separate column.

Right now, the test data is comprised of 90 of these transcripts, as that is how many Sarah, Arif and I have cleaned from Haydon's transcripts. However, in the future, we intend to clean more transcripts so that we have more data. In addition, I intend to use the transcripts Eimara has annotated for even more testing data.

In addition to the list of transcripts, the EMS ontology used to generate regular expressions for concepts can also be considered part of the dataset. How this ontology was used is described in the previous section "Current approach."

Evaluation metrics and methods

The metric that is used for evaluation is average recall. When pyConTextNLP is run on a transcript, it outputs a list of negated concepts in the transcript. Each transcript in the testing data also has a list of negated ground truth concepts. Therefore, recall for a transcript can be determined by computing:

$$\frac{number\ of\ detected\ ground\ truth\ concepts}{total\ number\ of\ ground\ truth\ concepts}$$

Average recall can then be calculated by averaging all recall values from each individual transcript.

Current results

My program currently achieves an average recall of 0.862 when run on the current dataset of 90 transcripts. Out of 234 total ground truth concepts throughout all transcripts, 209 were detected. This is very promising so far, and I hope this recall will be similar when I clean and include more transcripts into my dataset.

Table 2. Program output for one transcript.

Detected negated concepts	Ground truth negated concepts	Recall for this transcript
{'nausea', 'pain', 'vomiting', 'tenderness', 'trauma', 'shortnessofbreath', 'diarrhea', 'wheezing', 'chestpain', 'respiratoryrate', 'lossofconsciousness'}	['pain', 'chestpain', 'shortnessofbreath', 'lossofconsciousness', 'tenderness', 'nausea', 'vomiting', 'diarrhea', 'trauma']	1.000

Table 2 shows the output of the program for a single transcript. In this table, the left column is the list of negated concepts the program detected, the middle column is the list of ground truth negated concepts, and the right column is the recall for this transcript.

```
Transcript 87:
Detected negated concepts:
{'nausea', 'vomiting', 'pale', 'tenderness', 'trauma', 'shortnessofbreath', 'cough', 'diarrhea'
Expected negated concepts:
['lossofconsciousness', 'chestpain', 'diarrhea', 'nausea', 'vomiting', 'tenderness']
Number of negated concepts detected / Total negated concepts: 6/6



Transcript 88:
Detected negated concepts:
{'trauma', 'fever', 'headache', 'bloodpressure', 'malaise', 'respiratoryrate'}
Expected negated concepts:
['fever', 'hyperthermia', 'trauma']
Number of negated concepts detected / Total negated concepts: 2/3



###################################################
Total number of negated concepts detected / All negated concepts: 209/234
Average recall: 0.8622594997594997

Process finished with exit code 0
```

Figure 5. Actual program output for 90 annotated transcripts in the current test data.

Figure 5 shows a screenshot of the current program's output. For each individual transcript, the program outputs the detected negated concepts, the expected list of negated concepts, and the recall for the transcript. Finally, at the end, the program prints out the average recall over all the transcripts.

Challenges during the past semester

One challenge I encountered while looking at the dataset was that some pairs of concepts in the list of required concepts were related to each other, and appeared together in the ground truth list of concepts for transcripts when certain phrases appeared in text. These pairs of concepts include "breath" and "shortness of breath," "lightheadedness" and "dizziness," "blood pressure" and "hypertension," and "hyperthermia" and "fever." For example, for the sentence:

“patient denies any SOB associated w/ pain.”

The ground truth annotations by Haydon included both of the following:

(breath, False, sob)

(shortness of breath, False, sob)

This was a recurring problem every time a word related to these different pairs of concepts appeared in a transcript. Because pyConTextNLP would only detect the concept “shortness of breath” from the phrase “sob,” it would fail to detect the “breath” concept, and thus the recall for that transcript would go down. I believe this is a problem with the ground truth data rather than pyConTextNLP, since when the phrase “sob” appears, I think only the concept “shortness of breath” should be detected as it is the most relevant. Thus, much of the ground truth data still needs to be cleaned to account for these related pairs of concepts.

Another challenge that I ran into while implementing my program using pyConTextNLP had to do with negations that do not use any negative phrases or modifiers. For example, in the sentence “patient is warm and dry,” the concept “pale” should be detected as negated. However, no word in the sentence indicates negation. Right now, I believe I may have a solution to this problem that is discussed in the section below of this report, however I have not actually implemented it yet due to time constraints.

Future work next semester

The biggest goal I have for next semester is to work with Sarah and Elizabeth on adding my negation detection program into the rest of the EMS pipeline. Last week, I met with

Elizabeth and got access to the Github repository for the EMS pipeline. Since the pipeline is run on a Linux system, it should be relatively trivial to add what I have to the pipeline.

Another goal I have is to add more transcripts and annotations to my existing dataset. I will be working with Eimara to include her annotations into the dataset, and then I will be cleaning more of Haydon's annotations to add as well.

Finally, I want to implement a way to detect negations that do not use negative words. My idea on how to do this is to introduce more concepts. The new concepts would effectively be the opposites of the concepts I have right now. For example, since we have a concept called "loss of consciousness", we would introduce a new concept "consciousness." The sentence "patient is conscious" means that "loss of consciousness" should be false, but we can't detect the negation because there are no negative words. But, since we can detect the "conscious" concept, we can infer that "loss of consciousness" should be negated. Another example is the sentence "patient is warm and dry." In this sentence, the "pale" concept is false. If we introduce a concept called "not pale," we can relate the words "warm" and "dry" to it. So, when we detect the words "warm" and "dry," we can infer that the "pale" concept has been negated. These additional opposite concepts would only be used to detect negations for our existing concepts, and would not serve as additional concepts that need to be accounted for. I can do this by setting a "type" for them in the YAML file containing the concepts and their regular expressions, or by adding a certain substring in the concept name. When they are detected, I should return the opposite concept as being negated instead.