ANLP Project

Andrés Lamilla

Problem to solve

Given a word in a text detect if it is negate:

word: gallops

text: No murmurs, GALLOPS, or rubs.

Or not:

word: vomiting

text: The patient was admitted on **DATE[Sep 25 2007], complaining of nausea and VOMITING.

Using negEx as a tagger

- [**PREN**] - Prenegation rule tag
- [**POST**] - Postnegation rule tag
- [**PREP**] - Pre possible negation tag
- [**POSP**] - Post possible negation tag
- [**PSEU**] - Pseudo negation tag
- [**CONJ**] - Conjunction tag
- [**PHRASE**] - Term is recognized from the term list
- [**NEGATED**] - Term was recognized from term list, and it was found being negated

Using negEx as a tagger

getNegTaggedSentence:

- **sentence1** = '[**PREN**]no[**PREN**] murmurs, [**NEGATED**]GALLOPS [**NEGATED**], or rubs.'

- **sentence2** = 'The patient was admitted on **DATE[Sep 25 2007], complaining of nausea and [**PHRASE**]VOMITING[**PHRASE**].'

- **sentence3** = 'His [**PHRASE**]NAUSEA[**PHRASE**] and vomiting [**PREN**]resolved[**PREN**] .'

reTag

- **sentence1** = '**PREN** murmurs, **NEGATED**, or rubs.'

- **sentence2** = 'The patient was admitted on **DATE[Sep 25 2007], complaining of nausea and **PHRASE**.'

- **sentence3** = 'His **PHRASE** and vomiting **PREN** .'

reTag

Replace **NEGATED** by **PHRASE**:

- **sentence1** = '**PREN** murmurs, **PHRASE**, or rubs.'

- **sentence2** = 'The patient was admitted on **DATE[Sep 25 2007], complaining of nausea and **PHRASE**.'

- **sentence3** = 'His **PHRASE** and vomiting **PREN** .'

reTag

Use additional may be useful tags:

- **POINT**: .
- **COMMA**: ,
- **OR**: or
- **AND**: and

reTag

- **sentence1** = '**PREN** murmurs **COMMA PHRASE COMMA OR** rubs **POINT**'

- **sentence2** = 'The patient was admitted on **DATE[Sep 25 2007] **COMMA** complaining of nausea **AND PHRASE POINT**'

- **sentence3** = 'His **PHRASE AND** vomiting **PREN POINT**'

reTag

Replace all non tagged words by **WORDS**

- **sentence1 = 'PREN WORDS COMMA PHRASE COMMA OR WORDS POINT'**
- **sentence2 = 'WORDS WORDS WORDS WORDS WORDS WORDS COMMA WORDS WORDS WORDS AND PHRASE POINT'**
- **sentence3 = 'WORDS PHRASE AND WORDS PREN POINT'**

reTag

Remove consecutive **WORDS**

- **sentence1 = 'PREN WORDS COMMA PHRASE COMMA OR WORDS POINT'**
- **sentence2 = 'WORDS COMMA WORDS AND PHRASE POINT'**
- **sentence3 = 'WORDS PHRASE AND WORDS PREN POINT'**

reTag

And finally split the sentence in pre PHRASE and post PHRASE

- **sentence1**
    - pre: **COMMA WORDS PREN**
    - post: **COMMA OR WORDS POINT**
- **sentence2**
    - pre: **AND WORDS COMMA WORDS**
    - post: **POINT**
- **sentence3**
    - pre: **WORDS**
    - post: **AND WORDS PREN POINT**

Features

- **Feature1**: The previous and next tag to the PHRASE
- **Feature2**: The two previous and next tags to the PHRASE
- **Feature3**: The three previous and next tags to the PHRASE
- **Feature4**: If there was a PREN tag before PHRASE and there wasn’t a POINT between them.

Features

sentence1

- **pre**: COMMA WORDS PREN
- **post**: COMMA OR WORDS POINT

Feature1:

- **pre:** COMMA
- **post:** COMMA

Feature2:

- **pre:** COMMA WORDS
- **post:** COMMA OR

Feature3:

- **pre:** COMMA WORDS PREN
- **post:** COMMA OR WORDS

Feature4: True

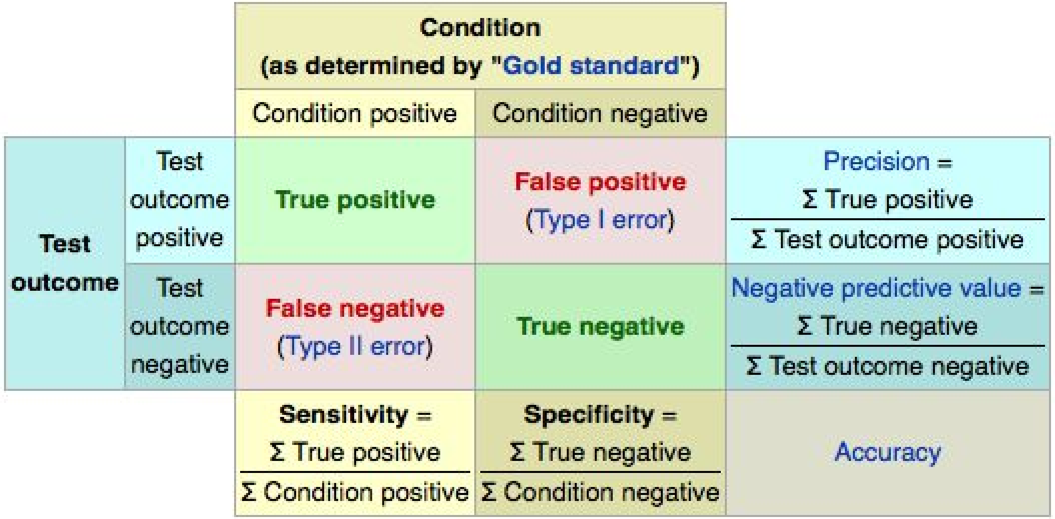
Naive Bayes

Then just count words and be happy :)

$$p(C, F_1, \dots, F_n) = p(C) \; p(F_1|C) \; p(F_2|C) \; p(F_3|C) \; \cdots$$

$$= p(C) \prod_{i=1}^{n} p(F_i|C).$$

Testing performance



		Condition (as determined by "Gold standard")		
		Condition positive	Condition negative	
Test outcome	Test outcome positive	**True positive**	**False positive** (Type I error)	Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$
	Test outcome negative	**False negative** (Type II error)	**True negative**	Negative predictive value = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$
		Sensitivity = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	**Specificity** = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Accuracy

Testing performance

Training set (2115):

- Results:
  - Correct: 2071
  - Negative Correct: 426
  - Negative Incorrect: 20
  - Positive Correct: 1645
  - Positive Incorrect: 24
- NegEx results:
  - Correct: 2056
  - Negative Correct: 406
  - Negative Incorrect: 40
  - Positive Correct: 1650
  - Positive Incorrect: 19

Testing performance

Training set

- Results:
  - sensitivity: 1645/(1645+20) = 0.9879
  - specificity: 426/(426+24) = 0.9466
  - precision(PPV): 1645/(1645+24) = 0.9856
  - NPV: 426/(426+20) = 0.9551
  - accuracy: 2071/2115 = 0.9791
- NegEx results:
  - sensitivity: 1650/(1650+40) = 0.9763
  - specificity: 406/(406+19) = 0.9552
  - precision(PPV): 1650/(1650+19) = 0.9886
  - NPV: 406/(406+40) = 0.9103
  - accuracy: 2056/2115 = 0.9721

Testing performance

Testing set (235)

- Results:
  - sensitivity: 188/(188+0) = 1
  - specificity: 44/(44+3) = 0.9361
  - precision(PPV): 188/(188+3) = 0.9842
  - NPV: 44/(44+0) = 1
  - accuracy: 232/235 = 0.9872
- NegEx results:
  - sensitivity: 187/(187+2) = 0.9894
  - specificity: 42/(42+4) = 0.913
  - precision(PPV): 187/(187+4) = 0.979
  - NPV: 42/(42+2) = 0.9545
  - accuracy: 229/235 = 0.9744

Improvements

- Add more features
- Add more useful tags