**COMP 6751 Natural Language Analysis**

**Project 3 Report 2 (Demo)**

Student: Yixuan Li 40079830

Table of Contents

[I. Input and Outputs files and patterns 2](#_Toc56781603)

[II. Input and Outputs presentation 3](#_Toc56781604)

[1. Positive test case 1 3](#_Toc56781605)

[2. Positive test case 2 3](#_Toc56781606)

[3. Positive test case 3 4](#_Toc56781607)

[4. Positive test case 4 4](#_Toc56781608)

[5. Positive test case 5 5](#_Toc56781609)

[**Positive test case 5 result explanation:** 6](#_Toc56781610)

[6. Positive test case 6 7](#_Toc56781611)

[7. Positive test case 7 9](#_Toc56781612)

[8. Positive test case 8 9](#_Toc56781613)

[9. Positive test case 9 9](#_Toc56781614)

[10. Negative test case 1 10](#_Toc56781615)

[11. Negative test case 2 11](#_Toc56781616)

[12. Negative test case 3 11](#_Toc56781617)

[13. Negative test case 4 12](#_Toc56781618)

[14. Negative test case 5 12](#_Toc56781619)

[**Negative test case 5 result explanation:** 13](#_Toc56781620)

[15. Negative test case 6 14](#_Toc56781621)

[**Negative test case 6 result explanation:** 15](#_Toc56781622)

[16. Negative test case 7 16](#_Toc56781623)

[17. Negative test case 8 16](#_Toc56781624)

[18. Negative test case 9 17](#_Toc56781625)

[19. Neutral test case 1 17](#_Toc56781626)

[20. Neutral test case 2 18](#_Toc56781627)

[21. Neutral test case 3 18](#_Toc56781628)

*Expectations of originality:*

*I, student 40079830, certify that this submission is my original work and meets the Faculty’s Expectations of Originality.*

*Date: November 20, 2020*

# **I. Input and Outputs files and patterns**

The positive input sentences are saved in *“data/positive.txt”*;

The negative input sentences are saved in *“data/negative.txt”*;

The neutral input sentences are saved in *“data/neutral.txt”*.

The sentences with correct labels are outputted to *“saved\_results/Good.txt”*;

The sentences with incorrect labels are outputted to *“saved\_results/False.txt”*.

The output pattern is shown as below

Input Sentence

Correct label | [ Prediction label(s) ]

parse tree(s)

Example:

It was too long but entertaining .  
positive | [positive]  
  
(S[-INV, SENTI='positive']  
 (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] It))  
 (VP[NUM='sg', PERSON=?p, SENTI='positive', TENSE='past']  
 (V[+AUX, NUM='sg', TENSE='past', -passive] was)  
 (ADJP[SENTI='positive']  
 (RB[SENTI='neutral', -negation] too)  
 (ADJP[SENTI='positive']  
 (JJ[SENTI='neutral'] long)  
 (CC[-and, +but, -or] but)  
 (JJ[SENTI='positive'] entertaining)))))

Comment:

If there are more than 1 sentiment predictions, then one parse tree with each sentiment prediction will be printed, and the order of parse trees will be same as the order of sentiment predictions.

And I consider that the sentence sentiment is labeled correctly if its correct sentiment is included in the sentiment prediction(s).

# **II. Input and Outputs presentation**

## **Positive test case 1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/positive.txt** | it's a compelling story . positive | [positive] | Console Output |
| Input sentence | it's a compelling story . | (S[-INV, SENTI='positive']  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] it))  (VP[NUM='sg', PERSON=3, SENTI='positive', TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive] 's)  (NP[NUM='sg', PERSON=?p, SENTI=?s] (DT[NUM='sg'] a))  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] compelling)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] story))))) | Parse Tree(s) |

## **Positive test case 2**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/positive.txt** | it has low impact but it's a compelling story . positive | [positive] | Console Output |
| Input sentence | it has low impact but it's a compelling story . | (S[-INV, SENTI='positive']  (S[-INV, SENTI='neutral']  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] it))  (VP[NUM='sg', PERSON=3, SENTI='neutral', TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive] has)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (JJ[SENTI='neutral'] low)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] impact)))))  (CC[-and, +but, -or] but)  (S[-INV, SENTI='positive']  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] it))  (VP[NUM='sg', PERSON=3, SENTI='positive', TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive] 's)  (NP[NUM='sg', PERSON=?p, SENTI=?s] (DT[NUM='sg'] a))  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] compelling)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] story)))))) | Parse Tree(s) |

## **Positive test case 3**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/positive.txt** | it has gut-wrenching impact and it is a compelling story . positive | [positive] | Console Output |
| Input sentence | it has gut-wrenching impact and it is a compelling story . | (S[-INV, SENTI='positive']  (S[-INV, SENTI='neutral']  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] it))  (VP[NUM='sg', PERSON=3, SENTI='neutral', TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive] has)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (JJ[SENTI='neutral'] gut-wrenching)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] impact)))))  (CC[+and, -but, -or] and)  (S[-INV, SENTI='positive']  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] it))  (VP[NUM='sg', PERSON=3, SENTI='positive', TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive] is)  (NP[NUM='sg', PERSON=?p, SENTI=?s] (DT[NUM='sg'] a))  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] compelling)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] story)))))) | Parse Tree(s) |

## **Positive test case 4**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/positive.txt** | this does not have gut-wrenching impact but it's a compelling story . positive | [positive] | Console Output |
| Input sentence | this does not have gut-wrenching impact but it's a compelling story . | (S[-INV, SENTI='positive']  (S[-INV, SENTI=?s]  (NP[NUM='sg', PERSON=?p, SENTI=?s] (DT[NUM='sg'] this))  (VP[NUM='sg', PERSON=3, SENTI=?s, TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive] does)  (RB[+negation] not)  (VP[NUM='sg', PERSON=?p, SENTI='neutral', TENSE='inf']  (V[-AUX, SUBCAT='trans', TENSE='inf', -passive] have)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (JJ[SENTI='neutral'] gut-wrenching)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] impact))))))  (CC[-and, +but, -or] but)  (S[-INV, SENTI='positive']  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] it))  (VP[NUM='sg', PERSON=3, SENTI='positive', TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive] 's)  (NP[NUM='sg', PERSON=?p, SENTI=?s] (DT[NUM='sg'] a))  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] compelling)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] story)))))) | Parse Tree(s) |

## **Positive test case 5**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/positive.txt** | this compelling story with gut-wrenching impact . positive | [positive, neutral] | Console Output |
| Input sentence | this compelling story with gut-wrenching impact . | (S[-INV, SENTI='positive']  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (DT[NUM='sg'] this)  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] compelling)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] story)))  (PP[SENTI='neutral']  (IN[-of] with)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (JJ[SENTI='neutral'] gut-wrenching)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] impact)))))) (S[-INV, SENTI='neutral']  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (DT[NUM='sg'] this)  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] compelling)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] story)))  (PP[SENTI='neutral']  (IN[-of] with)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (JJ[SENTI='neutral'] gut-wrenching)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] impact)))))) | Parse Tree(s) |

### **Positive test case 5 result explanation:**

1. **Sentiment label**

There is 2 sentiment prediction outputs: ***positive and neutral***. The reason is similar as the 2nd limitation described in the main report 1. The sentiment of a NP containing a “with” can be determined by the part before “with” or the part after “with”. I will make 2 examples below.

|  |  |  |  |
| --- | --- | --- | --- |
| **NP sentiment** | **the part before “with”** |  | **the part after “with”** |
| positive | a compelling story *[positive]* | with | common impact *[neutral]* |
| positive | a story *[neutral]* | compelling impact *[positive]* |

1. **Parse tree**

The parse trees are outputted 1 for positive sentiment and 1 for neutral sentiment. And the order of parse trees are same as the order of sentiment predictions.

|  |  |
| --- | --- |
| **Parse tree with positive sentiment** | (S[-INV, **SENTI='positive'**]  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (DT[NUM='sg'] this)  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] compelling)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] story)))  (PP[SENTI='neutral']  (IN[-of] with)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (JJ[SENTI='neutral'] gut-wrenching)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] impact)))))) |
| **Parse tree with neutral sentiment** | (S[-INV, **SENTI='neutral'**]  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (DT[NUM='sg'] this)  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] compelling)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] story)))  (PP[SENTI='neutral']  (IN[-of] with)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (JJ[SENTI='neutral'] gut-wrenching)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] impact)))))) |

## **Positive test case 6**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/positive.txt** | this may not have the dramatic gut-wrenching impact of other holocaust films , but it's a compelling story , mainly because of the way it's told by the people who were there . positive | [positive] | Console Output |
| Input sentence | this may not have the dramatic gut-wrenching impact of other holocaust films , but it's a compelling story , mainly because of the way it's told by the people who were there . | (S[-INV, SENTI='positive']  (S[-INV, SENTI=?s]  (NP[NUM='sg', PERSON=?p, SENTI=?s] (DT[NUM='sg'] this))  (VP[NUM='sg', PERSON=?p, SENTI=?s, TENSE='inf']  (V[+AUX, TENSE='inf', -passive] may)  (RB[+negation] not)  (VP[NUM='sg', PERSON=?p, SENTI='neutral', TENSE='inf']  (V[-AUX, SUBCAT='trans', TENSE='inf', -passive] have)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (DT[NUM='sg'] the)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (JJ[SENTI='neutral'] dramatic)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (JJ[SENTI='neutral'] gut-wrenching)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] impact)))))  (IN[+of] of)  (NP[NUM='pl', PERSON=?p, SENTI='neutral']  (JJ[SENTI='neutral'] other)  (NP[NUM='pl', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='negative'] holocaust)  (NP[NUM='pl', PERSON=?p, SENTI='neutral']  (N[NUM='pl', SENTI='neutral'] films))))))))  (COMMA[] ,)  (CC[-and, +but, -or] but)  (S[-INV, SENTI='positive']  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] it))  (VP[NUM='sg', PERSON=3, SENTI='positive', TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive] 's)  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (DT[NUM='sg'] a)  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] compelling)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] story))))  (COMMA[] ,)  (PP[SENTI='neutral']  (RB[SENTI='neutral', -negation] mainly)  (IN[-of] because)  (IN[+of] of)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (DT[NUM='sg'] the)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] way)))  (SBAR[SENTI='neutral']  (S[-INV, SENTI=?s]  (NP[NUM='sg', PERSON=3, SENTI=?s]  (PRP[NUM='sg', PERSON=3] it))  (VP[NUM='sg', PERSON=3, SENTI=?s, TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive]  's)  (V[-AUX, SUBCAT='trans', +passive] told)  (PP[SENTI=?s]  (IN[-of] by)  (NP[NUM='sg', PERSON=?p, SENTI=?s]  (DT[NUM='sg'] the)))))  (SBAR[SENTI='neutral']  (S[-INV, SENTI='neutral']  (NP[NUM='pl', PERSON=?p, SENTI='neutral']  (N[NUM='pl', SENTI='neutral'] people)))  (SBAR[SENTI=?s]  (WP[+wh] who)  (V[+AUX, NUM='pl', TENSE='past', -passive] were)  (EX[] there))))))))) | Parse Tree(s) |

## **Positive test case 7**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/positive.txt** | a perfect example . positive | [positive] | Console Output |
| Input sentence | a perfect example . | (S[-INV, SENTI='positive']  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (DT[NUM='sg'] a)  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] perfect)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] example))))) | Parse Tree(s) |

## **Positive test case 8**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/positive.txt** | well-intentioned movie making . positive | [positive] | Console Output |
| Input sentence | well-intentioned movie making | (S[-INV, SENTI='positive']  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] well-intentioned)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] movie)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] making))))) | Parse Tree(s) |

## **Positive test case 9**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/positive.txt** | It was too long but entertaining . positive | [positive] | Console Output |
| Input sentence | It was too long but entertaining . | (S[-INV, SENTI='positive']  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] It))  (VP[NUM='sg', PERSON=?p, SENTI='positive', TENSE='past']  (V[+AUX, NUM='sg', TENSE='past', -passive] was)  (ADJP[SENTI='positive']  (RB[SENTI='neutral', -negation] too)  (ADJP[SENTI='positive']  (JJ[SENTI='neutral'] long)  (CC[-and, +but, -or] but)  (JJ[SENTI='positive'] entertaining))))) | Parse Tree(s) |

## **Negative test case 1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/negative.txt** | it's a compelling story , but it has low impact . negative | [negative] | Console Output |
| Input sentence | it's a compelling story , but it has low impact . | (S[-INV, SENTI='negative']  (S[-INV, SENTI='positive']  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] it))  (VP[NUM='sg', PERSON=3, SENTI='positive', TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive] 's)  (NP[NUM='sg', PERSON=?p, SENTI=?s] (DT[NUM='sg'] a))  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] compelling)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] story)))))  (COMMA[] ,)  (CC[-and, +but, -or] but)  (S[-INV, SENTI='neutral']  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] it))  (VP[NUM='sg', PERSON=3, SENTI='neutral', TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive] has)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (JJ[SENTI='neutral'] low)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] impact)))))) | Parse Tree(s) |

## **Negative test case 2**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/negative.txt** | manipulative movie making . negative | [negative] | Console Output |
| Input sentence | manipulative movie making . | (S[-INV, SENTI='negative']  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (JJ[SENTI='negative'] manipulative)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] movie)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] making))))) | Parse Tree(s) |

## **Negative test case 3**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/negative.txt** | shamelessly manipulative movie making . negative | [negative] | Console Output |
| Input sentence | shamelessly manipulative movie making . | (S[-INV, SENTI='negative']  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (ADJP[SENTI='negative']  (RB[SENTI='negative', -negation] shamelessly)  (JJ[SENTI='negative'] manipulative))  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] movie)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] making))))) | Parse Tree(s) |

## **Negative test case 4**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/negative.txt** | well-intentioned but manipulative movie making . negative | [negative] | Console Output |
| Input sentence | well-intentioned but manipulative movie making . | (S[-INV, SENTI='negative']  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (ADJP[SENTI='negative']  (JJ[SENTI='positive'] well-intentioned)  (CC[-and, +but, -or] but)  (JJ[SENTI='negative'] manipulative))  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] movie)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] making))))) | Parse Tree(s) |

## **Negative test case 5**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/negative.txt** | a perfect example of well-intentioned but manipulative movie making . negative | [positive, negative] | Console Output |
| Input sentence | a perfect example of well-intentioned but manipulative movie making . | (S[-INV, SENTI='positive']  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (DT[NUM='sg'] a)  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] perfect)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] example))))  (IN[+of] of)  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (ADJP[SENTI='negative']  (JJ[SENTI='positive'] well-intentioned)  (CC[-and, +but, -or] but)  (JJ[SENTI='negative'] manipulative))  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] movie)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] making)))))) (S[-INV, SENTI='negative']  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (DT[NUM='sg'] a)  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] perfect)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] example))))  (IN[+of] of)  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (ADJP[SENTI='negative']  (JJ[SENTI='positive'] well-intentioned)  (CC[-and, +but, -or] but)  (JJ[SENTI='negative'] manipulative))  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] movie)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] making)))))) | Parse Tree(s) |

### **Negative test case 5 result explanation:**

1. **Sentiment label**

This issue is similar as the issue in Positive test case 5. The sentiment of a NP containing “of” can be determined by the part before “of” or the part after “of”. And this is discussed in the main report Limitation section.

1. **Parse tree**

The parse trees are outputted 1 for positive sentiment and 1 for neutral sentiment. And the order of parse trees are same as the order of sentiment predictions.

|  |  |
| --- | --- |
| **Parse tree with positive sentiment** | (S[-INV, **SENTI='positive'**]  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (DT[NUM='sg'] a)  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] perfect)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] example))))  (IN[+of] of)  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (ADJP[SENTI='negative']  (JJ[SENTI='positive'] well-intentioned)  (CC[-and, +but, -or] but)  (JJ[SENTI='negative'] manipulative))  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] movie)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] making)))))) |
| **Parse tree with negative sentiment** | (S[-INV, **SENTI='negative'**]  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (DT[NUM='sg'] a)  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] perfect)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] example))))  (IN[+of] of)  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (ADJP[SENTI='negative']  (JJ[SENTI='positive'] well-intentioned)  (CC[-and, +but, -or] but)  (JJ[SENTI='negative'] manipulative))  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] movie)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] making)))))) |

## **Negative test case 6**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/negative.txt** | a perfect example of rancid , well-intentioned , but shamelessly manipulative movie making . negative | [positive, negative] | Console Output |
| Input sentence | a perfect example of rancid , well-intentioned , but shamelessly manipulative movie making . | (S[-INV, SENTI='positive']  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (DT[NUM='sg'] a)  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] perfect)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] example))))  (IN[+of] of)  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (ADJP[SENTI='negative']  (JJ[SENTI='negative'] rancid)  (COMMA[] ,)  (ADJP[SENTI='negative']  (JJ[SENTI='positive'] well-intentioned)  (COMMA[] ,)  (CC[-and, +but, -or] but)  (ADJP[SENTI='negative']  (RB[SENTI='negative', -negation] shamelessly)  (JJ[SENTI='negative'] manipulative))))  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] movie)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] making)))))) (S[-INV, SENTI='negative']  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (DT[NUM='sg'] a)  (NP[NUM='sg', PERSON=?p, SENTI='positive']  (JJ[SENTI='positive'] perfect)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] example))))  (IN[+of] of)  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (ADJP[SENTI='negative']  (JJ[SENTI='negative'] rancid)  (COMMA[] ,)  (ADJP[SENTI='negative']  (JJ[SENTI='positive'] well-intentioned)  (COMMA[] ,)  (CC[-and, +but, -or] but)  (ADJP[SENTI='negative']  (RB[SENTI='negative', -negation] shamelessly)  (JJ[SENTI='negative'] manipulative))))  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] movie)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] making)))))) | Parse Tree(s) |

### **Negative test case 6 result explanation:**

1. **Sentiment label**

This issue is similar as Negative test case 5 and is discussed in the main report.

## **Negative test case 7**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/negative.txt** | He gave her an ugly sweater . negative | [negative] | Console Output |
| Input sentence | He gave her an ugly sweater . | (S[-INV, SENTI='negative']  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] He))  (VP[NUM='sg', PERSON=3, SENTI='negative', TENSE='past']  (V[-AUX, SUBCAT='dative', TENSE='past', -passive] gave)  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] her))  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (DT[NUM='sg'] an)  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (JJ[SENTI='negative'] ugly)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] sweater)))))) | Parse Tree(s) |

## **Negative test case 8**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/negative.txt** | I saw a dull and scary movie . negative | [negative] | Console Output |
| Input sentence | I saw a dull and scary movie . | (S[-INV, SENTI='negative']  (NP[NUM='sg', PERSON=1, SENTI=?s] (PRP[NUM='sg', PERSON=1] I))  (VP[NUM='sg', PERSON=?p, SENTI='negative', TENSE='past']  (V[-AUX, SUBCAT='trans', TENSE='past', -passive] saw)  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (DT[NUM='sg'] a)  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (ADJP[SENTI='negative']  (JJ[SENTI='negative'] dull)  (CC[+and, -but, -or] and)  (JJ[SENTI='negative'] scary))  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] movie)))))) | Parse Tree(s) |

## **Negative test case 9**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/negative.txt** | It was a mess and a hazard . negative | [negative] | Console Output |
| Input sentence | It was a mess and a hazard . | (S[-INV, SENTI='negative']  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] It))  (VP[NUM='sg', PERSON=?p, SENTI='negative', TENSE='past']  (V[+AUX, NUM='sg', TENSE='past', -passive] was)  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (DT[NUM='sg'] a)  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (N[NUM='sg', SENTI='negative'] mess)))  (CC[+and, -but, -or] and)  (NP[NUM='sg', PERSON=?p, SENTI=?s] (DT[NUM='sg'] a)))  (NP[NUM='sg', PERSON=?p, SENTI='negative']  (N[NUM='sg', SENTI='negative'] hazard)))) | Parse Tree(s) |

## **Neutral test case 1**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/neutral.txt** | it has low impact . neutral | [neutral] | Console Output |
| Input sentence | it has low impact . | (S[-INV, SENTI='neutral']  (NP[NUM='sg', PERSON=3, SENTI=?s] (PRP[NUM='sg', PERSON=3] it))  (VP[NUM='sg', PERSON=3, SENTI='neutral', TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive] has)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (JJ[SENTI='neutral'] low)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] impact))))) | Parse Tree(s) |

## **Neutral test case 2**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/neutral.txt** | this is an example . neutral | [neutral] | Console Output |
| Input sentence | this is an example . | (S[-INV, SENTI='neutral']  (NP[NUM='sg', PERSON=?p, SENTI=?s] (DT[NUM='sg'] this))  (VP[NUM='sg', PERSON=3, SENTI='neutral', TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive] is)  (NP[NUM='sg', PERSON=?p, SENTI=?s] (DT[NUM='sg'] an))  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] example)))) | Parse Tree(s) |

## **Neutral test case 3**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input file and content** | **Output on console and parse tree diagram** |  |
| Input file | **data/neutral.txt** | this is an example of movie making . neutral | [neutral] | Console Output |
| Input sentence | this is an example of movie making . | (S[-INV, SENTI='neutral']  (NP[NUM='sg', PERSON=?p, SENTI=?s] (DT[NUM='sg'] this))  (VP[NUM='sg', PERSON=3, SENTI='neutral', TENSE='pres']  (V[+AUX, NUM='sg', PERSON=3, TENSE='pres', -passive] is)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (DT[NUM='sg'] an)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] example)))  (IN[+of] of)  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] movie)))  (NP[NUM='sg', PERSON=?p, SENTI='neutral']  (N[NUM='sg', SENTI='neutral'] making)))) | Parse Tree(s) |