

## Report – Project 3

### Design:

As given, I took the basic feature grammar, which is “feat\_gram\_project\_2.fcfg” from Project 2. The submitted grammar in the file “feat\_gram\_project\_3.fcfg” is inspired from the above mentioned feature grammar. The grammar includes Source (S), Noun Phrases (NP), Verb Phrase (VP), Adverbial Phrases (AdvP), Adjective Phrases (AdjP), Prepositional Phrases (PP) and other rules to parse tree for a sentence as per singular, plural, noun, pronoun, tenses, active-passive voices, satisfying the Sentiment Features to know whether a sentence is positive (pos), negative (neg) and neutral (neut)

Overview of possible expansions:

Source can form itself with punctuations, and to NP, VP, PP. NP could further be parsed into Determinants, Adjectives, Pronouns, Proper Nouns, Nouns. Verb Phrases could expand to NP, PP and different kinds of verbs including infinitive, intransitive, transitive, stative verbs. PP (Prepositional Phrases) could be expanded into PP itself and NP.

### Feature grammar coverage:

The grammar from the file “feat\_gram\_project\_3.fcfg” could parse most of the sentences from the test file “test-data.txt” given with the project description. The test file consists of varieties of semantics having good amount of combinations of punctuations, conjunctions, noun phrases, verb phrases, prepositional phrases altogether.

As the requirement mentions about pos, neg, pos AND, neg AND, pos BUT, neg BUT. The grammar can parse all of them.

### Design better than SSAP (afinn) [in some aspects]:

Let's talk about how the design of feature grammar is better than SSAP?

Taking some examples such as “**This is not fun.**” It is basically a negative sentence and it is handled in the feature grammar as the negated (“not”) positive word (“fun”) makes it negative, but whereas the SSAP gives it a mark of 1.79 which makes it positive and it is not the correct sentiment inference for the mentioned sentence.

Another example where “**The controls are lousy.**” It is basically a negative sentence and it is handled in the feature grammar as the negated (“not”) positive word (“fun”) makes it negative, but whereas the SSAP gives it a mark of 1.79 which makes it positive and it is not the correct sentiment inference for the mentioned sentence.

## Ideas:

At first, feature grammar was not outputting correct results for such sentences where sentences were containing negation. SSAP was already performing poorly on such sentences.

Taking an example:

**"You will NOT be disappointed."**

Earlier,

Feature grammar was tagging it as a negative sentiment since **disappointed** is a negative verb. SSAP (afinn) gives it a score of **"-0.82"** which is also leaning towards negative sentiment.

Later,

Making some changes to the feature grammar handling this situation here. Where "NOT" would be tagged as negative sentiment as well as disappointed.

So, wrote the grammar rule in such a way that two negative parts of the phrases being siblings make it positive. In this manner we can analyze and solve many of the negation problems.

Taking another example:

**"You will NOT be happy."**

Earlier,

Feature grammar was tagging it as a positive sentiment since **happy** is a positive adjective. SSAP (afinn) gives it a score of **"1.22"** which is also leaning towards positive sentiment.

Afterwards,

Some changes to the feature grammar handling this situation here. Where "NOT" would be tagged as negative sentiment as well as disappointed.

So, wrote the grammar rule in such a way that one negative and one positive parts of the phrases being siblings make it negative.

Mostly, in cases of "BUT" phrases, the second part of the phrase is the negation of the sentiment of the first phrase. Talking about the "AND", in most of the cases sentiment of the second phrase is the similar is the first phrase.

These were some basic ideas to mention.

## Changes to the feature grammar after comparing with SSAP:

### Comparison and Analysis:

When we evaluate sentences on SSAP there are certain conditions where sentences were tagged correctly and incorrectly, I analyzed the performance of feature grammar from Project-3 keeping SSAP as baseline.

Example:

SSAP =, Feature Grammar Project 3 = neg

Sentences	SSAP Performance	Feature Grammar Performance
a) This cd was horrible.	-1.34	neg
b) The controls are lousy.	0.0	neg
c) This is not fun.	1.79	neg

As we can see 3 different scenarios where baseline has similar values or varies from the feature grammar output. Sentence a) is an actual negative sentence and it is tagged as neg by feature grammar and SSAP as well, but the sentences are actually tagged correctly by feature grammar as they are incorrectly tagged in baseline.

## **Improvements to the feature grammar than Project 2:**

### **Adding sentiment feature to feature grammar from project 2:**

Feature grammar from project 2 did not have sentiment feature. So, had to add the sentiment feature in each of the grammar rule where it was required. Where it can impact the sentiment of the whole phrase.

### **Handling sentences containing 's :**

Feature grammar from project 2 did not have procedures to handle “s”. So, had to add the text processing to handle the scenario to parse sentences.

### **Adding some rules:**

There were some rules missing from project 2 which were added on to the feature grammar of project.