# Deep Parsing & Multimodal Sentiment Analysis

Team 6
Mohammad Ali Rehan
Neel Aryan Gupta
Shreya Pathak

# DP to CP Transformer

The transformer is a complete functional tool that takes as input a sentence, generates a dependency parse and applies our top down algorithm to get the constituency parse. The code base is complete for both the tools.

We use the generic grammer given below to represent the CP tree and find these corresponding phrases from the dependency tree.

S-> NP VP  SBAR

NP->DET JJP NNS/NNP SBAR | (NP CONJ NP)+

VP->AUX* NEG* VBD/VBG ATTR* NP1* NP2* ACOMP* RB* PP* SBAR* | (VP CONJ VP)+

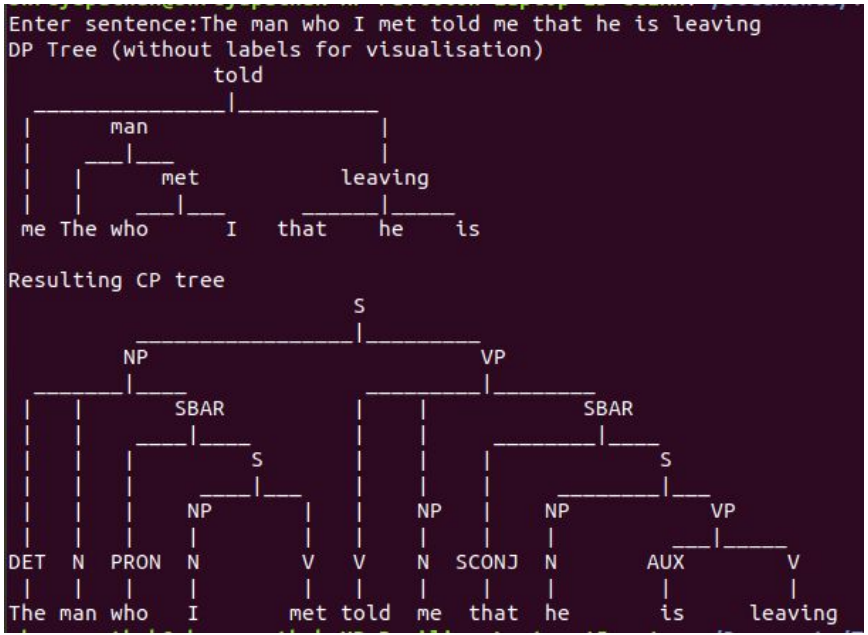
JJP->JJ+ | (JJP CONJ JJP)+

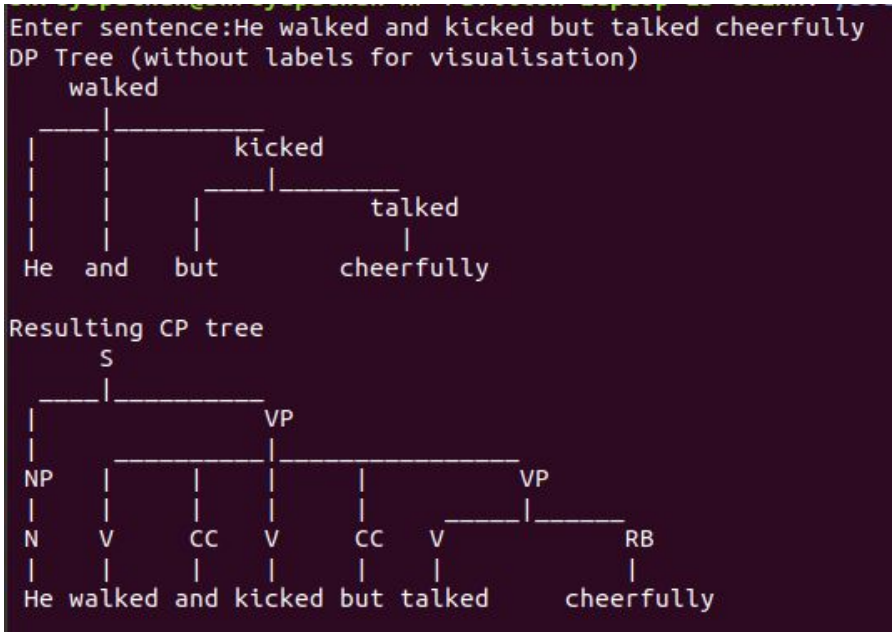
PP-> IN NP

SBAR->WHWORD S

# DP to CP Transformer

We have created an interface that takes as input an example and using the dependency parse output of the spacy library in python converts it to constituency parse tree. The tool is completely ready, we show I/O below



# Error Analysis

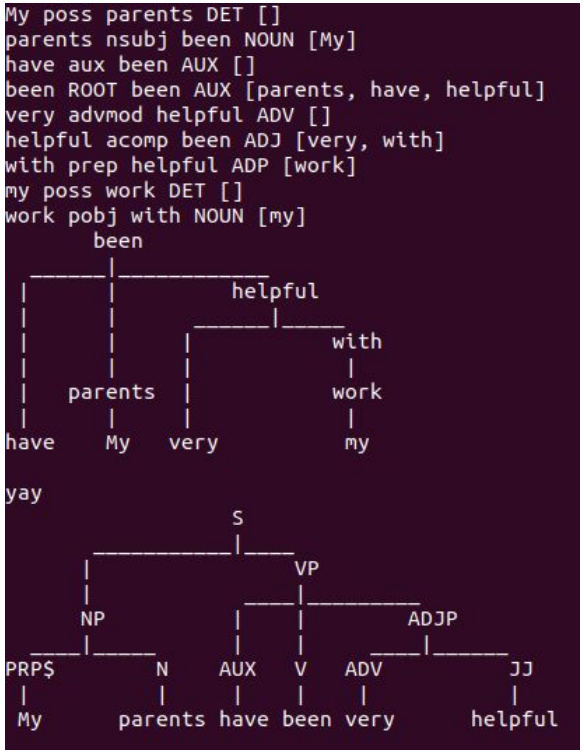
- Passive voice sentences are not handled properly as they involve tags like nsubjpass etc which we have not been encoded

```
The det seats DET []
seats nsubjpass sold NOUN [The]
were auxpass sold AUX []
sold ROOT sold VERB [seats, were, out, fast]
out prt sold ADP []
fast advmod sold ADV []
     sold
  ____|_________
 |    |    |   seats
 |    |    |     |
were out  fast  The

      S
      |
      VP
      |
      VP
  ____|___
 V        RB
 |        |
sold     fast
```

- Questions are not handled properly.
  Eg: when did you come back

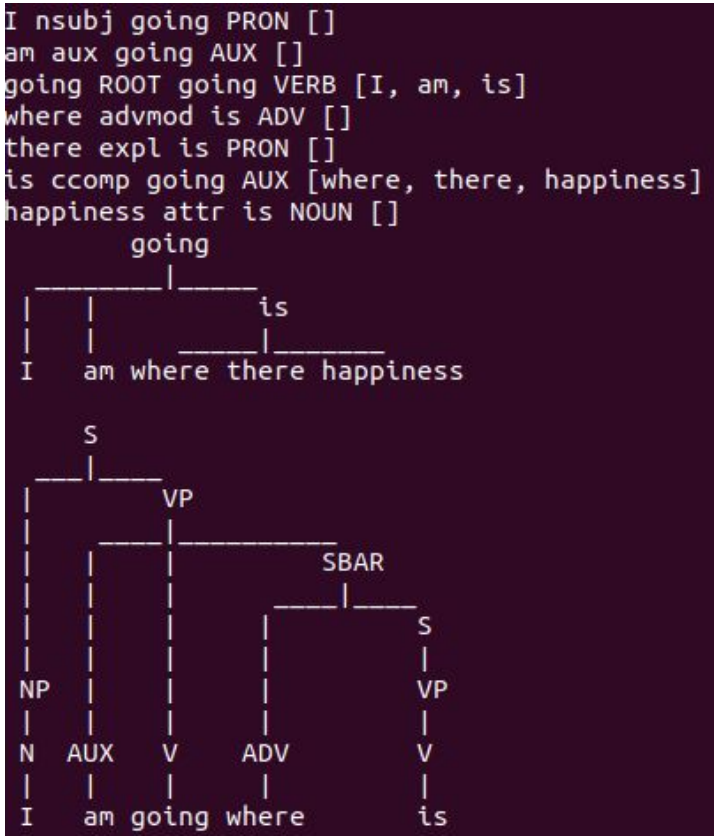
```
When advmod come ADV []
did aux come AUX []
you nsubj come PRON []
come ROOT come VERB [When, did, you, back]
back advmod come ADV []
     come
  ____|________
When did  you back

      S
   ___|___
 NP       VP
 |     ___|_________
 N   AUX  V    RB   RB
 |    |   |    |    |
you did come When back
```

- Subtree corresponding to acomp isn't expanded. Input: My parents have been very helpful with my work

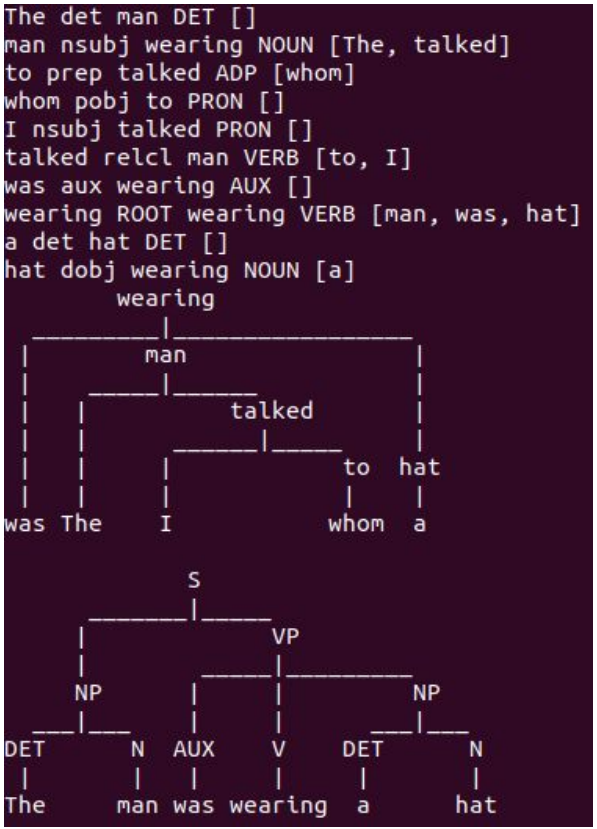


- The identification of label of whword and labels in its subtree is not always correct because of the various roles it can take in the clause like object, subject etc.

I am going where there is happiness




The man to whom I talked was wearing a hat



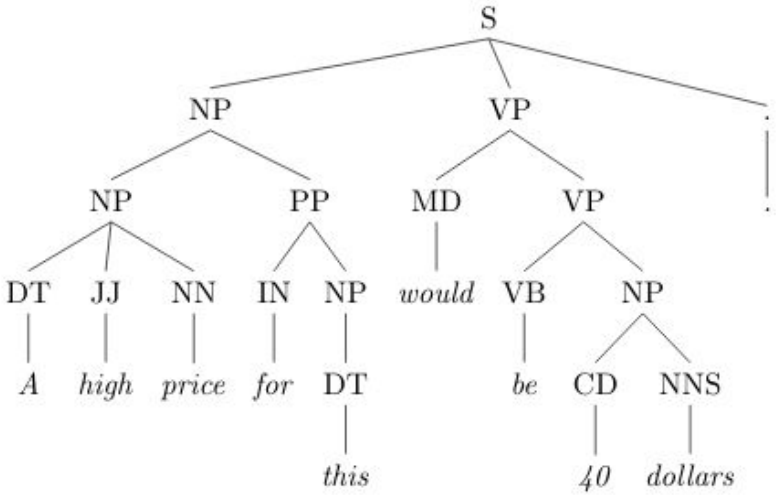
# CP to DP Transformer

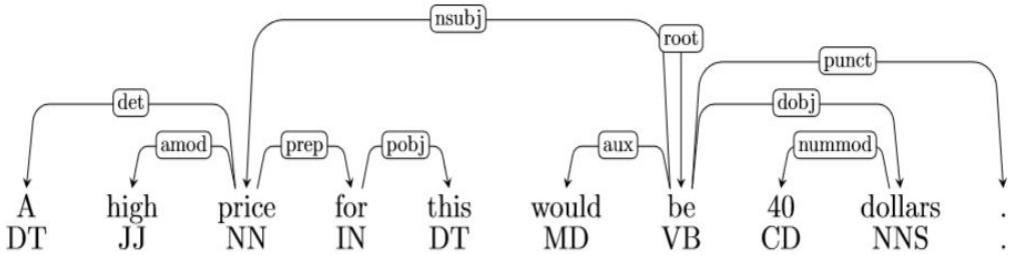
The transformer is a complete functional tool that takes as input a sentence, generates a constituency parse and applies the algorithm below to get the dependency parse.

We have created an interface that takes as input an example and using the constituency parse output of the benepar library in python converts it to constituency parse tree. The algorithm used for the transformation is based on Xia and Palmer (2001). Firstly, we assign lexical heads at each level of the tree using the head rules suggested by Collins (1999). The model uses local information to determine the edge labels in the resulting dependency tree. The local information comprises of the phrase chunk labels of the lexical heads at the ends of the given edge, along with their POS tags.

# CP to DP Transformer

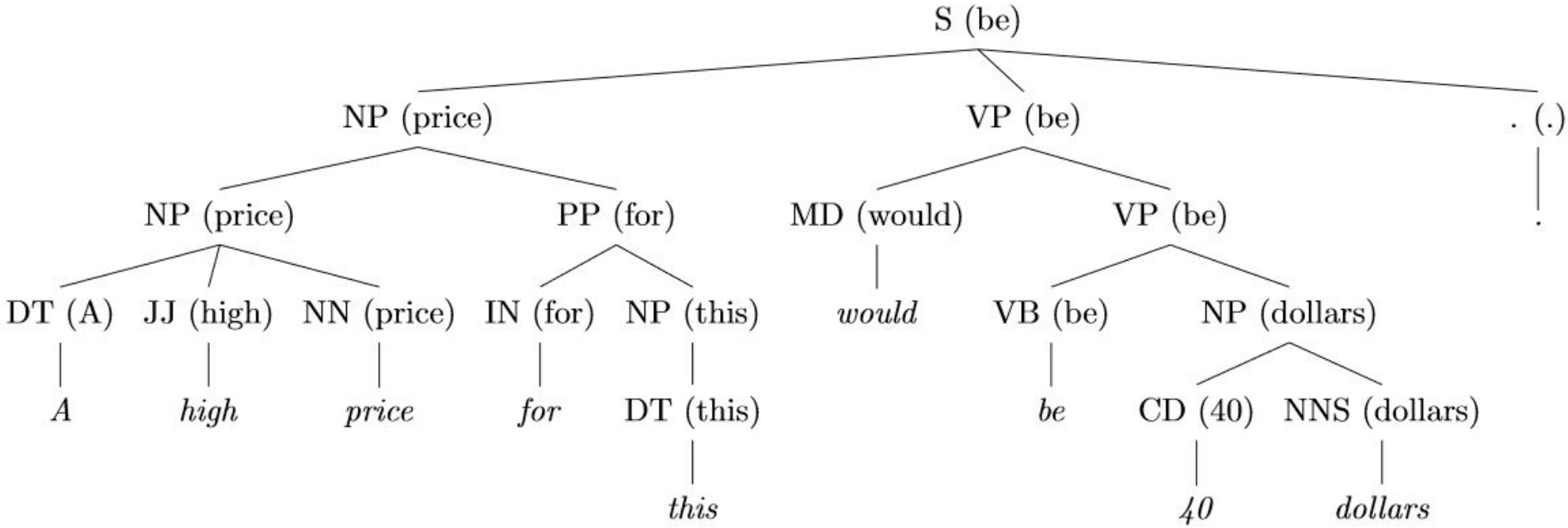
The tool is completely ready and we show input output below



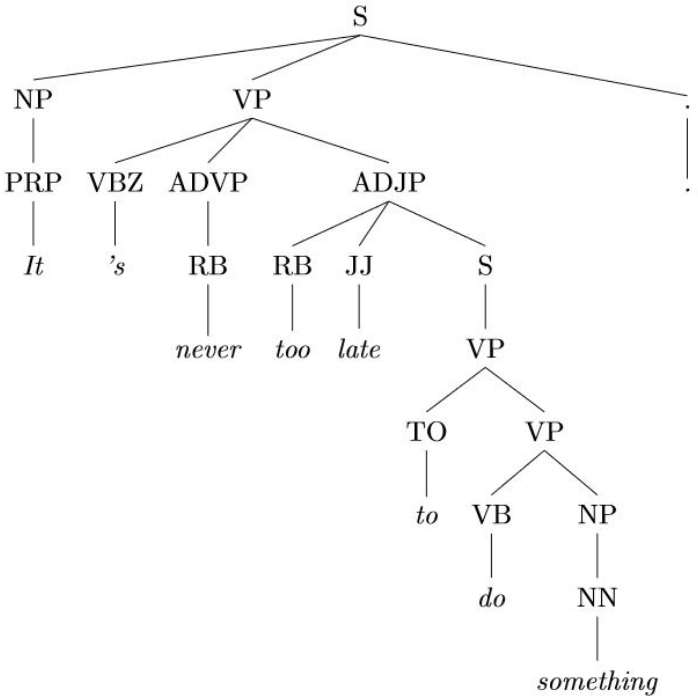
# CP to DP Transformer

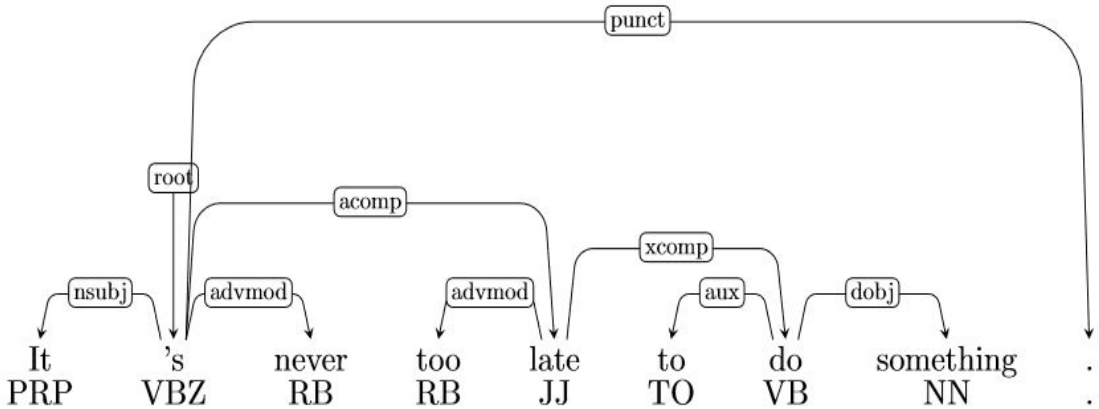
Here is the headified CP tree (using Collin's 1999 head-finding rules) of the previous sentence.



# CP to DP Transformer

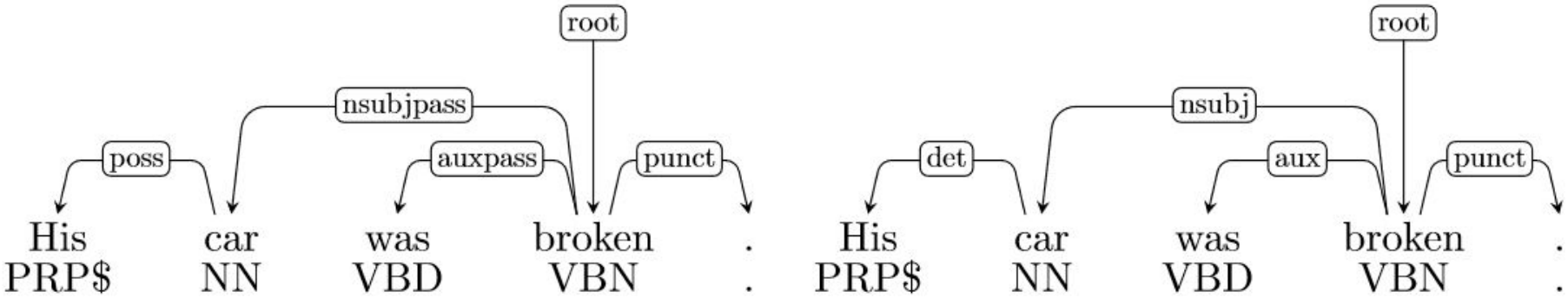
Here's another example output of the parser.



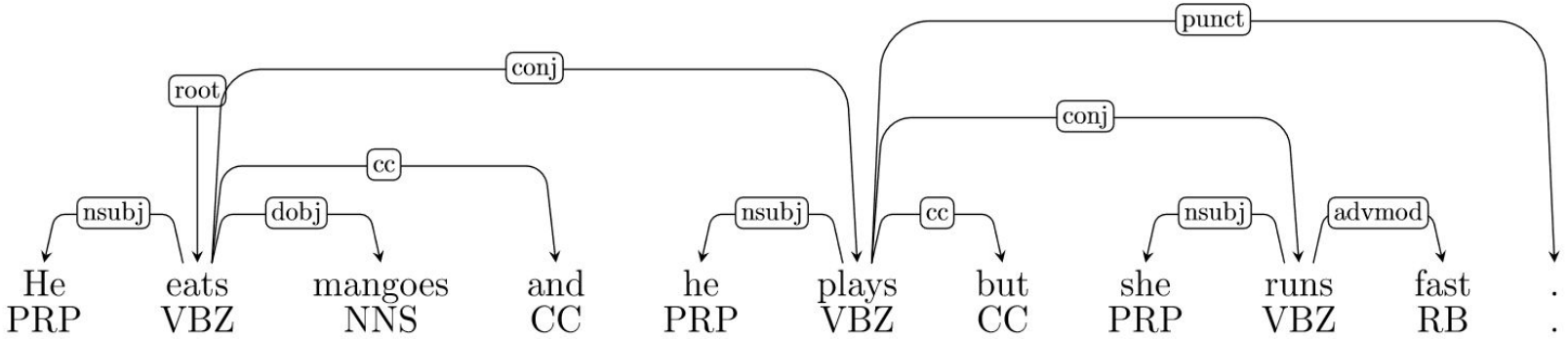



# Error Analysis

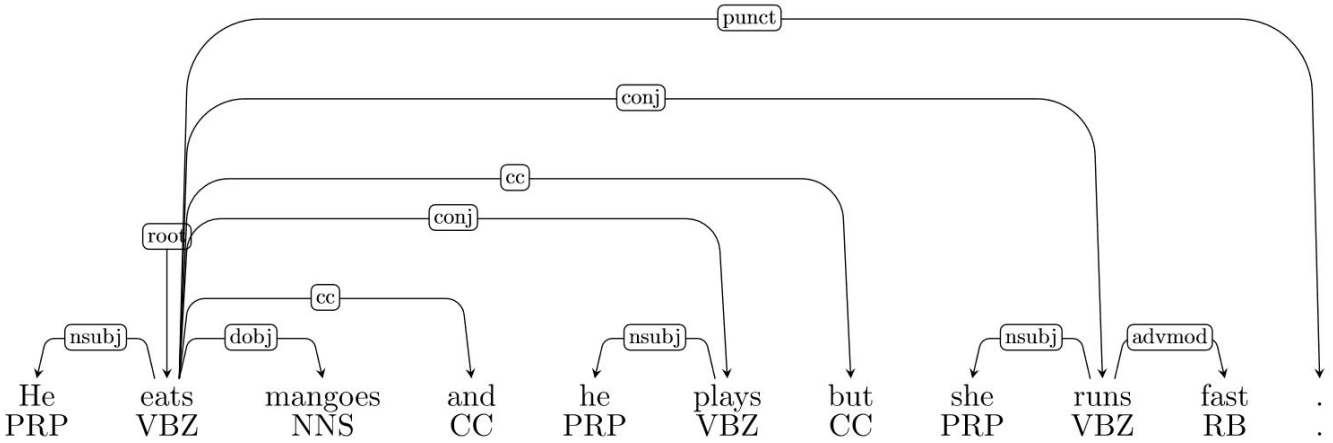
- Passive voice sentences for which the modifiers should ideally be marked as ’nsubjpass’, ’auxpass’ etc. are currently not handled by the model. For Example, the correct and the produced parses are respectively. Eg: His car was broken. On the left is the correct tree



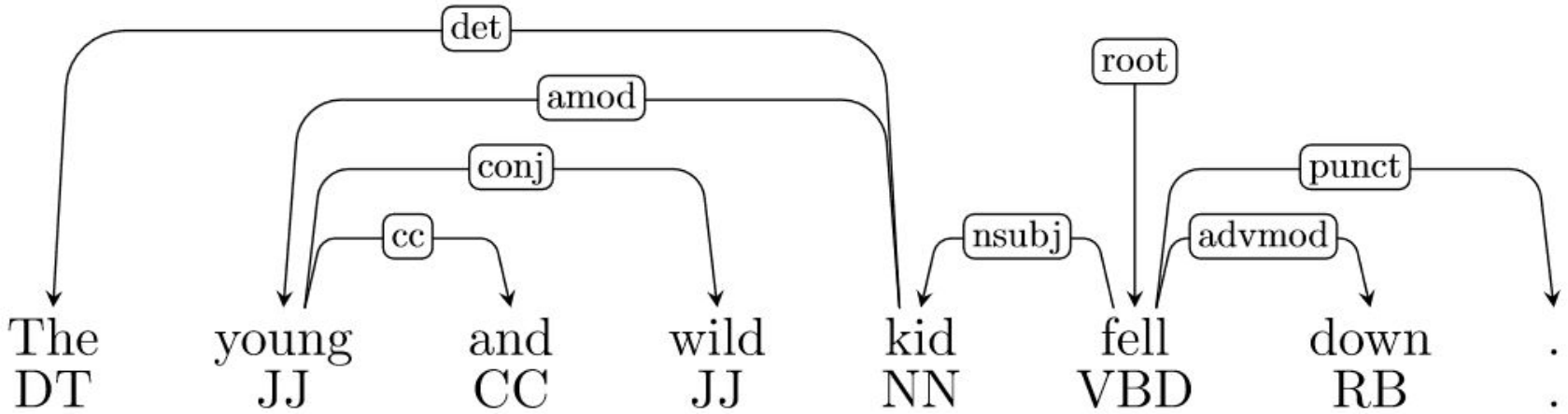
- Sentences involving multiple conjunctions are not handled properly, depending on the type of the phrase-chunks obtained from the parser. For the sentence, ‘He eats mangoes and he plays but she runs fast’, the correct tree is:



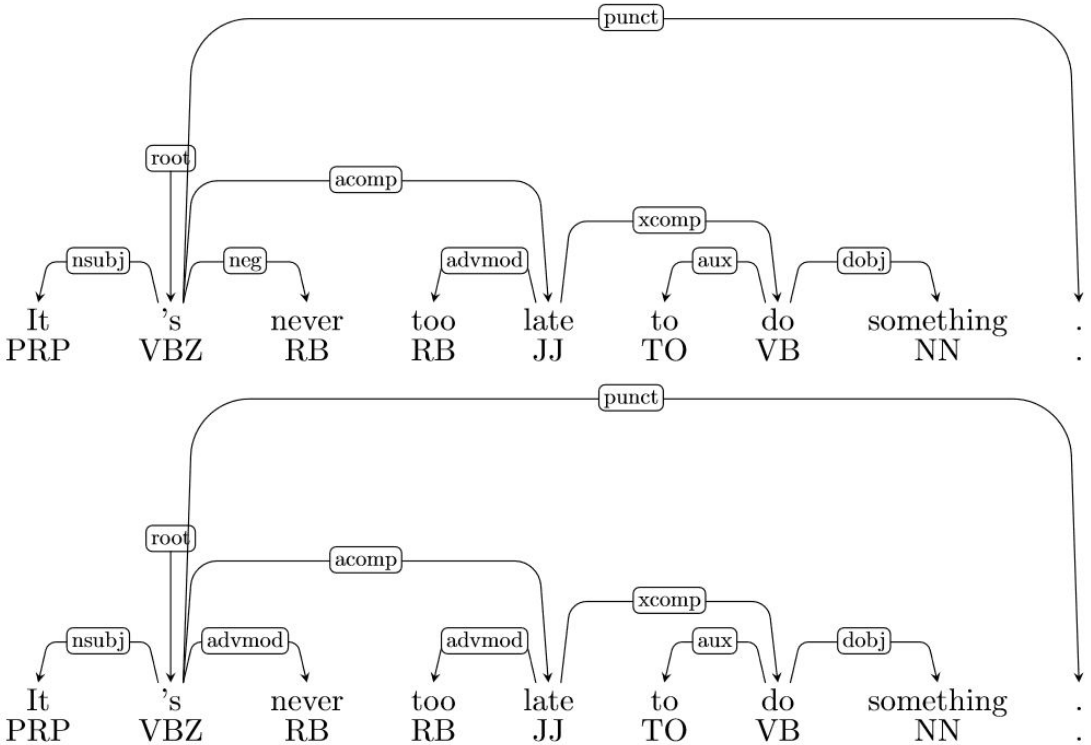

But we got



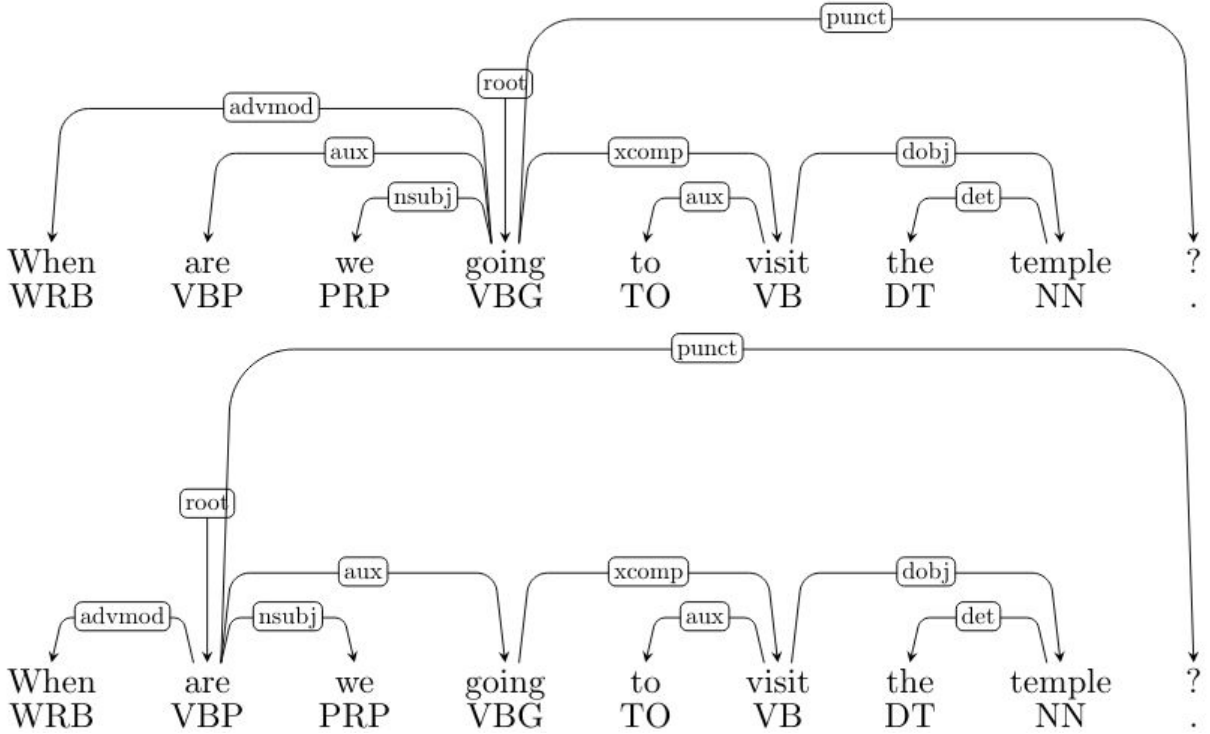
- Some complex tags such as 'dative', 'attr' (see the given example) are mispredicted because prediction of these relations requires much more information than just the local tags.
- ‘prt’ modifiers are mistaken as ‘advmod’ because this relation excludes literal/directional uses of prepositions/particles, in which case ‘advmod’ is used as the relation. Example : (Here ‘prt’ should be the relation instead of ‘advmod’)



- Model fails to identify the ’neg’ modifier, because the negative words like ‘not’, ‘never’ are usually tagged as ’RB’,and it’s hard to deduce such words from the tags themselves, but can be done with hard coding all the negative words from a large enough corpus. Example : top one is correct and at the bottom is ours.



- The model sometimes fails to identify the verbs properly because the fixed set of head-rules used in the model are not enough to capture all the ambiguities of the language. For example, here are the correct and the produced parses respectively. Example



# Project Description

- Study effect of adding audio data to text for sentiment analysis
- Study the underlying technique for semantic integration of the two sources of data
- Implement the techniques in the paper ‘Multimodal Sentiment Analysis using Hierarchical Fusion with Context Modeling’ by Majumder et al.
- Carry out training and evaluation on the datasets:

  1) CMU-MOSI
  2) IEMOCAP and more if time permits ...

- Compare with other baselines and more primitive models
- Carry out error analysis and find weak spots

# Project Progress

- Prepared script to extract audio features from the given dataset
- Implemented a basic model using text and audio cues individually
- Trained and tested on a subset of the CMU-MOSI dataset and
- Read about the techniques in the paper ‘Multimodal Sentiment Analysis using Hierarchical Fusion with Context Modeling’ by Majumder et al and summarized part of it.

## Work remaining to be done

- Implement the fusion layers to integrate the multimodal data as discussed in paper
- Finish summarising the paper and carry out extensive error analysis
- Try and compare different model architectures like BERT, LSTM etc.
- Train and test on other datasets eg, IEMOCAP with more possible outputs

# Unimodal model

We have implemented a very simple model which takes unimodal features as input and gives the the sentiments as output

The features are passed through an input layer which feeds the result into a GRU. The output of the GRU is fed into a time-distributed dense layer with softmax activation to give the desired result.

We will use a similar architecture in the final bimodal model (after fusing the features) and hence will be able to compare coherently the effects of adding additional data

The unimodal accuracies obtained are:

- Text: 68.87%
- Audio: 39.32% (41.45 % if Bidirectional LSTM used instead of GRU)

# Bimodal model

- We aim to implement a model similar to that given in the aforementioned paper
- In this model after passing the unimodal features through a GRU layer (similar to the unimodal model already implemented) and after equalising dimensions by passing through a Dense layer, we combine them by passing them through a Dense layer with tanh activation followed by another GRU layer
- Now these features with bimodel contexts are passed through a Dense layer with softmax activation to get the sentiments

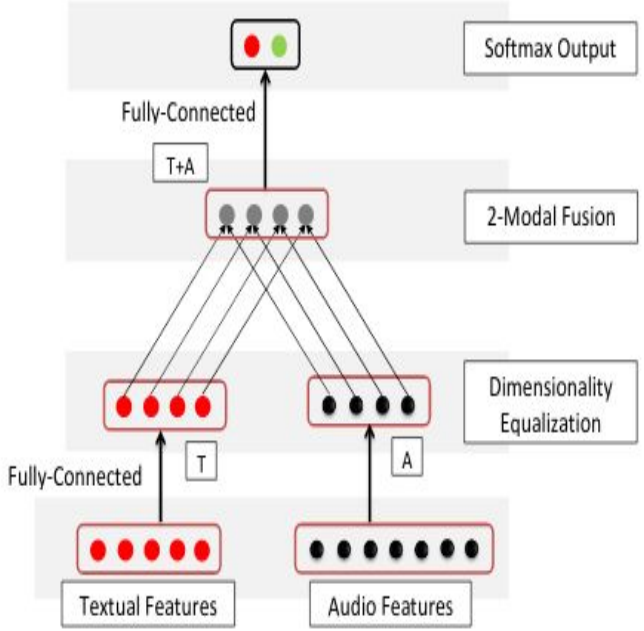



Figure 2: Utterance-level bimodal fusion