Bonus

Spanish:

Question 8.

1. For simplicity, we will focus on the instances which consist only of a single token and have been labeled as complex by at least one annotator.   
   Calculate the length of the tokens as the number of characters.

Calculate the frequency of the tokens using the wordfreq package (<https://pypi.org/project/wordfreq/>).

Provide the Pearson correlation of length and frequency with the probabilistic complexity label:

Pearson correlation length and complexity: 0.22

Pearson correlation frequency and complexity: -0.16

Provide 3 scatter plots with the probabilistic complexity on the y-axis.

X-axis: 1) Length 2) Frequency 3) POS tag

Set the ranges of the x and y axes meaningfully.

Scatter chart

Description automatically generated

A picture containing text

Description automatically generated

Chart, scatter chart

Description automatically generated

Question 12

Train the model on the data in *preprocessed/train* and *preprocessed/dev* by running the code in *train.py*.

Evaluate the model on the data in *preprocessed/test*.

The original code only outputs the accuracy and the loss of the model. I adapted the code for you, so that it writes the predictions to *model\_output.tsv*.

Implement calculations for precision, recall, and F1 for each class in *TODO\_detailed\_evaluation.py*. You can use existing functions but make sure that you understand how they work.

Provide the results for the baselines and the LSTM in the table below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Class N | | | Class C | | | Weighted Average |
|  | Precision | Recall | F1 | Precision | Recall | F1 | F1 |
| Random | 0.50 | 0.88 | 0.64 | 0.50 | 0.12 | 0.19 | 0.41 |
| Majority | 1 | 0.88 | 0.93 | 0.0 | 0 | 0 | 0.94 |
| Length | 0.82 | 0.94 | 0.87 | 0.60 | 0.31 | 0.41 | 0.77 |
| Frequency | 0.98 | 0.19 | 0.31 | 0.44 | 0.99 | 0.61 | 0.43 |
| LSTM | 0.87 | 0.95 | 0.91 | 0.67 | 0.43 | 0.53 | 0.83 |

Question 13

We see once again that the LSTM doesn’t outperform simple statistical models on F1 score. However, it is the model that works best without having a 0 in any category so I would prefer it when I would have to choose the best model for a general purpose.

Question 14.

Increasing the amount of nodes leads to a better F1 score. This suggests that we should use large hidden layers to capture the patterns that decide if a word is complex

Chart, line chart

Description automatically generated