# Computational Linguistics - Report Assignment 1

Felix Koole, Gaby Voorbraak, Sjors Lockhorst, Tijn Smedema

20 May 2022

## Results

In Table 1, an overview of the performance of the bigram and trigram models with the default limit of 200 is presented. In general, all models perform better on test samples with longer sentences. In addition, the trigram models outperform the bigram models for all sentence lengths.

Per sample size, we created a graph that plots the limit against the correct-%. We see that an increase in limit leads to an increase in the correct-%, which is as expected. For the lowest sample size (Figure 1), we see that the bigrams mostly outperform the trigrams, until a limit of approximately 75. For a sample size of 30 (Figure 2), we see the same, but with less fluctuation. For a sample size of 90 (Figure 3), we see little to no difference between the bigram and trigram performance until a limit of approximately 10. For limits higher than 10, we see that the trigrams clearly outperform the bigrams. In general, we notice the following trend: the higher the sample size, the smaller the difference between the bigram and trigram models.

If we tweak the $n$ of the n-grams a bit, we see different performances from the different sample-sizes, which is explainable since more data means more accuracy. In Figure 4 for example, we see that in all the sample sizes, the performance is relatively low with $n = 1$ (and the performance increases with higher test samples). That is easy to explain, since the n-grams only contain one letter each, which makes it easy to mismatch a language (it would still be possible since some languages contain more specific letters than others). If we increase the $n$, we see that the n-gram performance in all sample sizes peaks around $n = 4$ or $n = 5$. This makes sense, because the n-grams created by a high $n$ are way more specific and can barely be matched with a text-data-set to test it on. Also, the higher the $n$, the higher the chance of the gram containing white space, which decreases the probability of matching a language even more.

We do see that the performance goes up depending on the sample size, which is explainable since more data means more accuracy.

## Reflection

In general, the programming of the module and scripts went well. As the structure of the assignment was well-defined it was not very hard to write the scripts. However, due to this we also experienced some difficulties when trying to write an extra layer on top of the written scripts, to compare the data for the discussion points for the report. In the end, we wrote this in a separate module which fixed a lot of the problems.

As for the collaboration, we felt like the structure of the project made it a bit difficult to divide the work. For example, it was not possible for two people to write the module and the other two people to write the scripts at the same time, as everything builds on the module. In addition, not everyone was able to be present at all the work groups, which made it challenging to involve everyone at all times. Therefore, in each workgroup, the people that were present worked on the project as much as possible and so we managed to finish it in the end. We also made use of github so the code was easily accessible for each team member, which is something we will definitely continue to be doing in the future.

## Tables and graphs

| **ngram** | **sentence length** | **correct %** |
|---|---|---|
| Bigram | 10 | 63.33% |
| Bigram | 30 | 90.00% |
| Bigram | 90 | 96.67% |
| Trigram | 10 | 73.33% |
| Trigram | 30 | 96.67% |
| Trigram | 90 | 100.00% |

Table 1: Bigram vs Trigram performance

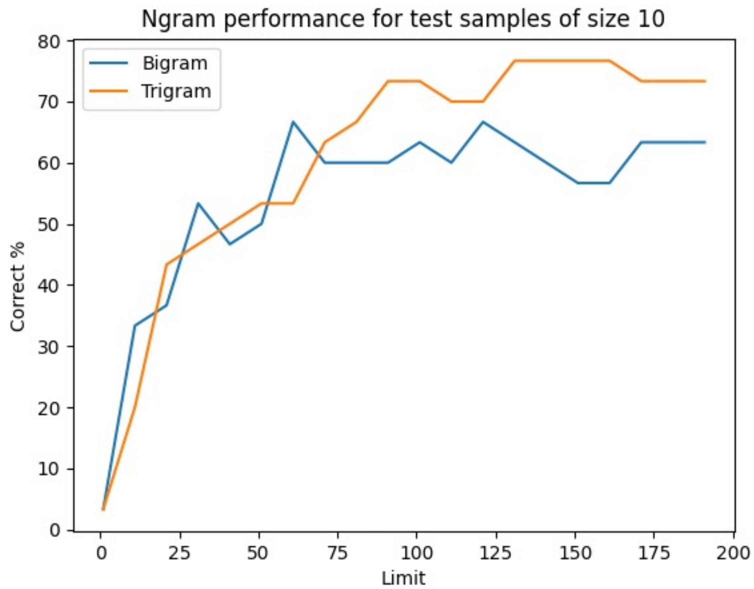


Figure 1: Bigram/trigram performance for test samples of size 10

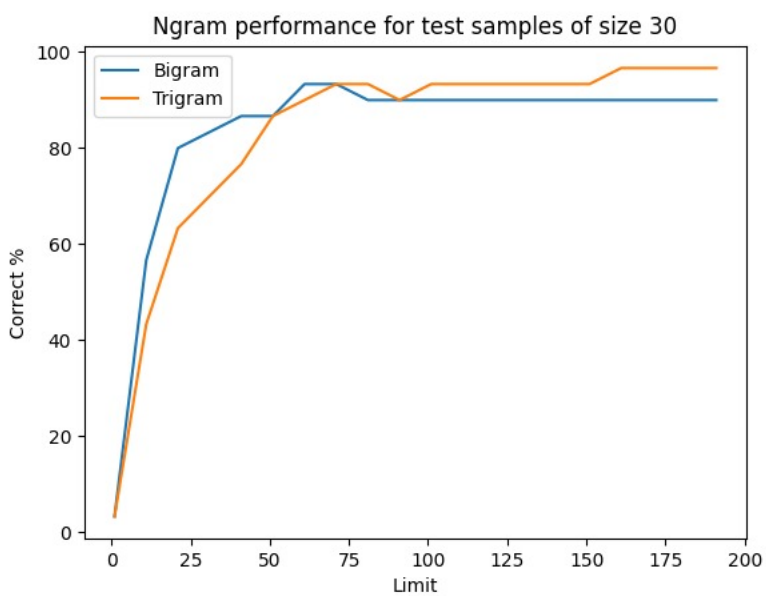


Figure 2: Bigram/trigram performance for test samples of size 30

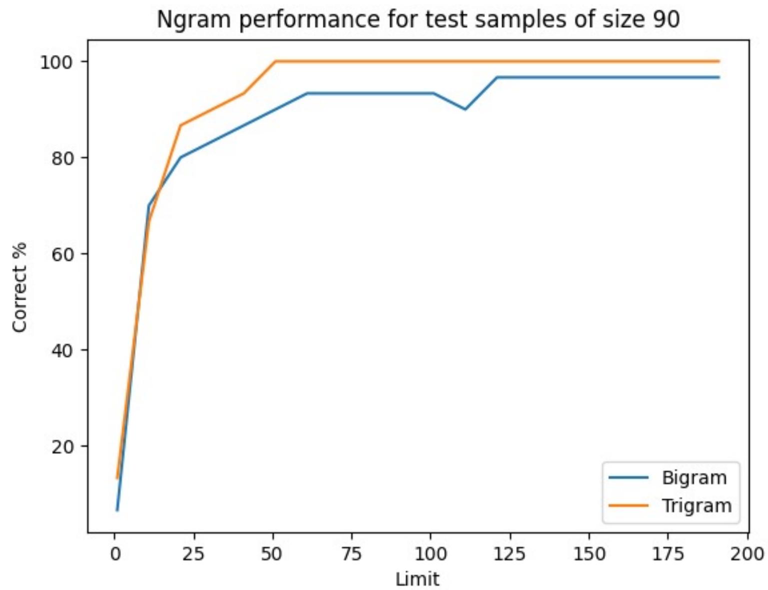


Figure 3: Bigram/trigram performance for test samples of size 90

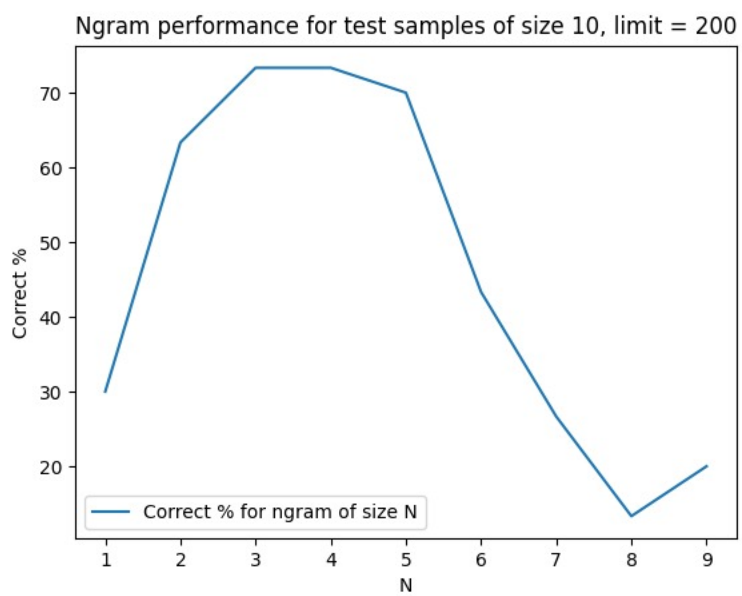


Figure 4: $N$gram performance for test samples of size 10

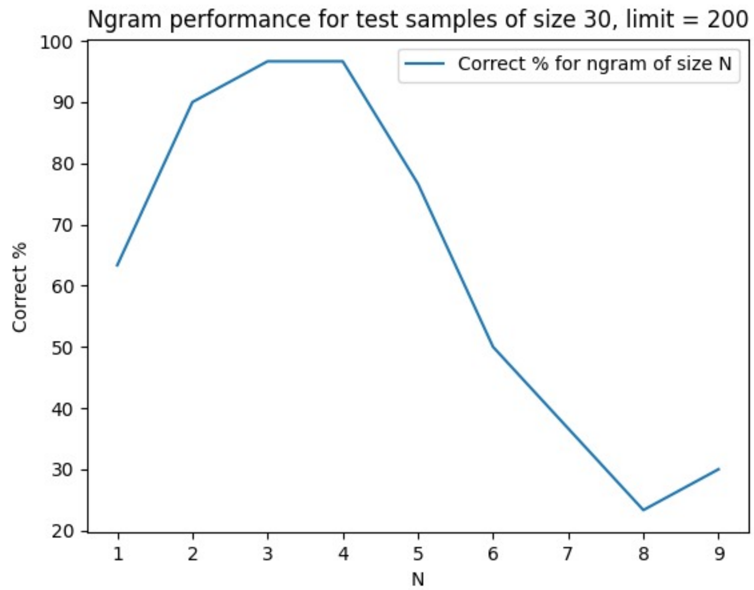


Figure 5: $N$gram performance for test samples of size 90

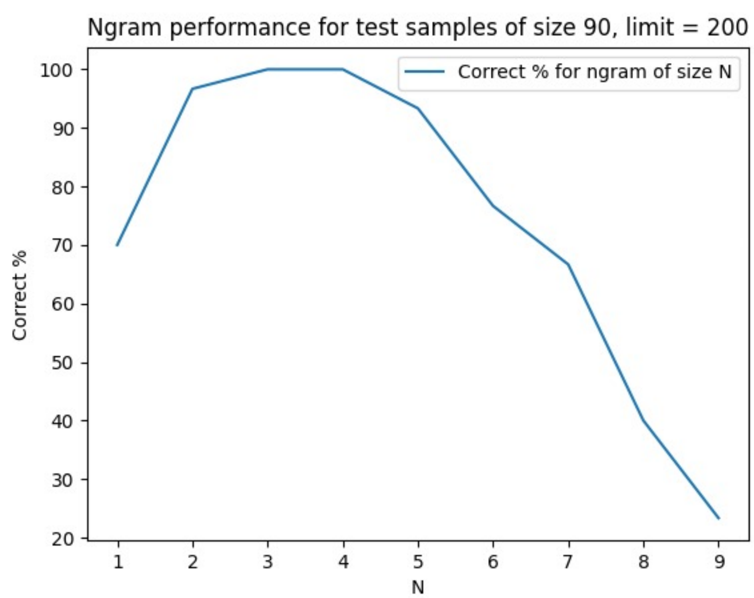


Figure 6: $N$gram performance for test samples of size 90