Similarity between French words based on their phonetic transcription using Needleman-Wunsch & graph clustering

Dominic Plein

February 16, 2025

Abstract

This project is part of the GP-GPU Computing course at UGA by Christophe Picard. Here, we present a proposal for the final project. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Contents

1 Introduction

The International Phonetic Alphabet (IPA) uses special symbols[1] to represent the sound of a spoken language. This is useful for language learners since the pronunciation of a word can be significantly different from its written form. For example, the French word *renseignement* (information) is pronounced /ʁɑ̃.sɛɲ.mɑ̃/. Based on this alphabet, one might wonder if we can construct a metric that quantifies the **distance between two words based on their phonetic transcription**. This would allow to construct a graph where nodes are words and edges are weighted by the distance between the words. This graph could then be used to find neighbors of a word based on their phonetic similarity. This opens up the possibility to apply clustering algorithms and other methods stemming from graph theory in order to analyze the phonetic structure of a language.

Calculating the distance between each pair of words corresponds to a fully-connected graph. Our dataset consist of around 600,000 French words and their IPA transcription, alongside their frequency in the French language. Including self-loops, we find a vast number of edges:

$$\#\text{nodes} = 600,000 \tag{1}$$

$$\#\text{edges} = \frac{600,000 \cdot 600,001}{2} \approx 3.60 \times 10^{11} \tag{2}$$

This high number and the independent nature of the distance calculation for each pair of words makes the problem well-suited for parallelization. In section 2, we present the Needleman-Wunsch algorithm used to calculate the distance between two words. In section 3, we discuss how to parallelize this algorithm on a CPU using the *Rayon* library in Rust and on a consumer Nvidia GPU using the CUDA framework with the *cudarc* Rust library. Finally, we provide visualizations of the obtained graphs in 4 and conclude in section 5.

In the following, by *word* we always refer to its phonetic transcription. That is, homophones like the French words *vert* /vɛʁ/ (green) and *verre* /vɛʁ/ (glass) are considered the same word.

[1] See for example the French list here.

Algorithm 2.1: Needleman-Wunsch

Input: $A = \{ A_0, \ldots, A_{\texttt{len}(A)-1} \}$, $B = \{ B_0, \ldots, B_{\texttt{len}(B)-1} \}$,
similarity: similarityScoreFunc, p: GapPenalty
Output: score

```
1  Function calculateScore():
2      Init scoreMatrix with dimensions (len(A) + 1) × (len(B) + 1)
3      for i ∈ {0, ..., len(A)} do
4          scoreMatrix[i][0] ← p · i
5      for j ∈ {0, ..., len(B)} do
6          scoreMatrix[0][j] ← p · j
7      for i ∈ {1, ..., len(A)} do
8          for j ∈ {1, ..., len(B)} do
9              cost ← similarity(A_i, B_j)
10             matchScore ← scoreMatrix[i − 1][j − 1] + cost
11             deleteScore ← scoreMatrix[i − 1][j] + p
12             insertScore ← scoreMatrix[i][j − 1] + p
13             scoreMatrix[i][j] ← max(matchScore, deleteScore, insertScore)
14     return scoreMatrix[len(A)][len(B)]
```

2 Needleman-Wunsch algorithm

TODO... which calculates the global alignment of two sequences and was originally used in bioinformatics to compare DNA sequences. Here, the alphabet will instead consist of the phonetic symbols. Out of all possible alignments of two words (including gaps), the Needleman-Wunsch algorithm finds the one with the smallest distance, i. e. the alignment with the highest score.

A good introduction to the algorithm can be found on the respective Wikipedia page.

3 Parallelized algorithm

test

4 Application

Based on the first 100,000 most frequently used words, we calculated the distance between every pair and visualized the graph using Gephi's ForceAtlas2 algorithm. The neighbors of the word "glace" and "prévoir" are shown in **??**. For bigger graphs than that, Gephi is not able to handle the amount of data anymore.

5 Conclusion

TODO

Glossary

IPA International Phonetic Alphabet. 1