CMPE300

MPI Application Project

Efe Önal

2016400267

Introduction

In this project we are to implement an MPI application which searches for the most semantically close words in a word embedding database. The code is written in c++ using the OpenMPI framework.

The program is compiled with the command "mpic++ -g ./main.cpp -o main".
The program is executed with the command "mpiexec -n NUMBER_OF_PROCESSES ./main".

Assumptions

- There are always 1000 words in the embedding.
- Maximum line length is 6000.
- Maximum word length is 50.
- Embedding dimension is always 300.
- Name of the embedding file is word_embeddings_1000.txt and it is always in the same directory as other project files.
- There is an extra command, COMMAND_WORD_NOT_FOUND = 3, which makes the slave processes do nothing.
- When user enters P+1 processors, P slave processors are in action throughout the program.
- Number of processors is at least 2.

Program Flow

The program start with initializing MPI. Then is sets the world_size and world_rank. Then runMasterNode and runSlavesNode functions are executed.

First, master process calls distributeEmbeddings function which distributes the embedding to the slave processors. It does so by giving all processors same number of lines except fisrt extra number of lines processes get an extra line. Slave processes then all recieve their parts of words and float values.

Then master process broadcasts a command to the slaves. If the command is;

- COMMAND_EXIT = 0, all proceesses, including the master, break their while(true) loops which lets the program execute again after each try until COMMAND_EXIT is given.

- COMMAND_QUERY = 1, all slaves wait for a query word to be broadcasted by the master.
- COMMAND_CALCULATE_SIMILARITY = 2, all slaves processes continue to the normal execution of the code.
- COMMAND_WORD_NOT_FOUND = 3, the word is not in the embedding, so all slaves do nothing.

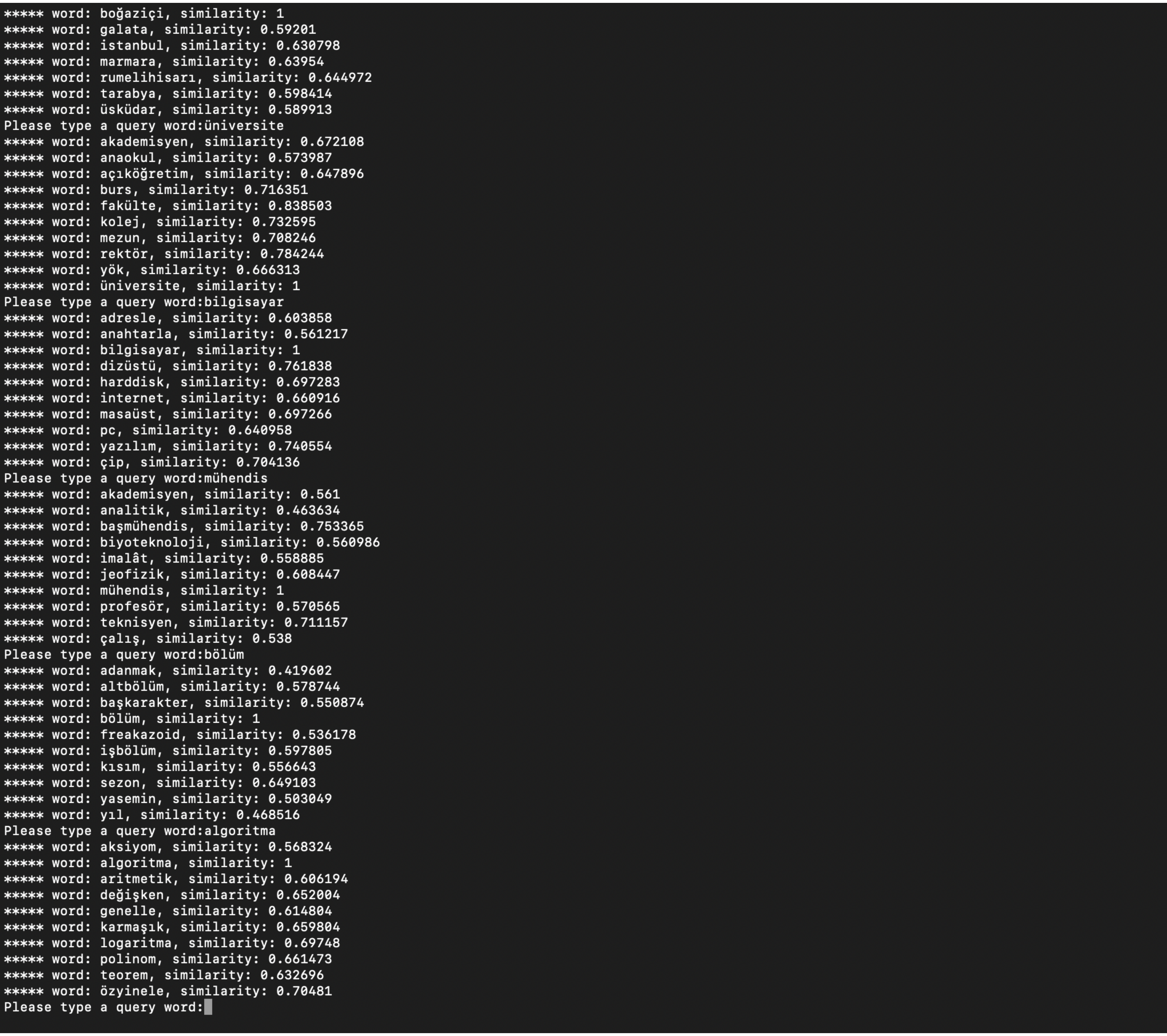
If the normal execution continues, user enters a word then master broadcasts COMMAND_QUERY, then master broadcasts the word and all slaves recieve it by MPI_Bcast method. After getting the word, all slaves search for it in their own "words". If they find it they return the index of the word, else, they return -1 to master by MPI_Send and MPI_Recv. Then if the word is not found master broadcasts COMMAND_WORD_NOT_FOUND but if it is found, master broadcasts COMMAND_CALCULATE_SIMILARITY. This command makes all slaves find similarities of all of their words with the query word. Then each slave finds the index of the word with the maximum similarity score and sends the word and the similarity score to master process.
Finally, master prints the similar words and values to screen and the loop executes again.

Outputs

When the input of the program is 11;

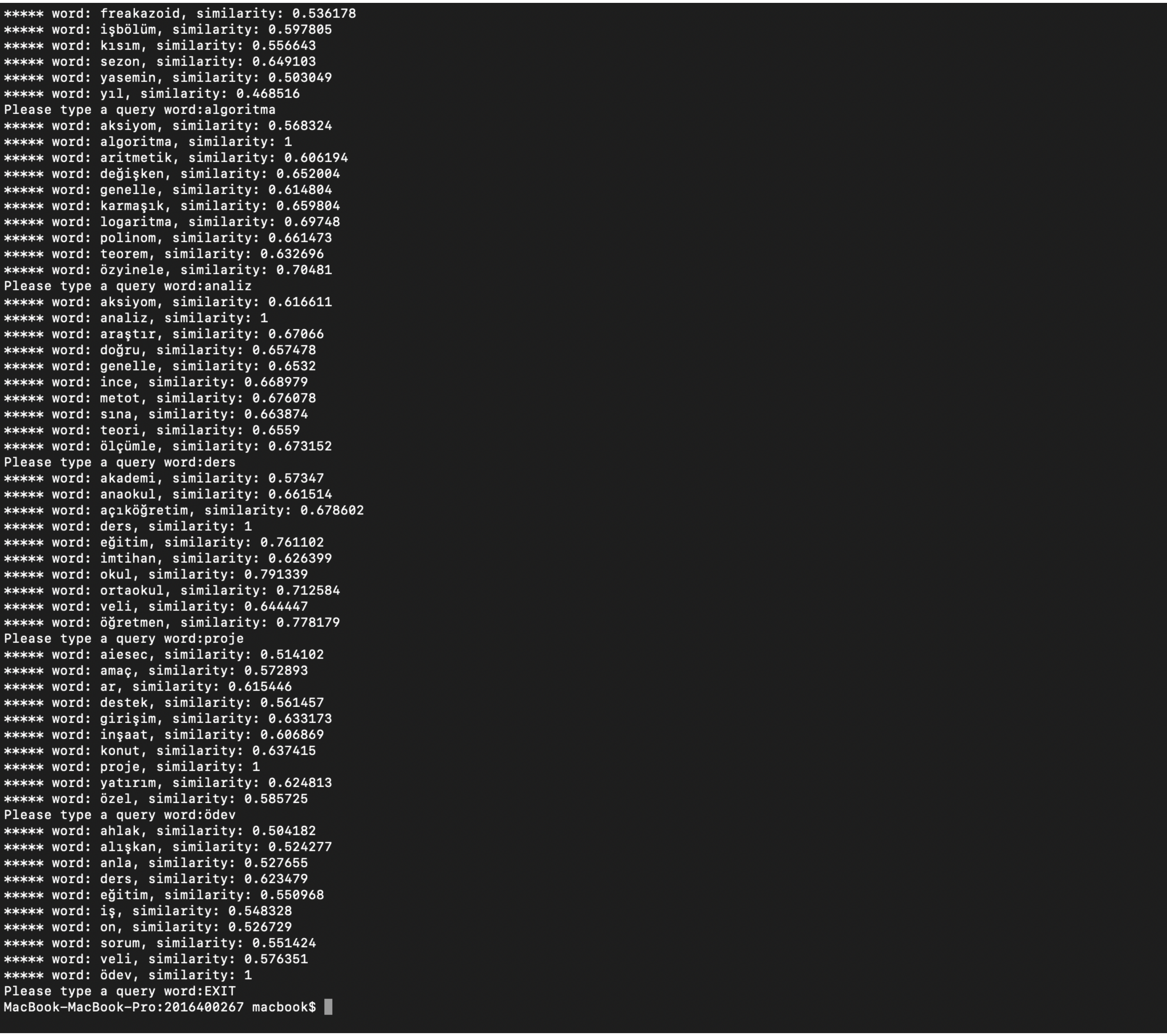
Concluesion

I have already hear of word embedding and I think it is a very interesting concept. Also, parallel programming experience was necessary for me and OpenMPI was a very useful framework to do so. Therefore implementing this program was both instructive and fun.



```
***** word: boğaziçi, similarity: 1
***** word: galata, similarity: 0.59201
***** word: istanbul, similarity: 0.630798
***** word: marmara, similarity: 0.63954
***** word: rumelihisarı, similarity: 0.644972
***** word: tarabya, similarity: 0.598414
***** word: üsküdar, similarity: 0.589913
Please type a query word:üniversite
***** word: akademisyen, similarity: 0.672108
***** word: anaokul, similarity: 0.573987
***** word: açıköğretim, similarity: 0.647896
***** word: burs, similarity: 0.716351
***** word: fakülte, similarity: 0.838503
***** word: kolej, similarity: 0.732595
***** word: mezun, similarity: 0.708246
***** word: rektör, similarity: 0.784244
***** word: yök, similarity: 0.666313
***** word: üniversite, similarity: 1
Please type a query word:bilgisayar
***** word: adresle, similarity: 0.603858
***** word: anahtarla, similarity: 0.561217
***** word: bilgisayar, similarity: 1
***** word: dizüstü, similarity: 0.761838
***** word: harddisk, similarity: 0.697283
***** word: internet, similarity: 0.660916
***** word: masaüst, similarity: 0.697266
***** word: pc, similarity: 0.640958
***** word: yazılım, similarity: 0.740554
***** word: çip, similarity: 0.704136
Please type a query word:mühendis
***** word: akademisyen, similarity: 0.561
***** word: analitik, similarity: 0.463634
***** word: başmühendis, similarity: 0.753365
***** word: biyoteknoloji, similarity: 0.560986
***** word: imalât, similarity: 0.558885
***** word: jeofizik, similarity: 0.608447
***** word: mühendis, similarity: 1
***** word: profesör, similarity: 0.570565
***** word: teknisyen, similarity: 0.711157
***** word: çalış, similarity: 0.538
Please type a query word:bölüm
***** word: adanmak, similarity: 0.419602
***** word: altbölüm, similarity: 0.578744
***** word: başkarakter, similarity: 0.550874
***** word: bölüm, similarity: 1
***** word: freakazoid, similarity: 0.536178
***** word: işbölüm, similarity: 0.597805
***** word: kısım, similarity: 0.556643
***** word: sezon, similarity: 0.649103
***** word: yasemin, similarity: 0.503049
***** word: yıl, similarity: 0.468516
Please type a query word:algoritma
***** word: aksiyom, similarity: 0.568324
***** word: algoritma, similarity: 1
***** word: aritmetik, similarity: 0.606194
***** word: değişken, similarity: 0.652004
***** word: genelle, similarity: 0.614804
***** word: karmaşık, similarity: 0.659804
***** word: logaritma, similarity: 0.69748
***** word: polinom, similarity: 0.661473
***** word: teorem, similarity: 0.632696
***** word: özyinele, similarity: 0.70481
Please type a query word:
```

Figure 1:



```
***** word: freakazoid, similarity: 0.536178
***** word: işbölüm, similarity: 0.597805
***** word: kısım, similarity: 0.556643
***** word: sezon, similarity: 0.649103
***** word: yasemin, similarity: 0.503049
***** word: yıl, similarity: 0.468516
Please type a query word:algoritma
***** word: aksiyom, similarity: 0.568324
***** word: algoritma, similarity: 1
***** word: aritmetik, similarity: 0.606194
***** word: değişken, similarity: 0.652004
***** word: genelle, similarity: 0.614804
***** word: karmaşık, similarity: 0.659804
***** word: logaritma, similarity: 0.69748
***** word: polinom, similarity: 0.661473
***** word: teorem, similarity: 0.632696
***** word: özyinele, similarity: 0.70481
Please type a query word:analiz
***** word: aksiyom, similarity: 0.616611
***** word: analiz, similarity: 1
***** word: araştır, similarity: 0.67066
***** word: doğru, similarity: 0.657478
***** word: genelle, similarity: 0.6532
***** word: ince, similarity: 0.668979
***** word: metot, similarity: 0.676078
***** word: sına, similarity: 0.663874
***** word: teori, similarity: 0.6559
***** word: ölçümle, similarity: 0.673152
Please type a query word:ders
***** word: akademi, similarity: 0.57347
***** word: anaokul, similarity: 0.661514
***** word: açıköğretim, similarity: 0.678602
***** word: ders, similarity: 1
***** word: eğitim, similarity: 0.761102
***** word: imtihan, similarity: 0.626399
***** word: okul, similarity: 0.791339
***** word: ortaokul, similarity: 0.712584
***** word: veli, similarity: 0.644447
***** word: öğretmen, similarity: 0.778179
Please type a query word:proje
***** word: aiesec, similarity: 0.514102
***** word: amaç, similarity: 0.572893
***** word: ar, similarity: 0.615446
***** word: destek, similarity: 0.561457
***** word: girişim, similarity: 0.633173
***** word: inşaat, similarity: 0.606869
***** word: konut, similarity: 0.637415
***** word: proje, similarity: 1
***** word: yatırım, similarity: 0.624813
***** word: özel, similarity: 0.585725
Please type a query word:ödev
***** word: ahlak, similarity: 0.504182
***** word: alışkan, similarity: 0.524277
***** word: anla, similarity: 0.527655
***** word: ders, similarity: 0.623479
***** word: eğitim, similarity: 0.550968
***** word: iş, similarity: 0.548328
***** word: on, similarity: 0.526729
***** word: sorum, similarity: 0.551424
***** word: veli, similarity: 0.576351
***** word: ödev, similarity: 1
Please type a query word:EXIT
MacBook-MacBook-Pro:2016400267 macbook$
```

Figure 2: