Environment Setup and DEMO

1 - Download Git

https://git-scm.com/

2 - Download Git LFS

https://git-lfs.github.com/

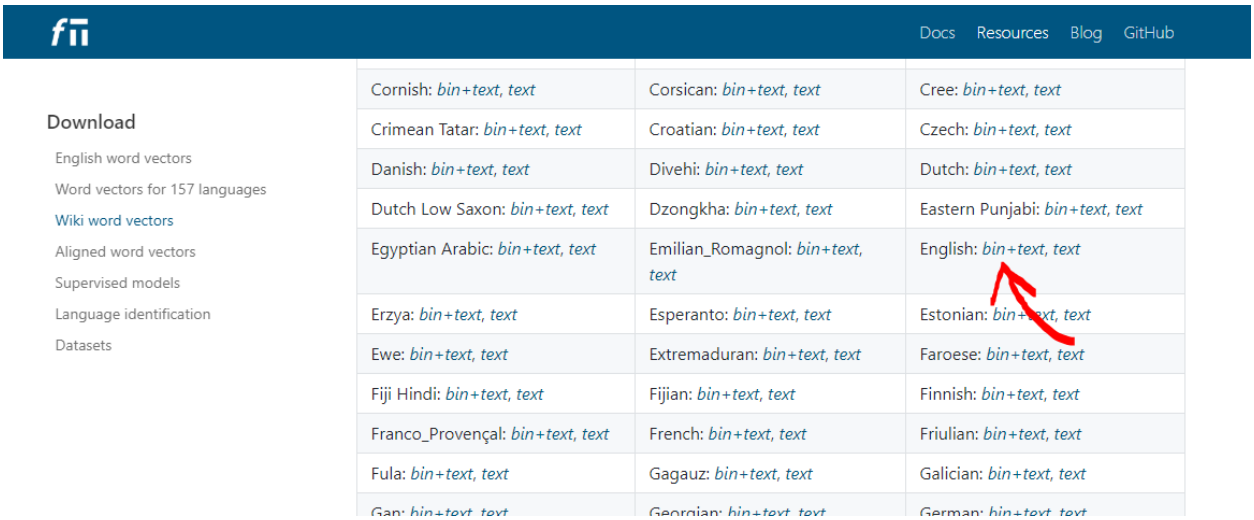
3 - Clone the project

Open the cmd at a desired location and type the command:

"git clone https://github.com/amandagdias/TRIVIR.git"

4- Download the pre-trained word vectors

https://fasttext.cc/docs/en/pretrained-vectors.html



5 - Place the file "wiki.en.bin" inside the scripts folder: TRIVIR/Server/scripts

6 - Download Node.js

https://nodejs.org/en/

7 - Download R (3.5.0), RStudio (recommended) and RTools

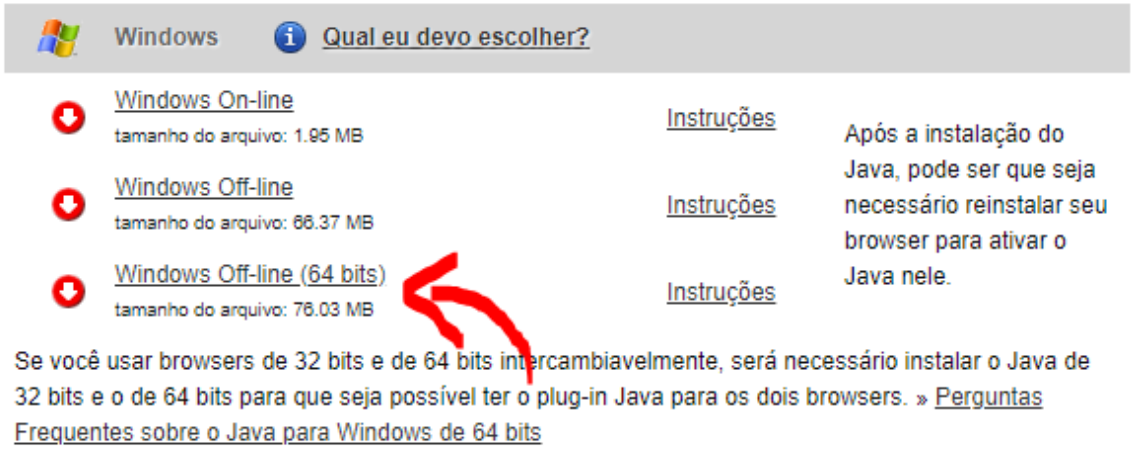
https://cran.r-project.org/bin/windows/base/old/

https://www.rstudio.com/

https://cran.r-project.org/bin/windows/Rtools/

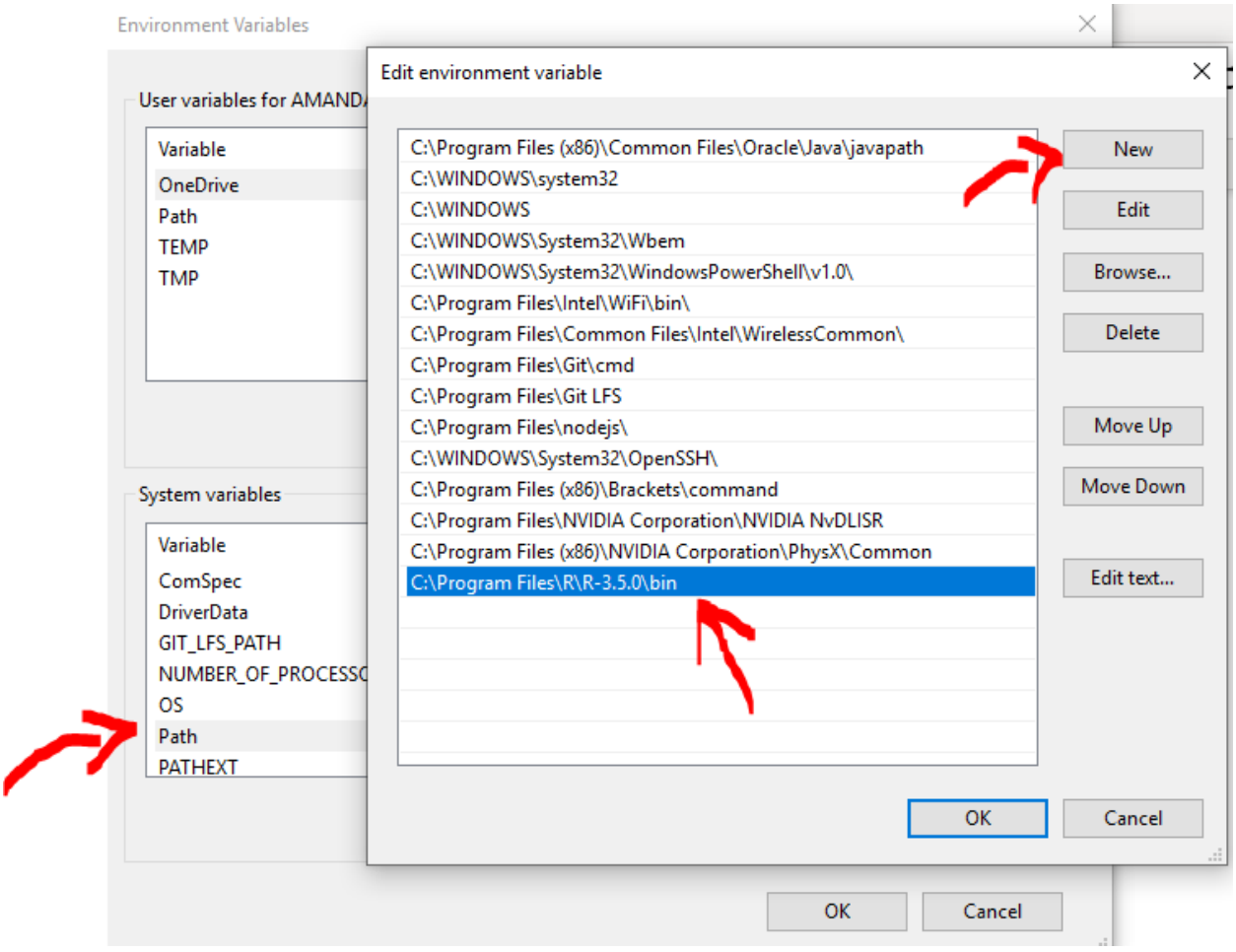
8 - Download Java jre

https://www.java.com/pt_BR/download/manual.jsp



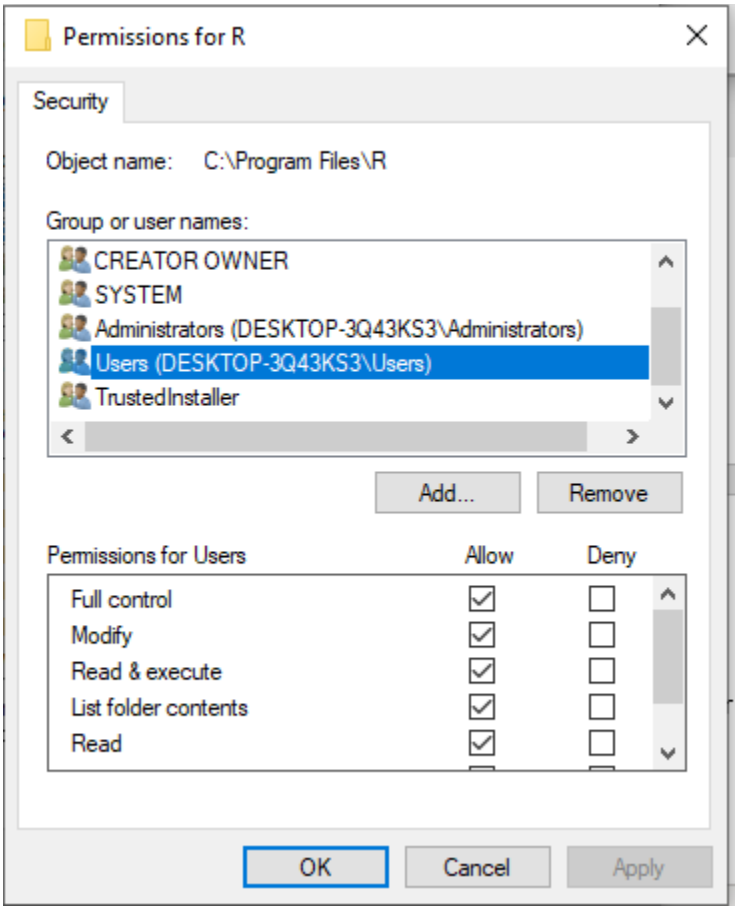
9 - Add R into your system's variables

This PC (right click) -> Properties -> Advanced system settings -> Environment variables…



10 - Give permission to the R folder

C:/Program Files/R (right click) -> Properties -> Security -> Edit -> Allow Full control



11 - Install node_modules

Go to TRIVIR/Server

Open cmd in that folder

Type command “npm install”

12 - Install R Packages

Run script installpackages.R at TRIVIR/Server/scripts (this might take a while)

13 - Select file location

Open index.js file at TRIVIR/Server

Change the *workingdir* variable to the location of the scripts folder

```
1
2   var express = require('express');
3   var app = express();
4   var bodyParser = require('body-parser');
5   var fs = require('fs');
6   var R = require('r-script');
7   var pathlib = require('path');
8
9   var base = "../../data/demo/CBR-837Aam274-288.txt";
10  var corpus = "../../data/demo";
11  var username = 'test';
12
13
14  var path_core = "../../core/"+pathlib.basename(corpus);
15  var path_users = "../../file/"+pathlib.basename(corpus)+"/"+username;
16  app.use(bodyParser.urlencoded({ extended: true }));
17
18  var projtech = "tsne";
19  var embtech = "bagofwords"; // or "word_embeddings"
20  var workingdir = "C:/Users/AMANDA-PC/Documents/TRIVIR/Server/scripts";
21
```

14 - Start server

In the same cmd, type command “npm start”

15 - Open client

Go to TRIVIR/Client/trivir.html

What you should see at the cmd:

```
PS C:\Users\AMANDA-PC\Documents\TRIVIR\Server> npm start

> server@1.0.0 start C:\Users\AMANDA-PC\Documents\TRIVIR\Server
> nodemon index.js

[nodemon] 1.19.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node index.js`
Listening on port 3000!
Initializing signature
Last Sync: 9/5/2019 @ 13:31:41
```

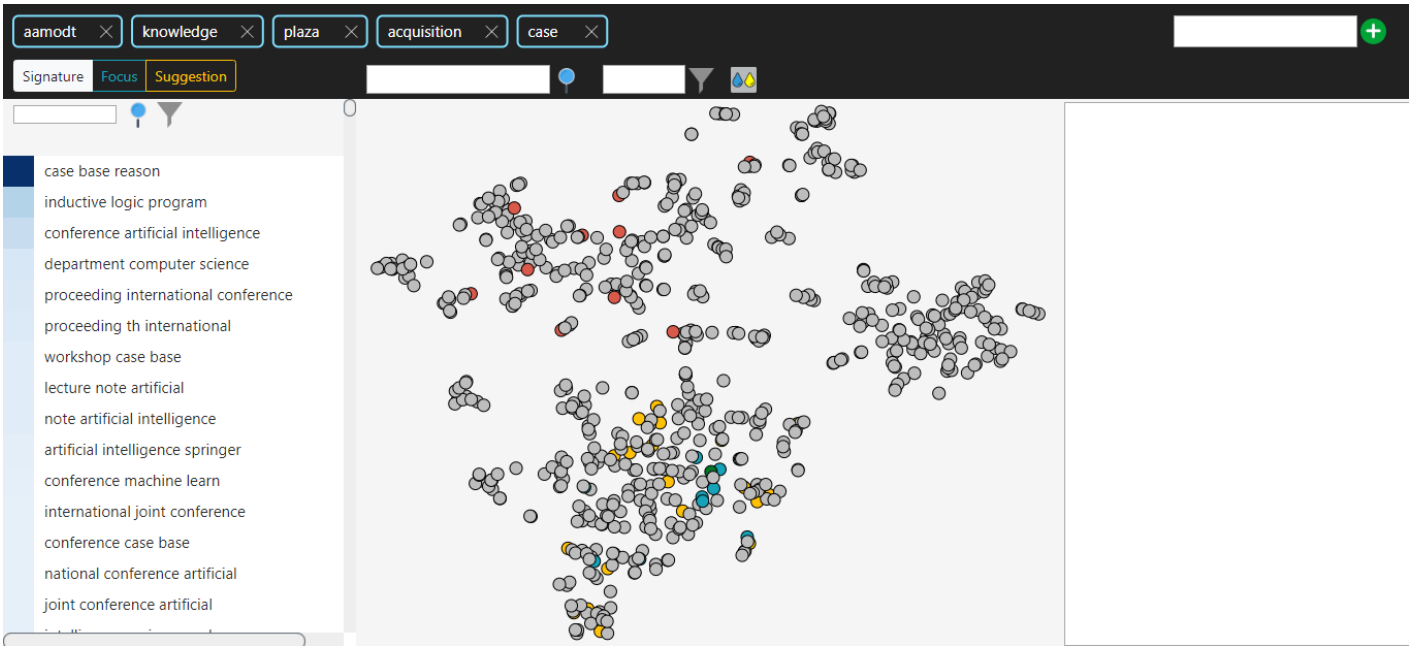
Note: At the first time running the server, it might install some R dependencies and take longer to load (approximately 12 minutes). If you receive these messages:

```
PS C:\Users\AMANDA-PC\Documents\TRIVIR\Server> npm start

> server@1.0.0 start C:\Users\AMANDA-PC\Documents\TRIVIR\Server
> nodemon index.js

[nodemon] 1.19.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching: *.*
[nodemon] starting `node index.js`
Listening on port 3000!
Initializing signature
Last Sync: 9/5/2019 @ 13:31:41


also installing the dependencies 'BH', 'plogr', 'digest', 'sylly', 'data.table', 'syuzhet', 'extrafont', 'fastmatch', 'ggplot2
', 'ggrepel', 'lubridate', 'network', 'RSpectra', 'RcppParallel', 'sna', 'spacyr', 'stopwords', 'xml2', 'yaml', 'RcppArmadillo
', 'english', 'mgsub', 'qdapRegex', 'slam', 'koRpus.lang.en', 'dplyr', 'hunspell', 'koRpus', 'lexicon', 'quanteda', 'SnowballC
', 'textclean', 'textshape'

also installing the dependencies 'openNLPdata', 'rJava'

also installing the dependency 'XML'

also installing the dependencies 'sys', 'askpass', 'curl', 'mime', 'openssl', 'gtools', 'httr', 'bitops', 'xlsxjars', 'qdapDic
tionaries', 'qdapTools', 'chron', 'gdata', 'gender', 'gridExtra', 'igraph', 'plotrix', 'RCurl', 'reports', 'stringdist', 'venn
euler', 'wordcloud', 'xlsx'
```

Just restart the server with command “rs” and refresh the trivir.html page

Expected outcome:

```
rs
[nodemon] starting `node index.js`
Listening on port 3000!
Initializing signature
Last Sync: 9/5/2019 @ 13:45:16
out
success
success
Last Sync: 9/5/2019 @ 13:45:24
Retrieving terms
success
Initializing focus list
success
Initializing suggestion list
success
Initializing scatterplot
Getting list of not relevant documents
success
success
Last Sync: 9/5/2019 @ 13:46:17
```


aamodt
knowledge
plaza
acquisition
case
Signature
Focus
Suggestion
case base reason
inductive logic program
conference artificial intelligence
department computer science
proceeding international conference
proceeding th international
workshop case base
lecture note artificial
note artificial intelligence
artificial intelligence springer
conference machine learn
international joint conference
conference case base
national conference artificial
joint conference artificial

Understanding the files and running other examples

Index.js at TRIVIR/Server is where all communications between the Client, files, and the R scripts happens.

The corpus should be placed as .txt files inside the data folder at TRIVIR/data/name_of_corpus/

In order to run the system considering a different corpus, open the index.js script and change the variables *base* (which is the query document) and *corpus.* You can also choose a *username*.

```
var express = require('express');
var app = express();
var bodyParser = require('body-parser');
var fs = require('fs');
var R = require('r-script');
var pathlib = require('path');

var base = "../../data/demo/CBR-837Aam274-288.txt";
var corpus = "../../data/demo";
var username = 'test';


var path_core = "../../core/"+pathlib.basename(corpus);
var path_users = "../../file/"+pathlib.basename(corpus)+"/"+username;
app.use(bodyParser.urlencoded({ extended: true }));

var projtech = "tsne"; // or "lsp"
var embtech = "bagofwords"; // or "word_embeddings"
var workingdir = "C:/Users/AMANDA-PC/Documents/TRIVIR/Server/scripts";

```

The projection technique and document representation can be chosen by changing the variables *projtech* and *embtech*.

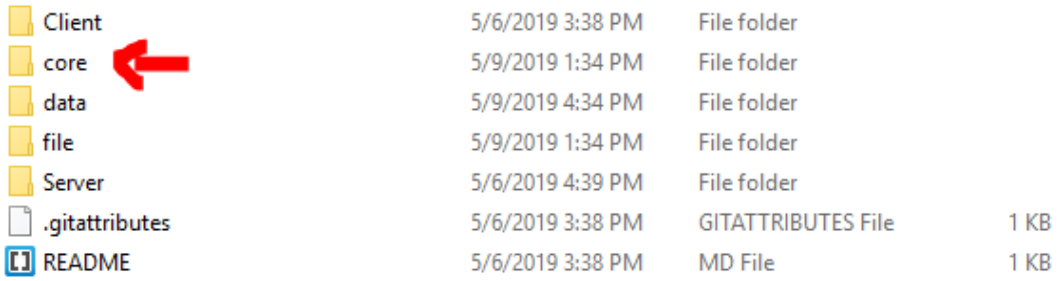
```
var express = require('express');
var app = express();
var bodyParser = require('body-parser');
var fs = require('fs');
var R = require('r-script');
var pathlib = require('path');

var base = "../../data/demo/CBR-837Aam274-288.txt";
var corpus = "../../data/demo";
var username = 'test';


var path_core = "../../core/"+pathlib.basename(corpus);
var path_users = "../../file/"+pathlib.basename(corpus)+"/"+username;
app.use(bodyParser.urlencoded({ extended: true }));

var projtech = "tsne"; // or "lsp"
var embtech = "bagofwords"; // or "word_embeddings"
var workingdir = "C:/Users/AMANDA-PC/Documents/TRIVIR/Server/scripts";

```

In case you have a .csv from IEEE, Parsifal, Scopus or a .bib from Web of Science and want to include this data into the system, open the script “extractCorpus.R” at TRIVIR/Server/scripts and change the last line (158) that calls the function *extractCorpus* where the first parameter is the location of your .csv or .bib file, the second parameter is a name for the corpus, and the third parameter is the source from which you retrieve the corpus. For now, the available sources are: “ieee”, “parsifal”, “scopus”, and “wos”.

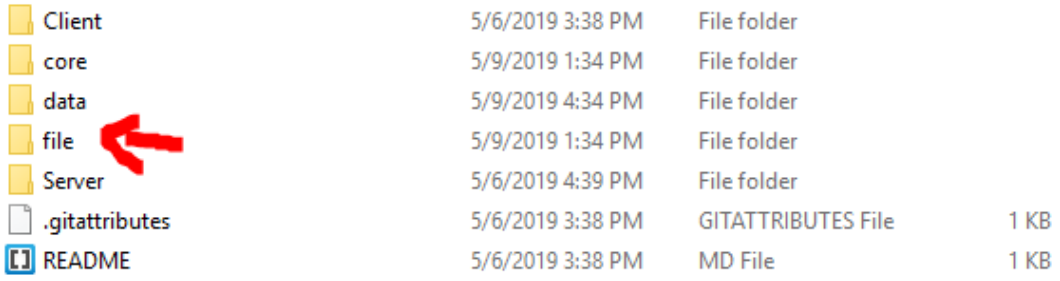
```
143     keywords <- iconv(M[i,  DE ], to =  utf8 );
144     #article <- paste(title, author, year, abstract, keywords, references, sep = "\n");
145     article <- paste(title, author, year, abstract, keywords, sep = "\n");
146
147     title <- defaultPreprocess(title, FALSE)
148     if (nchar(title) > 100){
149       title <- substring(title, 1, 100);
150     }
151     conn<-file(sprintf("../../data/%s/%s.txt",dirName, title), encoding = "utf8")
152
153     writeLines(article, conn)
154     close(conn)
155   }
156 }
157
158 extractCorpus("C:/Users/AMANDA-PC/Documents/savedrecs.bib", "WOS corpus", "wos")
159
```

After you run the script, you should see the folder with the name of your corpus at TRIVIR/data

Inside the “core” directory, are all files related to each corpus.



Inside the “file” directory, are all user related files (each user gets a directory).



The directories: core, data, file, and Server are only accessible through the server side.

Obs: In the future, we want to set up a public server and allow the user to access TRIVIR from a web browser. Interfaces for the user to add the corpus and other information would be necessary.