Data visualization project

Etymology graph

Nicolas Pradignac, Pierre Mahmoud, Christopher Benz

Overview

None of us are linguists per se, but we all had a great interest on trying to display in a graph the etymologies of words. The very original idea was to display a giant graph representing all the words of the English language, interconnected by their etymologies, roots and history. For reasons explained later, the final idea diverged and we came up with a visualization that allows the user to enter a word and see all its ancestors. He can then click on an ancestor which display the descendants of the

An initial research showed that visualization of etymologies is very difficult to find on the internet. The main reason for this is the lack of database with clear etymologies and links between words.

The main target audience is obviously linguists, as a dynamic tool for etymology visualization and exploration can be of an immense help instead of searching through dictionaries or clicking links on Wiktionary. But it is also interesting for language learners to help them understand the history of a word, or find words that can be etymologically related, which can make the learning process easier for beginners and advanced learners. It is also a fun way for any speaker to explore common words in his own language and discover links that we might not have suspected.

For all these reasons, we went on with the goal of displaying etymology links between words in a graph visualization.

Technologies and dataset

We use the basic techs introduced during the course and build our project upon: ES6, Babel, Webpack. Our website consists of a Github Pages website with Jekyll,
For visualization we explored many different possibilities such as d3, d3-dagre, cytoscape, cola.js.

At the very beginning of the project, we had a dataset comprised of a single text file[1] with every line representing an etymological link between two words, in many languages. This dataset was quickly abandoned for the following reasons:

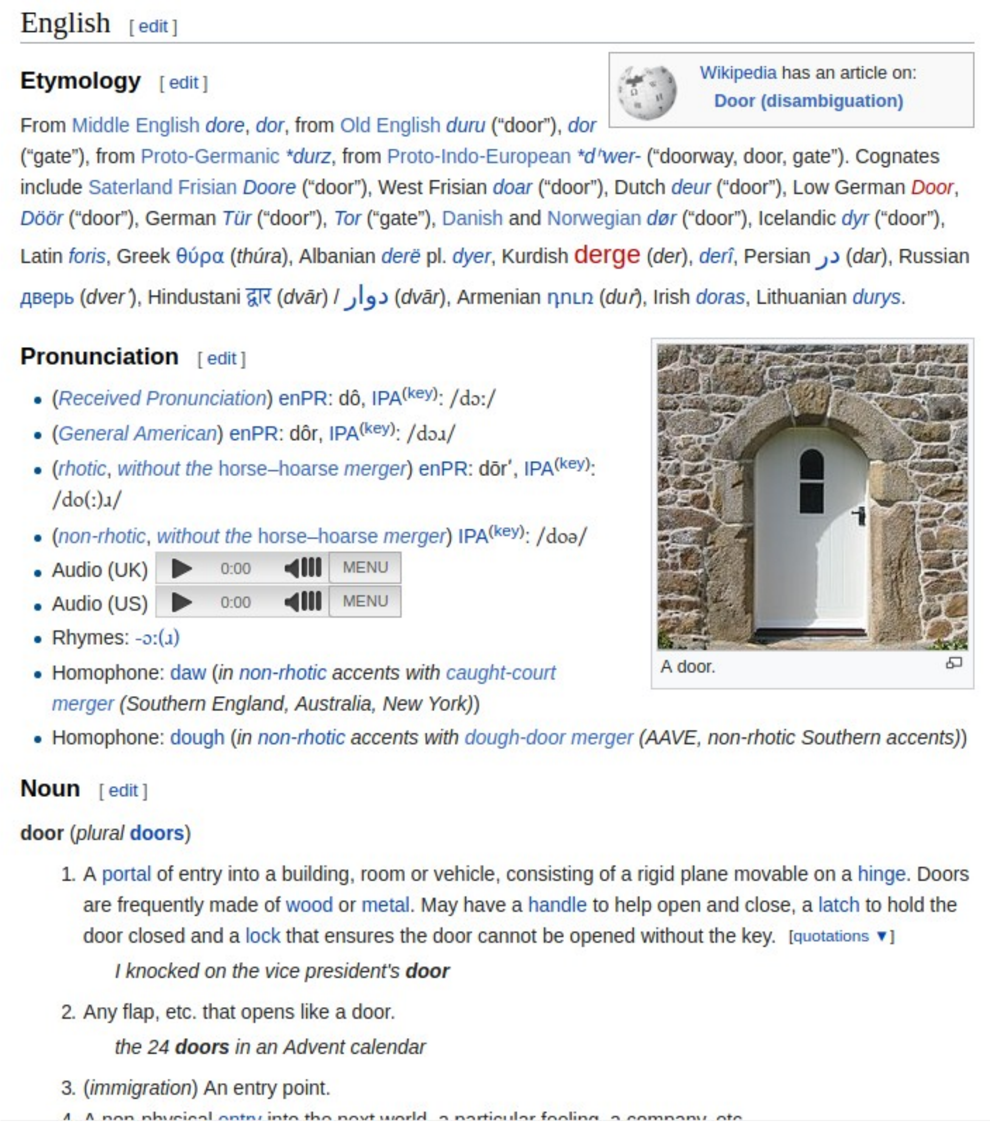
- Not up-to-date: the dataset was created in 2013 from the English Wiktionary, which is a platform that changes and evolves consistently from user contributions. Most of the data wasn't consistent with the current state of data on Wiktionary (end 2017)
- Not much information about the type of the etymology:

[1] http://icsi.berkeley.edu/~demelo/etymwn/

- No extra metadata such as date, link to the Wiktionary page, etc.
- Homonyms were not differentiable, because the plain word was given without ID
- Another, more powerful dataset was found, as we describe below

Therefore, after trying to work with this file, we rapidly understood it wasn't enough for our goals.

We searched more and found a better dataset, from the work of an Italian PhD who is currently working, in collaboration with the Wikimedia Foundation, on the extraction, creation and visualization of etymologies[2]. As far as we know, this is one of the only projects working on the visualization of etymologies.
The main difficulty is that no real database of etymology links exists. This PhD is creating such a database by scrapping the Wiktionary website and extracting information with Natural Language Processing based on text patterns under the 'etymology' field. As seen in Illustration 1, such entries are often quite complex and are written by other linguists, but not consistently with the same text patterns.



English [edit]

Etymology [edit]

Wikipedia has an article on:
Door (disambiguation)

From Middle English *dore*, *dor*, from Old English *duru* ("door"), *dor* ("gate"), from Proto-Germanic **durz*, from Proto-Indo-European **dʰwer-* ("doorway, door, gate"). Cognates include Saterland Frisian *Doore* ("door"), West Frisian *doar* ("door"), Dutch *deur* ("door"), Low German *Door*, *Döör* ("door"), German *Tür* ("door"), *Tor* ("gate"), Danish and Norwegian *dør* ("door"), Icelandic *dyr* ("door"), Latin *foris*, Greek θύρα (*thúra*), Albanian *derë* pl. *dyer*, Kurdish derge (*der*), *derî*, Persian در (*dar*), Russian дверь (*dverʹ*), Hindustani द्वार (*dvār*) / دوار (*dvār*), Armenian դուռ (*duṙ*), Irish *doras*, Lithuanian *durys*.

Pronunciation [edit]

- (*Received Pronunciation*) enPR: dô, IPA(key): /dɔː/
- (*General American*) enPR: dôr, IPA(key): /dɔɹ/
- (*rhotic*, *without the horse–hoarse merger*) enPR: dōr', IPA(key): /do(ː)ɹ/
- (*non-rhotic*, *without the horse–hoarse merger*) IPA(key): /doə/
- Audio (UK) ▶ 0:00 MENU
- Audio (US) ▶ 0:00 MENU
- Rhymes: -ɔː(ɹ)
- Homophone: daw (*in non-rhotic accents with caught-court merger (Southern England, Australia, New York)*)
- Homophone: dough (*in non-rhotic accents with dough-door merger (AAVE, non-rhotic Southern accents)*)

A door.

Noun [edit]

door (*plural* **doors**)

1. A portal of entry into a building, room or vehicle, consisting of a rigid plane movable on a hinge. Doors are frequently made of wood or metal. May have a handle to help open and close, a latch to hold the door closed and a lock that ensures the door cannot be opened without the key. [quotations ▼]
   I knocked on the vice president's ***door***
2. Any flap, etc. that opens like a door.
   the 24 ***doors*** *in an Advent calendar*
3. (*immigration*) An entry point.
4. A non-physical entry into the next world, a particular feeling, a company, etc.

Illustration 1: Wiktionary entry for the English word 'door'. We can observe that the etymology information is dense and not simple.

[2] https://meta.wikimedia.org/wiki/Grants:IEG/A_visual_representation_of_the_etymology_of_words_using_trees

The disadvantage of this dataset is that because it is an ongoing project, the database is still in its early phase, and we had to create complex custom queries to get the information we needed, and have to query this external database for each word.

User experience flow

The user enters a word in a search box. We query the database for all the ancestors of the word recursively, then display the graph. The user can then observe the etymology tree, can access the Wiktionary page for each word to get further information and the word's definition.

When he clicks on a node, we query the database for the descendants of the clicked word, then query again for the ancestors of the descendants and create a descendant tree which is displayed on top of the existing graph. The user can continue exploring the etymologies this way.

Process

After finding our dataset as described above, we still had to do a lot of exploration on the PhD's work to find our the queries that had to be done to get etymology information about words.
We could query the database with a word, and it sent back first-degree etymology links for that word.
Therefore, to create the full ancestor tree for a given word, we have to recursively query the database to get the ancestors

We needed the following features for our graph:

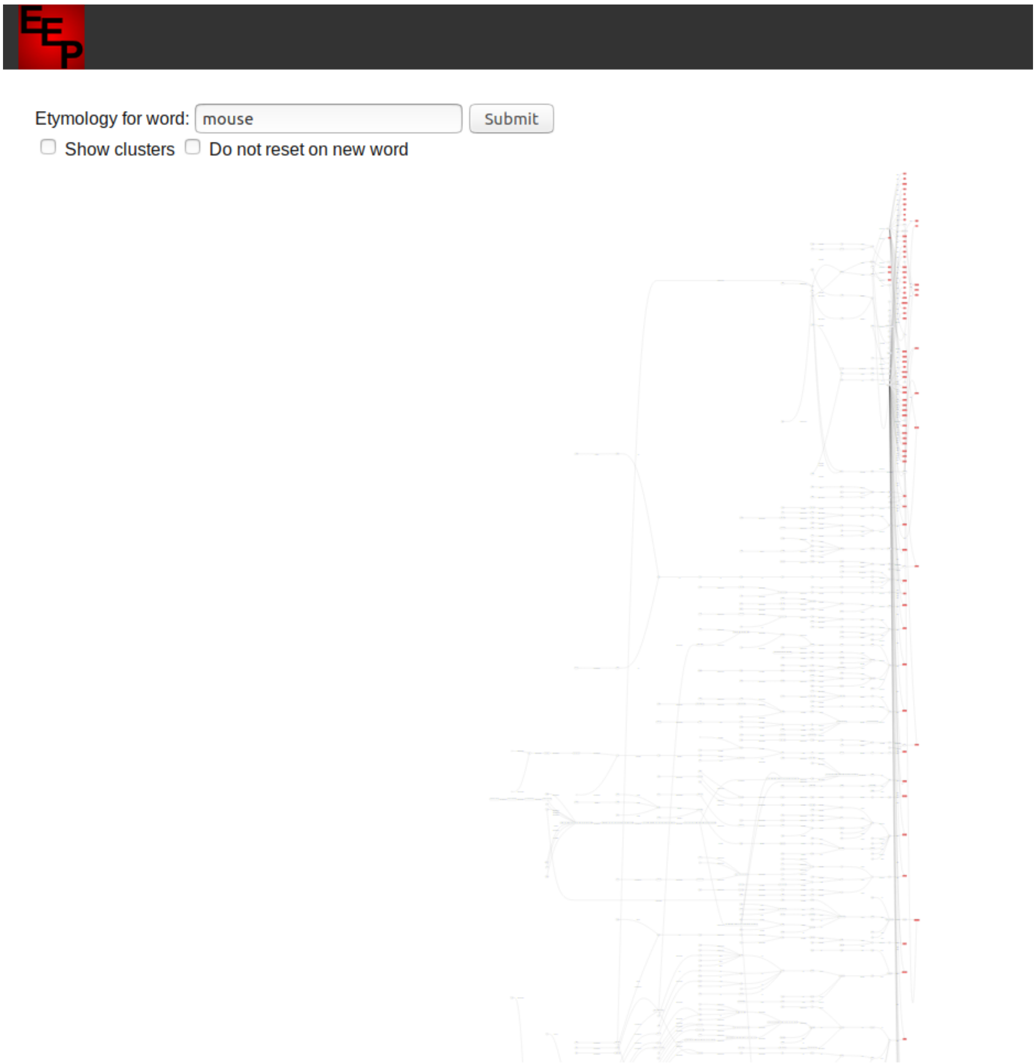
1. Possibility to display the tree by keeping a certain structure, as etymologies have a direction (newest to oldest). We therefore couldn't let a force layout take entirely care of the graph form, or the direction of the etymologies were difficult to understand
2. Cluster nodes, to group them by language
3. Letting the user click and explore the graph
4. Display a text/link/sound when hovering over a node, for example show the Wiktionary definition or extract.
5. Animation/transitions when exploring the graph and showing ancestors/descendants

At first, we tried using the basic d3 tree for our graph. It became quickly evident that it was lacking some features we needed. First of all, it's a strict tree, in the sense that there can't be any cycles, which are needed for etymologies. Secondly, the node layouts had to be done manually and were difficult to work with: centering texts in nodes, adding arrows to edges, etc.

We needed to use an external library that helps us displaying graphs as we needed: some root words have ancestors that themselves have the same ancestor, and the graph should be directed, with direction going from the root word to its ancestor. What we needed were Directed Acyclic Graphs (DAG).

We found a library that manages this type of structure[3], *dagre* by cpettitt, and the same developer even has a library to make it work with d3[4], *dagre-d3*! It was written for d3v3, but somebody created a version compatible with d3v4. We still had issues because of webpack compatibility but fortunately another fork was created for webpack and d3v4[5].

Now we wanted to glue the data and the view together. We wrote the code, and for some reason, because of a misunderstanding due to our inexperience, we were creating a JSON string representing the tree data structure from the database queries, then iterating over the JSON and creating the nodes and edges for our graph. This was a bad idea – and a completely useless one – that led to many complications: it wasn't possible with our mechanism to easily add another tree data structure to the graph, for example for descendants of a word.
Also, the graph was often very long or large, which we thought didn't make a good user experience: it was required to pan the graph a lot during exploration. See Illustration 2 for an example.



Etymology for word: mouse

□ Show clusters □ Do not reset on new word

Illustration 2: Extreme example of a large graph

[3] https://github.com/cpettitt/dagre
[4] https://github.com/cpettitt/dagre-d3
[5] https://github.com/mmoscher/dagre-d3v4-webpack

We therefore wanted to modify our graph visualization. Our idea was to show a radial graph with the root word at its center, and the ancestors distributed concentrically by level. This way, the graph would be more easily visible in its entirety. We could also create other radial graphs for different words that are interconnected by their common ancestors. See Illustration 3 and Illustration 4 for what we had in mind.

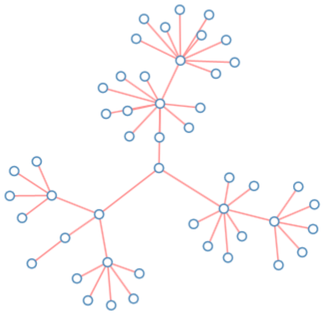


Illustration 4: What we were thinking of displaying

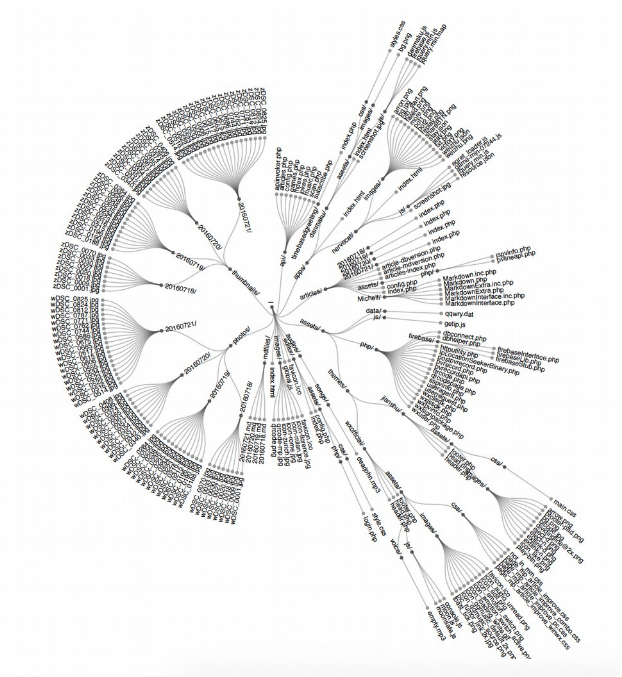
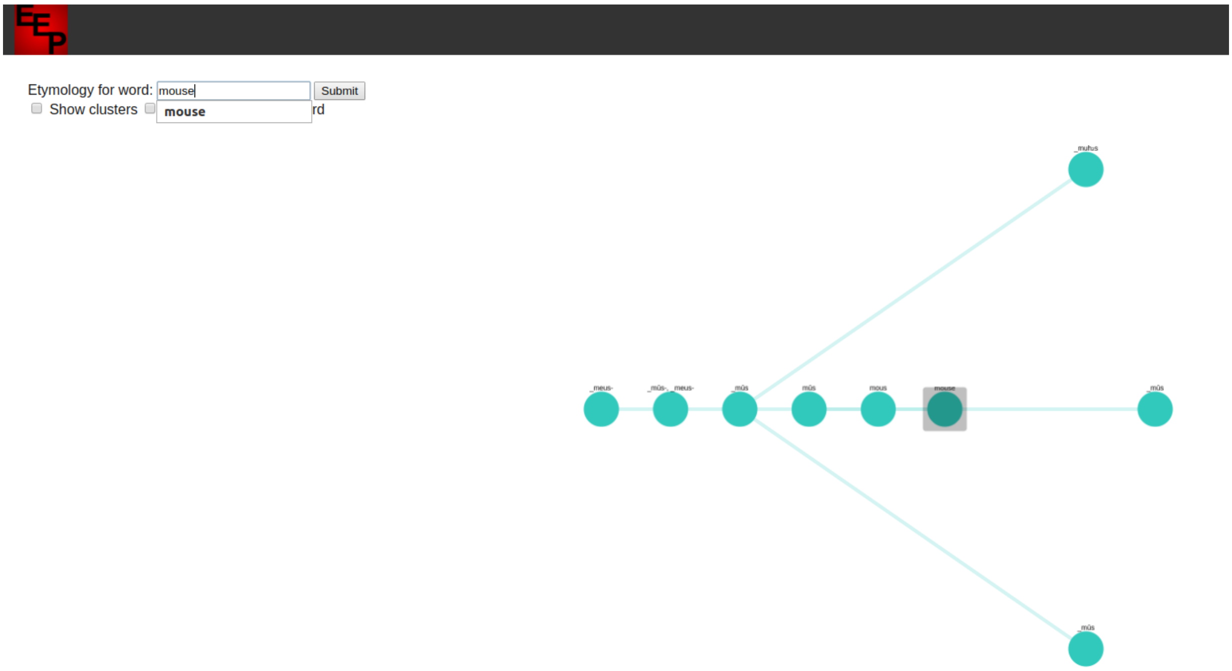


Illustration 3: What we were thinking of displaying

We liked this idea but it created other issues: would nodes still be readable while showing the full graph? And how would we manage the case where the users clicks on an ancestor of the root word: put the clicked node at the center of another radial graph that is connected to the first? But then the center of a radial graph represents both root ancestor and root descendant, which is not intuitive.

After discussion with the course’s TA, he also gave us the idea of displaying two graphs: on the right a radial graph with all the nodes and clicked nodes and ancestors and descendants; on the left only the clicked word and its ancestors.



Etymology for word: mouse

□ Show clusters □ rd

Illustration 5: Cytoscape.js radial graph for word ‘mouse’: not what we expected

To create this we used another library that was advertised in class: cytoscape.js. We had to modify and migrate parts of the code to make it work with cytoscape.

Unfortunately, the result wasn’t as expected (see Illustration 5). The etymology trees don’t have enough children at each node to display elegantly on a radial graph. They often have a single ancestor so the graph didn’t look radial but more linear in a direction.

We abandoned the idea of using a radial or concentric graph. We still tried using cytoscape because now we had migrated the code and cytoscape offered other solutions to our visualization, as well as built-in layouts. See Illustration 6 and Illustration 7.

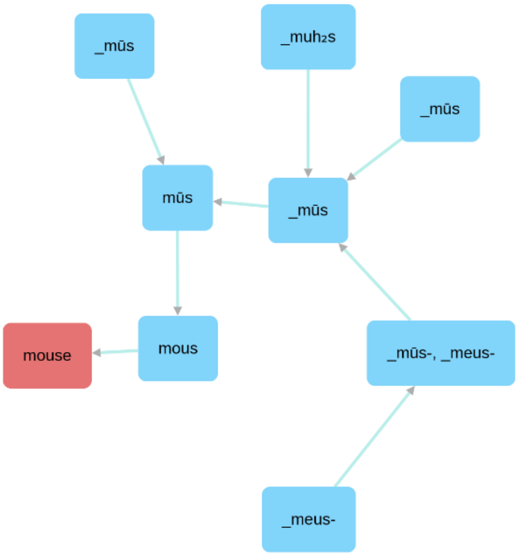


Illustration 6: cytoscape cose layout

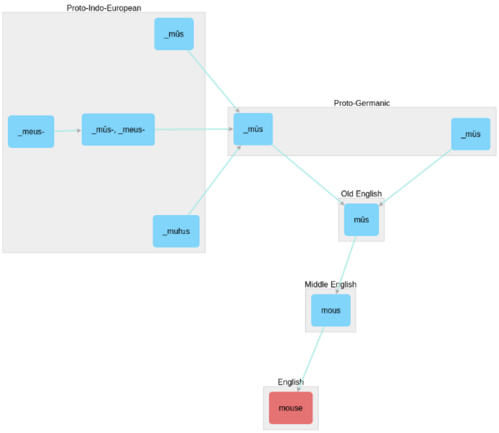


Illustration 7: cytoscape cose layout with clusters

As we can observe, cose layouts are less spread over the canvas, but are not intuitive for etymology trees: we cannot easily infer the history of the word.

We turned ourselves to yet another library, cola.js. We migrated the code and tried making it work with cola. Unfortunately, after hours of trying, we still don’t understand why but we didn’t make it work with version 4 of d3, and didn’t want to use d3v3, as d3v4 was a requirement for this project.

After discussion with the professor, he convinced us simply using d3 should allow us to do everything we needed. We thus came back to d3 and saw indeed that stuff we didn’t think was possible was in fact possible. There we were back at our starting point.

We finally settled with using the dagre-d3 library for the layout and d3v4 for the rendering to obtain the result at Illustration 8.

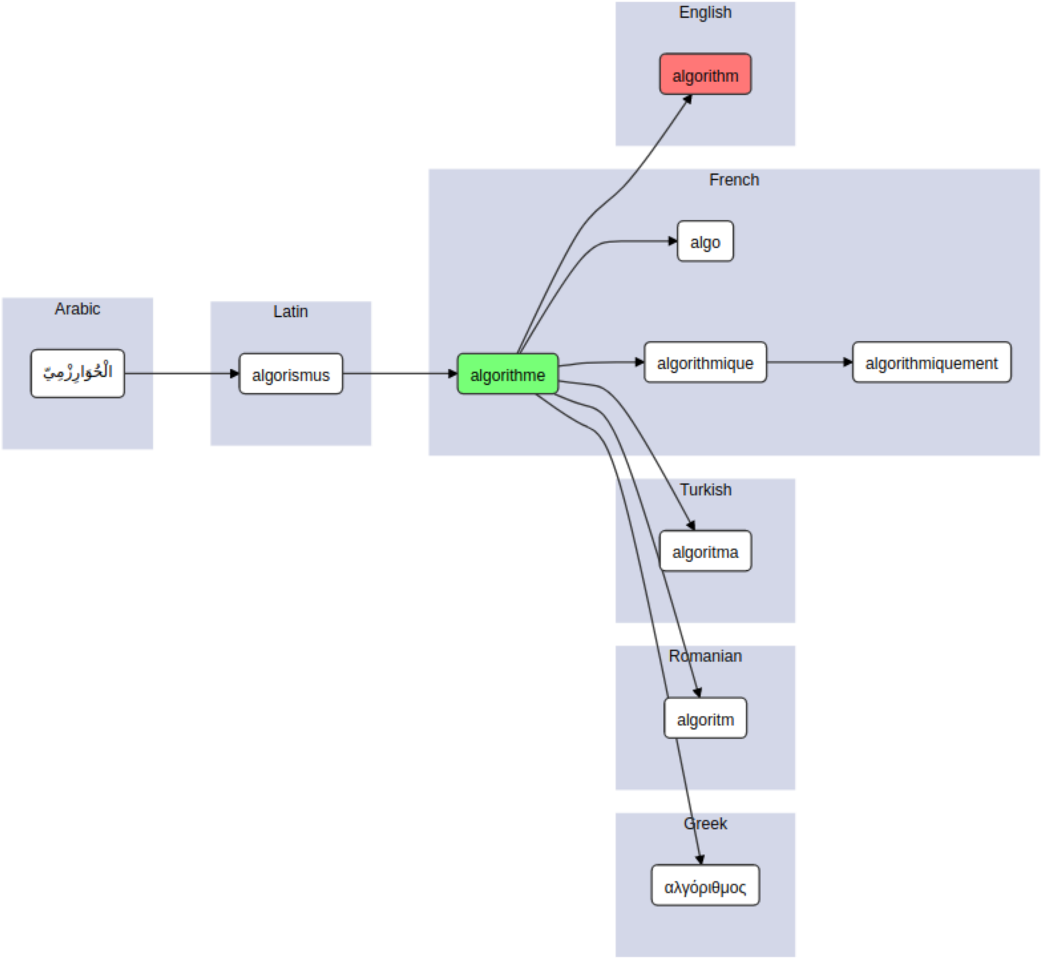


Illustration 8: Final layout(search for ‘algorithm’)

We colored nodes in red to indicate a root ancestor word, or a word which ancestors were searched. Thus, the initial inputed word is a red node.

When the user clicks on a node, we search the descendants of the word and colorize the clicked node in green. This lets the user keep track of how he is exploring the graph.

We also transition the view to center it on the new root word, so that the user doesn't get lost.

There is an option to cluster the nodes by language. This allows to regroup the nodes and visualize dependencies language-wise.

We added a 'Center on root node' button that allows the user to center the visualization on the current root word at any moment, in case he wants to come back to it.

If wanted, the user can check the 'do not reset on new word' box to keep the graph when inputting a new word.

Code

We have 4 main files (which can be found in /assets/):

- ***dagre_vizu***, which represents our visualization. It initializes d3, has methods to add nodes and edges to the graph. It is the view component
- ***etymology_tree.js***, which takes care of querying the database and creating the data structure for ancestor look-ups
- ***descendants.js***, which queries the database for descendants and create the descendant tree structure
- ***utils.js***, which contains utility methods

Conclusion

While that was not demanded for this project, the data exploration took a great part of our time (~25%) due to the lack of existing etymology information with a relational model. We had to search for a capable database, find out the correct queries to make to it by reverse engineering, and build the etymology tree ourselves.
After that, we encountered many problems due to limitations (or incomprehension from our part) of the used libraries, and exploring different solutions took another extra time because we had to rewrite parts of the code and the data had to be reformatted every time.

While the end result could be improved if we gave it additional time, we are satisfied with the work done during the semester; this project offered problems that were at the core of visualization of etymology data.

We estimate to have completed the goal of offering an intuitive way of searching and exploring etymologies, but stay inspired with additional features and ideas we have that could improve our project onwards.
Our visualization can greatly help linguists or language learners by letting them explore etymologies in a new way with an interactive graph.

Further work

From the current state of the project, we see many extra ideas and features that we could add to it:

- A minimap on the edge of the screen, which shows the current display position in the entire graph, so that the user doesn't get lost.
- A horizontal scale of the year, and each node is placed accordingly on this scale. This would allow to see how far in time different words and etymologies are related. But technical issues we see with this solution are for example how to display when multiple words that follow each other etymologically are from the same language, or how to show the graph if the languages are very far from each other (e.g. if a word jumps from modern English to a very old language).
- Support multiple languages. Currently the project only works with English words as input. By modifying the query we could use other languages.
- Display more information in the tool-tip, such as synonyms, antonyms, definitions, pronunciations, or anything customizable by options for the user.

Other issues

- The dataset we use for the whole project is based on a database hosted by Wikimedia Foundation and is still under construction. If the project is consequently modified or abandoned, our project could be thorn of its source and would be completely destroyed. At the time of writing the final version of this report, we are lucky and the database is still online and working correctly.
- There is still a issue when words have extremely many ancestors/descendants. The graph becomes very large and it is difficult to navigate through it efficiently. The solution would be to cluster the words like a word cloud, *only* when there are too many on the same level horizontally. But even then, other issues arise for example outgoing edges can still be too many.
- Due to a shortage of time before the deadline, we didn't manage to polish the design of our website layout (loading pop-up, buttons, checkboxes, other). We recognize that with a better organization we could have had time to make it more beautiful.

Peer assessment

Preparation

Saw Pierre only twice during the semester, was late or didn't come to most meetings.

Contribution

Pierre did not contribute to the coding. He did setup the website template at the beginning though.

Respect for others' ideas

OK

Flexibility

OK