Title of the LREC 2014 Paper

Name of author

Address - Line 1
Address - Line 2
Address - Line 3

Abstract

Each article must include an abstract of 150 to 200 words in Times 9 pt with interlinear spacing of 10 pt. The heading Abstract should be centred, font Times 10 bold. This short abstract will also be used for printing a Booklet of Abstracts containing the abstracts of all papers presented at the Conference.

Keywords: keyword A, keyword B, keyword C

1. Introduction

2. CLiCS

2.1. Colexification

2.2. Data

2.3. Network modeling

3. Visualization

The CLiCS database is available online and offers its users a search interface to all concepts and cross-linguistic polysemies between concepts. The wealth of information in the database and the various possibilities of exploring the polysemies in the network call for an additional component that makes potentially interesting observations more easily accessible to the researcher. The idea was to equip the database with a visualization component that provides various interactive functionalities that enables users to navigate through the networks of polysemies while at the same time providing more detailed information on the actual language data.

3.1. Web-based visualization

We opted for a web-based implementation of the CLiCS visualization in JavaScript using the D3 library (Bostock et al., 2011). The main benefits of a web-based visualization are its platform independence and the fact that users can access it from any device with a browser supporting JavaScript. There is no need for the installation of additional software or for maintenance of the system on the part of the user (Murray, 2010). In addition, links to the descriptions of the external resources can easily be included to allow users to explore the CLiCS data in more detail on demand.

3.2. Interactive functionalities

The visualization features various interactive functionalities that are designed to enhance the exploration of the CLiCS data. The main component is a flexible force-directed graph layout that displays the concepts as nodes and the cross-linguistic polysemies as edges (see Figure 1). The strength of the force in the edges of the graph is dependent on the number of cases that can be attested in the languages for the respective concepts that are linked through the edge. We decided to have separate graphs for all communities, which the user can select from a drop-down menu. As described above, the communities have been automatically generated from the whole network of concepts and links.

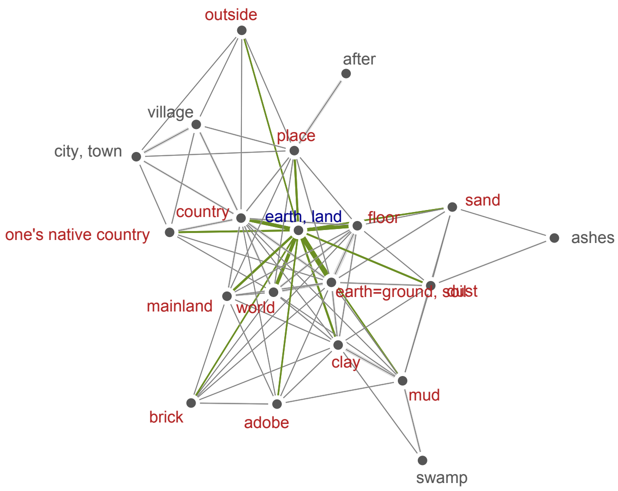

Figure 1: Force-directed graph with mouse-over functionalities highlighting all connected concepts

The force-directed graph layout ensures that all concepts are neatly arranged according to their similarity as defined by the number of cross-linguistic polysemies. As a result, concepts that are highly connected are located close to each other. To make it easier for users to explore the network that is depicted in the graph, concepts can be dragged to different positions where there is less overlap. The dragging behavior of a concept is activated when mousing over the respective node in the graph (when the cursor turns into a crosshair).

As mentioned above, the edges of the graph represent the number of cases of cross-linguistic polysemies for the linked concepts. For a more detailed view on which languages contribute to the strength of the connections, the user can mouse over the links in the graph to see a list of languages featuring polysemous words for the respective link (Figure 2). The list includes additional information on the languages such as their ISO 639-3 language code and

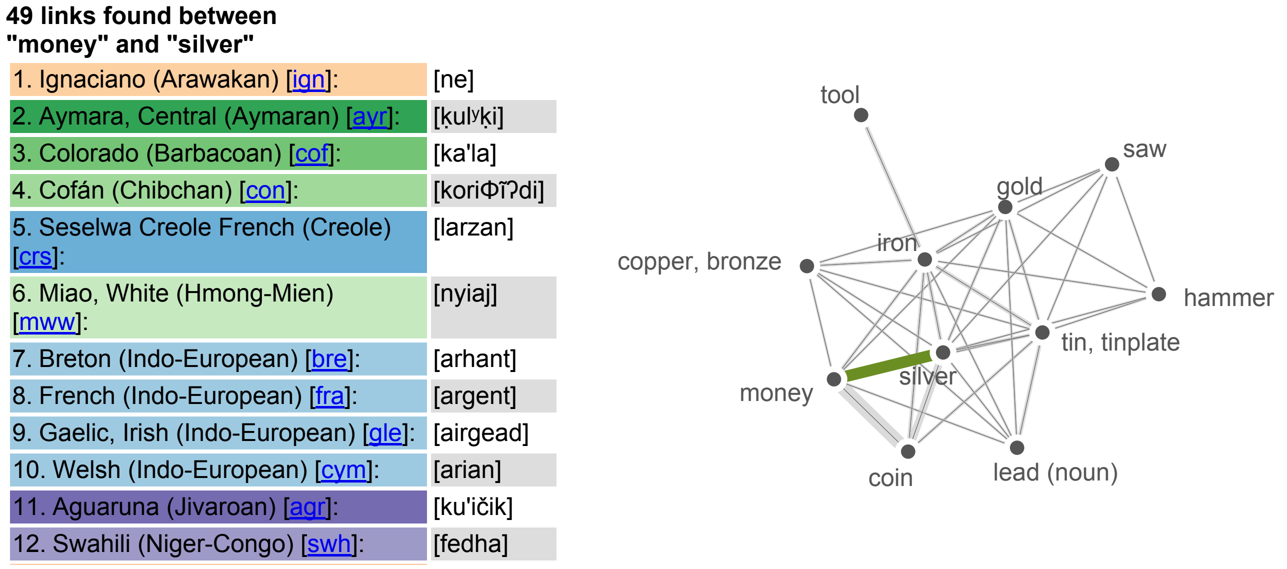

Figure 2: Force-directed graph with mouse-over functionalities showing a subset of the list of words contributing to the cross-linguistic polysemies. The entries have different background colors depending on their language family.

family. Furthermore, each entry in the list provides a hyperlink to the original source from where the information is taken.

Each language in the list is attributed a different background color depending on its language family or location in order to allow for an at-a-glance overview for all languages in the list. The user can choose from a drop-down menu whether to include the genealogical or areal information as the background color. For the genealogical information, all language families are attributed a different color value. Languages belonging to the same language families are therefore given the same background color. Moreover, the list is sorted according to language families. In this way, the user can immediately see how many languages of a given family contribute to the overall strength for the connection at hand.

As to the areal information, the world map is provided with a color gradient as shown in Figure 3. To this end, each position in the world map is attributed a color value using the L*a*b* color space. The color hue thereby indicates the position on the map in terms of the longitude (East-West) whereas the lightness of the color represents the position in terms of the latitude information (North-South).[1] The mapping from geolocation to color values allows for an easier evaluation of areal patterns in the selected connection. In this regard, users can directly detect whether a certain cross-linguistic polysemy is restricted to a certain region of the world or constitutes a more widespread colexification pattern (see the case study in Section 3.4. below).

In addition to the interactive functionalities described above, the visualization also features a variety of further components that allow for an easier exploration of the database. The graph layout is equipped with panning and zooming functionality that enables the user to navigate

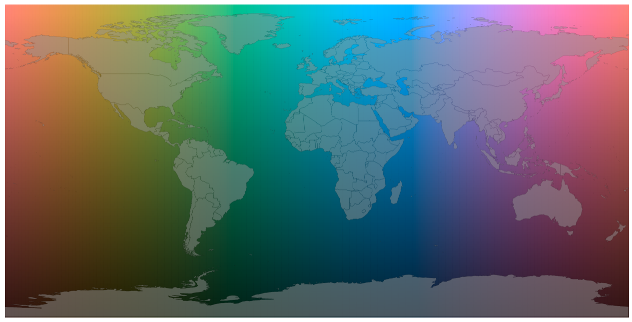

Figure 3: World map with color gradient

through the network graph. Panning is enabled when the cursor changes into a hand symbol when mousing over a link of the graph. The whole graph can then be dragged to a new position. The zooming behavior is activated with the scroll wheel. When mousing over a concept (node) in the graph all connected links and concepts are highlighted in order to provide a better overview of the connectivity of certain concepts (see Figure 1). The control panel of the visualization also includes a slider button that allows the user to show only those edges in the graph with a minimum number of cross-linguistic polysemies.

3.3. Implementation

The visualization is implemented in JavaScript using the D3 library (Bostock et al., 2011).[2] The force-directed graph is generated with the `force()` function from the `d3.layout` module. The layout implementation uses position Verlet integration for simple constraints (Dwyer, 2009).[3] In order to ensure that the concept labels are lo-

[1] See (Mayer et al., 2014) for a different approach for a linguistically informed color gradient of the world map.

[2] `http://d3js.org`

[3] See `https://github.com/mbostock/d3/wiki/Force-Layout` for a description of the implementation.

cated close to the concept nodes, a second force layout (with a static weight of 1) for each concept link to the node is set up.

The color values for the world map gradient scale are computed from the two-dimensional geographical coordinates that are given as an input. The latitude [-90;90] and longitude [-180;180] values are thereby normalized between [0;1] and serve as the input for the function `cl2pix`.[4]

```
function cl2pix(c,l){
    var TAU = 6.2831853
    var L = l*0.61 + 0.09;
    var angle = TAU/6.0 - c*TAU;
    var r = l*0.311 + 0.125
    var a = Math.sin(angle)*r;
    var b = Math.cos(angle)*r;
    return [L,a,b];
 };
```

The actual HTML color code is generated with the function `d3.lab` from the D3 library, which takes as input the three values for `[L,a,b]`. The main reason for choosing the L*a*b* color space is a smoother transition between different color hues without any visible boundaries.[5] For the coloring of the language families, the background colors are generated with the categorical scale functions of the `d3.scale` module.

The dragging and panning functionalities of the graph are implemented with the `drag()` function from the `d3.behavior` module and the SVG `transform` and `translate` attributes.

3.4. Case study

In order to illustrate the usefulness of the visualization for the purposes of exploring the database, consider the graph in Figure 2. Among other things, it contains the connection between the concepts "money" and "silver". A subset of the languages and words contributing to this connection are shown on the left where the background color represents the language families. For instance, French contributes to the cross-linguistic polysemy because both concepts are realized by the same word (viz. *argent*) in that language. When looking at the areal distribution of the languages, a clear pattern emerges at a glance (see Figure 4). Most of the languages contributing to the polysemy are from two major regions: Caucasus (marked in blue) and South America (marked in green).

49 links found between "money" and "silver"

1. Ignaciano (Arawakan) [ign]:	[ne]
2. Aymara, Central (Aymaran) [ayr]:	[ḳulʸḳi]
3. Colorado (Barbacoan) [cof]:	[ka'la]
4. Cofán (Chibchan) [con]:	[koriΦĩʔdi]
5. Seselwa Creole French (Creole) [crs]:	[larzan]
6. Miao, White (Hmong-Mien) [mww]:	[nyiaj]
7. Breton (Indo-European) [bre]:	[arhant]
8. French (Indo-European) [fra]:	[argent]
9. Gaelic, Irish (Indo-European) [gle]:	[airgead]
10. Welsh (Indo-European) [cym]:	[arian]
11. Aguaruna (Jivaroan) [agr]:	[ku'ičik]
12. Swahili (Niger-Congo) [swh]:	[fedha]
13. Akhvakh (Northern) (North Caucasian) [akv]:	[ачи]
14. Akhvakh (Southern) (North Caucasian) [akv]:	[арчи]
15. Andi (North Caucasian) [ani]:	[орси]
16. Andi (Muni) (North Caucasian) [ani]:	[орси]
17. Archi (North Caucasian) [aqc]:	[arsi]
18. Archi (Var1) (North Caucasian) [aqc]:	[арси]
19. Archi (Var2) (North Caucasian) [aqc]:	[арси]
20. Avar (Andalal) (North Caucasian) [ava]:	[гlарац]
21. Avar (Antsukh) (North Caucasian) [ava]:	[гlарас]
22. Avar (Batlukh) (North Caucasian) [ava]:	[гlарач]
23. Avar (Hid) (North Caucasian) [ava]:	[гlарас]
24. Avar (Karakh) (North Caucasian) [ava]:	[гlарац]
25. Avar (Kusur) (North Caucasian) [ava]:	[гlарац]
26. Avar (Standard) (North Caucasian) [ava]:	[гlарац]
27. Bagvalal (North Caucasian) [kva]:	[ас]
28. Bezhta (North Caucasian) [kap]:	[окко]
29. Botlikh (North Caucasian) [bph]:	[арси]
30. Chamalal (North Caucasian) [cji]:	[ас]
31. Dargwa (Itsari) (North Caucasian) [dar]:	[арц]
32. Dargwa (Kajtak) (North Caucasian) [dar]:	[арц]
33. Dargwa (Kubachi) (North Caucasian) [dar]:	[ас]
34. Dargwa (Muiri) (North Caucasian) [dar]:	[арц]
35. Dido (Mokok) (North Caucasian) [ddo]:	[мицхир]
36. Dido (Sagadin) (North Caucasian) [ddo]:	[мицхир]
37. Ghodoberi (North Caucasian) [gdo]:	[арси]
38. Hunzib (North Caucasian) [huz]:	[окро]
39. Karata (North Caucasian) [kpt]:	[гlарсе]
40. Karata (Tokitin) (North Caucasian) [kpt]:	[гlарци]
41. Khvarshi (Inxokvari) (North Caucasian) [khv]:	[ос]
42. Khvarshi (Khvarshi) (North Caucasian) [khv]:	[ос]
43. Lak (North Caucasian) [lbe]:	[арцу]
44. Tindi (North Caucasian) [tin]:	[аси]
45. Shipibo-Conibo (Panoan) [shp]:	[koriki]
46. Tacana (Tacanan) [tna]:	[čipilo]
47. Thai (Tai-Kadai) [tha]:	[ŋən]
48. Siona (Tucanoan) [snn]:	[kutˀi]
49. Pumé (Unclassified) [yae]:	[čɛrɛ]

Figure 4: Languages and words contributing to the connections of polysemies for the concepts "money" and "silver"

4. Conclusions

5. References

Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D3: Data-driven documents. *IEEE Transactions on Visualization & Computer Graphics (Proc. InfoVis)*, 17(12):2301–2309.

Tim Dwyer. 2009. Scalable, versatile and simple constrained graph layout. *Eurographics/IEEE-VGTC Symposium on Visualization*, 28(3).

Thomas Mayer, Bernhard Wälchli, Christian Rohrdantz, and Michael Hund. 2014. From the extraction of continuous features in parallel texts to visual analytics of heterogeneous areal-typological datasets. In *Language Processing and Grammars. The role of functionally oriented computational models*, pages 13–38. John Benjamins.

Scott Murray. 2010. *Interactive Data Visualization for the Web*. O'Reilly Media, Inc.

[4]The code was adapted from the GNU C code by David Dalrymple (`http://davidad.net/colorviz/`) and translated into JavaScript.

[5]See `http://davidad.net/colorviz/` for the difference between using the L*a*b* and HSV color space.