Visualizing Etymology: A Radial Graph Displaying Derivations and Origins

Chinmayi Dixit
Stanford University
cdixit@stanford.edu

Filippa Karrfelt
Stanford University
filippak@stanford.edu

ABSTRACT
Study of words and their origins, or Etymology has been a long-standing area of interest. Although it has been well researched, there aren't many satisfactory visualizations to help understand the data. In this paper we bridge this gap by creating an interactive tool where users can see how words are related to each other at different depth levels. The approach we take also improves the current tools available for language visualizations by providing a better spatial usage, minimizing clutter, while still giving the user the full picture by showing the etymological data.

Author Keywords
Etymology, graph visualization, radial graph, language visualization

INTRODUCTION

There are many online tools for visualizing synonyms and smaller networks of words. However, these all say basically the same thing, and are usually only including words from the same language as the searched word. The etymology of languages is researched and mapped [8], but it is yet to be well visualized. The nominal hierarchical tree structure of the etymological data brings many possibilities for effective visualization.

The structure of etymological data is similar to the structure of a dictionary. Therefore, the tools that are used to visualize dictionaries should be fit to visualize etymology.

From the tools that we have found, there are some properties that we thought was important for effective visualization. This includes a radial layout, with prominent connections between the words. However, we noticed that many applications for word visualization become very cluttered when there are many connections, so for our visualization we focused on providing a cleaner distribution and better usage of space.

RELATED WORK

Our project extends two streams of prior work; etymological research and word relation (language) visualizations.

Etymology

For etymology the most relevant prior work is Etymological Wordnet: Tracing the History of Words, by De Melo, G [1]. This work provides an extensive dataset to see the connections between words, across languages. The data set is quite extensive and makes a great source of information. However, the tools available to visualize it are rudimentary at best [2]. It provides the results in textual format as a list of different categories of relations. As this dataset is rich in content, it became our inspiration to build an interface to access it in a format that is easy to navigate.

Word relation visualization

In the world of visualizing language data there has been extensive work done on showing the relation and the structure of Wordnet data. The visualizations focus on connections within the English language; relations such as synonyms, antonyms, usage, meanings etc. [4,5,6,7].

However, none of these includes a satisfactory visualization for the etymology of words. One common theme that we noticed in these prior works is that radial trees were the most frequent visualization. The visualizations are clear to understand and well represented when the number of nodes is small. In the case of Visualizing Wordnet structure, by Jaan Kamps, we could see that as the number of nodes grew large, the graph turned cluttered (see figure 1).

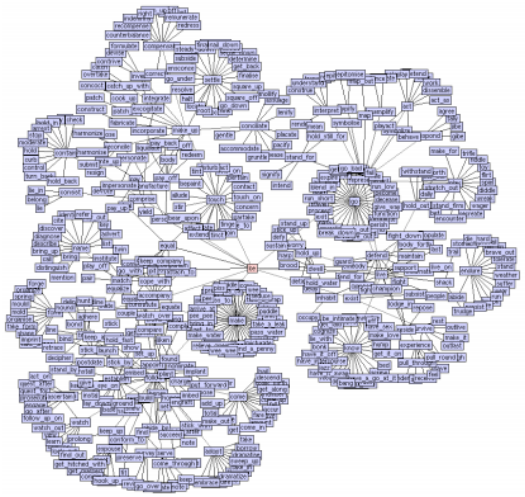

Figure 1: Visualizing Wordnet Structure, J. Kamps

Abrate et al., in Wordnet Atlas: a web application for visualizing WordNet as a zoomable map, address the layout of nodes for better perception [3]. They draw the WordNet out as a spatially arranged radial graph, with nodes arranged in concentric circles, (see figure 2). This made large data sets more manageable and easier to navigate. One drawback of this strategy is that all the spatial location data has to be pre-calculated without real-time updates.

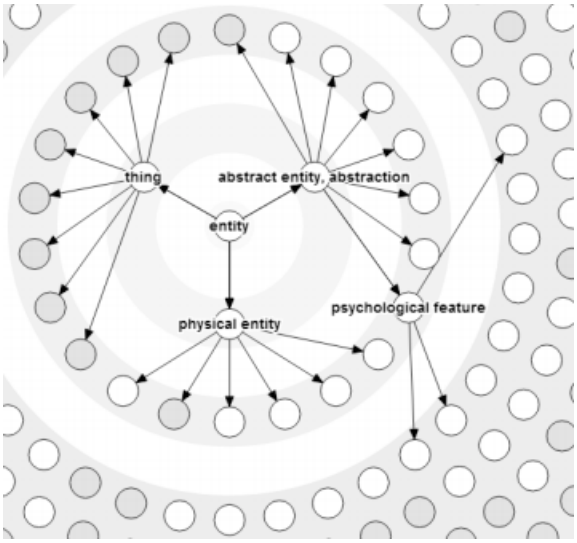

Figure 2: Visualizing WordNet as a zoom-able map. Picture of the structure from the web application.

We saw a possibility to develop a technique building on these prior works to develop an application fit to show more nodes, with a cleaner layout and provide interactive capabilities.

METHODS

Data manipulation

In order to effectively be able to structure a visualization with the data set at hand, the data had to be put in a format that would be efficient for lookup and structuring. In the original data set each entry was a word linked via a relation (such as derived from or origin of) to another word. The first step to making the data more useful and effective for lookup was to structure so that each word was a key and all its relations was stored as values of the key (see figure 3).

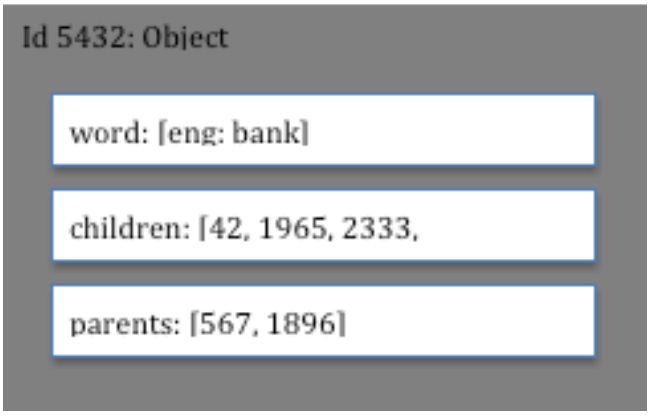

Figure 3: An entry in the data structure. Organized by id, fields holding word, parents and children.

The values are properties such as language and vectors of all the words that that word is related to. Instead of having the word be the key for lookup, we decided to have each word have an id. The ids could then be put in an array with each id in the corresponding index of the vector. While accessing the vector of related words, we could also store those as ids (see figure 3). This meant that we could easily index the words, and find all the properties of any related word. Organizing the data this way was crucial for having the visualization perform within a reasonable time limit.

Also, the dataset contained a large number of suffix words (for example, -less, -er, -ing). These caused many words to be related to each other just through the suffixes. Also, they caused the tree to have a large number of nodes at small values for depth. To make sure that only the relevant data was displayed, we cleaned out the dataset by removing all of the suffixes, phrases with special characters etc.

Traversing the data structure

When searching for a word in the data structure, the whole data set is traversed. This is done once, since the words are stored by their id's and not the word itself. Once the search word is found the rest of the lookup and structuring is done by id. The algorithm for creating the tree-structure search result is as follows:

- Find id of search word (linear search through the data structure; comparing each word to the search word to find the id with the matching word)
- Set the id as the root
- Recursively add the children and parent nodes. (for each children and parent, find its children and parent, repeat). The recursion stops once the selected depth of the search is reached.
- Make sure there are no repeating nodes.
- Return the tree structure with the root and all child and parent nodes.

Traversing the data is definitely a bottleneck in terms of performance in the interface. Especially the linear search for the first word takes a long time compared to the other steps towards viewing the graph.

Visualization

Once the data is organized in a manner identifying the parent and child nodes of the word, the visualization is created using the JavaScript d3 library. Each of the nodes in the tree represents a word/phrase. Each of the connections represents a relationship. The nodes and links are colored based on their type. If a node is the center node, or the one we are focusing on, it has a fill color of black to make it stand out. It also has a center positioning. All other nodes are arranged radially based on their depth – more the depth, farther the node from the center. They also have the same coloring as their incoming links. The links themselves can have one of the two colors chosen – lighter shade of blue for older/origin words and a darker shade for newer/derived words. This is meant to represent the timeline of the word usage.

Hovering over a word highlights the path from that node to the center node. The color of the path represents the relationship between the two nodes. If the outer node is an origin, highlights are done with dark blue and if it is a derived word, they are in black. Hovering over a word also makes information about the word such as the language code, pop up in the graph area. The main idea behind choosing the location of the text box was to not create more noise on the graph.

To reduce clutter when there are a large number of nodes, we change the size of the graph to slightly bigger than the window size. The tool allows users to scroll the window to look at the details of the graph.

The layout of the web application is done with HTML/CSS. The site is not responsive to the display size. We considered making it interactive by screen size but decided to keep it static for two main reasons. The first reason is to keep the same screen configuration and the second is to keep the graph be big enough so that all connections and labels show clearly.

Label placement

To utilize the area of the graph optimally, we use a radial layout, with the text values on the leaf nodes arranged radially. The text values on the nodes within the circle are arranged horizontally for clarity. (see figure 4 and 5).

Figure 4 Horizontal labeling

Figure 5 Radial labeling for the leaf nodes

RESULTS

The result is a web application for visualizing etymology. The data is limited to the data set that is available from the Etymological Wordnet study at Berkeley [1]. Figure 6 shows the web-page layout of the application. The interface is only fit for a computer screen, as the layout is static. We saw no meaning in rescaling the view to a smartphone view since viewing the graph there would be unclear, at best.

We tested the tool by making multiple searches with different depths and words. The performance of the tool is similar for all searches, which verifies that most time is spent on the initial lookup of the word. The performance depends on the depth of the search, but it is marginal to the time it takes to find a word in the system. To load a new word takes between 30 seconds to 60 seconds. However, to change the center node of a view (by clicking any node in the graph) is instantaneous. Adding a new word to the existing view takes up to 60 seconds as well.

Functionality

Functionality of the application includes:

Searching for a word to show the word tree of that word.

Users can type in words to draw the tree around, and we create a tree with the first match we find.

Selecting the depth of the search, ranging from one to seven. This depth limit was imposed as the number of nodes grew very large over this limit, which caused the graph to be impractical.

Clicking on any word in the current word tree to center the tree around that node.

Hovering over a word to display the path from the root to the word color indicating the relation. (Fig 10)

Hovering over a word to display the language and the depth from the root.

Adding a new word to the graph: once the user enters a word to add to the tree, we look for connections from the current center node to the new word node, within the depth limit of 7. If there is such a path, we draw the tree with updated depth and include the new word. If not, we display a message to communicate the limit to the user.

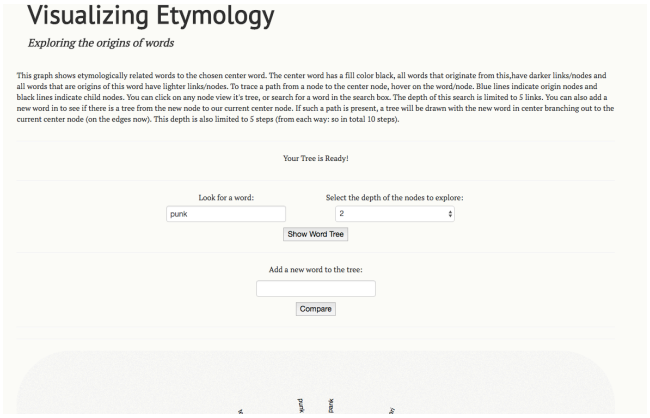

Figure 6 Screen layout of the application

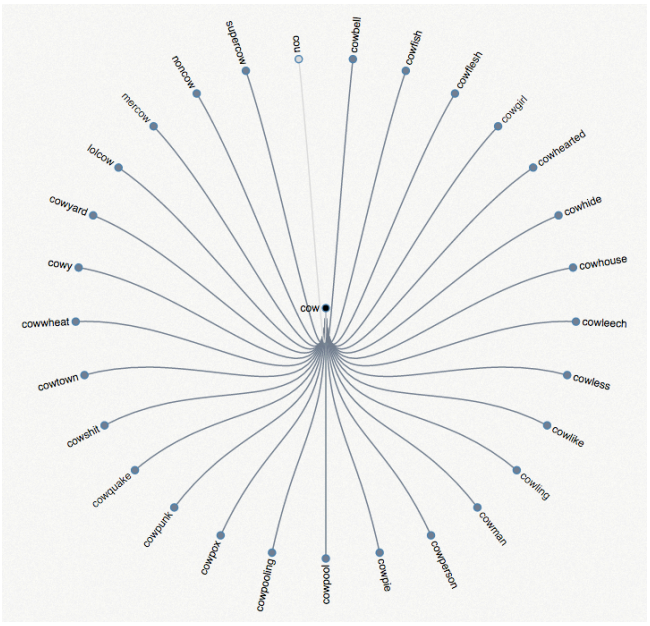

Figure 7 Graph for "cow" depth 1

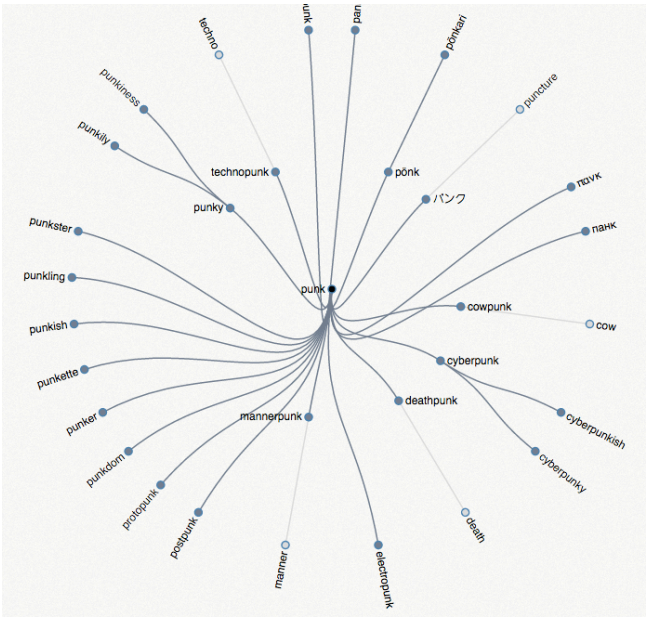

Figure 8 Graph for "punk" depth 2.

Figure 9 Highlighted path.

In order to make sure there is sufficient space for the graph, we decided to put the search bar at the top of the page and assign the complete width of the window under it to the graph. The layout of the graph is spherical which makes the view of the data more condensed, which is something that we were hoping to accomplish. However, there is a certain limit to the number of relations that can be effectively displayed, in a compact area.

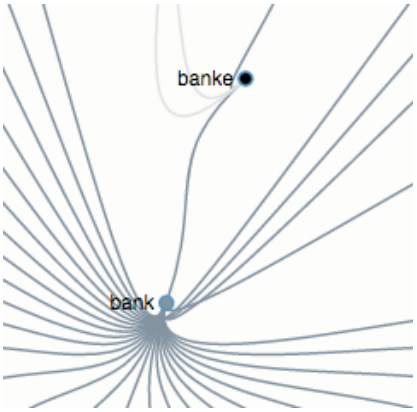

Figure 10 Highlighting the center node.

DISCUSSION

The interface that we have created can help people understand etymology by introducing an overview of the relations between words. The tool is also a step towards effective word relation visualizations in general.

Exploring etymology

Anyone who uses the application that we have built can explore the world of etymology. The tool is useful when exploring words of close etymological proximity, since the interface can switch between words graphed in the same word tree instantaneously.

Professionals or researchers as well as anyone who is not as familiar with the topic can use this application.

Word and language visualization

The interface that we built can easily be applied to other word visualizations. There is a possibility to display both dictionaries and translations effectively this way. The interface is effective at making visualizations less cluttered, which can be very useful when visualizing large data sets such as Wordnet.

Even though the application is currently designed to show two different types of connections (derived from and origin of) this can easily be extended to include more categories.

FUTURE WORK

We have identified three sectors for future work. The first one is data selection and filtering, the second is data traversal and the third is extensions that could be done on the graph.

The interface

Right now there is some ambiguity in how a word is selected during a search. Many words are the same across many languages, but the current program picks one and does the visualization for that word in that language. To overcome this, the user could be presented with options to specify the search (selecting the search word from a list of the options matching the search), or make the system fit to display multiple trees simultaneously.

Data traversal

To make the visualization more useful a work for the future is to speed up the data traversal. Introducing buckets or a more sophisticated map could for example speed up finding a word. This would make it possible to handle data sets that are larger than the one currently in use.

This system could be extended to span more than just etymological relations. A possibility would be to introduce filtering of what type of relations the user would like to see for the searched word.

Another possibility is to instead of extending the actual database with more entries and relations is to make the explanation given for the relation more extensive. Currently, the user is given the language and the depth (the number of words between the current node and the searched word) but there is much more information that could be useful for understanding the etymology of that word.

The graph

With the graph work can be done on making the graph more dynamic. Examples of this are to remove the titles on the radial of the circle if there are many connections.

REFERENCES

1. De Melo, G. (2014). Etymological Wordnet: Tracing The History of Words. In*LREC* (pp. 1148-1154). http://www1.icsi.berkeley.edu/~demelo/etymwn/
2. http://www.lexvo.com/
3. Abrate, M., Bacciu, C., Marchetti, A., & Tesconi, M. (2012). WordNet Atlas: a web application for visualizing WordNet as a zoomable map. In *GWC 2012 6th International Global Wordnet Conference* (p. 23). http://www.iit.cnr.it/sites/default/files/gwa2012_submission_65.pdf
4. Kamps, J., & Marx, M. (2002). Visualizing wordnet structure. In *Proc. of the 1st International Conference on Global WordNet* (pp. 182-186). http://humanities.uva.nl/~kamps/webart/publications/2002/kamp:visu02.pdf
5. Kievit-Kylar, B., & Jones, M. N. (2012). Visualizing multiple word similarity measures. *Behavior research methods*, *44*(3), 656-674. http://www.indiana.edu/~clcl/Papers/KK_Jones_2012.pdf
6. Smith, A., Chuang, J., Hu, Y., Boyd-Graber, J., & Findlater, L. (2014). Concurrent Visualization of Relationships between Words and Topics in Topic Models. *Sponsor: Idibon*, *79*. http://nlp.stanford.edu/events/illvi2014/papers/smith-illvi2014a.pdf
7. Collins, C. (2006). Wordnet explorer: Applying visualization principles to lexical semantics. *Computational Linguistics Group, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada*. https://www.researchgate.net/profile/Christopher_Collins5/publication/228400344_Wordnet_explorer_Applying_visualization_principles_to_lexical_semantics/links/02e7e523b459e66867000000.pdf
8. C. T. Onions, G. W. S. Friedrichsen, R. W. Burchfield, (1966, reprinted 1992, 1994). *Oxford Dictionary of English Etymology*. (ISBN 0-19-861112-9)