Interactive Visualization of Kinship Terms

Mohammed Muqeeth*
University of North Carolina at Chapel Hill

Kevin Chen†
University of North Carolina at Chapel Hill

Abstract

Visualizing a family tree helps to understand the underlying abstract relationship terms. However, most visualizations of family trees are static and are mostly designed for English, which falls under the Inuit Kinship system. There are a handful of kinship systems in the world, and most languages follow one of them. We have developed a tool to create family trees interactively and currently support two different languages: English and Chinese. These languages fall under two different kinship systems: Inuit and Sudanese. We start by building a basic block for every node in the family tree and provide buttons to build any possible family tree. We also provide an option to switch between languages once a tree is built, which would help learn relationship terms in multiple languages. Finally, our tool also makes it easier to add new languages from other Kinships with the easy-to-use API we have provided.[1]

Index Terms: FamilyTrees—Visualization—Visualization techniques—Interactivity; Kinships—Multilingual—Relationship calculators

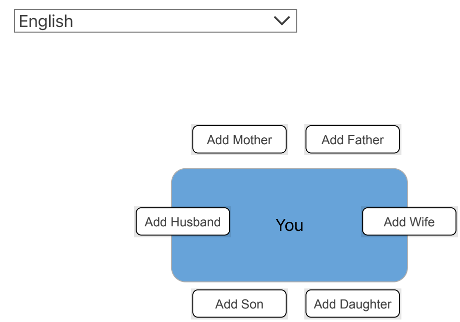

Figure 1: Basic building block

1 Introduction

Visual aids have long been known to help with learning fast, comprehension and knowledge retention. They do it through simplifying complex concepts and make them easier to understand.

One example of how visualization is used to teach young children is in the learning of the alphabet. Teach the alphabets using visual aids can be highly effective in improving children's learning outcomes. Using interactive visual aids, such as videos or animations, significantly improves children's letter-sound correspondence skills compared to traditional teaching methods. Another example is the use of visual aids in teaching basic mathematics concepts. For instance, the use of manipulatives such as counting blocks, number lines, and geometric shapes can improve children's understanding of basic mathematical concepts such as addition, subtraction, and geometry. In both of these examples, visualization is used to represent abstract concepts in a way that is more concrete and understandable to young learners. Not only does visualizing abstract concepts improve learning performance, but interactivity serves to reinforce the learned concepts by associating an action with such concepts. In this manner, the concepts are learned from multiple sensory systems from verbal to visual to touch [9].

Our project is built on the above ideas to simplify the process of learning kinship terms in different languages. Kinship refers to the relationships between family members, and different languages have unique terms to describe these relationships. Among thousands of languages, there are only a handful of Kinship systems, including Inuit, Hawaiian, Sudanese, Iroquois, Crow, and Omaha. English language uses the Inuit Kinship system, where the close family is treated differently from collateral kins. People in the same generation as us are called cousins/brother/sister, while people from the above generation of mother and father are referred to as aunt and uncle. In contrast, in the Hawaiian Kinship system, no such distinction exists. All people above one generation are either mother and father, and those in the same generation are brother and sister. In the Sudanese Kinship system, every person is called with a unique term, or in other words, they have distinct terms for aunts and uncles from maternal and paternal sides. The Latin language also follows the Sudanese Kinship system. Chinese languages are closer to the Sudanese Kinship, but they also have distinct terms based on the person's age. The Iroquois kinship system differentiates parallel cousins and cross cousins, where parallel cousins are considered siblings. The Crow system considers the father's sister's son as the father, while the father's sister and father's sister's daughter are both called aunt. In the Omaha system, the mother's brother's daughter is considered as the mother, while the mother's brother and mother's brother's son are both called uncle. There are other kinship systems that are hybrids of the above ones. Each Kinship systems uses a different term to represent the same relation. For instance, while English speakers refer to the brother of their mother as "uncle," Chinese speakers may use "jiu jiu". Despite these linguistic differences, the underlying concept of the relationship remains the same. We plan to use visualization to help users understand abstract concepts and learn kinship terms across languages.

There are many kinship visualizers available for multiple languages, but most of them are static, making it difficult to interpret with all the edges connecting multiple people. Our research seeks to develop an interactive visualization of kinship terms that reduces complexity and makes it easier to understand the relationships between family members. We believe that an interactive kinship visualizer will be more engaging and enjoyable to use, leading to better retention and understanding of these abstract concepts.

To begin, we start with a basic tree centered around "you," as illustrated in Figure 1. This tree offers several options for selection, including: Add Mother, Add Father, Add Husband, Add Wife, Add Son, and Add Daughter. Depending on the selection made, a node is added to the tree and the user can continue to build the tree iteratively. The name of the selected node indicates its relationship to the center node, "you." For example, if the user selects a node representing

*e-mail: muqeeth@cs.unc.edu
†e-mail: kekevi@live.unc.edu

[1]`https://github.com/kekevi/kinship-terms-viz`

their mother's mother, we will label that node as "grandmother."

2 COMPONENTS

Here we describe the components that we used to build our tool. We used two main components: a relationship calculator and a JavaScript API for visualization. The relationship calculator is a component which helps users determine the relationship term of an individual relative to you. We used a English relationship calculator developed by Steve Morse [7] to get the relationship terms based on the path traversed in the tree. Similarly, the Chinese relationship calculator developed by [8] calculates the relationship give a traversed path.

Along with relationship calculators, there are also various APIs available for visualizing family trees. Balkan Family Tree JS [1] is a popular JavaScript library for creating interactive family trees. It supports various layouts and customization options, and provides an easy-to-use API for adding nodes and links to the tree. Similarly, DTreeJS [4] and GoJS [5] are other JavaScript libraries that allows users to create family trees. We started with Balkan JS library and did not experiment with others in this project.

3 RELATED WORK

Family tree visualizers are mostly static in different languages [2]. Even though the APIs listed in Section 2 support interactively building family trees, they are restricted to only English terms. Our work uses different relationship calculators mentioned in Section 2 to determine terms in multiple languages for a given abstract concept denoted as a path in the visualization.

Trees are not only used for visualizing family relationships, but also for various other purposes. Evolutionary relationships between species, for example, can be visualized as trees, where each node represents a species or a group of species, and the branches represent the evolutionary relationships between them. Similarly, file systems can be represented as trees, where directories and files are organized in a hierarchical structure. Taxonomies, which classify objects into categories and subcategories, can also be visualized as trees, where each node represents a category or a subcategory, and the branches represent the relationships between them.

There are a variety of visualizations for genealogical graphs like family trees. Though simple relationships can seem simple to draw, families can quickly become complicated with many nodes and with the added complexity that spousal relationships are not static, any visualization must make a choice on whether to support showing historical (i.e. exes) relationships in the first place. That is not to mention any instances where nodes might need to be reduplicated across generations of even within the same generation.

Bokhare and Nazmee [2] reviews a variety of different genealogical tree visualizations. One of which is the tree map which partitions a rectangle into separate rectangles representing subtrees where the area is the weight of that node or sum of the weights of that subtree's nodes. Of course, as there are no inherent weights in kinship terms and such a visualization is unfamiliar to most people, such a representation is not useful for our goals. Hyperbolic trees however are an alternative for showing kinship relationships and are used on websites of certain DNA genealogy services, particularly to highlight who are ones' closest relatives. In these graphs, the root node is at the center of the visualization with peripheral nodes having higher degrees of separation located further out from the center, which works for DNA sites as they often want to match users to relatives that have also sent them their DNA samples. Typically, nodes farther out also appear smaller to show that something is less relevant. Though having nodes further separated from the root being less salient matches with how terms farther out are less likely to be used, that would make it more difficult to see the emergence of patterns within kinship terms. Another problem with this type of tree is that it is more difficult to capture generational structure, which all follow a pattern. We could have it separate horizontally, but this would make it more difficult once we introduce spousal relationships.

From this analysis of alternative graph representations, we decided to continue researching traditional family tree graphs where the vertical axis represents generations. Many of the libraries that we researched tended to be smaller scale projects without much documentation. For example, one of the libraries we looked at, donatso's family-chart [3], had many features we were looking for, but the lack of documentation made it difficult to change some of the behavior.

Within more traditional genealogical tree representations, there are problems inherent to this type of representation, along with alternating ways to show edges, as described in McGuffin & Balakrishnan analysis of genealogical trees specifically [6]. One of the large problems is the issue with crowding due to having too many nodes, especially once spousal relationships with the root are introduced. This also leads to exceedingly long lines that overall make the tree harder to read. Another source of this problem is due to how genealogical trees may not in fact be "trees" in the graph sense. Given that one goes back enough generations, there is a high exponentially increasing probability that there is intermarriage between different sides of the family. In these scenarios, it may be beneficial to introduce crossing edges to allow for sibilings to be kept together as much as possible. Regardless, any visualization we make must take into consideration these factors.

4 TASK ANALYSIS

4.1 Learning and Winnowing

To better understand the tool that we are setting out to make, we learned about what people want to do when they set out to find terms for their family members. We noticed that when people do not know a specific term to call their relatives, they tend to explain their relationship through the most basic terms, that is brother/sister, mother/father, son/daughter, and husband/wife, and avoid using more complicated terms like uncle/aunt, grandfather/grandmother, etc. though "cousin" is often used as well. This means that even if a relative has multiple paths, for example, one's granduncle could be referred to as "grandpa's brother" or "mom/dad's uncle," often one will say "my mom/dad's dad's brother" instead as it is the most specific (reduces the number of possible people) and is most easy to comprehend when listening. For example, "grandpa's brother" is more ambiguous because it could be one's mother's father's brother or it could be from the paternal side.

In the case of learning languages, people can easily translate an individual term using a dictionary, but it can quickly get confusing when learning a bunch of terms, and when using one's non-native language, especially when talking about interaction between relatives. In this case, a visual aid is important to act as a quick reference when studying, while being potentially useful for showing someone else.

The data we use for our visualization comes from the standard dialects of each language we support. This comes with advantages and disadvantages of course:

Advantages:

- The standard dialect are typically what most language-learners learn.
- The standard dialect is spoken by the majority of people.
- The standard dialect contains the most resources (to easily get machine-readable data) and least variation across speakers.

Disadvantages:

- Non-standard dialects continue to be underrepresented with less resources to learn them.

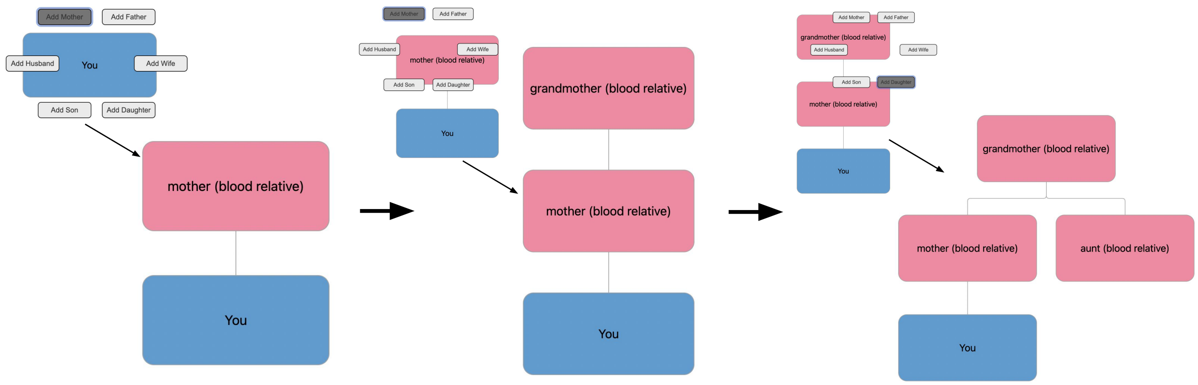

Figure 2: Steps taken by the user to get “Aunt” relation in English

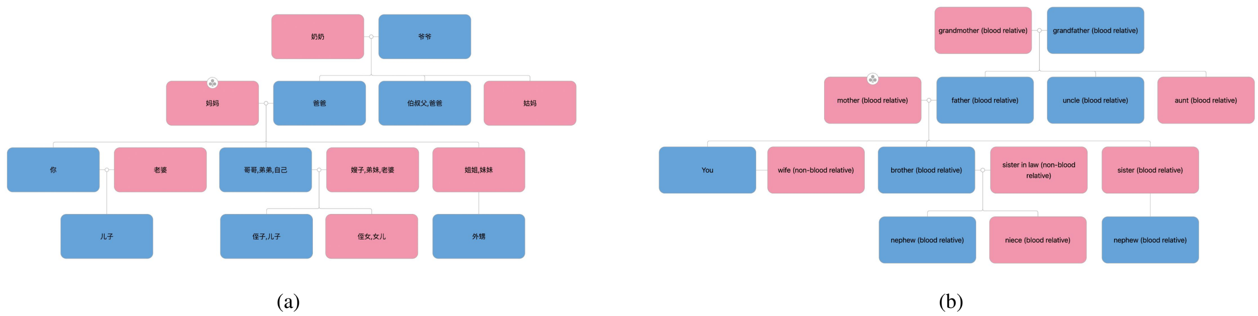

Figure 3: A family tree made using our tool in Chinese and English languages

- Real speakers will often use non-standard dialect to refer to family members (eg. “nana” for grandmother).

Family tree data tends to be presented as a path of these basic first-level terms similar to how people describe their family relationships. The problem is that there are multiple paths to refer to the same family even when using the most basic terms, for example one’s maternal uncle could be referred to as one’s mother’s brother or one’s mother’s mother’s son, or potentially even a loop like one’s brother’s brother’s ... mother’s brother [10]. This means its still easy, albeit more difficult than with more than just first-level terms, to get ambiguous paths. Compounding this difficulty is the semantic differences people make highlight when they purposely choose a longer path, it makes it seem as if the family member is more distant or potentially a step or half-relative. Hence any visualization on a family tree must make a choice on whether to only support nuclear family terms or to allow for divorce/non-nuclear families making a potentially more confusing but more informative tool.

4.2 User Tasks and Prototypes

One task that this visualization accomplishes is as an educational tool. As described in the previous section, foreign language learners need the ability to quickly translate terms between their native language and their target language. Again, more complex terms are built on a chain of simpler terms that need to be done by memorization. However, more complex terms also tend to have a naming system that is most easily learned by visualizing the system to their relative relationships as a visual pattern is more easily recognizable.

Another task that this visualization accomplishes is as a tool for visualizing an user’s own family. For example, during holidays like Chinese New Year, a user may see many distant relatives but be unsure what to refer to them as. Sometimes, this can be perceived of as disrespectful as well. In this scenario, users need to get the proper terms of many relatives at once and will likely group relatives of the same immediate family together, which means the visualization needs a way to show multiple individuals.

From looking at these two tasks, we were able to come up with the following prototype in Figure 4.

In this figure, the triangles and squares represent female and male respectively and the green underlines denote possible options to choose relative to the last chosen node. This tree structure was chosen as it is likely mirrors the most common type of family trees that most users are exposed to whether that be from school or other sources. Hence, people would likely intuitively recognize what each node represents, which is important as educational tools like these should focus on teaching the material rather than focusing on how to use the tool itself. Like what we learned from our tasks, only nodes with one-degree of separation are shown; in addition, it keeps the tree from getting too cluttered when only a few relationships are shown.

5 System

Our system currently works on two different languages, and we provide users with the option to select their preferred language from a dropdown menu. Upon selecting a language, a node appears whose term corresponds to the language chosen as shown in 1. In English, this node is referred to as the “You” node, and our system builds on this node as the center and builds all other relations interactively. In our system, we assume that the gender of each node can be categorized as either male or female. To help differentiate between the two, we use blue color for male gender and pink for female gender, as they are semantically relevant to users. By using different colors for different genders, users can easily identify the gender of

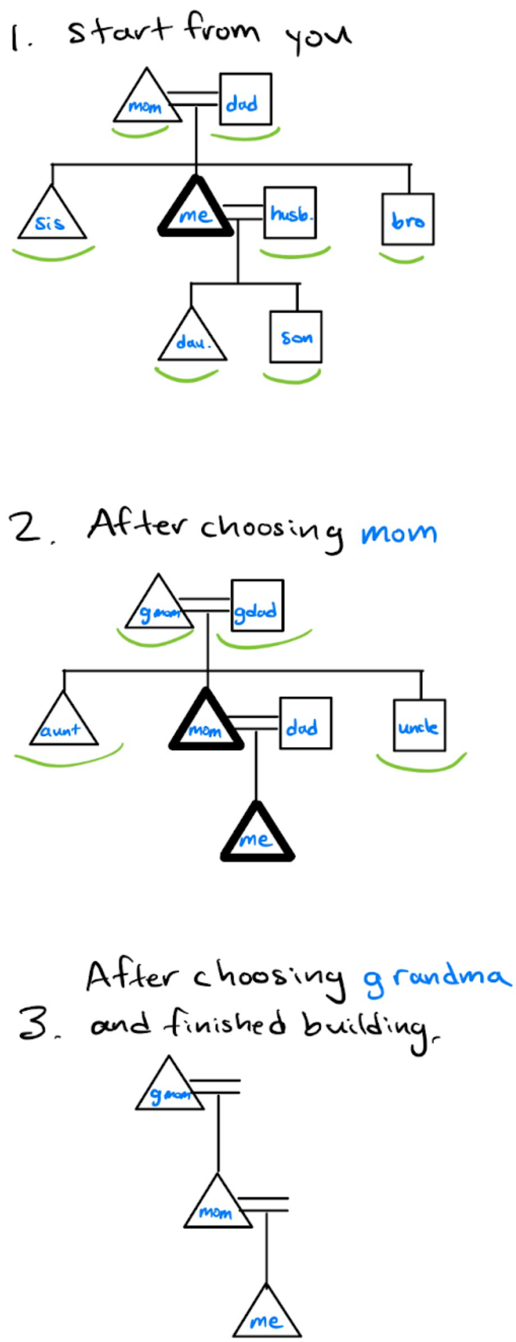

Figure 4: Prototype sketch

each node, which can be helpful in understanding the family tree. For example, users can quickly identify if a given node is a father or a mother based on the color of the node.

Each node in the family tree contains text that describes its relation to the "You" node. The language of this text is based on the user's initial language selection. For instance, if a user chooses Chinese as their preferred language, the text inside the nodes will be in Chinese, making it easier for them to navigate and comprehend the family tree in their preferred language To assist in constructing the family tree, we have incorporated basic buttons that enable users to add various family members to the tree, such as husband, wife, son, daughter, mother, and father. Using these basic buttons, we can build the complete family tree in any language. We did not include buttons for siblings as part of our basic systems, as it makes our basic tree more manageable, and siblings nodes can be created using the basic buttons provided. For example, adding a brother would entail adding a mother and then adding a son to the mother node. Furthermore, we offer an additional option to switch languages at any time during tree construction.

We employ relationship calculators built for English and Chinese to obtain the text that describes the relationship of a node. Each node in the tree is indexed based on the traversed path leading to that node. We then translate this path into the appropriate format required by our relationship calculator to calculate the relationship. This approach allows us to easily expand our tool to new languages by adding new translating functions that take the traversed path and output the correct format required by the relationship calculator in that language.

6 Example

In this section, we provide a step-by-step walkthrough of how a typical user can navigate to the "Aunt" relation on the maternal side, using our tool. We have chosen English for this example, but the same steps can be followed for any language. First, the user clicks on the "You" node, which generates navigation buttons as shown in Figure 1. The user then selects "Add Mother" as highlighted in Figure 2. Upon selecting this button, a node is added to the current tree, with text that describes the relation to the "You" node, which in this case is "Mother". Since we did not include an option to add siblings in the current implementation to keep our basic building block simple, the user must add a "Grandmother" node and then add a daughter to it to add a "Sister" node to the "Mother" node We did include an option to add siblings in the current implementation in order to have our basic building block simple. The complete process of reaching the "Aunt" relation is illustrated in Figure 2. Additionally, we have created trees that show all the members of the immediate family built using our tool in both English and Chinese, as shown in Figure 3.

7 Future Work

Our current implementation works only with English and Chinese languages. In the future, we can extend our tool to other languages by adding programmatic APIs that can determine relationships in these languages. In cases where relationship calculators do not exist, we could allow users who are native to that language to use our tool and provide an option to annotate the nodes as they interact with it. This way, we can collect annotated data that can be stored in a data store and used further without the need for annotations every time.

In future work, we plan to include additional attributes such as age and images in the metadata of the nodes in our family tree. Currently, our tree only accounts for the gender of the nodes. Incorporating age would provide a more detailed and specific relationship since it is considered in certain languages such as Chinese and some Indian languages. We could include additional attributes like age and images. Age would help determine a more detailed and specific relationship since it is accounted for in some languages like Chinese and some Indian languages. Furthermore, adding images as an attribute would enhance the visual representation of the family tree, making it more personalized and appealing.

We also plan to add ability to determine the relation between any two nodes in the tree using our tool. Currently, all the relationships shown have the "You" node as the center. In the future, we aim to support choosing any node in the current tree as the center and thus display the relationship between any two nodes in the family tree.

REFERENCES

[1] BalkanFamilyTreeJS. Balkan familytreejs. `https://balkan.app/FamilyTreeJS`.

[2] S. Bokhare and W. Zainon. A review on tools and techniques for family tree data visualization. *J. Theor. Appl. Inf. Technol*, 97(1):121–132, 2019.

[3] donatso. family-chart. `https://github.com/donatso/family-chart`.

[4] dTree. dtree. `https://github.com/ErikGartner/dTree`.

[5] gojs. gojs. `https://gojs.net/latest/index.html`.

[6] M. McGuffin and R. Balakrishnan. Interactive visualization of genealogical graphs. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 16–23, 2005. doi: 10.1109/INFVIS.2005.1532124

[7] S. P. Morse. Calculating relationships in one step, 2005. `https://stevemorse.org/relation/calculator.html`.

[8] S. mumuy, hustcc. Chinese relationship calculator github. `https://github.com/mumuy/relationship/`.

[9] D. Schweitzer and W. Brown. Interactive visualization for the active learning classroom. In *Proceedings of the 38th SIGCSE technical symposium on Computer science education*, pp. 208–212, 2007.

[10] C. Xu and Q. Zhang. The shortest kinship description problem. *Information Processing Letters*, 138:61–66, 2018. doi: 10.1016/j.ipl.2018.06.005