LANGDIST: ALGORITHMS AND EVALUATION OF NOVEL STRUCTURAL LANGUAGE SIMILARITY METRICS

Huxley Marvit
Princeton University
huxley@princeton.edu

ABSTRACT

There has been considerable discussion about "language equality:" are two given languages really the same language? How many languages are there? To answer questions of this sort, we must be able to define what it means for languages to be equivalent. The key determinant of language equality has been described as "mutual intelligibility." While a potentially useful concept, mutual intelligibility is not a suitable mathematical measure of equality as it is not transitive (an inherent trait of equality). Thus, we argue that the notion of language equality must give way to language similarity. That is, all languages exist along a continuum from identical to completely dissimilar. To address this, we propose LangDist, which provides three novel language similarity metrics based on: shared language features, shared language structure and grammar, and shared representation as language vectors — a generalization of word vectors. We then examine the results of two of these methods, and discuss directions for future work. We also provide open-source implementations of these metrics at github.com/TheEnquirer/LangDist.

1 Introduction

It has been argued that language is "in many ways the ultimate human artefact." (Clark, 1998). But, of course, there are many languages. So whether we agree with Clark or not, this gap between a single "artefact" and a multiplicity of them must be reckoned with. The concept of "language equality" relies on a test to determine if two languages are the same and, in fact, really only one. If the languages are "mutually intelligible," that is, if speakers of language A can understand language B, and speakers of language B can understand language A, it seems reasonable to say that the languages are equal, that the two languages (here A and B) are the same.

Sadly, the world is seldom so simple. One of the characteristics of equality is transitivity. Simply put, if A is equal to B and B is equal to C, then A is equal to C. For any entities to be equal, they must satisfy the test of transitivity. However, if languages A and B are mutually intelligible, and languages B and C are mutually intelligible, it does not follow that languages A and C are mutually intelligible. Thus, mutual intelligibility does not map to language equality.

So where does this leave us? It forces us to recognize that languages exist on a continuum, that simply describing them as either the same or not the same is too simplistic. Instead, we need to elevate our thinking of language similarity from zero dimensional to one dimensional: rather than thinking of the relationship between two languages as having zero dimensions and thus being a single point (the same or not the same) we can think of the relationship as lying somewhere along a line, a one dimensional continuum from identical to completely different.[1]

A richer understanding of language similarity can guide us to develop methods of quantifying similarity between any two languages. This has a number of potential applications. First, it is reasonable to expect that novel approaches might make contributions to phylogenetic and evolutionary analysis of languages. (Greenhill, 2015)

Second, language similarity can serve as a proxy for mutual intelligibility (which is of course itself a spectrum). Thus, a language similarity metric can be a guide for which languages are easier or harder to understand or even learn given knowledge of certain languages as a starting point.

Third, understanding the relationships between languages may make it easier to design new languages that would be simpler for speakers of some existing languages to understand and even master. When it comes to constructing languages — whether for modern use, pedagogical purposes, or even predicting how languages will look in the future (Sanchez-Stockhammer, 2015) — having rigorous and quantitative similarity metrics is vital.

Fourth, a similarity metric can provide a better understanding of language equality, allowing formalizations of equality and subclasses of languages. This stands in contrast to using mutual intelligibility which is not formal (in the mathematical sense).

1.1 Prior Work

Of course others have examined language similarity. In general, this metric is referred to as Lexical Similarity (LS) (Ahmed et al., 2020). These similarity metrics focus on the lexicon of a language (which we believe to be lacking, as there is much more than lexicons that can inform assessments of language similarity).

The currently accepted methods for determining LS are primarily based on the Swadesh word-list (Swadesh, 1950). This has expanded and bifurcated over the years but the most common baseline is his list of 207 words (Swadesh, 1952). By translating this list, different languages can be compared and lexical similarity assessed. Methods of using this word list range from simply tabulating the number of overlapping words (lexical overlap) (Glot, n.d.), to using many variants of Levenshtein Edit Distance between words (Nerbonne and Heeringa, 2002), and even edit distance on the IPA representation of words (Mutabazi, 2020).

These methods for Lexical Similarity — as a proxy for Language Similarity — work by aggregating the pairwise similarities of words given some word similarity metric (Hamming distance, Levenshtein Distance, WNSim (Do et al., 2010, based on Fellbaum, 1998), etc.). While fruitful, this still has many drawbacks. Not only do lists such as Swadesh's inherently distort the language by selecting such a small fraction of its words, but the failure to capture anything about the language's structure and attributes leaves a gaping hole. One can easily imagine languages that have similar words across Swadesh's or similar lists, but with an entirely different grammar. LS approaches would consider these identical while they could easily be constructed to be completely mutually unintelligible. Instead, we need to take a broader view and move beyond a Lexical Similarity metric to a Language Similarity metric. To do so, we must capture language attributes beyond a limited wordlist and include elements such as structure and grammar.

[1] It may be that we can develop a still deeper understanding by moving to higher dimensions. That is beyond the scope of this work, but remains an area for potentially fruitful work.

2 Methods

2.1 Guiding Principles

We would like our methods to capture as rich a representation of any given language as possible. This should include representations of lexicon as well as structure. We want to represent as many dimensions of the language as possible. And above all, when comparing languages, we don't want to be misled by some abstraction that might make languages appear similar but not represent actual similarities between the languages themselves. In computer science terms, we do not want to be fooled by an overfitting.

2.2 Feature Similarity

One way to formally encode structure is through manual featurization. One can name a feature (such as possessing "Future Tense", or not) and then label languages according to these features. The World Atlas of Language Structures (WALS) does just this (Dryer and Haspelmath, 2013). It includes a set of almost 200 features, labeling 2660 languages. Using this database, we can construct a language similarity metric based on shared features.

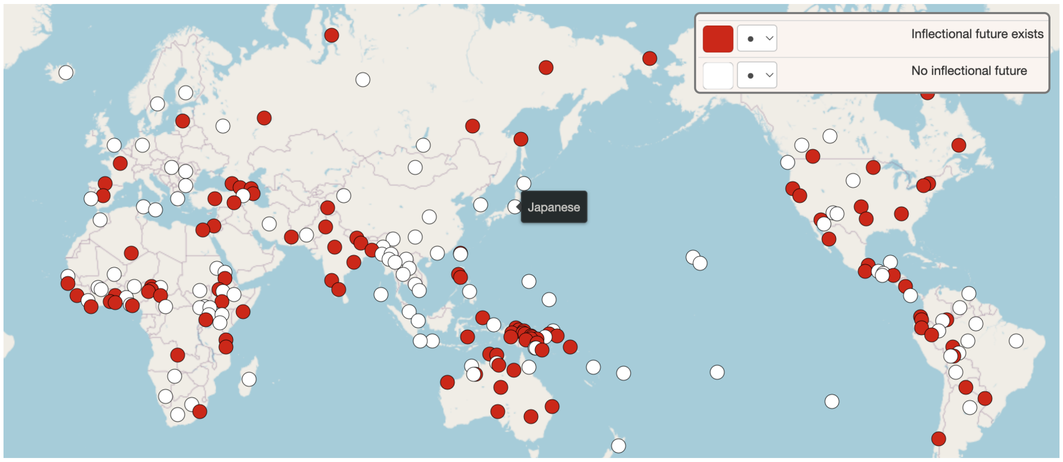

Figure 1: An example of an entry in the WALS database.

2.2.1 Specification

We provide a definition for Shared Feature Similarity $\mathrm{sf}(X, Y)$. To find the similarity between language X and language Y, we consider a weighted sum of all their shared features:

$$\sum_{f \in X \cap Y} w(f_i)$$

We generate these weights based on the rarity of the features. Similar to the IDF portion of TF-IDF, we employ the heuristic that rarer features are more indicative of similarity. Thus, for a given feature f, the corresponding weight of that feature will be the probability of that feature occurring in a given language. However, we want rarer features to be weighted more heavily, so we invert this probability:

$$w(f_i) = 1 - P(f_i \in X)$$

where P represents the probability function, f_i represents the ith feature, $w(f_i)$ or w_i represents the weight for the ith feature, and X is a generalized arbitrary language, modeled here as a set of features. In the future, we may denote this as $P(f_i)$.

The first issue we encounter is that languages in the WALS database do not always share the same set of measurements. For example, while we may know that a certain languages X does in fact have future tense, there may not be an entry at all for future tense in Y. Thus, when comparing two languages, we only consider the overlap between measurements of each language. (A measurement of a language determines the state of a feature for that given language.) Each given set of languages (X, Y) may not share a large set of similar measurements, and may also share an entirely distinct set of measurements from another set (P, Q). Thus, in order for this metric to be valid, we must normalize it.

To calculate this normalization across different numbers of measurements and different set of measurements entirely, we divide by the sum of expected values of each of our measurements, where a "value" of a measurement would be the associated weight of the feature determined by the measurement. (Note that each measurement can collapse into many feature states, not just binary ones. For example, the "Rhythm Type" measurement can lead to features "Trochaic, Lambic, Dual, Undermined, and Absent".) Thus, we normalize with the following:

$$\sum_m \mathbb{E}(m_i)$$

where m_i represents the ith measurement, and the expected value $\mathbb{E}$ of a measurement is defined as

$$\mathbb{E}(m) = \sum_{f \in m} w_i \cdot P(f_i) = \sum_{f \in m} P(f_i)[1 - P(f_i)]$$

Thus, our final metric can be represented as

$$\text{sf}(X, Y) = \left(\sum_{f \in X \cap Y} (1 - P(f_i)) \right) \cdot \left(\sum_m \sum_{f \in m} P(f_i)[1 - P(f_i)] \right)^{-1}$$

$$= \left(\sum_{f \in X \cap Y} w(f_i) \right) / \sum_m \mathbb{E}(m_i)$$

We additionally return the number of overlapping features between X and Y as a measure of confidence in our measure of similarity for a given pair (X, Y).

2.3 Structure Similarity

While the Feature Similarity in section 2.2 captures the structure of a language, it does so within predefined features created by imperfect humans. The featurization used when calculating sf(X, Y) has been manually named and constructed, and are certainly far from complete descriptors of the structure, features, and grammar of a language.

Thus, in this section, we propose the usage of a more "natural" way to encode the structure of a language: syntactic trees. Here, we turn to the Parallel Universal Dependencies (PUD) dataset, which was constructed for the CoNLL-U 2017 shared task of multilingual parsing (CoNLL-17, 2017). The PUD dataset contains a shared list of 1000 sentences in the same order across 20+ languages, all encoded in CoNLL-U format. This format has been constructed by the Universal Dependencies (UD) project — an open source community with 500+ contributors — to encode the structure of a given sentence.

From here, we can convert this CoNLL-U encoding of these sentences into trees, and compare how similar these trees are. With some measure of tree-similarity, we can compare the tree structure of a sentence in language X with the tree representing the same sentence translated into language Y. Finally, we can aggregate over many such sentence pairs to create a measure of language similarity (which increases in accuracy as our sentence bank of 1000 increases in length).

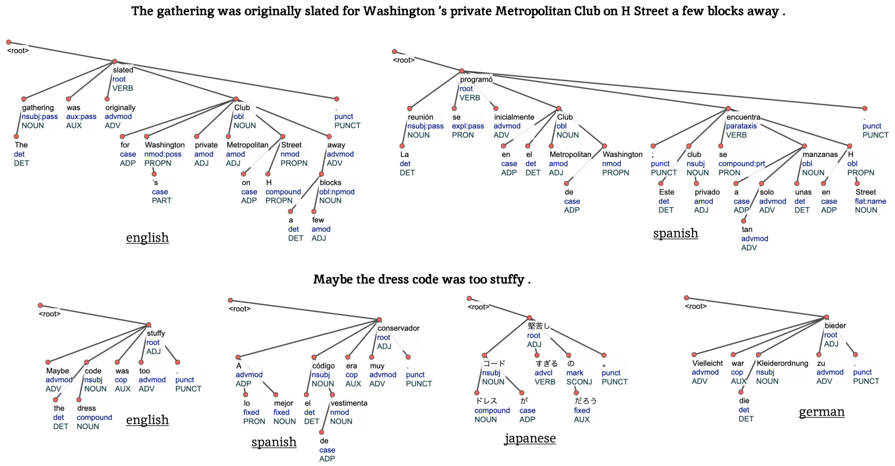
The gathering was originally slated for Washington 's private Metropolitan Club on H Street a few blocks away .
english
spanish
Maybe the dress code was too stuffy .
english
spanish
japanese
german

Figure 2: Visualizations of the CoNLL-U Trees for the same sentence across languages

2.3.1 Specification

We provide a definition for Shared Structure Similarity $\text{ss}(X, Y)$. To find the similarity between language X and language Y, we consider the normalized average tree distance between all parallel sentence pairs within the PUD dataset. In order to find the distance between two trees, we utilize ordered Tree Edit distance (Zhang and Shasha, 1989). Tree Edit Distance (TED) is a generalization of Levenshtein string edit distance to ordered trees, where left to right order is important (such as with sentences). It describes the minimum number of edits (in this case, insertions, deletions, and replacements) to transform a tree $x \in X$ into a tree $y \in Y$.

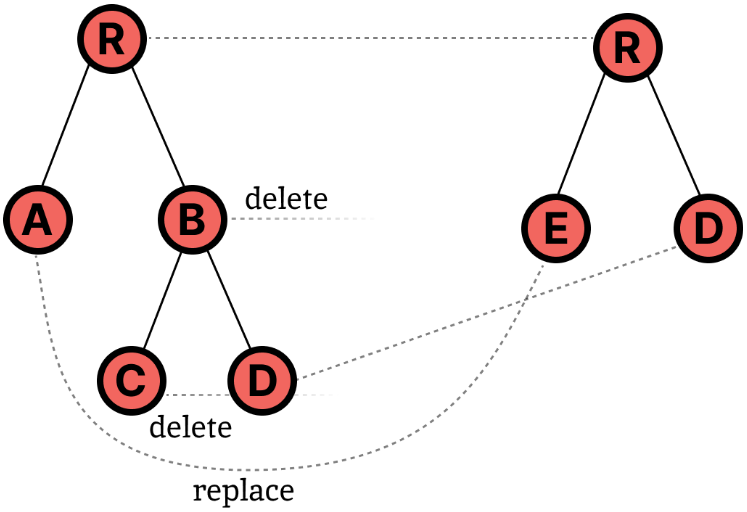

Figure 3: Visualization of a tree $x \in X$ being edited into a tree $y \in Y$

In order to convert a CoNLL-U tree into one we can perform TED on, we swap node labels from words to UPOS labels: Universal Parts of Speech tags. Thus, nodes representing the same part of speech can be preserved through editing. We also perform other transformations as necessary to our tree representation (such as formatting in depth-first-search order) so we can apply our tree edit distance algorithm.

With these trees, we iterate through all pairs $(x, y) \in X \times Y$, and find the average TED of our transformed CoNLL-U trees. (Note that we use standard TED, in which all edits are weighted equally. This should be addressed in future work.) Let $T(x)$ represent the transformation from a sentence x to the tree representing it. Next, we normalize this value between 0 and 1. To do this, we use the following:

$$\mathrm{ss}(X, Y) = \exp\left(-\frac{1}{\lambda} \cdot d\right)$$

where d represents our average edit distance, and λ represents some normalization constant. (Experimentally, we set this constant to be equal to the max average edit distance between languages.) Thus, we can formally represent our Shared Structure Similarity ss(X,Y) as:

$$\mathrm{ss}(X, Y) = \exp\left(-\frac{1}{\lambda} \cdot \sum_{x \in X, y \in Y} \mathrm{TED}[T(x), T(y)] \;/\; |X|\right)$$

where we divide by the size of our sentence bank $|X|$ to generate an average TED.

2.4 Embedding Similarity

Finally, we propose (but do not implement due to time constraints) Embedding Similarity. In search of the most natural representation of a language, free from human encoding as features or even as tree-structures, we can automate the encoding process entirely with an algorithm. We propose the usage of a machine learning model to construct a latent space with which we can compare languages. With this latent space, we can form "Language Vectors," conceptually equivalent to word vectors but representing languages. We can then compare languages X and Y through the cosine similarity of their vector representations:

$$S_C(X, Y) = \frac{X \cdot Y}{\|X\| \|Y\|}$$

where "$\cdot$" represents the dot product.

To construct this latent space, we could use the pre-training task of classifying languages: given some very large sentence bank, we could train a machine learning model to classify what language sentences belong to. Using this to construct our latent space, we could find vectors that represent a given language through some aggregation of the sentence vectors corresponding to that language (for example, we could average all the sentence vectors in a given language, perhaps first pruning outliers, to generate a language vector). We could also use existing model architectures to perform this task, such as the LangId model made by the Stanza group at Stanford (github.com/stanfordnlp/stanza/blob/main/stanza/models/langid/model.py).

However, this method may violate some of our guiding principles for good language similarity metrics: it may easily overfit or construct a low-dimensional representation of the languages that works for differentiating between them but is not suitable for accurate language vector similarity.

2.5 Evaluation

In order to check the efficacy of our similarity metrics, we create rudimentary "sanity-check" metrics that we can compare against. One such metric is the Shared Geography metric, $\mathrm{sg}(X, Y)$ for two languages X and Y. This metric is calculated simply, as:

$$\mathrm{sg}(X, Y) = \sqrt{\text{distance between X and Y}}$$

We take the square root of the distance based off of the assumption that languages disperse roughly evenly from a center point, and thus the effect of languages on one another will not be linear with distance but rather falloff as a function of area.

Next, we compare this Shared Geography metric with our similarity metrics by finding the correlation between our root distance and similarity. (In actuality we compare this with dis-similarity, or negative similarity, as this should increase with distance.) We do this by calculating by Pearson product-moment correlation coefficients, and then selecting the relevant matrix entries. For comparing with Shared Feature Similarity, we use consider all pairwise similarities of the top 15 languages with the most measurements. For Shared Structure Similarity, we compare with all pairwise similarities available (441 datapoints).

3 Results and Discussion

The results of our various experiments are presented in the graphs below.

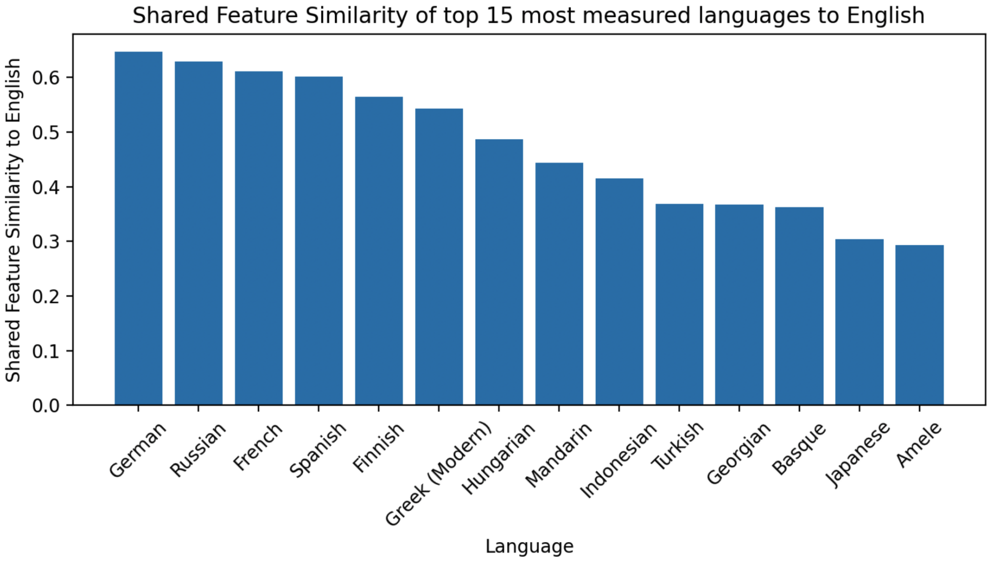

Figure 4: Graph of language similarity to English based upon Shared Feature Similarity
Experiments here compare the 15 languages that have the most measurements listed in the WALS data set. Conducted as an initial test to see if results were reasonable, we ordered all 15 languages by the closeness of their feature set to English. The result does match intuition, with German and French being fairly closely related to English while Japanese, Amele, and Basque are more distantly related.

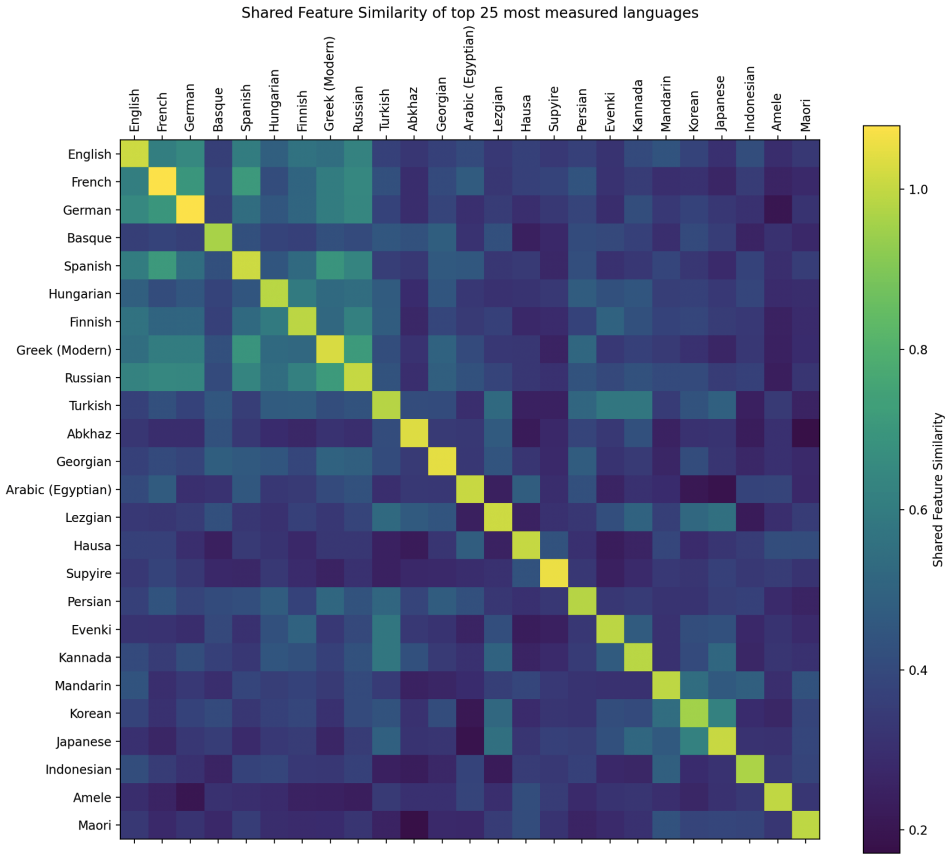

Figure 5: Shared Feature Similarity of the top 25 most measured languages.
This figure illustrates the similarity of the 25 most measured languages based upon Shared Feature Similarity. They are ordered (roughly) by their geographical distance to English to illustrate structure out of block-matrices.

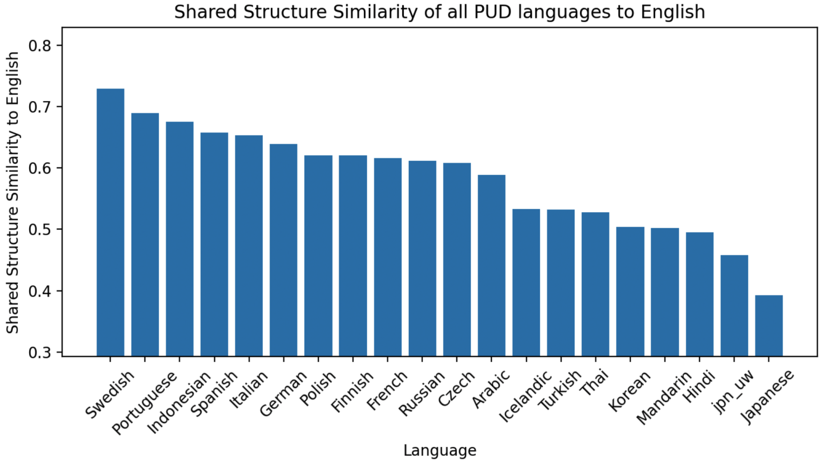

Figure 6: Graph of language similarity to English based upon Shared Structure Similarity
Experiments here compare all of the languages in the PUD data set to English. Conducted as an initial test to see if results were reasonable, the result does roughly match intuition, with Spanish and German being fairly closely related to English while Japanese and Hindi are more distantly related. It is worth noting that the results are still somewhat different from those in Figure 4.

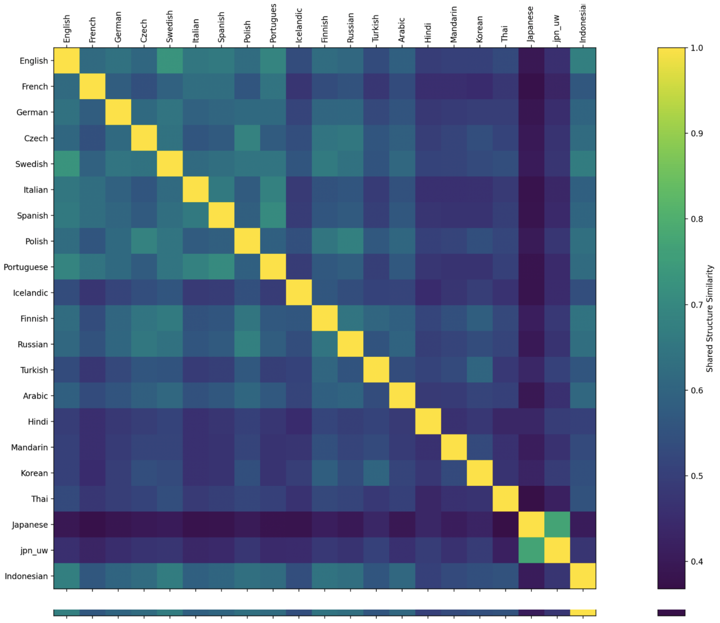

Figure 7: Shared Structure Similarity 21 languages in the PUD dataset.
This figure illustrates the similarity of the 21 languages in the PUD dataset based upon Shared Structure Similarity. They are ordered (roughly) by their geographical distance to English to illustrate structure out of block-matrices.

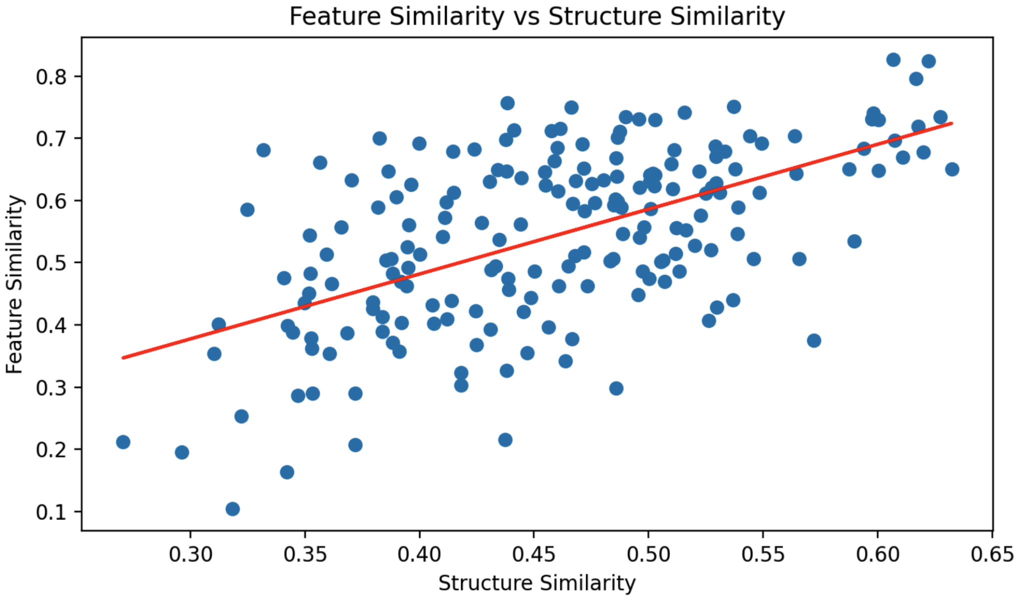

Figure 8: Relationship between Feature Similarity and Structure Similarity
As noted earlier, both the Feature Similarity and the Structure Similarity rankings of languages against English roughly match intuition. However, they do provide different rankings. This is to be expected since the methods they use are so distinct. Here languages are being compared with both methods and the correlation between those methods graphed. Each dot is a single language comparison (such as Japanese vs. Mandarin). The overall result reflects a correlation of 0.577 between the two different systems of similarity assessment.

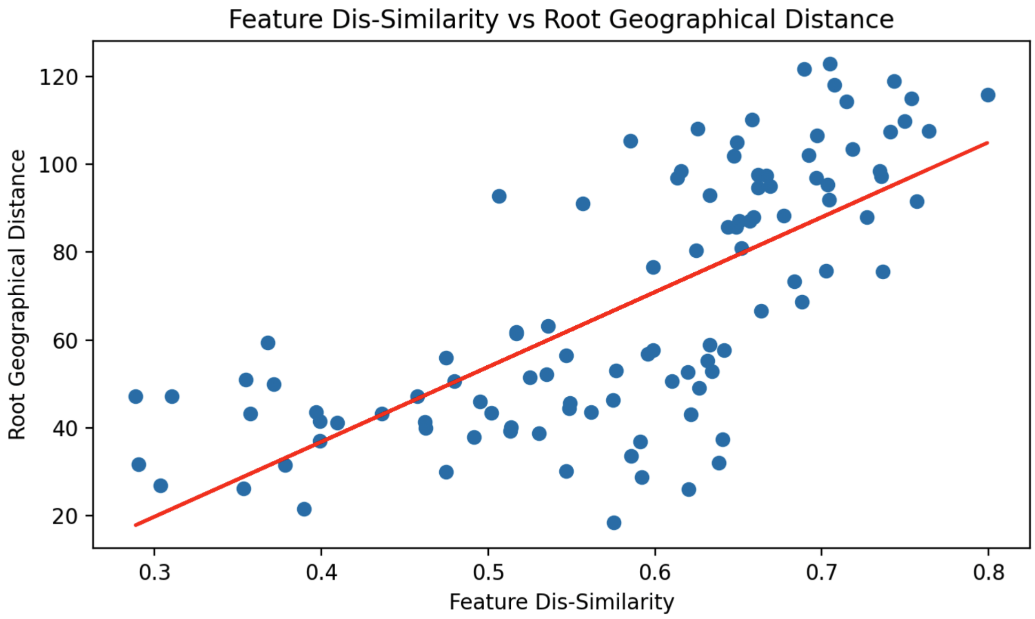

Figure 9: Feature Dis-Similarity vs. Square root of geographic distance
Based upon the assumption that languages originating more closely geographically will be more closely related (and, conversely, those that are more distant geographically will be less closely related) we attempted to use geographic location as a simple test for the (rough) validity of our approach based upon the WALS database. We found a 0.719 correlation, suggesting that the Feature Similarity does relate to actual language similarity.

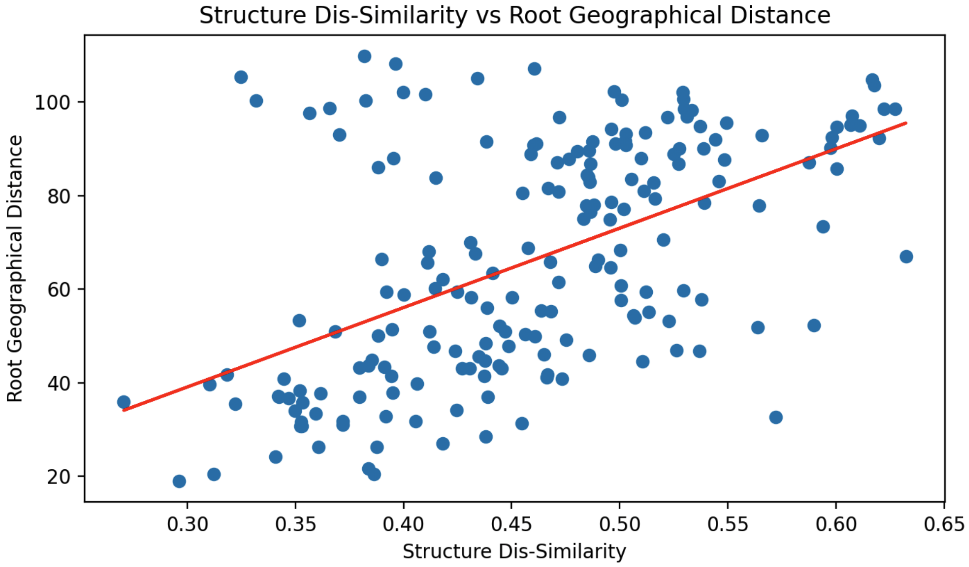

Figure 10: Structure Dis-Similarity vs. Square root of geographic distance
As with the Feature Dis-Similarity graph earlier, we attempted to use geographic location as a simple test for the (rough) validity of our approach — structure dis-similarity in this case. We found a 0.532 correlation, suggesting that the Structure Similarity does relate to actual language similarity.

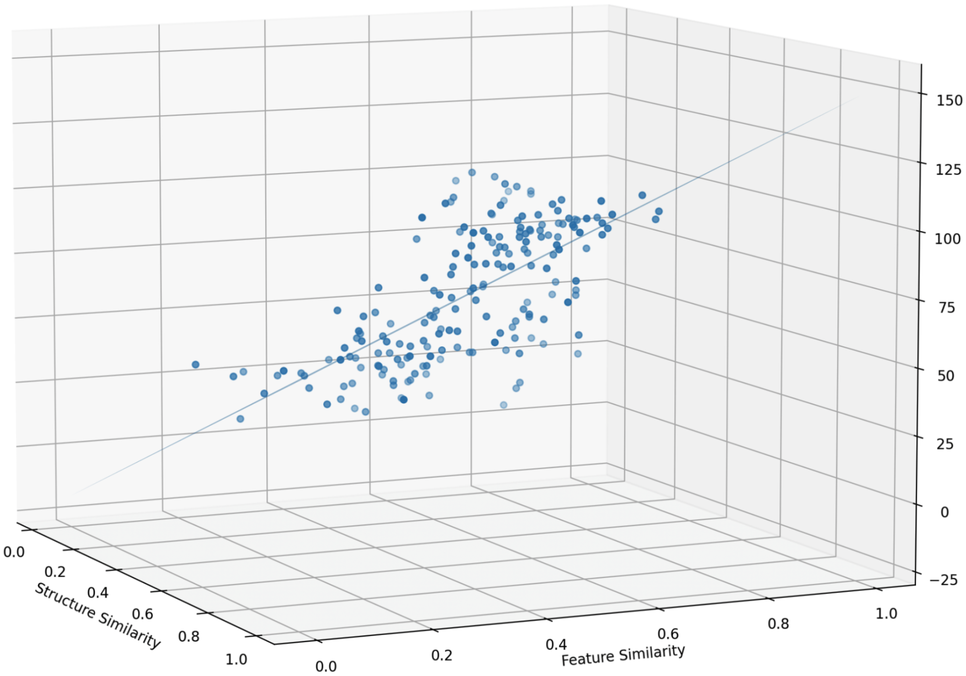

Figure 11: 3-Dimensional scatterplot
This graph demonstrates the relationship between all three metrics that have been examined, Structure Similarity, Feature Similarity, and the square root of geographical distance.

4 Extensions

Due to the limited time and resources available, and the richness of the subject matter, we were unable to explore many possible extensions to the current work. Below are listed a few of the additional directions we would like to pursue.

Combining IPA and Structural Edit Distance: In an effort to incorporate insights from both phonetic variation and structural variation, we could combine IPA edit distance with structure tree edit distance. We would need to (among other challenges) determine an appropriate weighting α and $1 - \alpha$ for the two measurements so that the combination would be maximally meaningful.

Weighting IPA Phoneme Transitions: Currently most IPA edit distance assumes that all phonetic edits are of equal significance, or have very rudimentary weighting (either 0.5 or 1). But, of course, not all phonemes are equally dissimilar. A transition between phoneme A and phoneme B might be much less probable (and hence should be weighted more heavily) than a transition between phonemes A and C. We would like to construct a model which uses much finer granularity of weights for phoneme transitions.

Vector Representation of Languages: Completing the implementation, and subsequent testing of the algorithm described in section 2.4.

Python Library of Tools: Several tools have been created in the development of this work. There may be value in making a python library of these tools and sharing them publicly.

Weighted Tree Edit Distance: Much as not all phoneme transitions are equally probable, not all structural changes (and differences) are equally probable. As such the elements of tree edit distance should be weighted differently. Doing so would represent an improvement over the work described in this paper (where currently all tree edits are weighted equally, and use the default set of edit operations).

In-Between Languages and Language Continua: By thinking of languages as lying on a continuum of similarity, and understanding the manipulations that can move languages along that continuum, one can create languages at arbitrary points between existing languages. This is one of the abilities that edit distance based methods allows. This enables a rich set of possibilities in which the approaches for continuous features and functions can be applied to languages that have, hitherto, been considered strictly as discreet (if ill defined) entities.

Reference Standard: It is challenging to assess the accuracy of language similarity assessments. If there was some better standard to measure against, or if we could create a better standard to measure against, that would enable us to improve our algorithms and to compare approaches. Candidates include: leveraging maps of degrees of mutual intelligibility (assuming such maps exist), looking at existing maps of language evolution, creating a standard from agreement between experts in the field, and more.

Setting Weights for Feature Similarity: Some language features are less common than others. As such, when computing language similarity, sharing less common features should weigh more heavily than sharing more common ones. Although setting the weights based upon the commonality of the feature may be valid, another approach may also be valid. Assuming the existence of a reference standard (as described above), we could set the weight of the shared features such that the resulting similarity assessments match the reference standard.

This could be done through a multitude of classical search methods, optimizing the following: we seek a set of weights $\hat{W} \subset \{w_i\}$ such that the loss $l = [\text{sf}(X,Y) - \text{metric}(X,Y)]^2$ (where ss denotes Shared Feature similarity) is minimized:

$$\hat{W} = \underset{W \subset \{w_i\}}{\arg\min} \{l(W; X, Y)\}$$

Then those weights could be applied to language pairs not yet assessed.

Phoneme Histogram Comparison: The frequency of phoneme usage may provide a fingerprint for any given language. The distance between such histograms might form a basis of comparison to assess language similarity.

5 Conclusion

Language is as multidimensional as the billions of people who use it. It is a product of unique needs and circumstances across all of human history. While it is impossible to represent them fully, when comparing languages we need to represent as many of their important dimensions as possible. In this paper we have attempted to go beyond strict Lexical Analysis and open the door to the beginnings of Language Analysis. While there is much more to be done, we hope that the work done here will prove useful and, maybe, inspire future efforts.

Acknowledgments

We would like to thank Professor Christiane Fellbaum for her inspiration and guidance, as well as all of my classmates in FRS116.

References

Ahmed, T., Nizami, M. S., & Khan, M. Y. (2020). Discovering lexical similarity through articulatory feature-based phonetic edit distance. *CoRR*, *abs/2008.06865*. https://arxiv.org/abs/2008.06865

Clark, A. (1998). *Being there: Putting brain, body, and world together again*. MIT press.

CoNLL-17. (2017). https://universaldependencies.org/conll17/

Do, Q., Roth, D., Sammons, M., Tu, Y., & Vydiswaran, V. (2010). Robust, light-weight approaches to compute lexical similarity.

Dryer, M. S., & Haspelmath, M. (Eds.). (2013). *Wals online (v2020.3)*. Zenodo. https://doi.org/10.5281/zenodo.7385533

Fellbaum, C. (1998, May). *WordNet: An electronic lexical database*. The MIT Press. https://doi.org/10.7551/mitpress/7287.001.0001

Glot, E. (n.d.). https://www.ezglot.com/

Greenhill, S. (2015). Evolution and language: Phylogenetic analyses. *International Encyclopedia of the Social Behavioral Sciences*. https://doi.org/10.1016/B978-0-08-097086-8.81035-1

Mutabazi, B. (2020). An algorithm for building language superfamilies using swadesh lists.

Nerbonne, J., & Heeringa, W. (2002). Measuring dialect distance phonetically.

Sanchez-Stockhammer, C. (2015). Can we predict linguistic change? an introduction. *Studies in Variation, Contacts and Change in English*. https://varieng.helsinki.fi/series/volumes/16/introduction.html

Swadesh, M. (1952). Lexicostatistic dating of prehistoric ethnic contacts. *Proceedings of the American Philosophical Society*, *96*, 452–463.

Swadesh, M. (1950). Salish internal relationships. *International Journal of American Linguistics*, *16*(4), 157–167. Retrieved March 17, 2024, from http://www.jstor.org/stable/1262898

Zhang, K., & Shasha, D. (1989). Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, *18*(6), 1245–1262. https://doi.org/10.1137/0218082

Appendix

Code can be found in https://github.com/TheEnquirer/LangDist