Deciphering Language Using Relative Entropy and Zipf's Law

Ryan Mahtab
Carnegie Mellon University
Department of Statistics and Data Science
rmahtab@andrew.cmu.edu

Abstract

Deciding whether a body of text is just a random array of symbols or an actual representation of meaningful language is a complex and subjective process. The approach used in this analysis utilizes relative entropy and Zipf's Law to determine whether a corpus of text represents language. The relative entropy of a text is the ratio between its symbol-based entropy and the maximum entropy possible given the symbols. A text's closeness to Zipf's distribution is measured using Kullback-Leibler divergence with a fitted Pareto distribution. The final decision to classify a set of symbols as language is made using a score based on these two calculations.

1. Data Preprocessing

Minimal preprocessing is performed on the given set of 13 text files. Spaces and new lines are kept in and interpreted as characters as they represent breaks in tokens and carry information. However, these whitespaces are not accounted for in the word token frequencies since they are used to separate the tokens themselves. Unigram, bigram, and trigram frequency dictionaries are constructed, but only the unigram dictionary is relevant for the calculated scores. An example unigram frequency table is shown in Table 1.

Tokens are not lower-cased, since it is unspecified whether the 13 given texts have capitalizations, and if so what rules they follow.

Table 1: Unigram Frequency Table of Text File 2

Word	Rank	Frequency
żỤŵ	1	34
štŷť	2	31
űŏŚŴ	3	31
ļŃŞňũ	4	31
ŜŐŪ	5	31
...	...	...

2. Relative Entropy

The notion of entropy in the context of the characters of a language is the uncertainty with which symbols are selected from a set with given probabilities.[1] In a language whose characters have maximum entropy, in other words all characters are equally likely, choosing the next character in a sequence is essentially random. Using this logic, as a language's entropy decreases, it is becoming less random.

Calculating the relative entropy of a given text gives us an indication of the structure of the spelling of language. It gives a picture of whether words are random permutations of characters or if there are certain rules governing how characters can be put together to form words, sentences, and ultimately ideas.

The first step in calculating a text's character-based relative entropy is calculating its own entropy. For each text, symbol frequency dictionaries were constructed and normalized over the number of unique symbols to give a probability density function for the distribution of symbols. Note that these dictionaries also take whitespace characters into account. An example of such a dictionary is given in Table 2. These probabilities are then used in the entropy equation to calculate the entropy of the given text.

Next, the maximum entropy of each language is computed. Maximum entropy occurs when each symbol has the same probability of occurring no matter what the previous symbols in the

[1] Claude Shannon

sequence are. Therefore this occurs when each symbol has the exact probability of 1/n where n is the number of unique symbols (including whitespaces) in the given text. Again, these probabilities are used in the entropy equation to calculate the *maximum* entropy of each given text.

Table 2: (Unranked) Symbol Frequency Table for Text File 1

Symbol	Normalized Frequency
Ë	0.007218
C	0.0456038
M	0.031490
ǫ	0.067544
...	...

Finally, the relative entropy is the ratio between the initially computed entropy and the maximum entropy. This number will always be between 0 and 1. The closer a text's relative entropy is to 1, the more random its word structure is. A table of each given text's relative entropy is shown below.

Table 3: Relative Entropy Values of Each Given Text

Text	Relative Entropy	Text (continued)	Relative Entropy (continued)
1	0.799648	8	0.875227
2	0.878982	9	0.699176
3	0.455717	10	0.872847
4	0.849670	11	0.872228
5	0.875801	12	0.900045
6	0.735950	13	0.705617
7	0.734137		

3. Zipf's Law

Another characteristic of the structure of languages can be observed in their word token distributions. The word token distributions should follow Zipf's Law, which is a probability distribution where events, in this case word frequencies, are inversely proportional to their ranks. While it is difficult to fit Zipf distributions since they do not have closed form maximum likelihood estimates for their parameters, it is possible to find parameter estimates for Pareto distributions, which are analogous for the purposes of this analysis.

Using the unigram frequency dictionaries from earlier to supply the word token frequencies and ranks, we can estimate the parameters, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ of the fitted Pareto distribution for each corpus of text:

$$\boldsymbol{\alpha} = n / (\sum \ln(x_i) - n\ln(\beta)) \qquad \boldsymbol{\beta} = \min_i X_i$$

Both the word frequency/rank data and the modeled Pareto function are transformed on a log-log scale. After Pareto distributions have been fitted to the data of each corpus, we need to determine how well the actual data follow it to see how closely the texts abide by Zipf's Law. This is done through the use of Kullback-Leibler divergence, which helps quantify the distance between two distributions.[2] The computed divergence values are shown below in Table 4.

Table 4: Kullback-Leibler Divergence Values for Each Given Text

Text	KL Divergence	Text (continued)	KL Divergence (continued)
1	2036.307892	8	4.239918
2	1446.905943	9	3171.729715
3	1335.869791	10	13.146736
4	743.678471	11	1108.437377
5	14.497046	12	751.700201
6	9097.490202	13	3631.133715
7	410.340914		

[2] KL Divergence code adapted from Cory Maklin

4. Scoring and Results

Now that relative entropy and KL divergence values have been calculated for each given text, we can begin the process of classifying them as languages or not. A reference corpus from known English language is used as a threshold to make this decision, specifically the first 10 chapters of Jane Austen's "Pride and Prejudice"[3]. This text's relative entropy and KL divergence from a fitted Pareto divergence are computed: 0.744035 and 2867.128641 respectively.

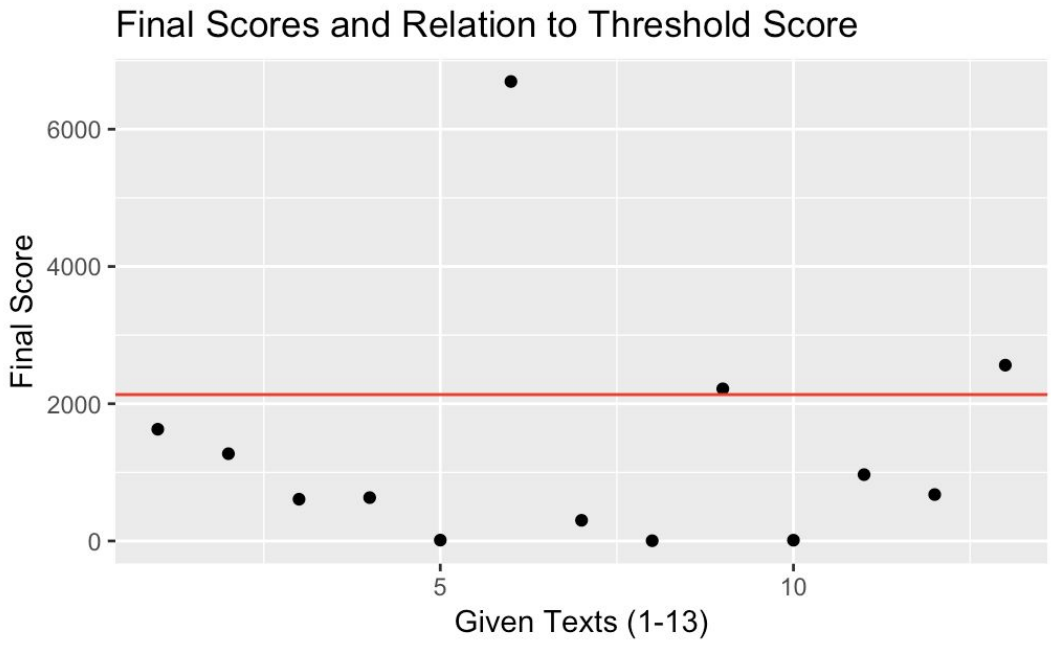
The scoring assigned to each text (for both the reference corpus and the 13 given texts) is simply the product of the relative entropy and KL divergence. Both values increase as bodies of text become more random and decrease as texts become more structured and meaningful. Thus, scaling the divergence, which is a function of words, by the relative entropy, which is a function of characters, should give a clearer description of the structure of the language overall. These scores are rounded to 6 decimal places and are displayed in Table 5 below. A plot with their relation to the value of the reference corpus's score is shown in Figure 1.

Table 5: Final Scores Assigned to Each Given Text

Text	Final Score	Text (continued)	Final Score (continued)
1	1628.329533	8	3.710891
2	1271.804280	9	2217.597295
3	608.778574	10	11.475089
4	631.881286	11	966.810116
5	12.696527	12	676.564007
6	6695.297914	13	2562.189679
7	301.246448		

[3] Jane Austen, Project Gutenberg EBook

Figure 1: Final Scores of Each Given Text and Relation to Threshold Score



As can be seen from the plot, texts 6, 9, and 13 are above the threshold score. By the definition of the scoring system, we classify these as "Not Language". That means texts 1, 2, 3, 4, 5, 7, 8, 10, 11, and 12 are classified as "Language" by our analysis. A final table of results is provided below in Table 6.

Table 6: Final Language Classifications

Text	Classification	Text (continued)	Classification (continued)
1	Language	8	Language
2	Language	9	Not Language
3	Language	10	Language
4	Language	11	Language
5	Language	12	Language
6	Not Language	13	Not Language
7	Language		

5. References

[1] “Entropy and Redundancy in English”. *Claude Shannon & Information Theory*. https://people.seas.harvard.edu/~jones/cscie129/papers/stanford_info_paper/entropy_of_english_9.htm

[2] Maklin, Cory. (2019) “KL Divergence Python Example”. *Towards Data Science*. https://towardsdatascience.com/kl-divergence-python-example-b87069e4b810

[3] Austen, Jane. (1813) Produced by Widger, David et al. (2019) “Pride and Prejudice”. https://www.gutenberg.org/files/1342/1342-h/1342-h.htm