Significance of Network Metrics

Complex and Social Networks (CSN)

Leandra Moonsammy, Odysseas Kyparissis

October 19, 2023

Facultat d'Informàtica de Barcelona (FIB)
Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

Contents

List of Figures

List of Tables

1 Introduction

Network metrics play a vital role in understanding the structure and properties of complex systems. In this report, we delve into the significance of network metrics by employing collections of global syntactic dependency trees from various languages. In these networks, vertices represent words, and links signify syntactic dependencies between words in syntactic dependency treebanks[1].

Syntactic dependency treebanks are comprised of collections of sentences along with their corresponding syntactic dependency trees. For simplicity, we focus on undirected versions of global syntactic dependency networks. A fundamental metric of a network is the global clustering coefficient which measures the tendency of nodes to form tightly-knit clusters. It quantifies the extent to which nodes in the network tend to form triangles. In other words, it assesses how likely it is for a node's neighbors to also be connected to each other. Our objective is to investigate whether the clustering coefficient of real language networks is significantly large compared to the clustering coefficient of random networks.

The analysis conducted involves data preparation, hypothesis testing, and the implementation of a Monte Carlo procedure to estimate the significance of the network metrics. We begin by processing the raw data, removing loops and multi-edges, and calculating fundamental network properties. We then construct two null hypotheses: one based on Erdős-Rényi graphs and the other using a switching model to generate random networks with the same degree sequence. Our goal is to test whether the observed network metrics in the real data significantly deviate from those expected under these null hypotheses. We also discuss optimizations for both exact and approximate results and the rationale behind them.

This report serves as a comprehensive exploration of the significance of network metrics in syntactic dependency networks and aims to shed light on the underlying structural properties and the differences observed in various languages.

The structure of the document is as follows: the results are presented in Section2. Next, Section3 covers the discussion, which involves commentary on the results and also some observations regarding the optimization of the calculations. Lastly, Section4 presents the methodology followed during the implementation of the solution.

2 Results

In this section the obtained results are included. Firstly, some basic statistics on the structure of the languages' networks are presented (after applying the preprocessing steps, see Section4), followed by visualizations depicting the results of the hypotheses testing.

2.1 Insights

We first introduce Table1 to showcase the information related to the network of each language. This table summarizes some statistical information about the syntactic dependency networks, as well as, the loops removed during the preprocessing steps.

Language	N	E	$\langle k \rangle$	δ	*Loops removed*
Arabic	21531	68742	6.385397799	0.000296581	25
Basque	12207	25541	4.184648153	0.000342835	17
Catalan	36865	197075	10.69171301	0.000290031	243
Chinese	40297	180924	8.979527012	0.000222839	157
Czech	69303	257254	7.424036477	0.000107126	41
English	29634	193078	13.03084295	0.000439741	108
Greek	13283	43961	6.619137243	0.000498354	13
Hungarian	36126	106681	5.906051044	0.000163489	35
Italian	14726	55954	7.599348092	0.000516085	88
Turkish	20409	45625	4.471066686	0.000219084	17

Table 1: Summary of the properties of the language networks. N is the number of vertices of the network, E is the number of edges, $\langle k \rangle = \frac{2E}{N}$ is the mean degree and $\delta = \frac{2E}{N(N-1)}$ is the network density of edges.

Moreover, Table2, presents the value of the clustering coefficient in the real networks, together with the two estimated p-values, one for the null hypothesis of a binomial graph and another for the null hypothesis of the switching model.

Language	C_{WS}	$p-value(binomial)$	$p-value(switching)$
Arabic	0.188490684	0.0000	0.7160
Basque	0.046710615	0.0000	0.4500
Catalan	0.22105654	0.0000	0.8120
Chinese	0.170799365	0.0000	0.5280
Czech	0.121714084	0.0000	0.8180
English	0.235280796	0.0000	0.8480
Greek	0.1337727	0.0000	0.8520
Hungarian	0.050847188	0.0000	0.9600
Italian	0.143705715	0.0000	0.9700
Turkish	0.223585095	0.0000	0.8280

Table 2: Summary of the hypotheses testing results on clustering coefficient C_{WS}.

2.2 Visualizations

In this section we include visualizations of the analysis' results, in order to better understand the hypotheses tests and derive conclusions. In this section only two of the ten languages are covered: ***Arabic*** (Figures 1, 2, 3) and ***Chinese*** (Figures 4, 5, 6). The visualizations of the

remaining languages are included in the Appendix. For each language the following plots were generated:

- visualization of the degree sequence of the network, in 4 different scale combinations (linear-linear, linear-logarithmic, logarithmic-linear and, finally, logarithmic-logarithmic),
- histogram of the values of x_{NH} together with the real value of x for the Erdős-Rényi model null hypothesis,
- histogram of the values of x_{NH} compared to the real value of x for the Switching model null hypothesis.

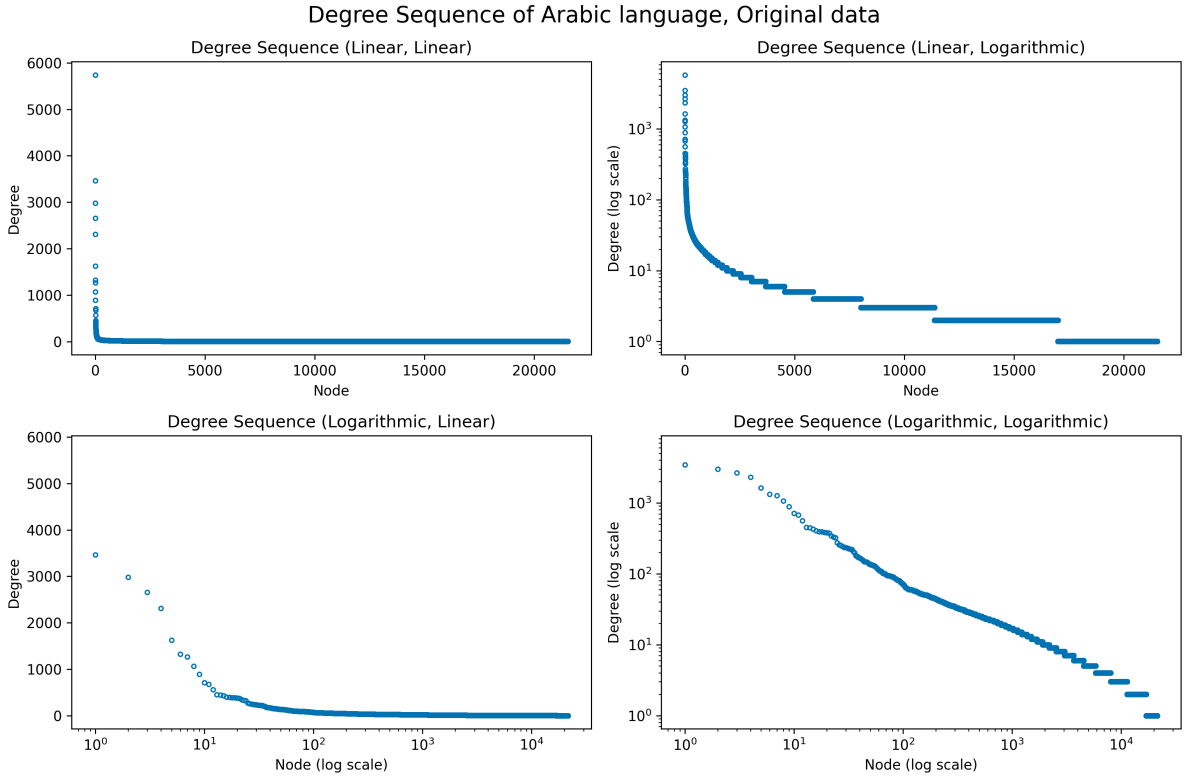

Figure 1: Arabic Language: Degree Sequence of Original Data

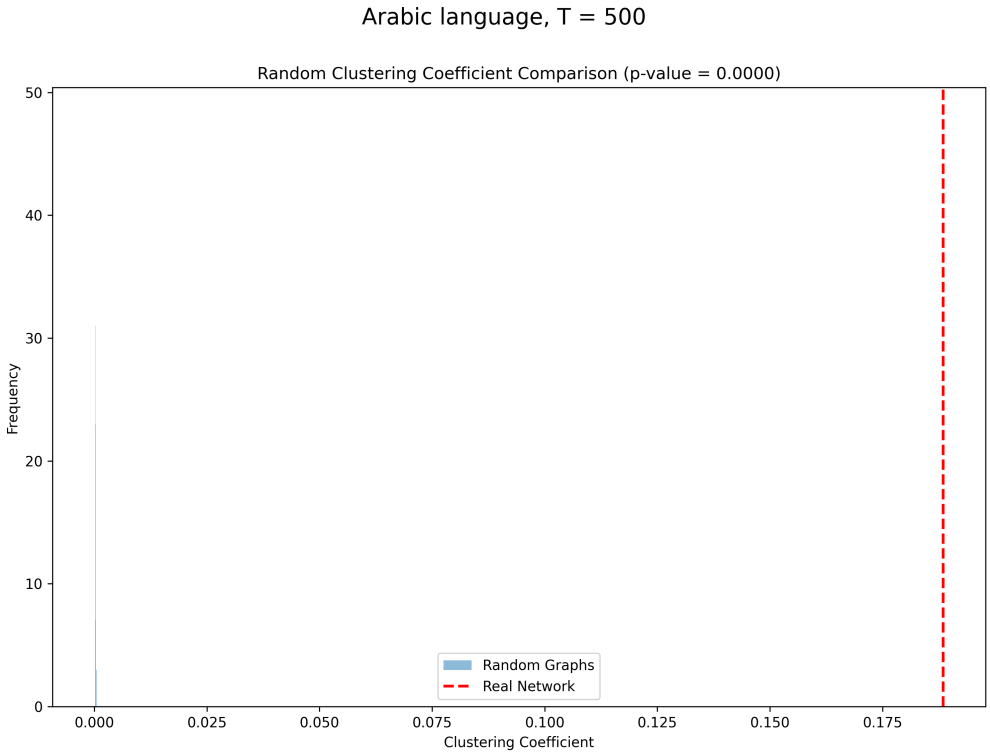

Figure 2: Arabic Language: Erdős-Rényi null hypothesis

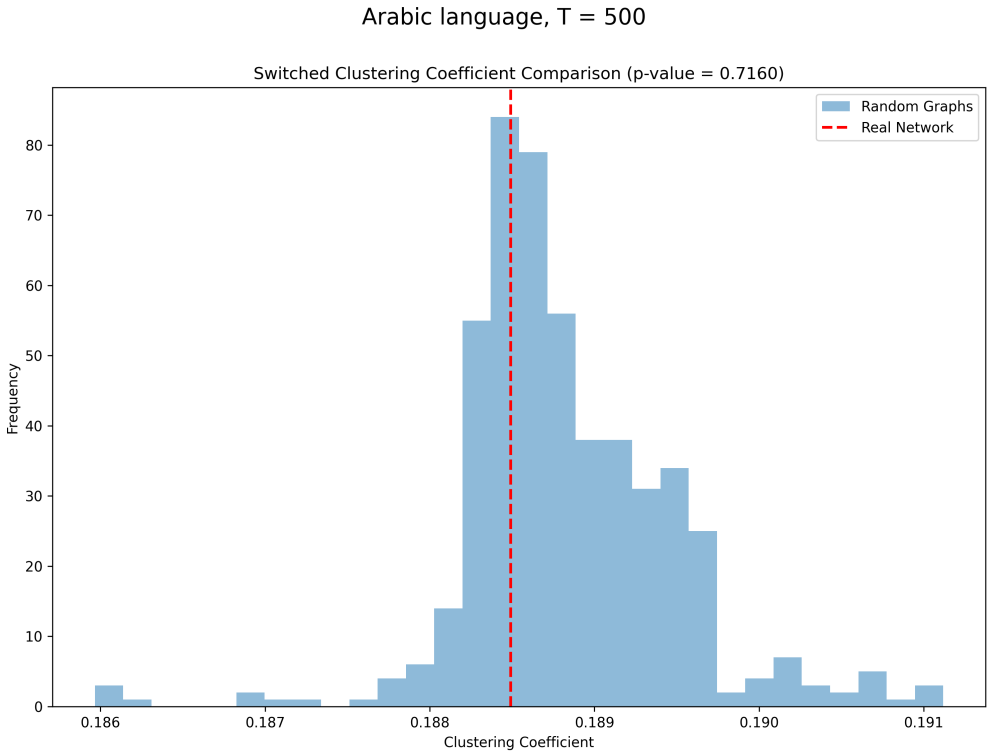

Figure 3: Arabic Language: Switching model null hypothesis

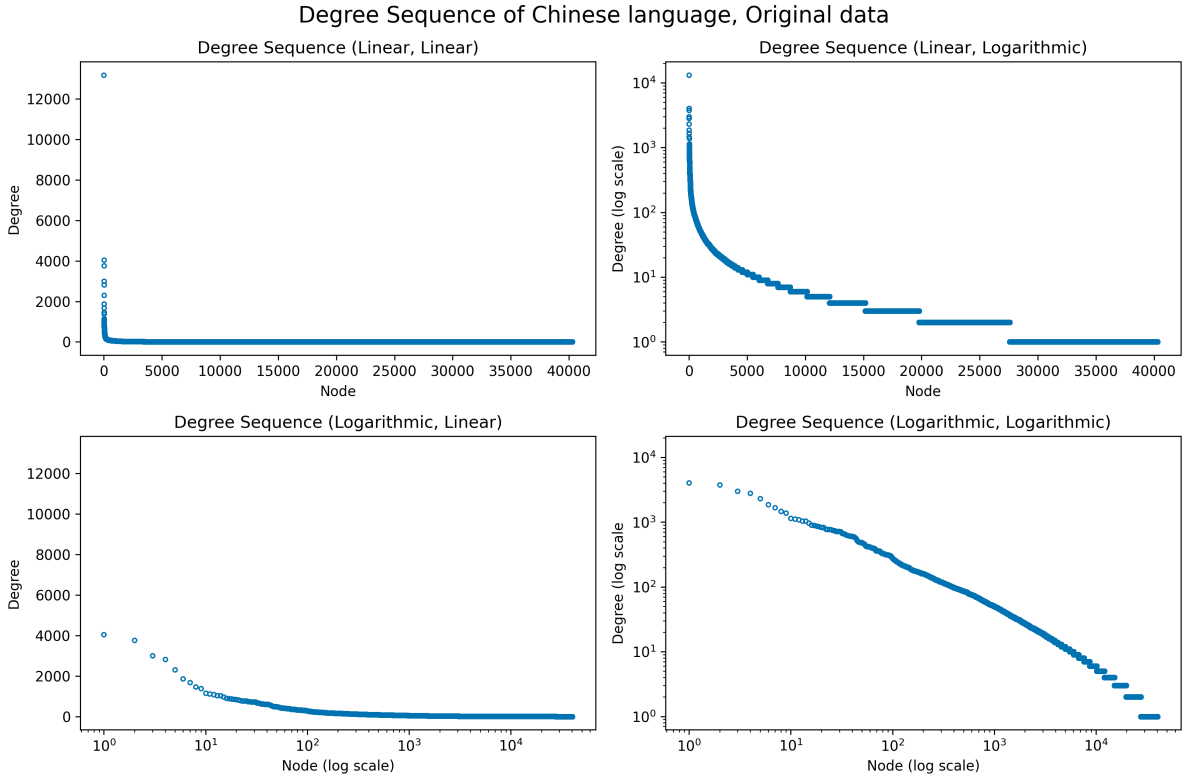

Figure 4: Chinese Language: Degree Sequence of Original Data

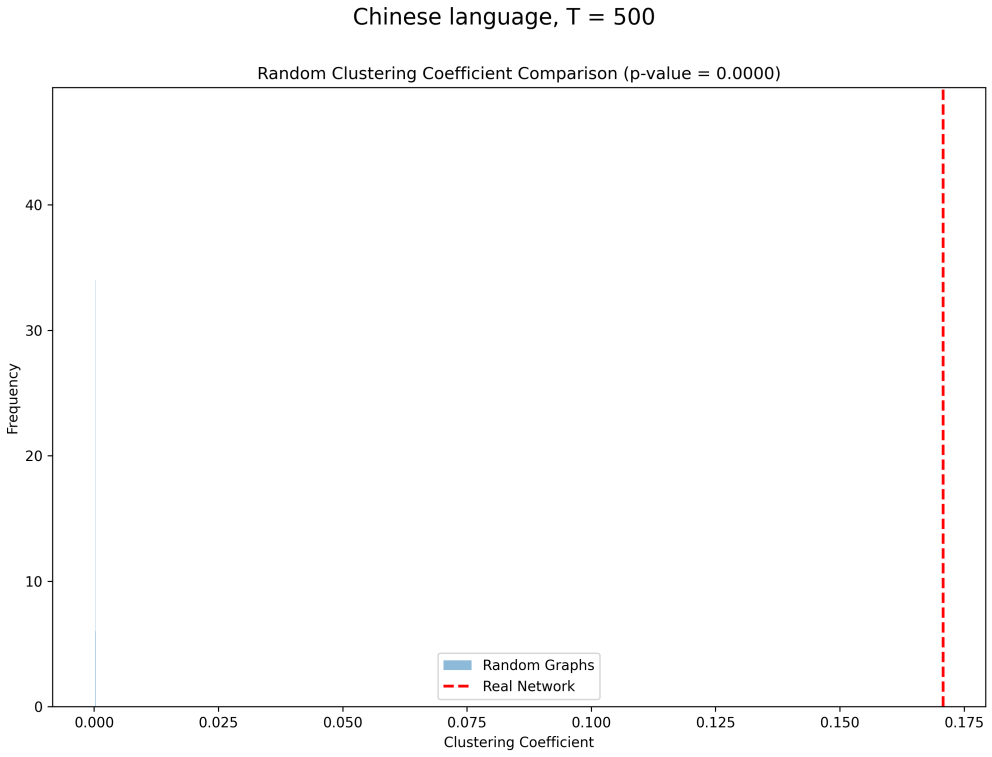

Figure 5: Chinese Language: Erdős-Rényi null hypothesis

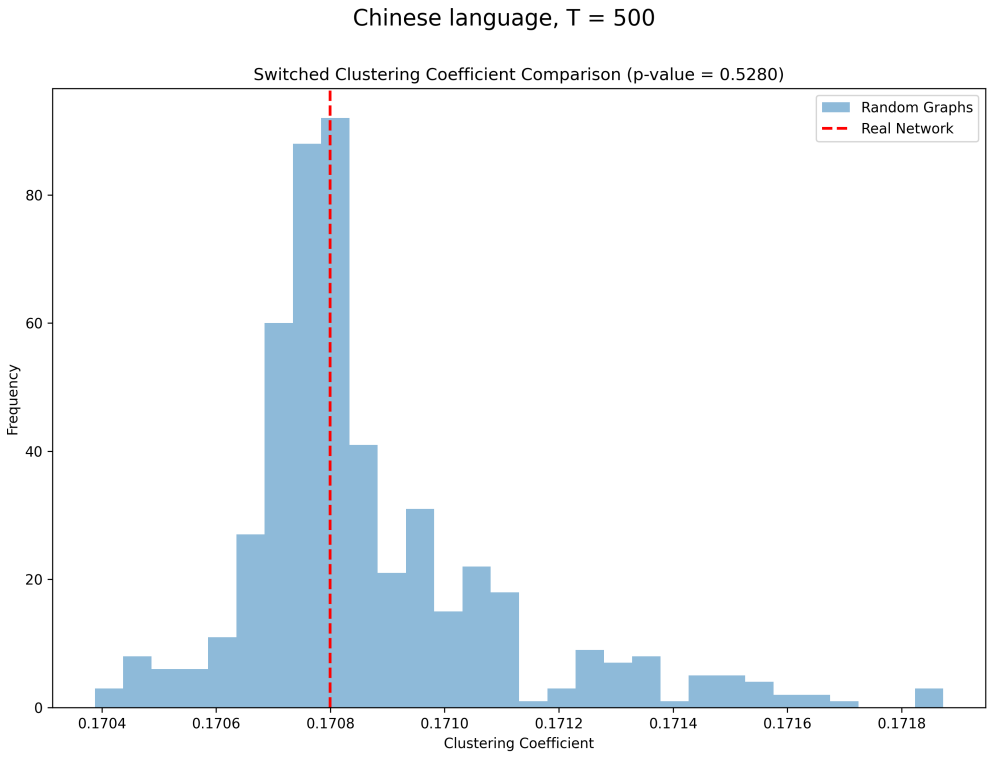

Figure 6: Chinese Language: Switching model null hypothesis

3 Discussion

Visualizing the degree sequence of the network revealed that each of the languages resemble a power law as the log-log plots resemble straight lines. In particular the Hungarian language stands out. All of the languages have many low degree nodes and few high degree nodes as expected in a power law degree distribution. This also impacts clustering coefficients.

For all languages examined in this study, the p-values ($p(x_{NH} \geq x)$) calculated under the Erdős-Rényi null model are smaller than the chosen significance level ($\alpha = 0.05$). This indicates that the observed clustering coefficient (x) is significantly larger than what would be expected in a random network. In essence, our hypothesis testing consistently reveals that the clustering coefficient of the syntactic dependency networks is significantly higher than one of the Erdős-Rényi random graph model, across all languages under investigation.

The fact that all p-values are precisely 0.0000 when compared to the Erdős-Rényi null model provides compelling evidence for the strong rejection of the null hypothesis. This result seems to be a sound conclusion, since in the sentences of a language, the words (nodes) are not connected with each other in a binomial random manner, but instead, there is a structured procedure which humans follow in order to form sentences. These findings underscore the need for more sophisticated network models capable of capturing the intricate, non-random patterns present in linguistic data. The consistent presence of higher clustering coefficients across languages shows that there are syntactic rules and patterns that exhibit universality across these diverse linguistic contexts.

In contrast, for the Switching model null hypothesis, we observe that $p(x_{NH} \geq x)$, the p-value, is not smaller than $\alpha = 0.05$. Thus, we cannot conclude that x is significantly large in this case. When preserving the degree distribution using the switching algorithm, the p-values vary across languages, but they all remain considerably higher than the significance level. For all of the languages the p-values are above 0.05 and has a range of 0.4500 to 0.9700 indicating

that the observed clustering coefficients in the language syntactic dependency networks do not significantly deviate from what would be expected under the Switching model null hypothesis. This suggests that, unlike the Erdős-Rényi random model, the observed clustering coefficients align more closely with the expectations of the Switching model.

More specifically, by taking a look at the results of Table2, it can be seen that most of the p-values of the switching model, tend to be close to 1 (70% of the cases values are higher than 0.8). The only languages that occur values of p-value close to 0.5 and not 1 is the Basque and the Chinese language. Also, by consulting the properties of the networks for those two languages from Table1, there is not a clear pattern that would indicate a reasoning behind this result. The reasoning behind this difference, would need furhter investigation.

However, the reasoning behind the idea that the observed clustering coefficients is more alligned with the expectations of the Switching model, compared to the Erdős-Rényi one is analysed here. The Switching model, unlike the Erdős-Rényi model, primarily focuses on preserving the degree distribution of the original network while rewiring edges. This means that it maintains the same number of connections for each node. In syntactic dependency networks, where nodes represent words and edges represent grammatical relationships, the degree distribution is an essential property since it reflects the inherent structure of language. Moreover, syntactic dependency networks are characterized by a strong degree of local structure. In these networks, words (nodes) are connected based on grammatical relationships, which tend to create clusters and small-world phenomena. The Switching model, by preserving local degree distributions, may naturally capture these clusters and preserve the network's local structural properties, including clustering. Furthermore, the formation of sentences in a language follows specific syntactic and grammatical rules. Thus the Switching model, by preserving the degree distribution while rewiring edges, might as well, inadvertently maintain some of the syntactic rules that govern sentence construction. This could lead to clustering patterns that align with the underlying linguistic principles.

In summary, these results imply that the clustering coefficients in the language networks are not random, and they are significantly different from those generated by the Erdős-Rényi null model. However, when you preserve the degree sequence in the Switching model, the clustering coefficients are not significantly different from those in the real network, which suggests that the degree distribution is a key factor explaining the clustering behavior in these languages. This information can be used to gain insights into the structural properties of the language networks.

3.1 Conclusion

In conclusion, our analysis of syntactic dependency networks across various languages has highlighted the presence of non-random clustering coefficients, significantly deviating from Erdős-Rényi null model expectations. However, when preserving the degree distribution using the Switching model, the observed clustering coefficients align more closely with the model's expectations. This underscores the importance of the degree distribution in explaining clustering behavior in linguistic networks, revealing underlying structural properties of language. Further investigation is needed to understand the subtle variations observed, particularly in the Basque and Chinese languages. These findings emphasize the necessity for advanced network models capable of capturing the intricate, non-random patterns in linguistic data.

4 Methodology

In this section a detailed description of the methodology that followed during the implementation of the current assignment is taking place.

4.1 Data preparation

To begin with, the file *dependency_networks.tar.gz* included in this lab's package contains the description of the global syntactic dependency graphs from different languages. More specifically, each file consist of a header and a list of edges. The header row contains the number of vertices and the number of edges of the network; although the other rows indicate the pairs of linked vertices.

As those networks contain loops and multi-edges the first step of the solution provided was to preprocess the data. The preprocessing steps included the removal of loops and multi-edges, as well as the calculation of the new number of nodes, number and density of the edges, the average degree of the network, and finally the respective clustering coefficient. The removal of self-loops and multi-edges was conducted to allow a more simple analysis of the problem. The results of the basic statistics on the languages' networks that obtained after processing the graphs have already been presented in Table 1.

4.2 Clustering coefficient

To continue with, as the goal of the specific solution was the hypotheses testing on the networks' clustering coefficient, it is necessary to describe how the calculations were completed. More specifically, the global clustering coefficient of a network can be defined as [5]:

$$C = \frac{number\ of\ closed\ paths\ of\ length\ 2}{number\ of\ paths\ of\ length\ 2} \tag{1}$$

Alternatively, a mean local clustering was defined by Watts & Strogatz (WS) as [6]:

$$C_{WS} = \frac{1}{N}\sum_{n=1}^{N} C_i \tag{2}$$

where N is the number of nodes in graph G, and C_i is the local clustering coefficient of node v_i (Eq. (3). For unweighted graphs, the local clustering coefficient of a node v_i is the fraction of possible triangles through that node, and is given by the equation:

$$C_i = \frac{2T(v_i)}{deg(v_i)(deg(v_i)-1)} \tag{3}$$

where $T(v_i)$ is the number of triangles through node v_i, and $deg(v_i)$ is the degree of v_i. Finally, the value of C_i is assigned to 0 if $deg(v_i) < 2$. The mentioned formulas are already implemented and contained in the average_clustering method of NetworkX[2] library of Python[1].

However, given a network G, C and C_{WS} can differ substantially. C can be dominated by the contribution of vertices of high degree (which have many adjancent nodes). C_{WS} can be dominated by the contribution of vertices of low degree (which are many in the majority of networks). For that reason, the value of C_{WS} is set to zero when $deg(v_i) < 2$ occurs. The conclusion is that C and C_{WS} measure transitivity in different ways (different assumptions/goals).

[1] `https://www.python.org/`

4.3 Test of significance

Moreover, as previously mentioned the goal of this study is to analyse the clustering coefficient of real networks, and produce hypotheses tests, which will compare the metric of the real network, with ones belonging to random generated graphs. In this session, the different approaches of random graphs generation, lead to two different null hypotheses:

- A hypothesis test focused on binomial graphs (Erdős-Rényi model) with the same number of vertices and edges as the real network. This null model has no free parameter.
- A hypothesis test including randomized graphs with the same degree sequence of the original graph. The *switching model* is the randomization technique used here to generate random graphs with the same degree sequence. This null model has **two parameters**: the original network structure (the list of edges) and Q. The number of random switchings tried is $Q * E$, where E is the number of edges. The number of trials QE has to include cases where the random switching could not be performed. Consequently, the value of Q was set to be equal to 20, although based on the solution of the coupon collector's problem, and the largest number of edges (257253 for Czech language), a value of 5 was enough.

Furthermore, the way a hypotheses testing is conducted is presented here. As we want to determine if the value of a network metric x is significantly large, with regard to a certain null hypothesis, x_{NH} is calculated on the random generated graphs following the null hypothesis and it is compared to the real value of x. We say that x is significantly large if $p(x_{NH} \geq x)$ the so-called p-value, is small enough, e.g. smaller than a significance level α (which is usually set as: $\alpha = 0.05$). The following subsections include details on the two different hypothesis tests.

4.3.1 Erdős-Rényi null hypothesis

One of the families of null hypotheses includes the procedure of pairing of vertices chosen uniformly at random (Erdős-Rényi model). In this family two different approaches exist:

- Testing the hypothesis on Erdős-Rényi graphs with variable number of edges (parameters N and π).
- Testing the hypothesis on Erdős-Rényi graphs with constant number of edges (parameters N and E).

During the implementation of this specific solution, the second approach was followed. However, this approach introduces unrealistic degree distribution on the random generated graph. In greater detail, once the number of nodes (N) and the number of edges (E) of the original language networks are extracted from the provided data files, a binomial graph (Erdős-Rényi model) with the same number of vertices and edges, as the real network, can be constructed. To do so, firstly, it is necessary to calculate the probability of randomly connecting two nodes of an Erdős-Rényi model graph. This probability can be calculated as:

$$p = \frac{2E}{N(N-1)} \tag{4}$$

Then by applying the method erdos_renyi_graph(num_nodes, p) of NetworkX[2], the generation of a random graph with the same number of nodes and edges as the original one is completed. In the following subsections, it is described in detail how those random models were used in order to conduct the hypotheses test, the results of which were presented in section2.

4.3.2 Switching model null hypothesis

An other family of null hypotheses includes processes of random pairing of vertices constraining the degree distribution. One of those processes is the following:

- Given a degree sequence: $k1, k2, ...k_{N_{max}}$ the switching model is being produced.

The generation of the switching model follows the algorithm below:

1. Input: a network of E edges and Q (a parameter).
2. Repeat $Q * E$ times:
   - Choose two edges uniformly at random: $u - v$ and $s - t$.
   - Exchange the ends to give $u - t$ and $s - v$ if they are well-formed according to given constraints (on loops, multiple edges,...).
   - Failures must be counted for detailed balance [4]

The implementation of the algorithm described above is implemented by the method double_edge_swap of NetworkX[2]. More precisely, from the documentation of the respective package, the following details are mentioned concerning the way of generating the switches and retaining exactly the same degree distribution.

- A double-edge swap removes two randomly chosen edges $u - v$ and $x - y$ and creates the new edges $u - x$ and $v - y$.
- If either the edge $u - x$ or $v - y$ already exist no swap is performed and another attempt is made to find a suitable edge pair.
- Instead of choosing uniformly at random from a generated edge list, this method chooses non-uniformly from the set of nodes with a probability weighted by degree.
- A trick to avoid periodicities is to pick two random edges without creating an edge list and then choose the source node indices from a discrete distribution.
- When a switch to be performed has same source ($u = x$) it is being skipped.
- Also, there is a check for not generating parallel edges.
- Finally, the target is chosen uniformly from neighbors

Given the procedure above, it is sure that the degree sequence will be preserved and that no self-loops or multi-edges are going to exist.

4.4 Implementation

So far, we have assumed that x or x_{NH} have to be calculated exactly. However, it is possible to estimate x or x_{NH} faster but with some error through a **Monte Carlo** procedure. The key is that the error is small. The Monte Carlo procedure follows the idea described here:

$$p(x_{NH} \geq x) \approx \frac{f(x_{NH} \geq x)}{T} \tag{5}$$

where T is the number of random graphs produced and $f(x_{NH} \geq x)$ is the number of those graphs where $x_{NH} \geq x$. To be more precise, the Monte Carlo procedure follows the algorithm:

1. Initialize counter: $f(x_{NH} \geq x) = 0$.
2. For i from 1 to T:
   - Generate random network following the null hypothesis
   - Calculate x_{NH} (clustering coefficient) on that network
   - Check if $x_{NH} \geq x$. If true, increment $f(x_{NH} \geq x)$.
3. Estimate $p(x_{NH} \geq x)$ as $\frac{f(x_{NH} \geq x)}{T}$.

However, T must be large enough, following the convention $\frac{1}{T} \ll \alpha$. As we use 0.05 as the value of α, T was set to be equal to 500.

4.4.1 Optimizations keeping results exact

Imagine that we have a way to bound x_{NH} below and above, x_{NH}^{min} and x_{NH}^{max} respectively just by having explored a subset of the vertices and/or edges of the network. Then if $x_{NH}^{min} \geq x$ it can be concluded that $x_{NH} \geq x$. The same is true for performing analysis on the relationship $x_{NH} \leq x$, although in this case we compare with the x_{NH}^{min}.

So, instead of computing 2, an estimate of C_{WS} is calculated from a mean of C_i over a small fraction of randomly selected vertices. This approach can lead to high precision exploring a small fraction of nodes (e.g., 5%). Also, imagine that the vertices of the network have been sorted producing a uniformly random permutation of the original vertices. Then, good estimates of the metrics can be obtained by just computing local metrics only for the M first vertices as:

$$C_{WS} \approx \frac{1}{M} \sum_{n=1}^{M} C_i \tag{6}$$

Obviously, the estimation is perfect when $N = M$. Interestingly, a good estimation can be obtained even when $M \ll N$ (e.g., $100M/N = 10\%$ or even smaller). However, an attempt of the mentioned approaches was not performed, but it is considered as future work. On the other hand, the generation of the results was feasible due to the usage of parallel computing and multi-threading.

4.4.2 Parallel computing and multi-threading

To minimize the computational time of the Monte Carlo procedure and produce the presented results, parallel computing and multi-threading was used. In greater detail, the Pool method of the multiprocessing[3] package was utilized. In this way, the 500 ($T = 500$) Monte Carlo estimations on each language, was possible to be executed in parallel, while taking advantage of a CPU architecture consisting of 16 threads. To conclude, the approach followed, did not transform the problem from $O(N^2)$ to $O(N)$, as it is possible with the methods presented in theory, however the generation of the results were completed in a couple of hours.

References

[1] Ramon Ferrer i Cancho, Ricard V Solé, and Reinhard Köhler. "Patterns in syntactic dependency networks". In: *Physical Review E* 69.5 (2004), p. 051915.

[2] Aric Hagberg, Pieter Swart, and Daniel S Chult. *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[3] Michael M McKerns et al. "Building a framework for predictive science". In: *arXiv preprint arXiv:1202.1056* (2012).

[4] Ron Milo et al. "On the uniform generation of random graphs with prescribed degree sequences". In: *arXiv preprint cond-mat/0312028* (2003).

[5] Mark Newman. *Networks*. Oxford university press, 2018.

[6] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world'networks". In: *nature* 393.6684 (1998), pp. 440–442.

A Appendix

As mentioned in Section2, here we present the respective figures for the remaining languages.

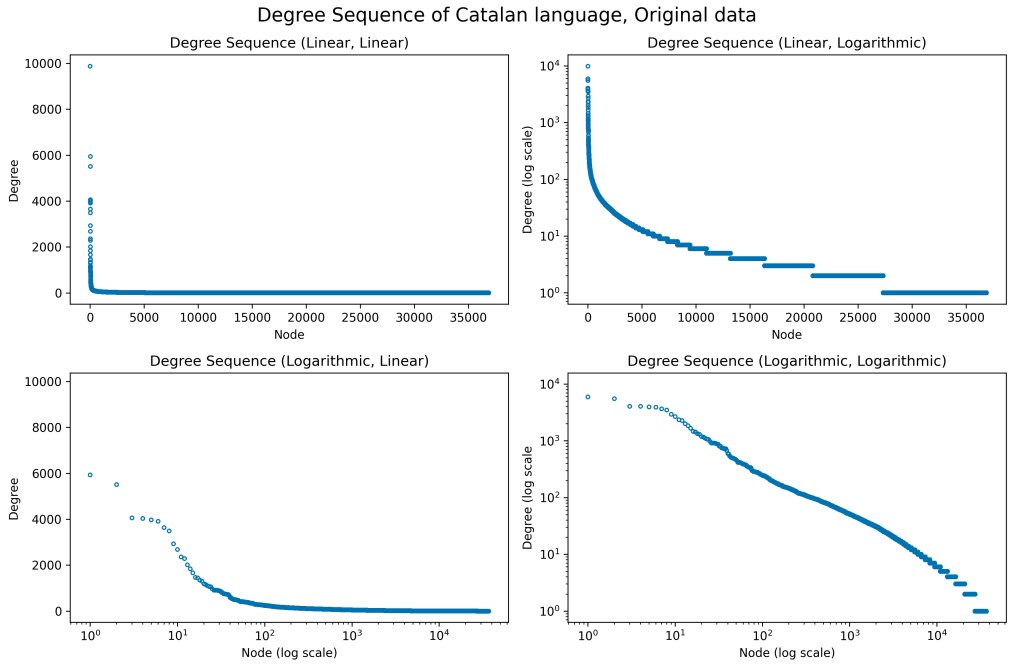

Figure 7: Catalan Language: Degree Sequence of Original Data

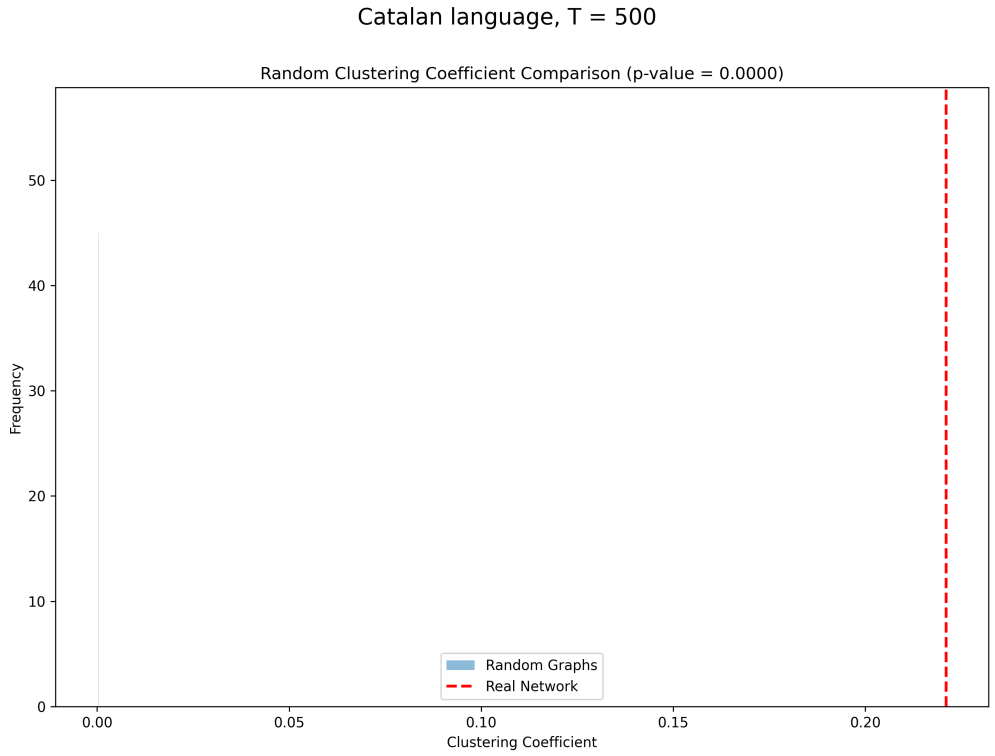

Figure 8: Catalan Language: Erdős-Rényi null hypothesis

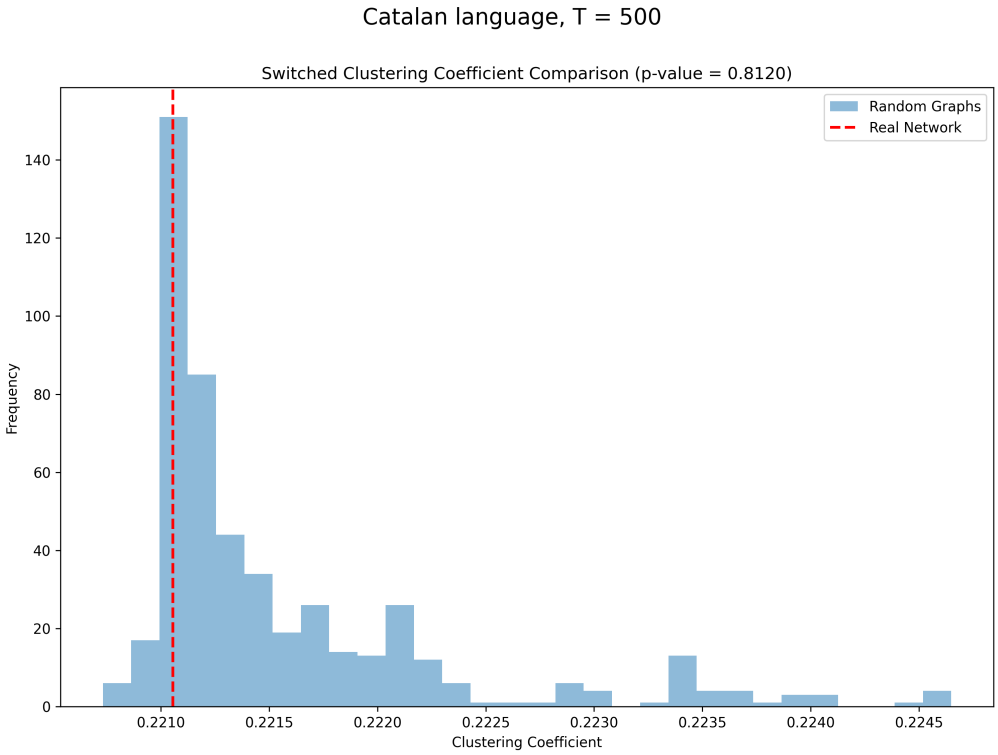

Figure 9: Catalan Language: Switching model null hypothesis

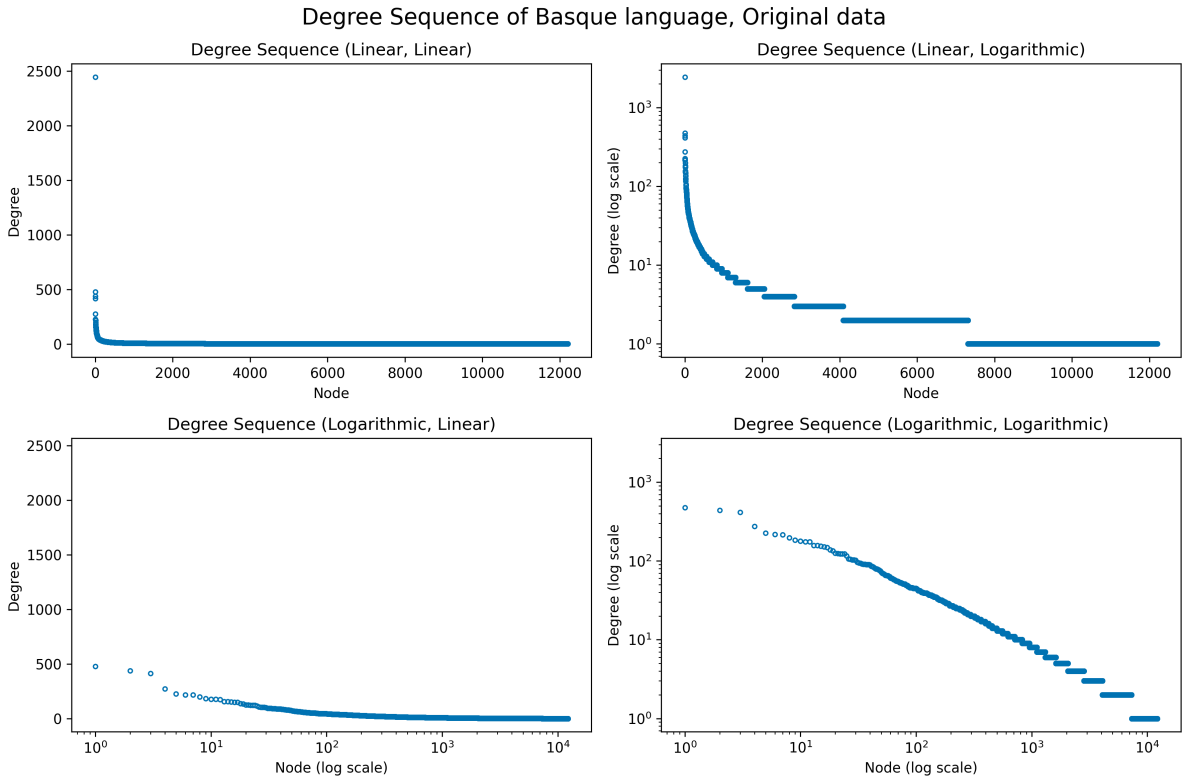

Figure 10: Basque Language: Degree Sequence of Original Data

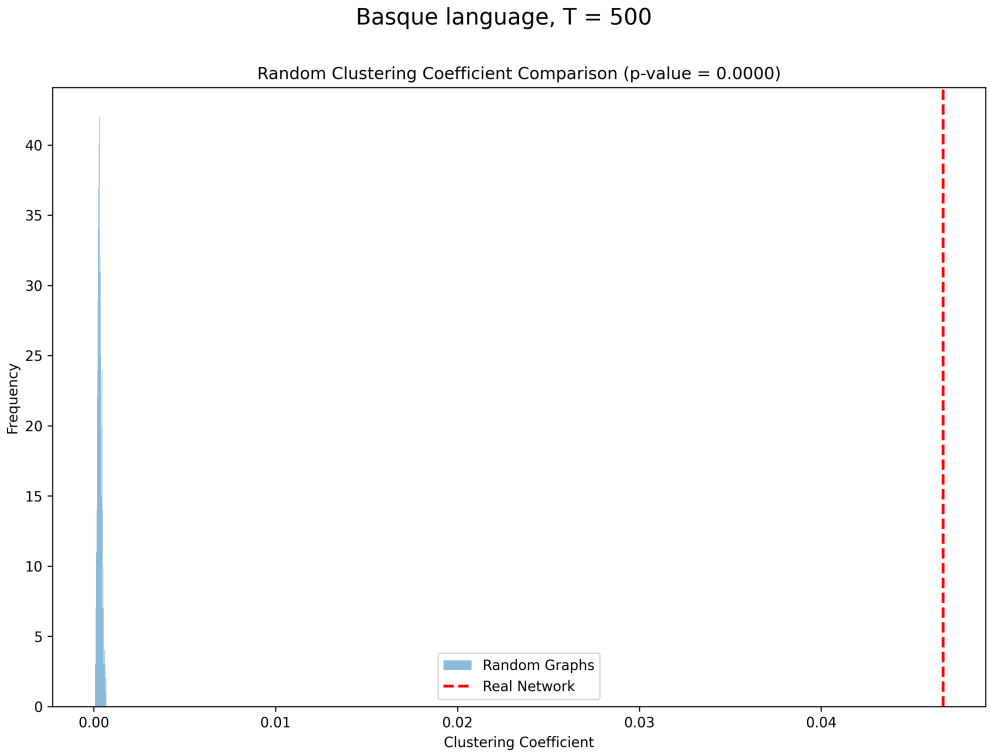

Figure 11: Basque Language: Erdős-Rényi null hypothesis

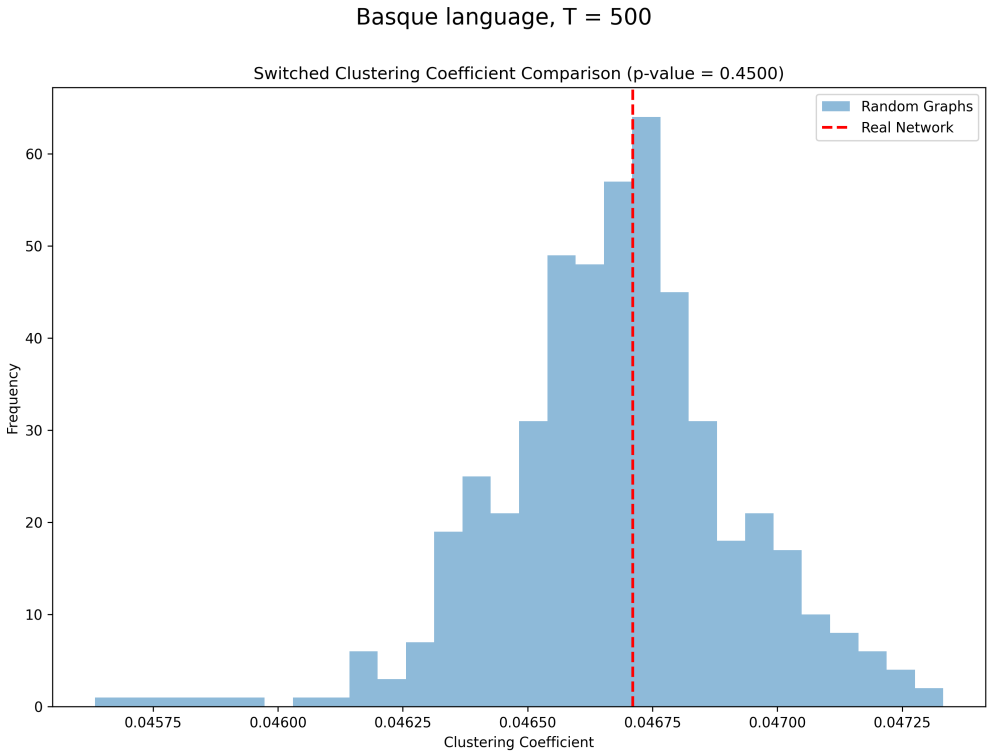

Figure 12: Basque Language: Switching model null hypothesis

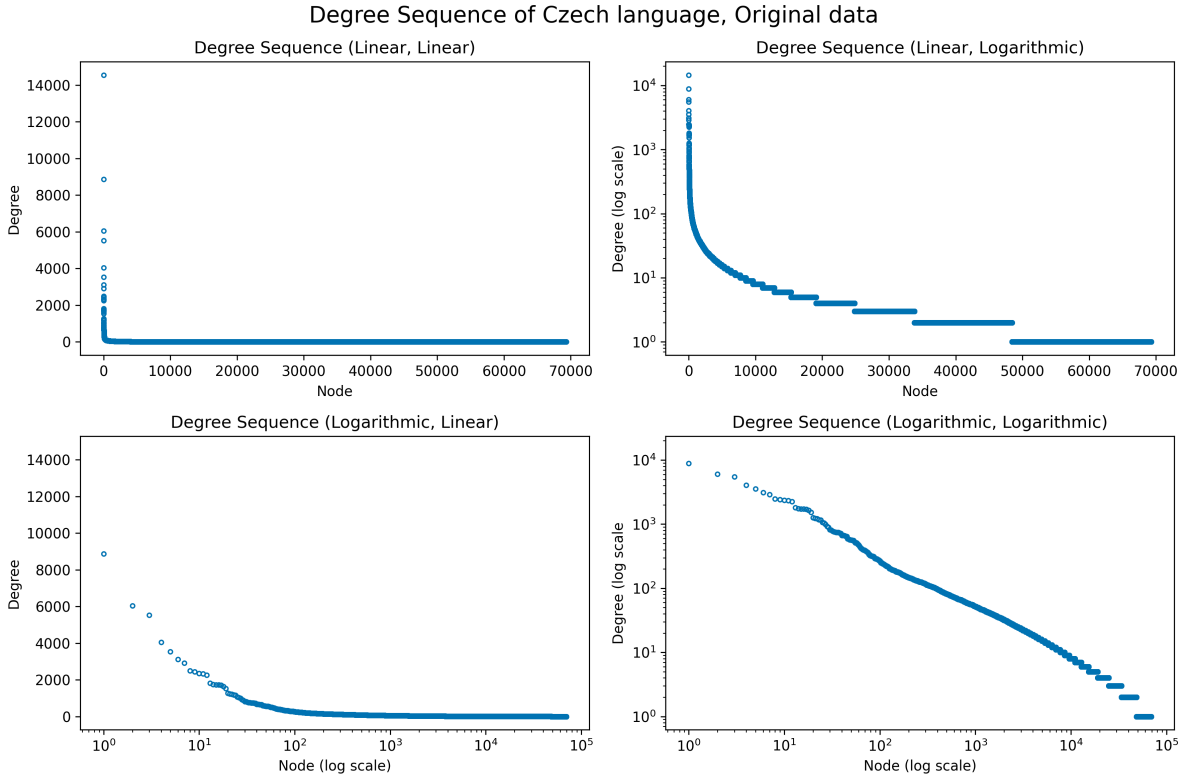

Figure 13: Czech Language: Degree Sequence of Original Data

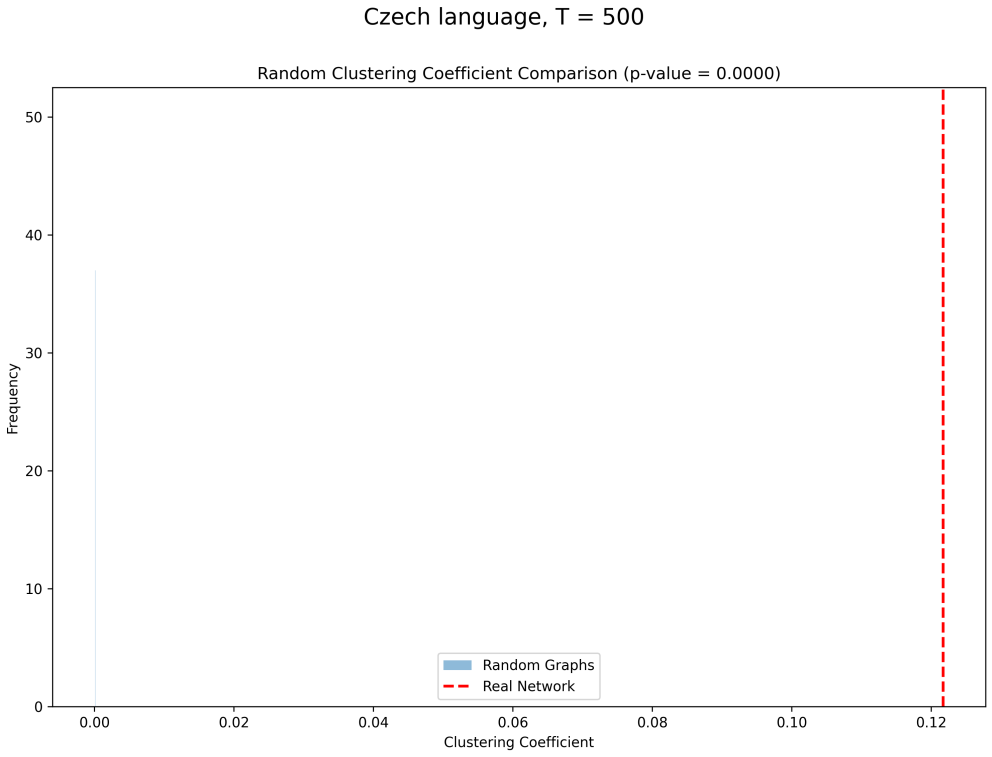

Figure 14: Czech Language: Erdős-Rényi null hypothesis

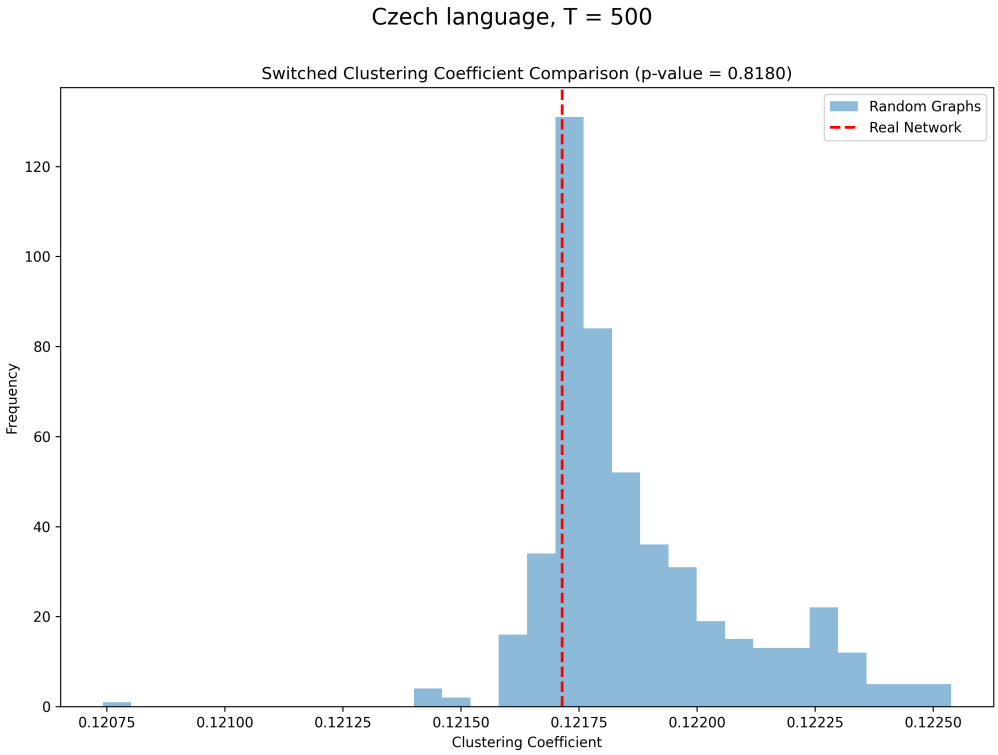

Figure 15: Czech Language: Switching model null hypothesis

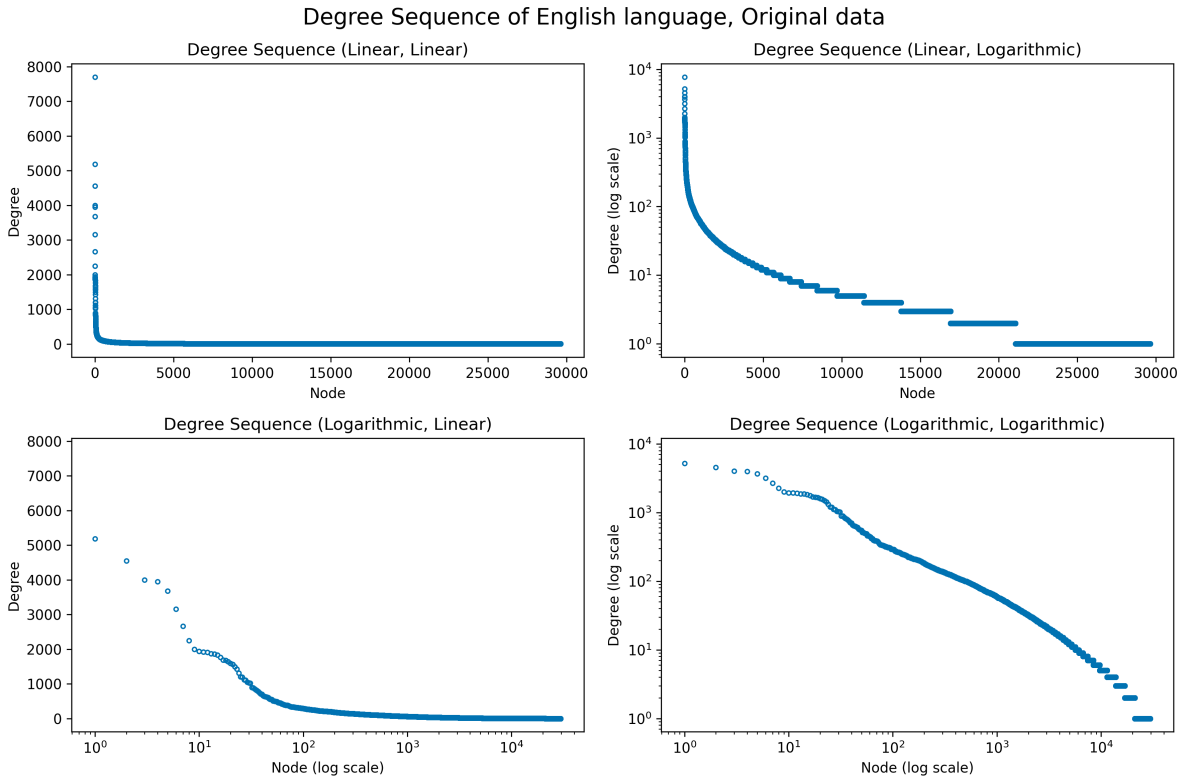

Figure 16: English Language: Degree Sequence of Original Data

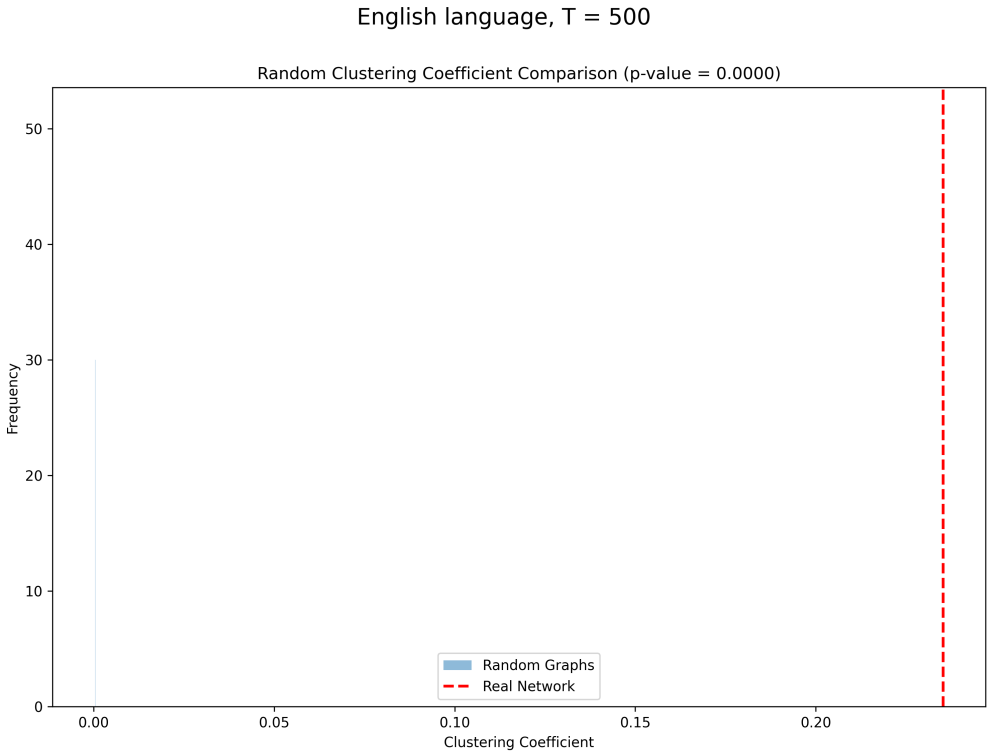

Figure 17: English Language: Erdős-Rényi null hypothesis

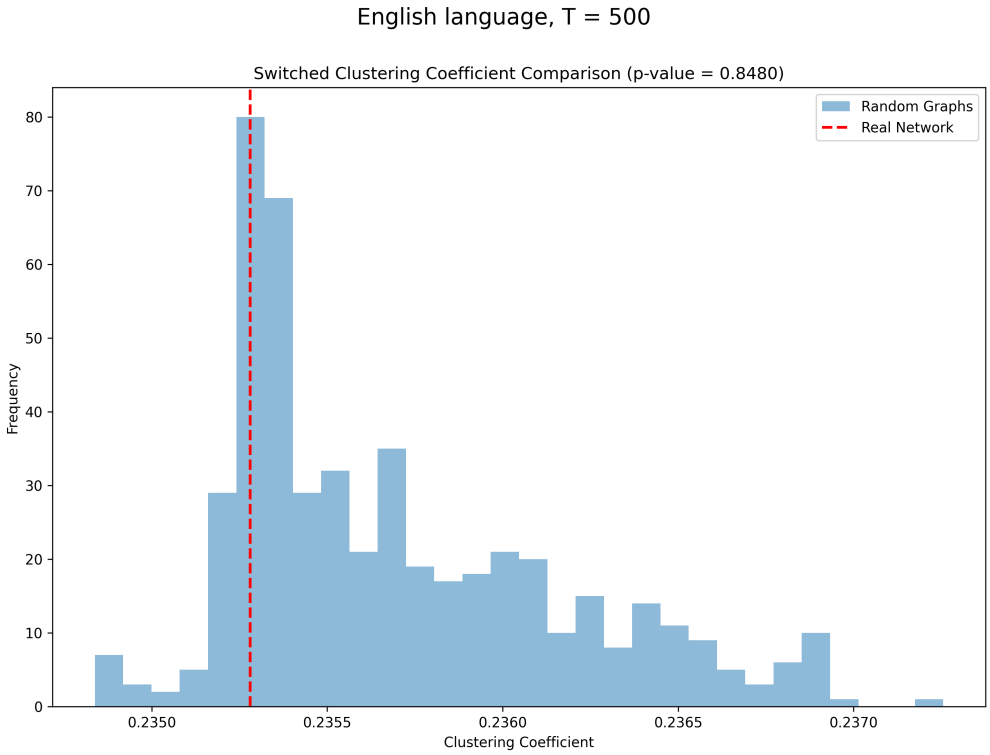

Figure 18: English Language: Switching model null hypothesis

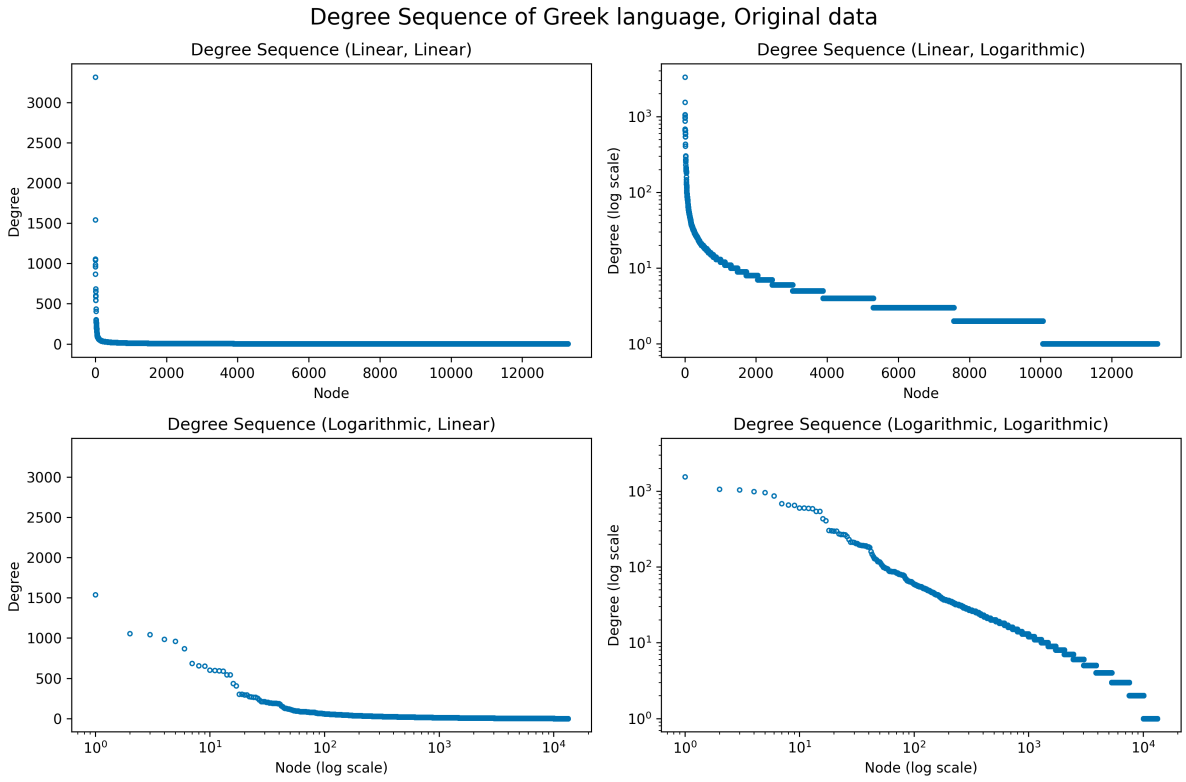

Figure 19: Greek Language: Degree Sequence of Original Data

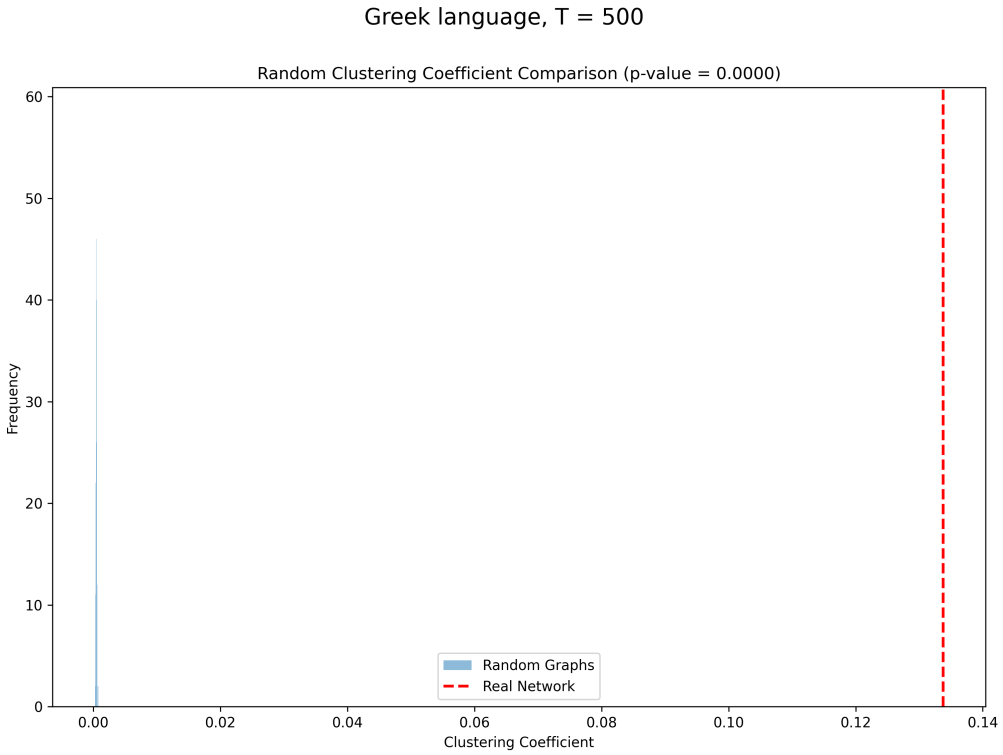

Figure 20: Greek Language: Erdős-Rényi null hypothesis

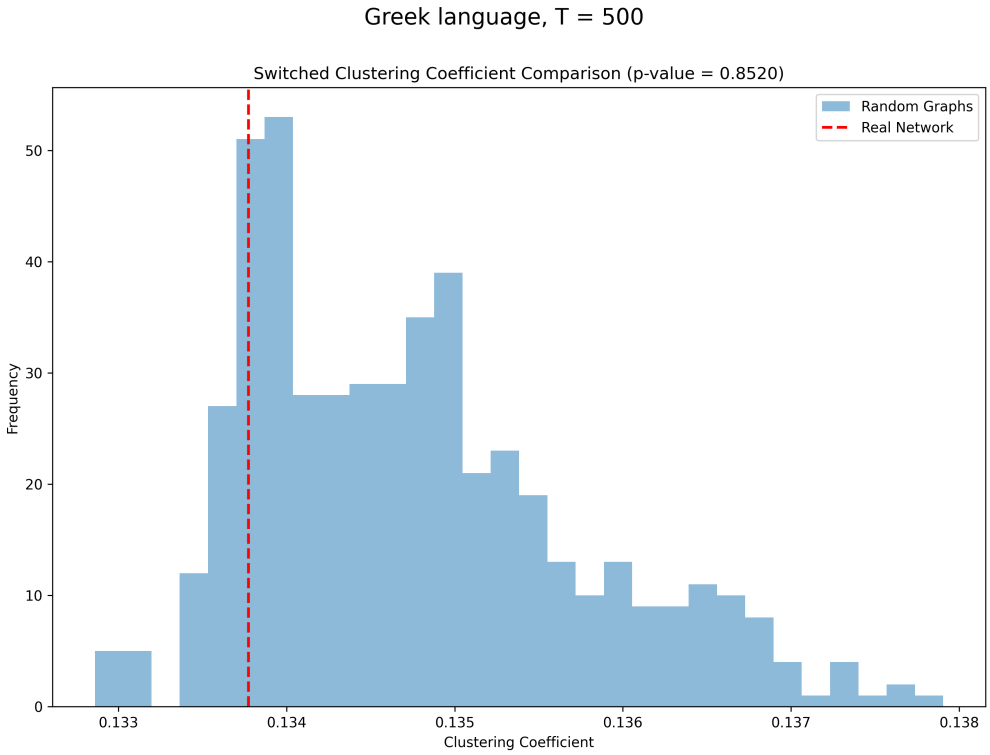

Figure 21: Greek Language: Switching model null hypothesis

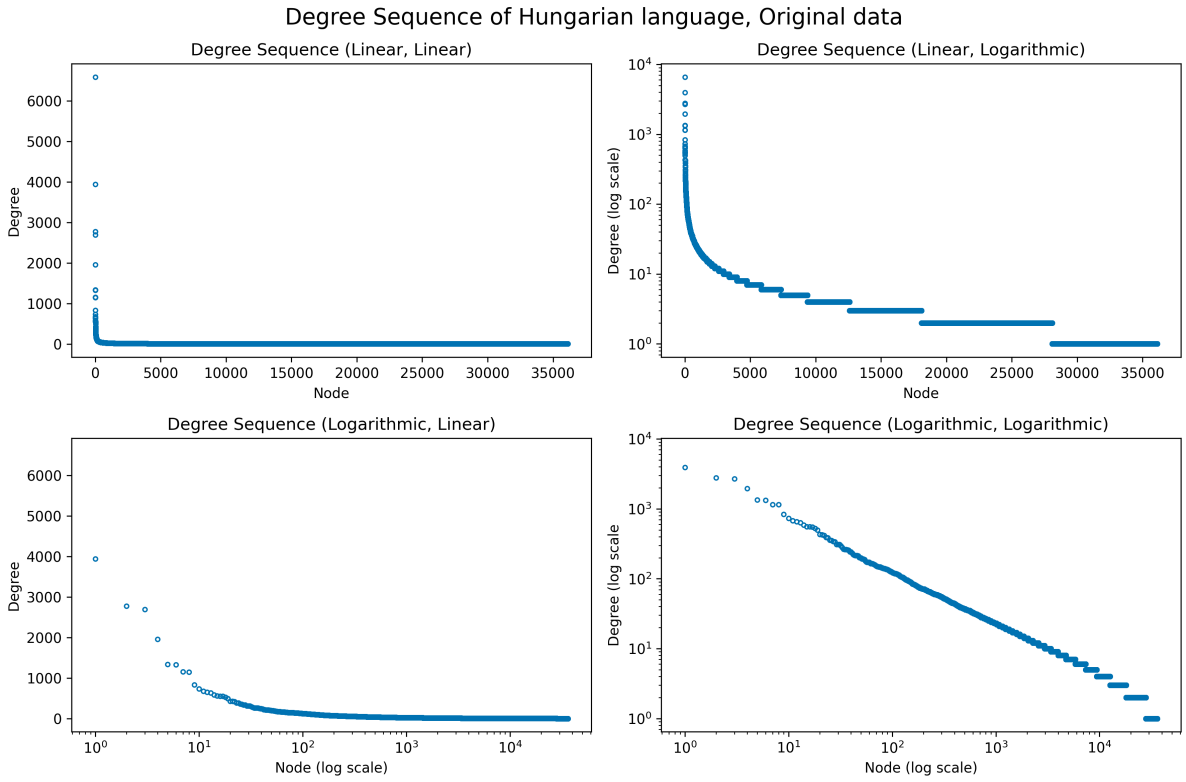

Figure 22: Hungarian Language: Degree Sequence of Original Data

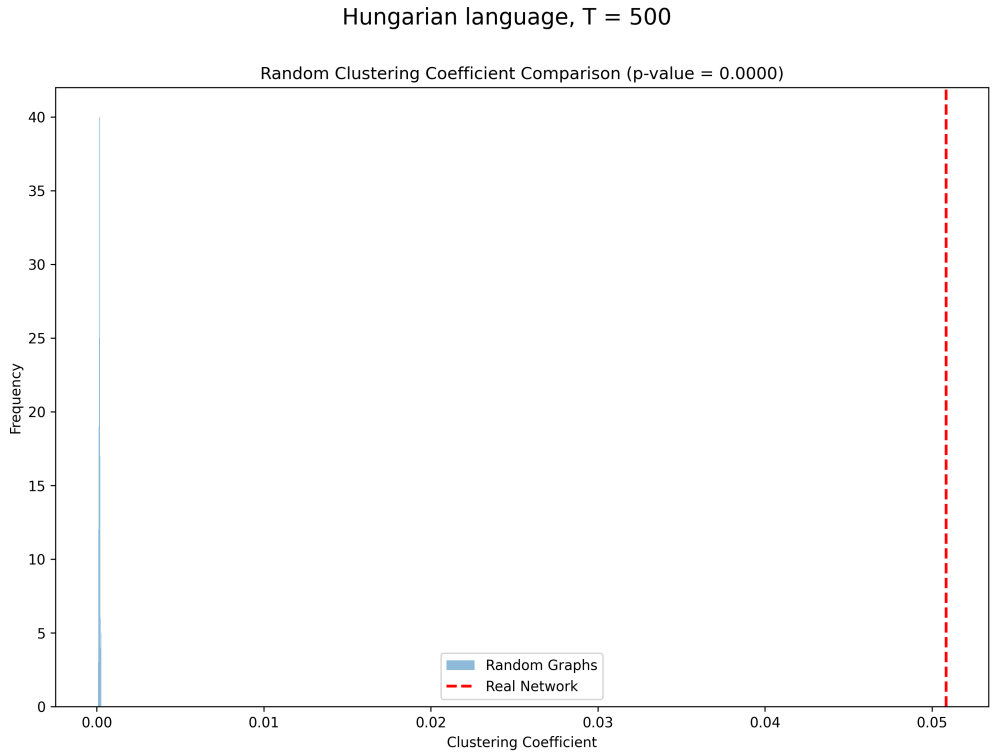

Figure 23: Hungarian Language: Erdős-Rényi null hypothesis

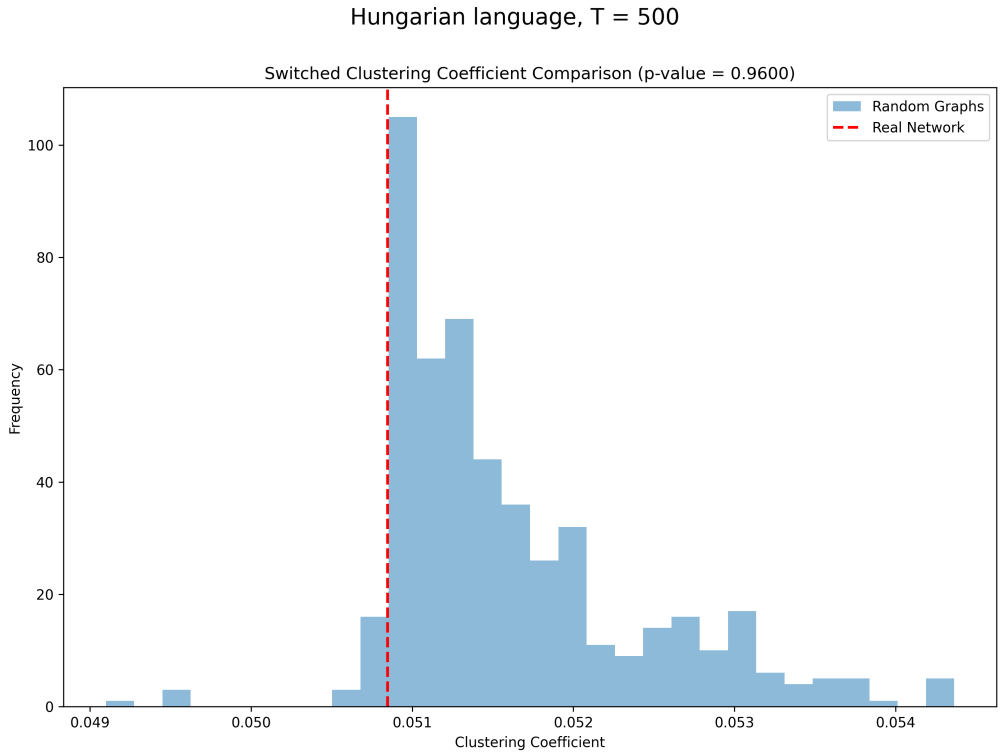

Figure 24: Hungarian Language: Switching model null hypothesis

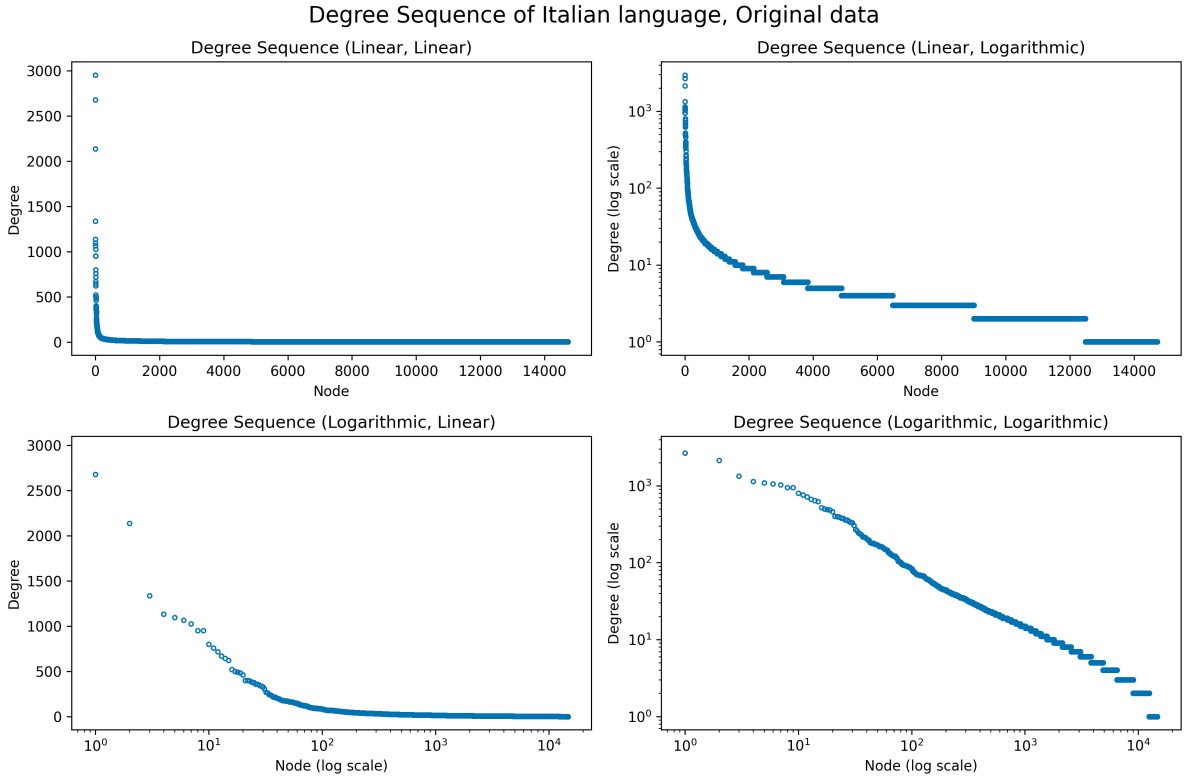

Figure 25: Italian Language: Degree Sequence of Original Data

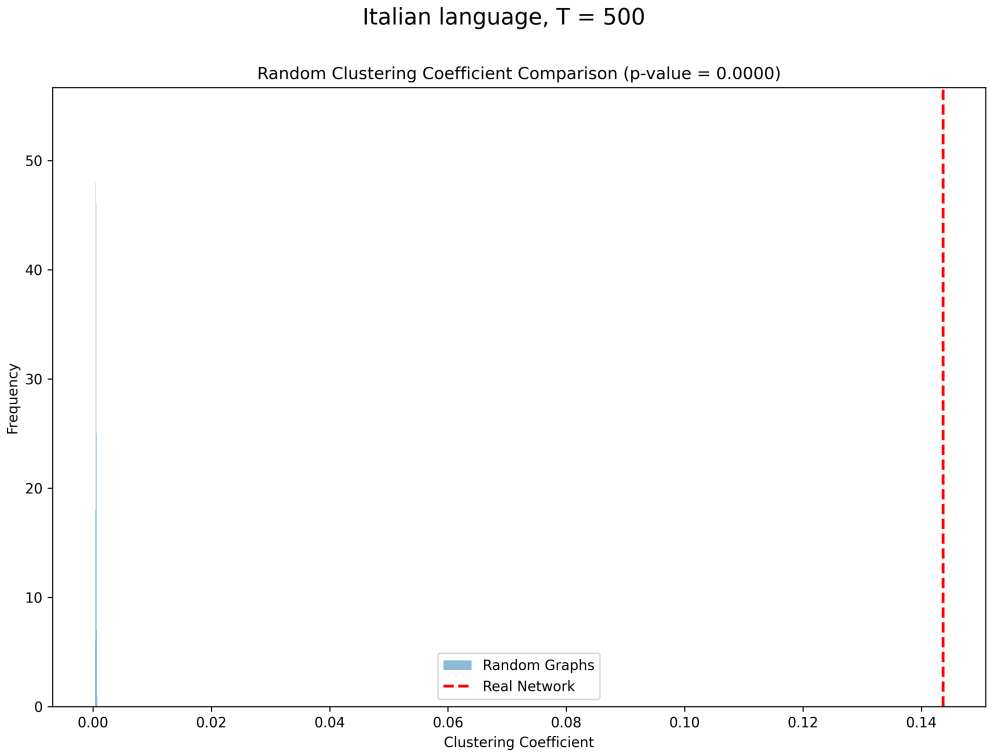

Figure 26: Italian Language: Erdős-Rényi null hypothesis

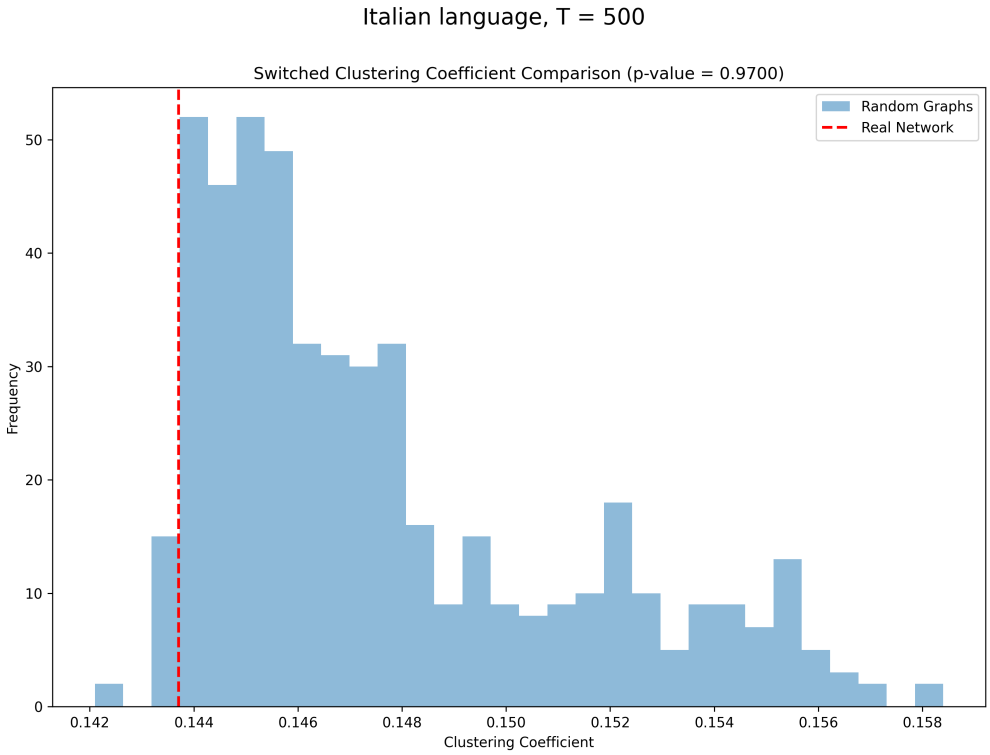

Figure 27: Italian Language: Switching model null hypothesis

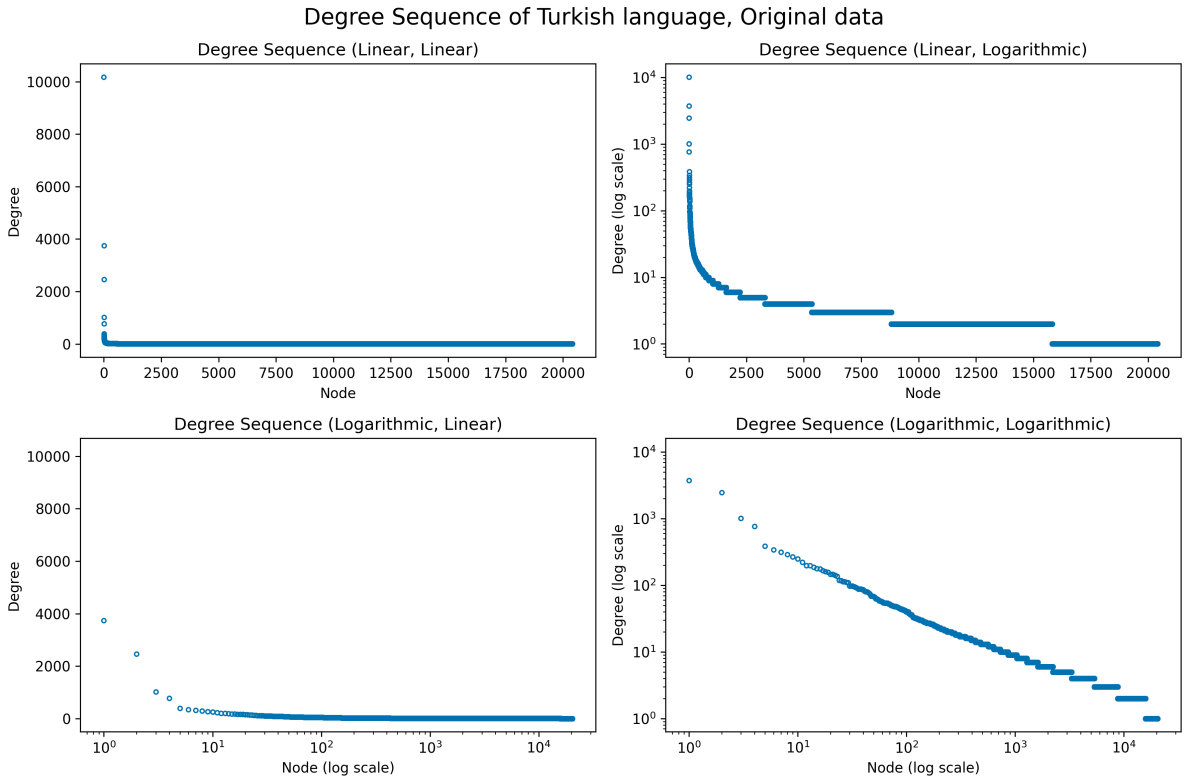

Figure 28: Turkish Language: Degree Sequence of Original Data

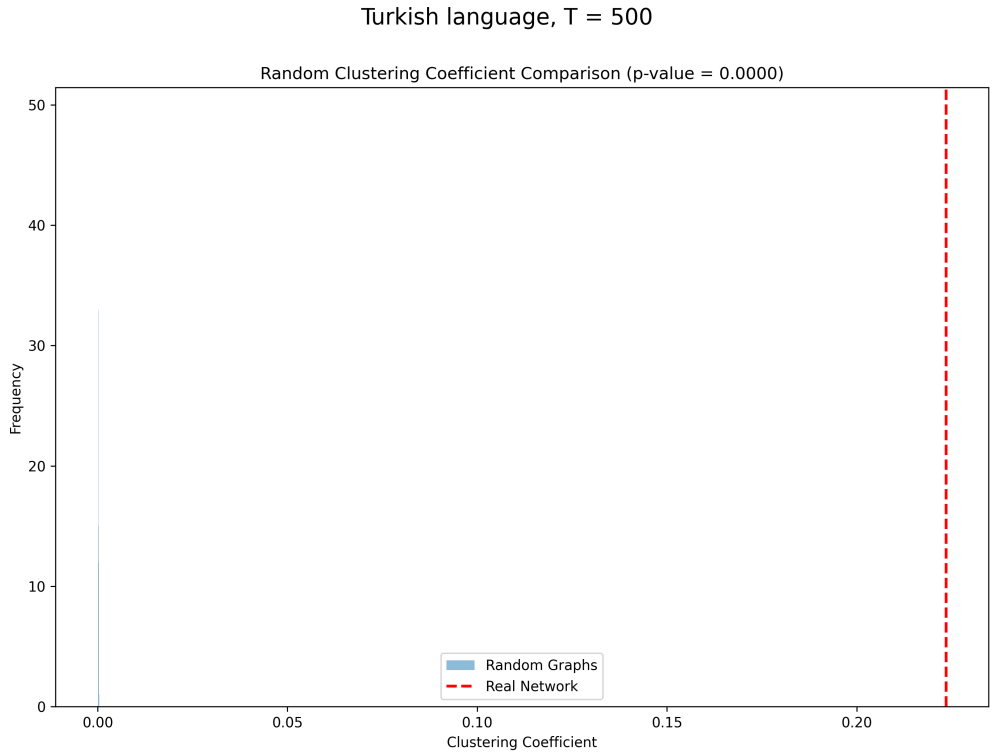

Figure 29: Turkish Language: Erdős-Rényi null hypothesis

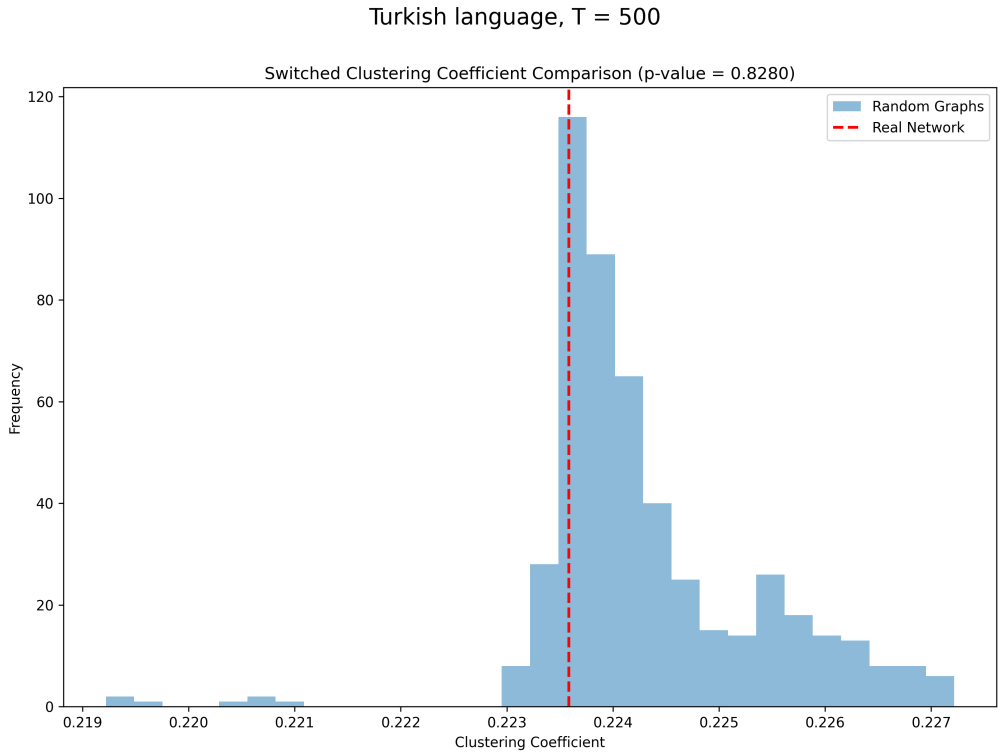

Figure 30: Turkish Language: Switching model null hypothesis