Complex and Social Networks

Laboratory 3

Significance of Metrics

Davide Volpi & Raffaele D'Agostino

Academic Year 2025/2026

1 Introduction

The goal of this laboratory session was to assess the statistical significance of the average local clustering coefficients computed on sets of global syntactic dependency networks across different languages. To this end, we compared the clustering coefficients of each real network with those obtained from two types of randomized models: the Erdős–Rényi model, which preserves the number of nodes and edges of the original network, and the switching model, which generates a randomized version of the original network while maintaining its degree sequence.

2 Results

Table 1 summarizes the basic topological properties of the syntactic dependency networks extracted for each language. For each network, we report the number of nodes (N), the number of edges (E), the mean degree ($\langle k \rangle$), and the network density (δ). These metrics provide a first overview of the structural variability across languages, which will be further analysed in the following sections.

Language	N	E	$\langle k \rangle$	δ
english	4665.0	16908.0	7.248875	0.001554
arabic	4785.0	16251.0	6.792476	0.001420
czech	5318.0	14984.0	5.635201	0.001060
german	5385.0	17316.0	6.431198	0.001195
spanish	4528.0	17396.0	7.683746	0.001697
finnish	4953.0	12733.0	5.141530	0.001038
french	4632.0	18186.0	7.852332	0.001696
galician	4476.0	17438.0	7.791778	0.001741
hindi	4420.0	15319.0	6.931674	0.001569
indonesian	3726.0	14961.0	8.030596	0.002156
icelandic	4837.0	14786.0	6.113707	0.001264
italian	4814.0	18072.0	7.508101	0.001560
japanese	4906.0	19901.0	8.112923	0.001654
korean	6371.0	13679.0	4.294145	0.000674
polish	5033.0	14840.0	5.897079	0.001172
portuguese	5109.0	18128.0	7.096496	0.001389
russian	5155.0	15432.0	5.987197	0.001162
swedish	5006.0	15676.0	6.262885	0.001251
thai	4043.0	15988.0	7.908978	0.001957
turkish	4831.0	13708.0	5.675016	0.001175
chinese	5446.0	16919.0	6.213368	0.001141

Table 1: Language Network Statistics. Where: N is the number of nodes; E is the number of edges; $\langle k \rangle$ is the mean degree; δ is the density of the network.

Table 2 presents the P-values obtained from statistical tests assessing whether the clustering coefficient observed in the real syntactic dependency networks was significantly lower that their Monte Carlo (MC) counterparts. The binomial test evaluates the probability that the MC-generated average local clustering coefficients exceed the real networks ones, while the switching test provides an alternative estimation based on randomized network configurations. P-values were computed over $T = 10000$ iterations, setting the resolution limits of the analysis.

Language	Metric	P-Value (Binomial)	P-value (Switching)
english	Average Local Clustering	$< 10^{-4}$	$> 1 - 10^{-4}$
arabic	Average Local Clustering	$< 10^{-4}$	0.9977 ± 0.0005
czech	Average Local Clustering	$< 10^{-4}$	$> 1 - 10^{-4}$
german	Average Local Clustering	$< 10^{-4}$	0.9921 ± 0.0009

Table 2: P-values for testing the null hypothesis $H_0 : C_{MC} > C_{real}$. Values are estimated over $T = 10000$ Monte Carlo iterations, setting the effective resolution of the test.

Language	Metric	P-Value (Binomial)	P-value (Switching)
spanish	Average Local Clustering	$< 10^{-4}$	0.9999 ± 0.0001
finnish	Average Local Clustering	$< 10^{-4}$	0.9989 ± 0.0003
french	Average Local Clustering	$< 10^{-4}$	0.9943 ± 0.0008
galician	Average Local Clustering	$< 10^{-4}$	0.6903 ± 0.0046
hindi	Average Local Clustering	$< 10^{-4}$	$> 1 - 10^{-4}$
indonesian	Average Local Clustering	$< 10^{-4}$	0.0139 ± 0.0012
icelandic	Average Local Clustering	$< 10^{-4}$	$> 1 - 10^{-4}$
italian	Average Local Clustering	$< 10^{-4}$	$> 1 - 10^{-4}$
japanese	Average Local Clustering	$< 10^{-4}$	$> 1 - 10^{-4}$
korean	Average Local Clustering	$< 10^{-4}$	0.0002 ± 0.0001
polish	Average Local Clustering	$< 10^{-4}$	$> 1 - 10^{-4}$
portuguese	Average Local Clustering	$< 10^{-4}$	0.1988 ± 0.0040
russian	Average Local Clustering	$< 10^{-4}$	0.9998 ± 0.0001
swedish	Average Local Clustering	$< 10^{-4}$	$> 1 - 10^{-4}$
thai	Average Local Clustering	$< 10^{-4}$	$< 10^{-4}$
turkish	Average Local Clustering	$< 10^{-4}$	0.3547 ± 0.0048
chinese	Average Local Clustering	$< 10^{-4}$	0.8856 ± 0.0032

Table 2: P-values for testing the null hypothesis $H_0 : C_{MC} > C_{real}$. Values are estimated over $T = 10000$ Monte Carlo iterations, setting the effective resolution of the test.

3 Discussion

3.1 Relationship Between Density and Expected Local Clustering in ER Graphs

An interesting observation arises when comparing the local clustering coefficients of our graphs to the expected values in Erdős–Rényi (ER) random graphs. As discussed in Section 4.7, for an ER graph with N nodes and edge probability p, the expected local clustering coefficient of any node is approximately equal to the graph density δ, up to minor fluctuations due to finite-size effects. While Newman [1] provides a derivation for the global clustering coefficient, the same reasoning applies at the local level, with the expected value of C_i essentially given by the probability that two neighbours of a node are connected, which is equal to p.

In our empirical analysis, we computed the local clustering coefficients for each language network and compared them to the corresponding network density. As shown in Figure 1, the observed distributions of C_i are tightly concentrated around the density: in all cases, the density lies within one standard deviation from the mean of the distribution.

3.2 Clustering hypothesis testing: Real Networks vs. Erdős–Rényi Models

The results of the hypothesis test in Table 2 indicate that for all languages, the null hypothesis $H_0 : C_{ER} > C_{real}$ is rejected, with P-values consistently below the significance level of $\alpha = 0.05$. This outcome aligns with expectations, as the ER model is a random graph model characterized by a uniform probability of edge formation between any two nodes. Consequently, ER graphs lack the local clustering and community structures inherent in real-world networks.

It is important to note that the P-values were estimated using a Monte Carlo simulation with $T = 10000$ iterations. This approach introduces a lower bound on the P-value resolution, approximately 10^{-4}, meaning that P-values smaller than this threshold are indistinguishable within the current simulation setup. While this limitation constrains the precision of our statistical inference, it does not undermine the overall conclusion that ER graphs do not exhibit the clustering properties observed in real linguistic networks.

In summary, the rejection of the null hypothesis across all languages underscores the inadequacy of the ER model in capturing the structural complexities of real-world linguistic networks. This finding highlights the necessity for more sophisticated models that account for the inherent clustering and community structures present in such networks.

3.3 Clustering hypothesis testing: Real Networks vs. Switching Models

For the switching model, the analysis is more nuanced. As in the previous section, we use a significance level of $\alpha = 0.05$, corresponding to a 95% confidence level, and perform the same Monte Carlo-based P-value test

to compare the clustering coefficient of the randomized networks (with preserved degree sequence) to the real ones. The procedure for generating randomized networks using the switching model is described in Section 4.3.

Unlike the Erdős–Rényi case, the null hypothesis $H_0 : C_{SW} > C_{real}$ is rejected only for few languages, the ones where the P-value is below 0.05. Specifically, significant rejection is observed for only the following languages: Indonesian, Korean and Thai.

The switching model provides an alternative way to randomize networks while preserving the degree of each node, i.e., the full degree sequence. Unlike the Erdős–Rényi model, we cannot assert that graphs generated by the switching model always exhibit the small-world property, as the null hypothesis $H_0 : C_{SW} > C_{real}$ is rejected only for a subset of languages.

In certain cases, despite the randomization, the switching model appears capable of maintaining some local structural properties of the original networks. More precisely, it is helpful to consider the situation from a different perspective: the switching model preserves the exact degree sequence of each real syntactic network, which is exactly the primary structure it always (globally) maintains. Our real networks represent just one of the many possible realizations of a network with that specific degree sequence.

To visualize this, we provide in Figure 2 histograms of the distributions of the T clustering coefficients obtained from the Monte Carlo simulations for each language. These plots allow a clear comparison between the simulated and real clustering values.

It is evident that some of the real clustering coefficients fall far from the mean of the corresponding distributions, sometimes appearing as clear outliers, either substantially higher or lower.

This observation provides an interesting insight. For languages where the real clustering coefficient lies above the 97.5th percentile of the simulated distribution, the observed clustering appears unusually high compared to what would be expected from the degree sequence alone. Conversely, when the real clustering coefficient falls below the 2.5th percentile, the network exhibits a more tree-like structure than expected from the degree sequence. In both cases, this suggests that additional structural factors may be influencing the organization of syntactic dependencies beyond what is captured by the degree sequence.

For other languages, the real clustering coefficient falls within the 2.5th–97.5th percentile range of the simulated distribution. In these cases, the observed clustering is largely compatible with the expectations from the degree sequence alone. This indicates that, for these languages, the local network structure could be largely explained by the degree sequence, without necessarily invoking additional organizing principles. However, this does not rule out the presence of other structural patterns; it only shows that the clustering coefficient is not atypical relative to the ensemble generated by the switching model.

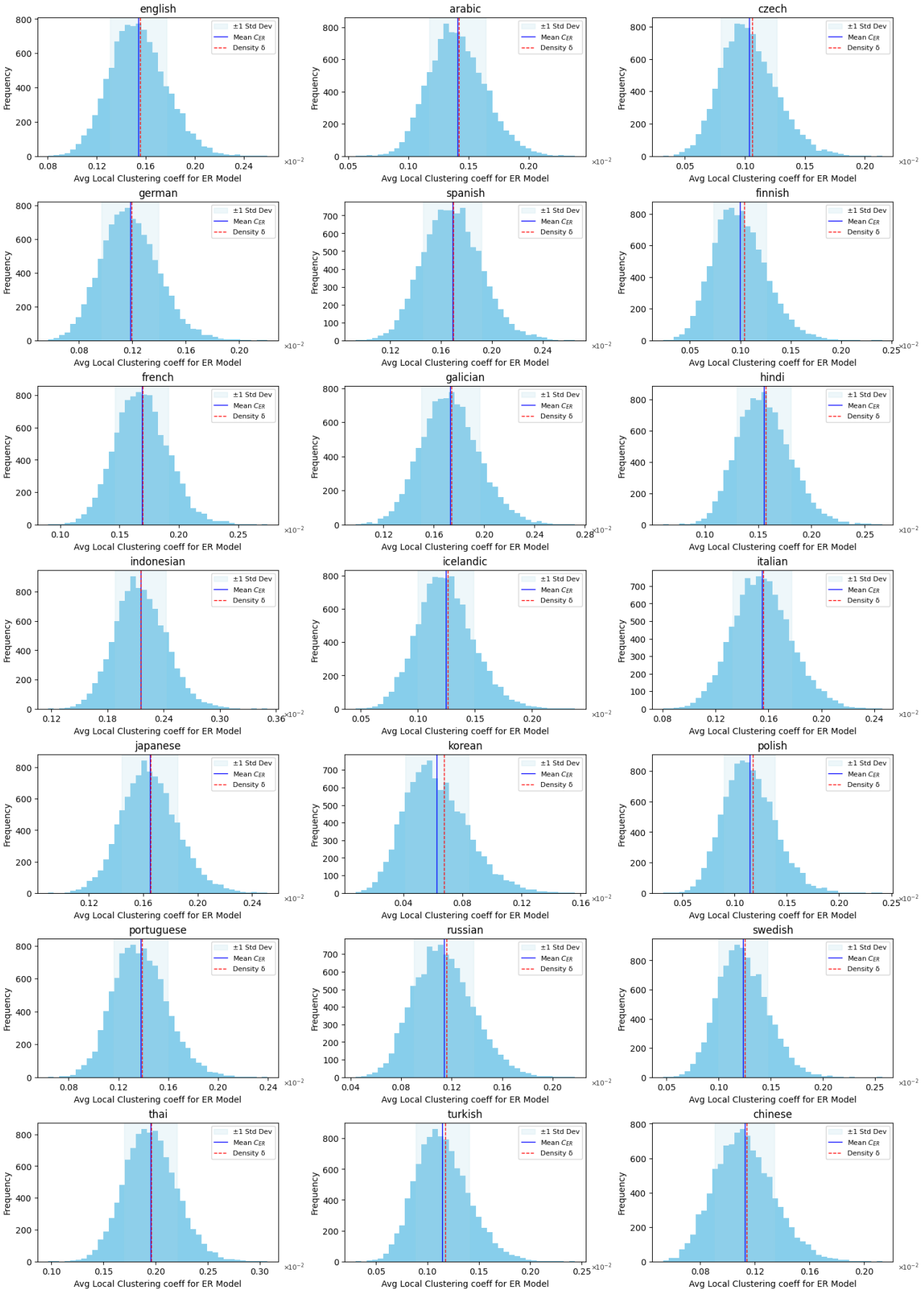

Figure 1: Histograms of the local clustering coefficients in Erdős–Rényi networks for each language. The red dashed line indicates the network density δ, which consistently lies within less than one standard deviation of the empirical distribution of clustering values, illustrating that the expected clustering closely matches the network density.

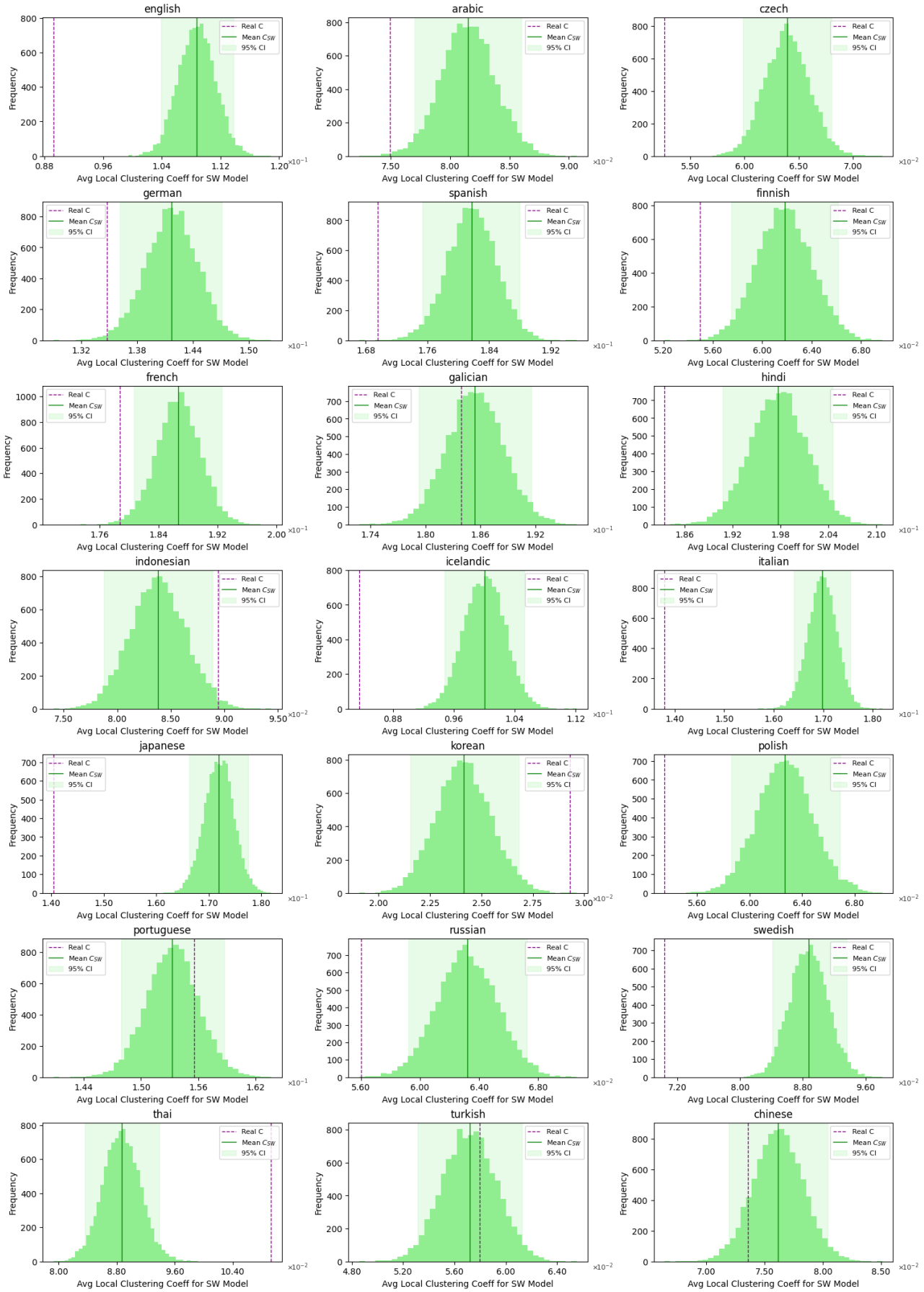

Figure 2: Histograms of the clustering coefficients obtained from $T = 10000$ Monte Carlo simulations of the switching model for each language. The purple dashed line indicates the real clustering coefficient of the network, the green solid line shows the mean of the simulated distribution, and the shaded area represents the empirical 95% confidence interval (2.5^{th}–97.5^{th} percentiles) of the simulated values.

4 Methods

In our experiments, we used the following python packages:

- `Numpy` for fast operations over arrays and random elements handling.
- `Pandas` for the management of the data-frames.
- `Os` for the functioning of the `.py` files
- `NetworkX` for the graphs implementation.
- `Multiprocessing` and `Tqdm` for parallelization and visualization of the operations on different available CPU cores.
- `Random` for a random extraction of an element from a list.
- `Math` for the $\epsilon_{Hoeffding}$ calculation.

4.1 Data Preparation

To generate the undirected graphs we proceeded as follow:

1. Retrieved the conllu files available from the past laboratory experiments (Laboratory 2).
2. Built the global syntactic dependency network for each PUD tree-bank thanks to the conllu files in each folder, that permit to extract the syntactic dependencies of each sentence for each language (each file is contained in the folder `data \PUD`). To do this we created a `parse_ud_conllu` function that, given the structure of the conllu file, permits to extract all the useful information about the dependencies such as the parent-child relation and all the sentences words.
3. Thanks to the python function previously mentioned, we extracted each word as a node and each father-child relation as an undirected edge and then constructed, thanks to the python library `NetworkX`, the entire undirected graph G for each language.
4. For each language, we set up a script that, given an undirected graph, extracted the three following files for each language:
   - `lang_degree_sequence.txt` with inside the degree sequence of the language `lang`.
   - `lang_dependency_network_edges.txt` with an header line that contained the number of vertices and the number of edges of the network, then each line showed an edge inside the graph of language `lang`.
   - `lang_dependency_network_adj.txt` with an header line that contained the number of vertices and the number of edges of the network, then each line showed a node with next to the adjacency list of that node in the graph of language `lang`.

The extracted networks could contain self loops, so we removed them before performing any analysis of the network properties.

In addition to these elements, we also extracted a data-frame containing , for each language network, the respective average local clustering coefficient.

4.1.1 Node Representation: Lexical Forms

In our dependency networks, edges connected heads to dependents following each sentence's syntactic structure. One important design choice was how to represent nodes: using surface forms (words as they appear) or lemmatized forms (base word forms).

We mainly used lemmatized forms, where available, since they group together inflected variants (e.g., run, runs, running), reducing data sparsity and giving a clearer picture of the network's structure. Lemmatization also helps capture semantic similarity and makes comparisons between languages more consistent, especially for highly inflected ones.

4.2 Erdős–Rényi Model

We created a function that, given a set of nodes and edges from a language:

1. Extracted the number of nodes and edges, respectively N and E.
2. Created a set of nodes composed by numbers from *1* to N and an empty set of edges.
3. While the number of edges was less than the original input number of edges:
   (a) We randomly sampled two node indices.
   (b) If they were different (avoiding self-loops), we created and added an undirected edge between them into the edges set.
4. Returned the set of nodes and edges generated.

The result of this process was a randomized binomial graph from the ER family without self loops or double edges.

4.3 Switching Model

We created a function that, given a set of edges from a language:

1. Attempted to perform edge rewiring $Q \times |E|$ times, where $Q = \lfloor \log(|E|) \rceil$.
2. For each attempt:
   (a) Randomly selected two distinct edges (u, v) and (x, y) from the edge set.
   (b) Checked if any node appears in both edges (i.e., $u \in \{x, y\}$ or $v \in \{x, y\}$). If so, skipped this iteration.
   (c) With probability 0.5, created new edges (u, x) and (v, y); otherwise, created new edges (u, y) and (v, x) (to avoid biased choices).
   (d) Checked if either of the new edges already exists in the graph (avoiding double edges). If so, skipped this iteration.
   (e) If all conditions were satisfied, removed edges (u, v) and (x, y), added the new edges, and updated the edge list accordingly.
   (f) Incremented the effective switches counter only when a switch was successfully performed.
3. Returned the rewired edge set and the fraction of successful switches (effective switches / total attempts).

This process preserves the degree distribution of the original graph while randomizing its structure.

4.3.1 Considerations

In order to implement calculations for the switching model successfully we had to answer the following questions: *"Given two edges* $(u \sim v)$ *and* $(s \sim t)$*, what are the switches that"*

1. Preserve the degree sequence?
   A switching replaces two edges (u, v) and (s, t) with either $(u, s), (v, t)$ or $(u, t), (v, s)$. Each involved node loses one connection and gains one, thus the degree of every node remains unchanged.
2. Preserve the degree sequence but produce edges that are not allowed (self-loops, multi-edges)?
   Even though degree preservation is guaranteed by construction, not all switches yield valid graphs in the sense of being *simple* (i.e., without self-loops or multiple edges). A switching is invalid if it creates a self-loop (e.g., $u = s$ leading to (u, u)) or if one of the new edges already exists in the edge set E, generating a multi-edge.

4.4 Exact Computation of Clustering Coefficients

Since we had sufficient computational resources to reliably estimate the clustering coefficients in both the switching model and the Erdős–Rényi model over a substantial number of iterations, we chose to avoid using approximations. While such approximations could have been useful for hypothesis testing, they would have limited our ability to perform further observations and analyses on the results. In the next paragraph, we also discuss that we were not able to confirm that random sampling-based approximations for these types of graphs are sufficiently precise.

4.5 Approximation Error of Clustering Coefficient Using Node Sampling

To speed up computations, it may be useful to test the same hypothesis on a smaller subset of nodes of the graphs, at the expense of introducing some approximation error. We decided to estimate a bound on this error using Hoeffding's inequalities, considering the random variable C_i, the local clustering coefficient of node i. This led to the following derivation.

4.5.1 Hoeffding Bound for Local Clustering Coefficient

Let $C_1, C_2, \ldots, C_N$ be the local clustering coefficients of the N nodes in a graph, with $0 \leq C_i \leq 1$ for all i. Let $S \subset \{1, \ldots, N\}$ be a random sample of size M, and let $\hat{C} = \frac{1}{M} \sum_{i \in S} C_i$ denote the sample mean. The goal is to bound the deviation of $\hat{C}$ from the true mean $\bar{C} = \frac{1}{N} \sum_{i=1}^{N} C_i$.

By Hoeffding's inequality, for independent bounded random variables $X_1, \ldots, X_m$ with $X_i \in [a_i, b_i]$, we have

$$\Pr\left(\left| \frac{1}{M} \sum_{i=1}^{M} X_i - \mathbb{E}[X_i] \right| \geq \epsilon \right) \leq 2 \exp\left(- \frac{2M^2 \epsilon^2}{\sum_{i=1}^{M} (b_i - a_i)^2} \right).$$

In our case, $X_i = C_i$ with $a_i = 0$ and $b_i = 1$, so the bound simplifies to

$$\Pr\left(|\hat{C} - \bar{C}| \geq \epsilon \right) \leq 2 \exp\left(-2M\epsilon^2 \right).$$

Solving for ϵ for a confidence level $1 - \alpha$ gives the $(1 - \alpha)$-confidence bound:

$$\epsilon \leq \sqrt{\frac{\ln(\alpha/2)}{-2M}}.$$

By setting $\alpha = 0.05$ and considering a fraction of 20% of the nodes, we can estimate the bound on the approximation error for the clustering coefficient of each language, as reported in Table 3. As can be seen, these bounds are too loose for our purpose: the values obtained might be too large to consider a 20% sample as a reliable approximation of the full clustering coefficient.

Language	$\varepsilon_{\text{Hoeffding}}$
english	0.0445
arabic	0.0439
czech	0.0416
german	0.0414
spanish	0.0451
finnish	0.0432
french	0.0446
galician	0.0454
hindi	0.0457
indonesian	0.0498
icelandic	0.0437
italian	0.0438
japanese	0.0434
korean	0.0380
polish	0.0428
portuguese	0.0425
russian	0.0423
swedish	0.0429
thai	0.0478
turkish	0.0437
chinese	0.0412

Table 3: Hoeffding bounds on the sampling error for the clustering coefficient of each language. Significance level $\alpha = 0.05$.

An interesting aspect arises from the fact that we initially set the clustering coefficient to lie between 0 and 1. In practice, this upper bound is overly pessimistic: the maximum possible value of C_i is much smaller than

1, since a node would need to be connected to all other nodes in its neighbourhood to reach it. To refine the bound, we considered structural constraints such as the total number of edges, the number of nodes, and the degree sequence of the network.

An intuitive and simple upper bound on the average clustering coefficient can be obtained by considering only the degree sequence of the network. Nodes with degree less than 2 cannot form any triangles, so their local clustering is necessarily $C_i = 0$. For all other nodes, one can imagine an idealized scenario in which every possible wedge among neighbours is closed, giving $C_i = 1$.

Under this approximation, the maximum average clustering is simply bounded by the fraction of nodes with degree at least 2:

$$C_{\max} \lesssim \frac{\#\{i \,|\, k_i \geq 2\}}{n}.$$

This simple reasoning highlights that low-degree nodes inherently limit the achievable clustering, while higher-degree nodes could, in principle, reach the maximum local clustering. This bound provides a rough estimate, it could be made tighter by considering additional structural constraints, such as the total number of edges or the arrangement of high-degree nodes. In this way, the degree distribution alone already sets natural limits on the network's clustering.

However, even using this straightforward bound, the approximation errors estimated via Hoeffding's inequality remain quite large, as shown in Table 4. This indicates that, for the networks we analysed, sampling a fraction of nodes—even 20%—does not guarantee a sufficiently precise estimate of the clustering coefficients.

More refined bounds on $C_{\max}$ may exist, taking into account the full degree sequence, edge distribution, and other structural details. Employing such tighter bounds would improve the confidence in the accuracy of approximations, reducing estimation errors even when using only a subset of nodes, and enable more reliable analysis with limited computational effort.

Language	$C_{\max,\ \text{tight}}$	$\varepsilon_{Hoeffding,\ tight}$
english	0.7820	0.0348
arabic	0.7720	0.0339
czech	0.7431	0.0309
german	0.8214	0.0340
spanish	0.8242	0.0372
finnish	0.6860	0.0296
french	0.8137	0.0363
galician	0.8246	0.0374
hindi	0.7597	0.0347
indonesian	0.7421	0.0369
icelandic	0.7703	0.0336
italian	0.8091	0.0354
japanese	0.7770	0.0337
korean	0.7325	0.0279
polish	0.7534	0.0323
portuguese	0.8285	0.0352
russian	0.7391	0.0313
swedish	0.7777	0.0334
thai	0.7341	0.0351
turkish	0.7295	0.0319
chinese	0.7431	0.0306

Table 4: Fraction of nodes with degree ≥ 2 and the tighter Hoeffding error for the new bounds for each language. Significance level $\alpha = 0.05$.

4.6 Monte Carlo Simulation

We implemented a Monte Carlo simulation framework to statistically compare the clustering coefficient of real dependency networks against two null models. For each language:

1. Performed $T = 10000$ independent simulation iterations.
2. For each iteration:

(a) Retrieved the real clustering coefficient C_{real} from the pre-computed dataset.

(b) Generated an Erdős-Rényi (ER) random graph with the same number of nodes and edges as the real network, preserving only the average degree.

(c) Applied the Switching Model (SW) to the real edge set, preserving the exact degree sequence while randomizing the network structure.

(d) Computed the clustering coefficients C_{ER} and C_{SW} for both null models.

(e) Incremented counters when $C_{\text{ER}} > C_{\text{real}}$ or $C_{\text{SW}} > C_{\text{real}}$.

(f) Recorded the fraction of effective switches achieved in the SW rewiring process, to ensure a sufficient number of switches.

3. Calculated P-values as the fraction of iterations where each null model's clustering exceeded the real value:

$$p = \frac{\text{counter}}{T} \pm \sqrt{\frac{p(1-p)}{T}} \tag{1}$$

The uncertainty was added only where the $P - value$ was not $> 1 - 10^{-4}$ or $< 10^{-4}$.

4. Computed the mean and standard error of effective switches across all SM iterations:

$$\mu_{\text{eff}} = \frac{1}{T}\sum_{i=1}^{T} s_i \quad \text{SE} = \frac{\sigma_{\text{eff}}}{\sqrt{T}} \tag{2}$$

5. Parallelized the process across all languages using multiprocessing to leverage multiple CPU cores.

This approach provides robust statistical evidence for whether the observed clustering in real dependency networks significantly differs from what would be expected under random null models.

4.6.1 Test of Significance for Monte Carlo simulations

We wanted to determine if the average clustering coefficient of each real language network is significantly larger than what would be expected from random null models. We formulated the following null hypotheses:

$$H_0^{ER} : C_{real} \leq C_{ER} \tag{3}$$

$$H_0^{SW} : C_{real} \leq C_{SW} \tag{4}$$

where C_{real} is the clustering coefficient of the real dependency network, C_{ER} is the clustering coefficient of the Erdős-Rényi random graph, and C_{SW} is the clustering coefficient of the Switching Model graph.

The alternative hypotheses are:

$$H_1^{ER} : C_{real} > C_{ER} \tag{5}$$

$$H_1^{SW} : C_{real} > C_{SW} \tag{6}$$

We computed P-values as the fraction of Monte Carlo iterations where the null model's clustering coefficient exceeded the real network's value:

$$p = \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}(C_{null}^{(i)} > C_{real}) \tag{7}$$

where $T = 10000$ is the number of simulations, $C_{null}^{(i)}$ is the clustering coefficient of the null model in iteration i, and $\mathbb{1}$ is the indicator function.

4.7 Expected Clustering in Erdős–Rényi Networks

Consider an Erdős–Rényi (ER) network with N nodes and E edges. Let us denote the network density by

$$\delta = \frac{2E}{N(N-1)},$$

which corresponds to the probability that any two nodes are connected. In an ER network, the expected local clustering coefficient of a node is equal to the probability that two of its neighbors are also connected. Since each edge is independently present with probability δ, the expected local clustering coefficient for any node is simply

$$\mathbb{E}[C_i] = \delta.$$

Averaging over all nodes gives the expected average clustering coefficient

$$\mathbb{E}[C_{\text{avg}}] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[C_i] = \delta.$$

This matches the intuition provided by Newman [1], who derives the same result for the global clustering coefficient; here, we see that the expected local clustering coincides with the network density as well.

References

[1] M. E. J. Newman, *The Structure and Function of Complex Networks*, SIAM Review, 45(2):167–256, 2003.