Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Analýza interpunkčných znamienok v rôznojazyčných textoch

Bakalárska práca

2025
Zdenko Németh

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Analýza interpunkčných znamienok v rôznojazyčných textoch

Bakalárska práca

Študijný program:	Aplikovaná Informatika
Študijný odbor:	Informatika
Školiace pracovisko:	Katedra aplikovanej informatiky
Školiteľ:	doc. RNDr. Mária Markošová, PhD.

Bratislava, 2025
Zdenko Németh

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Zdenko Németh
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Analýza interpunkčných znamienok v rôznojazyčných textoch
Analysis of punctuation marks in different languages texts

Anotácia: Študent naprogramuje aplikáciu, ktorá v textovom súbore spracuje distribúciu interpunkčných znamienok. V aplikácii je potrebná možnosť výberu, teda, či pôjde o distribúciu všetkých interpunkčných znamienok, alebo len tých, ktoré budú dopredu špecifikované (napr. len čiarky, len bodkočiarky ....). Aplikácia tiež musí umožniť rôzne typy zobrazení výsledných súborov, ako napr. log log zobrazenie, log lin zobrazenie a podobne. Študent porovná a vyhodnotí distribúcie, ktoré získa pre jednotlivé jazyky.

Cieľ: Cieľom je vytvoriť aplikáciu na analýzu distribúcií interpunkčných znamienok v texte.

Literatúra: Kulig and others. In narrative texts punctuation marks obey the same statistics as words, Information Sciences
Volume 375, 1 January 2017, Pages 98-113

Vedúci: doc. RNDr. Mária Markošová, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: doc. RNDr. Tatiana Jajcayová, PhD.

Dátum zadania: 22.02.2024

Dátum schválenia: 23.10.2024 doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Poďakovanie: V prvom rade by som sa rád poďakoval svojej školiteľke doc. RNDr. Márii Markošovej, PhD., za odborné vedenie a cenné rady počas písania tejto bakalárskej práce. Ďakujem tiež svojej rodine, priateľom a všetkým, ktorí ma počas celého štúdia podporovali a motivovali.

Abstrakt

Táto bakalárska práca sa zaoberá analýzou interpunkčných znamienok v rôznojazyčných textoch. Cieľom práce bolo vytvoriť aplikáciu umožňujúcu analyzovať vplyv interpunkcie na štruktúru a vlastnosti slovných sietí. Aplikácia poskytuje používateľovi možnosť výberu textu, jazyka, konkrétnych interpunkčných znamienok a následne vybrať typ analýzy, o ktorú má záujem. Podporuje tri hlavné typy analýz: výpočtovú grafovú a jazykovú analýzu, analýzu distribúcie stupňov vrcholov a analýzu závislosti zmeny exponentu mocninového rozdelenia od počtu vrcholov siete. S ohľadom na komplexnosť výpočtov bola analýza realizovaná na vybraných vzorkách textov, ktoré boli dostačujúce na získanie reprezentatívnych výsledkov. V práci je prezentovaný postup riešenia problematiky, návrh aplikácie a implementácia dôležitých častí, ako aj jej možné vylepšenia a rozšírenia. V závere práce sú zhrnuté hlavné zistenia a navrhnuté ďalšie smery výskumu.

Kľúčové slová: slovná sieť, analýza textu, interpunkcia, mocninové rozdelenie, grafová analýza, jazyková analýza, NetworkX

Abstract

This bachelor's thesis deals with the analysis of punctuation marks in different language texts. The aim of the thesis was to develop an application that allows analyzing the influence of punctuation marks on the structure and properties of word networks. The application provides the user with the ability to select text, its language, specific punctuation marks and then select the type of analysis they are interested in. It supports three main types of analysis: computational graph and language analysis, degree distribution analysis and the study of how the power-law exponent depends on the number of network nodes. Given the complexity of the calculations, the analysis was carried out on selected samples of texts that were sufficient to obtain representative results. The thesis presents the procedure for solving the problem, the design of the application and the implementation of important parts, as well as its possible improvements and extensions. The main findings are summarized and further research directions are proposed at the end of the thesis.

Keywords: word network, text analysis, punctuation marks, power-law distribution, graph analysis, language analysis, NetworkX

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úryvky kódu

Úvod

Put your introduction here.

Kapitola 1

Prehľad problematiky

Táto kapitola sa zaoberá základnými pojmami a definíciami, ktoré sú nevyhnutné pre pochopenie problematiky tejto práce.

Najskôr sa zameriame na teóriu grafov, ktorá je základom pre analýzu a modelovanie sietí. Definujeme si kľúčové pojmy, ako sú graf, vrchol, hrana, stupeň a ďalšie.

Následne sa pozrieme na rôzne typy modelov sietí a ich vlastnosti, predovšetkým na náhodné a bezškálové siete. Budeme sa sústrediť na ich vznik a charakteristiky, ako je distribúcia stupňov vrcholov. Taktiež sa budeme venovať slovným sieťam, ktoré sú špecifickým typom sietí, využívaným na analýzu jazyka a jeho štruktúry.

Na záver kapitoly sa zameriame na úlohu a význam interpunkcie v jazyku a jej vplyvu na štruktúru textu.

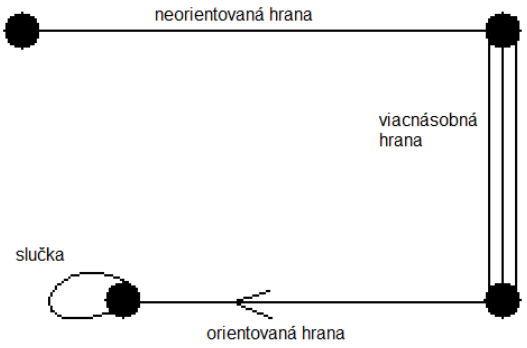
1.1 Teória grafov

Teória grafov je rozsiahla a komplexná oblasť matematiky a informatiky. V tejto kapitole sa zameriame na základné pojmy, definície a koncepty, ktoré sú nevyhnutné na pochopenie práce s grafmi, ich vlastností a rôznych aplikácií. Pre podrobnejšie informácie a hlbšie pochopenie odporúčam odbornú literatúru, najmä publikáciu *Graph Theory Fifth Edition od R. Diestela* [1] .

Graf G je reprezentovaný ako dvojica $G = (V, E)$, pričom V *(z angl. vertices)* je množina všetkých vrcholov a E je množina všetych hrán *(z angl. edges)*, ktoré tvoria spojenia medzi týmito vrcholmi. Vrcholy grafu sa taktiež nazývajú aj uzly.

Medzi dvojicou vrcholov grafu môže, ale nemusí existovať hrana. Ak medzi nimi hrana existuje, hovoríme, že sú navzájom prepojené. Jednotlivé hrany, ktoré obsahuje množina E, sú reprezentované ako dvojice vrcholov (u, v), pričom u a v sú vrcholy z množiny V.

Rozlišujeme niekoľko typov hrán. Ak má hrana grafu daný smer, nazýva sa orientovaná a záleží na poradí vrcholov v usporiadanej dvojici, prvý vrchol dvojice predstavuje



Obr. 1.1: Zobrazenie typov hrán v grafe.

uzol, od ktorého hrana vychádza a druhý vrchol dvojice predstavuje uzol, do ktorého hrana smeruje. Ak hrana nemá určený smer, tak sa jedná o neorientovanú hranu a poradie vrcholov v usporiadanej dvojici nie je dôležité. Hranu, ktorá má začiatok a koniec v rovnakom vrchole, nazývame slučka. Pojem viacnásobná hrana označuje prípad, kedy medzi dvoma vrcholmi existuje viac ako jedna hrana. Rôzne typy hrán je možné vidieť na ilustračnom obrázku č. 1.1 .

Hrany, ktoré majú priradenú číselnú hodnotu, nazývame vážené hrany. Tieto váhy môžu reprezentovať rôzne vlastnosti hrany, ako napríklad vzdialenosť medzi vrcholmi alebo náklady na prechod medzi nimi. Sú využívané hlavne v praktických aplikáciách, ako napríklad dopravné siete.

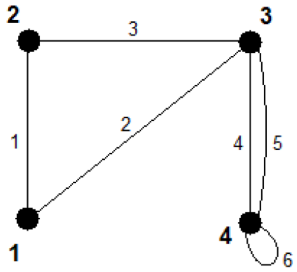
Pojem jednoduchý graf definuje taký graf, ktorý neobsahuje slučky ani viacnásobné hrany. Jednoduchý graf, ktorý neobsahuje orientované hrany a vrcholy sú poprepájané spôsobom každý s každým nazývame kompletný graf [2] .

Pri analýze grafov je potrebné poznať rôzne spôsoby, akými môžeme prechádzať cez vrcholy a hrany grafu. Prechod grafom, pri ktorom sa striedajú vrcholy a hrany, ktoré sa môžu opakovať, sa nazýva sled. Formálny zápis pre sled v grafe $G = (V, E)$ je postupnosť

$$v_0, e_0, v_1, e_1, \ldots, e_{k-1}, v_k,$$

kde $v_i \in V(G)$ pre všetky $i \in \{0, 1, \ldots, k\}$ a $e_j \in E(G)$ pre všetky $j \in \{0, 1, \ldots, k-1\}$ s podmienkou $e_i = (v_i, v_{i+1})$. Ťah je špeciálny typ sledu, pri ktorom sa hrany nemôžu opakovať, teda pre všetky $i \neq j$ platí $e_i \neq e_j$. Vrcholy sa v ťahu môžu opakovať. Sled a ťah majú špeciálny prípad, kedy sa počiatočný a koncový vrchol zhodujú, teda $v_0 = v_k$. Takýto sled sa nazýva uzavretý sled a ťah uzavretý ťah. Ešte existuje pojem cesta, ktorý predstavuje prechod grafom, pri ktorom sa nemôžu opakovať ani vrcholy ani hrany.

Spojitý graf je taký graf, v ktorom existuje cesta medzi každými dvoma vrcholmi. Nie každý graf je spojitý, pretože niektoré grafy sa skladajú z viacerých disjunktných častí, ktoré nie sú navzájom prepojené hranou, teda medzi nimi neexistuje žiadna cesta. Takýmto disjunktným častiam grafu hovoríme komponenty. Graf s viacerými



$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 2 \\ 0 & 0 & 2 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 2 \end{pmatrix}$$

Obr. 1.2: Matica susedností a incidenčná matica pre zobrazený graf.

komponentami je nespojitý graf.

Grafy sa dajú reprezentovať rôznymi spôsobmi. Najčastejšia matematická reprezentácia je pomocou matice susedností a matice incidencie. Nech $G = (V, E)$ je graf s $n = |V|$ vrcholmi a $m = |E|$ hranami. Matica susedností A je $n \times n$ matica, kde každý prvok a_{ij} predstavuje počet hrán medzi vrcholmi v_i a u_j. Incidenčná matica B je $n \times m$ matica, kde pre každý prvok platí $b_{ij} = 1$, ak je hrana e_j incidentná s vrcholom v_i, inak $b_{ij} = 0$. Príklad matice susedností a incidenčnej matice spolu s ilustračným grafom je zobrazený na obrázku č. 1.2 .

Teraz si predstavíme niekoľko základných vlastností, s ktorými budeme pracovať, ich definície z teórií grafov a ich aplikáciu. Budeme sa zaoberať iba jednoduchými, neorientovanými grafmi, ak nebude uvedené inak.

Jedna zo základných vlastností vrchola grafu je jeho stupeň. Stupeň vrchola v označovaný aj ako $deg(v)$ nám udáva počet hrán, ktoré sú incidentné s vrcholom v. Minimálna hodnota stupňa vrchola je 0, vtedy tvorí vrchol samostatný komponent, taktiež sa nazýva izolovaný vrchol. Maximálna hodnota stupňa vrchola je $n-1$, nastáva keď vrchol susedí s každým vrcholom grafu. Celkový počet hrán v grafe potom získame ako polovicu súčtu všetkých stupňov vrcholov grafu, pretože každá hrana je incidentná s dvoma vrcholmi. Tento vzťah môžeme zapísať ako:

$$|E| = \frac{1}{2} \sum_{i=1}^{n} deg(v_i), \tag{1.1}$$

kde $|E|$ predstavuje počet hrán v grafe a n je počet vrcholov [3] .

Taktiež často používanou charakteristikou grafu je priemerný stupeň uzla:

$$\bar{d} = \frac{1}{n} \sum_{i=1}^{n} deg(v_i) = \frac{2|E|}{n}, \tag{1.2}$$

kde $\bar{d}$ je priemerný stupeň uzla a n je počet vrcholov grafu [3] .

Hustota grafu určuje, ako blízko je graf ku kompletnému, teda akú časť z maximálneho počtu hrán obsahuje. Nadobúda hodnoty v intervale $[0, 1]$, pričom 0 znamená, že

graf je prázdny a 1 znamená, že graf je kompletný. Formálny zápis pre hustotu grafu je:

$$D = \frac{2|E|}{n(n-1)}, \tag{1.3}$$

kde D je hustota grafu, $|E|$ je počet hrán a n je počet vrcholov.

Ďalšou vlastnosťou grafu je koeficient zhlukovania, inak známy aj ako klasterizačný koeficient. Tento koeficient udáva ako veľmi sú prepojené susedné vrcholy uzla. Hodnota koeficientu sa pohybuje v intervale $[0, 1]$, kde 0 značí, že medzi susedmi vrcholu neexistujú žiadne hrany a 1 znamená, že všetci susedia sú medzi sebou navzájom prepojení. Nech v je vrchol grafu a $deg(v)$ je jeho stupeň, potom:

$$C_v = \frac{2|E_v|}{deg(v)(deg(v)-1)}, \tag{1.4}$$

kde C_v je koeficient zhlukovania vrchola v, $|E_v|$ predstavuje počet hrán medzi $deg(v)$ susedmi vrchola v. Týmto spôsobom vieme získať lokálny koeficient zhlukovania pre každý vrchol grafu, z ktorých potom vieme vypočítať priemerný koeficient zhlukovania pre celý graf:

$$C = \frac{1}{n}\sum_{i=1}^{n} C_{v_i}, \tag{1.5}$$

kde C je priemerný koeficient zhlukovania grafu a n je počet vrcholov grafu [3] .

Pojem najkratšia cesta v grafe definuje takú cestu, ktorá spája dva vrcholy grafu a obsahuje najmenší možný počet hrán. Priemerná najkratšia cesta medzi dvoma vrcholmi je definovaná ako priemer všetkých najkratších ciest medzi všetkými kombináciami dvojíc vrcholov grafu.

$$\ell = \frac{1}{n(n-1)}\sum_{\substack{i,j=1 \\ i\neq j}}^{n} d(v_i, v_j), \tag{1.6}$$

kde ℓ je priemerná najkratšia cesta, $d(v_i, v_j)$ je dĺžka najkratšej cesty medzi vrcholmi v_i a v_j a n je počet vrcholov grafu [3] .

Priemer grafu udáva najväčšiu dĺžku najkratšej cesty medzi dvoma rôznymi vrcholmi grafu.

$$diam = \max_{i\neq j} d(v_i, v_j), \tag{1.7}$$

Veľmi dôležitým parametrom pri analýze grafu je distribúcia stupňov vrcholov, značené ako p_{deg}. Poskytuje pravdepodobnosť, že náhodne vybraný vrchol grafu má práve deg susedov [3] . Pre graf s počtom vrcholov n a počtom vrcholov so stupňom deg označeným ako n_{deg} je distribúcia stupňov definovaná ako:

$$p_{\text{deg}} \cong \frac{n_{\text{deg}}}{n}, \tag{1.8}$$

kde p_{deg} predstavuje pravdepodobnosť výskytu daného stupňa, n_{deg} je počet vrcholov s daným stupňom a n je počet vrcholov grafu.

1.2 Modely sietí

Siete sú štruktúry, ktoré možno reprezentovať ako grafy skladajúce sa z uzlov (vrcholov) a prepojení (hrán), pričom uzly predstavujú jednotlivé entity a prepojenia reprezentujú vzťahy medzi nimi. Pojem graf sa používa matematickej reprezentácií, zatiaľ čo termín sieť často odkazuje na reálne systémy. Umožňujú modelovanie a analýzu komplexných systémov v rôznych odvetviach, ako informatika, biológia, sociológia a iné [3] . Medzi základné topológie sietí patria náhodné siete a bezškálové siete, ktoré predstavujú odlišné prístupy ku vzniku a distribúcii prepojení.

1.2.1 Náhodná sieť

Náhodná sieť predstavuje jeden zo základných konceptov v teórií sietí. Definujú sa dva základné modely náhodných sietí.

Prvý z nich je Erdős-Rényiho model, $G_{V,H}$, kde V je množina vrcholov a H je množina náhodne vybraných hrán [4][3] . Tento model generuje náhodnú sieť tak, že začne s $n = |V|$ izolovanými vrcholmi a následne náhodne pridáva $m = |H|$ rôznych hrán medzi nimi, ktoré sa vyberajú z $\frac{n*(n-1)}{2}$ všetkých možných hrán.

Druhý model je Gilbertov model, $G_{V,p}$, takisto obsahuje $n = |V|$ vrcholov, však hrany nie sú pridelené pevným počtom, ale nezávisle s pravdepodobnosťou p [5][3] . Model začína s n izolovanými vrcholmi a následne pre každú dvojicu vrcholov u a v pridá hranu medzi nimi s pravdepodobnosťou p, pričom výsledný počet hrán je náhodný a závisí od hodnoty p.

Priemerný stupeň vrchola v náhodnej sieti pre model $G_{V,H}$ vieme vypočítať jednoduchým vzorcom:

$$\bar{d} = \frac{2|H|}{n}, \tag{1.9}$$

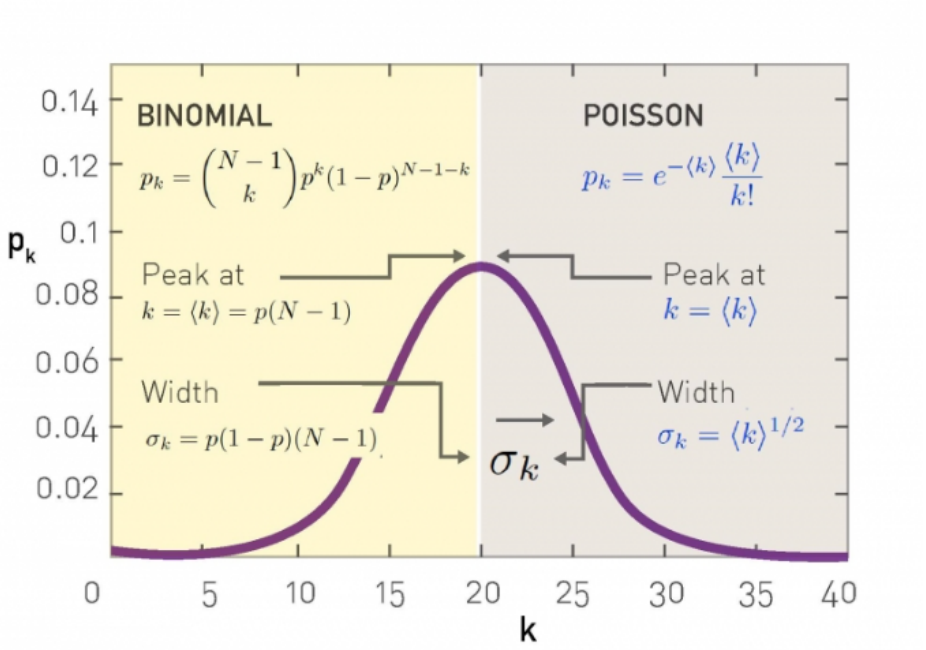
keďže poznáme počet všetkých hrán $|H|$, aj vrcholov n [3] . Pre výpočet priemerného stupňa vrchola v Gilbertovom modeli $G_{V,p}$ môžeme použiť vzorec [3] :

$$\bar{d} = \frac{2 < L >}{n} = p(n-1), \tag{1.10}$$

kde $< L >$ definuje súčin pravdepodobnosti p a počtu párov, ktoré sa snažíme spojiť [3] :

$$< L >= p\frac{n(n-1)}{2}. \tag{1.11}$$

Na tvorbu náhodných sietí nemáme veľký vplyv, preto nastáva, že niektoré vrcholy majú veľmi vysoký stupeň, zatiaľ čo iné majú veľmi nízky stupeň, dokonca aj nulový. Tieto rozdiely je možné pozorovať na distribúcii stupňov vrcholov p_k, ako je vidieť na obrázku č. 1.3 [3] .



Obr. 1.3: Binomická vs. Poissonová distribúcia stupňov. Presná forma distribúcie náhodnej siete je binomická distribúcia (ľavá strana), pri veľkom počte vrcholov sa dá priblížiť Poissonovou distribúciou (pravá strana) [3] .

Distribúcia stupňov vrcholov v náhodnej sieti sa riadi binomickou distribúciou, ktorá je definovaná ako [3] :

$$p_{\text{deg}} = \binom{n-1}{\text{deg}} p^{\text{deg}} (1-p)^{n-1-\text{deg}}, \tag{1.12}$$

pričom tvar binomickej distribúcie je daný počtom vrcholov n, počtom hrán m a pravdepodobnosťou p. Distribúcia stupňov vrcholov v náhodnej sieti sa dá priblížiť Poissonovou distribúciou, ak má veľký počet vrcholov, ktorá je definovaná ako [3] :

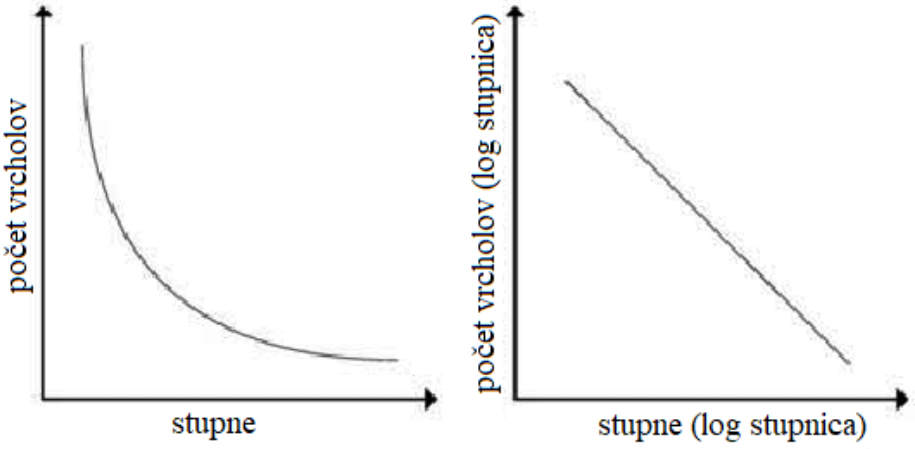
$$p_{\text{deg}} = \frac{\bar{d}^{\text{deg}} e^{-\bar{d}}}{\text{deg}!}, \tag{1.13}$$

$\bar{d}$ je priemerný stupeň vrchola, ktorý je daný vzorcom 1.10 .

Koeficient zhlukovania v náhodnej sieti je veľmi nízky, pretože väčšina vrcholov je izolovaných a nie sú prepojené. Vieme ho jednoducho vypočítať ako:

$$C = p = \frac{\bar{d}}{n}, \tag{1.14}$$

predstavuje pravdepodobnosť, že dva náhodne vybrané vrcholy sú prepojené [3] .



Obr. 1.4: Distribúcia stupňov vrcholov v bezškálovej sieti [7] .

1.2.2 Bezškálová sieť

Pri bezškálových sieťach sa jedná o špecifický typ grafu, ktorý sa vyznačuje tým, že existuje niekoľko uzlov s extrémne vysokým počtom prepojení a väčšina má malý počet prepojení. Tento typ siete sa v reálnom živote vyskytuje oveľa častejšie ako náhodné siete, najmä v kontextoch ako internet, biologické siete alebo sociálne siete [6] [3] .

Na rozdiel od náhodných sietí, kde sú stupne vrcholov rozdelené podľa binomickej alebo Poissonovej distribúcie, ako je vidno na obrázku č. 1.3 , bezškálové siete dodržujú tzv. mocninové rozdelenie (power-law distribution):

$$p_{\deg} \sim \deg^{-\gamma}, \tag{1.15}$$

kde γ je škálovací exponent v rozmedzí $2 < \gamma < 3$, ktorý určuje tvar [3] . Pri tomto rozdelení nás najviac zaujíma rozdelenie stupňov vrcholov, zobrazené v dvojitej logaritimickej mierke, kde produkuje priamku, ako je vidieť na obrázku č. 1.4 .

Najznámejší model na generovanie bezškálových sietí je Barabási–Albertov model, označovaný ako $G_{BA}(n, m)$, ktorý má škálovací exponent $\gamma = 3$. Pri generovaní využíva princíp preferenčného pripájania. Začína s malým počtom počiatočných uzlov a v každom kroku sa pridá nový uzol, ktorý sa pripojí k m už existujúcim uzlom. Pravdepodobnosť prepojenia nového uzla so starým závisí od stupňa starého uzla [3] :

$$p(\deg(\mathrm{i})) = \frac{\deg(\mathrm{i})}{\sum_j \deg(\mathrm{j})}, \tag{1.16}$$

Ďalším modelom na generovanie bezškálových sietí je Dorogovtsev–Mendes model, často označovaný ako $G_{DM}(n)$. Tento model taktiež generuje siete s mocninovým rozdelením stupňov. Od Barabási-Albertovho modelu sa odlišuje mechanizmom rastu siete. Model začína s trojuholníkom (kompletným grafom s tromi uzlami) a v každom kroku sa pridá nový uzol, ktorý sa pripojí na oba koncové body náhodne vybranej

existujúcej hrany [8]. Takýmto spôsobom sa v každom kroku vytvorí nový trojuholník, čo vedie k vysokému koeficientu zhlukovania. Model teda zabezpečuje vznik sietí, ktoré sú nielen bezškálové, ale majú aj malé priemerné vzdialenosti medzi uzlami a vysokú mieru zhlukovania.

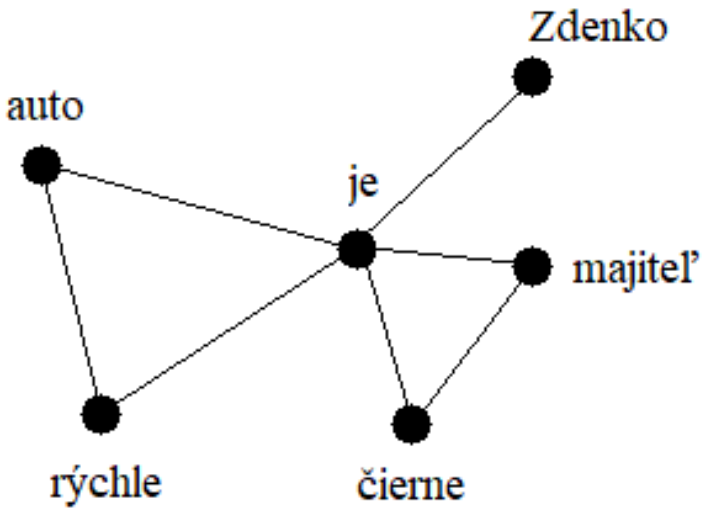
Zvýšený koeficient zhlukovania sa vyskytuje najmä pri analýze slovných sietí, kde sa rovnaké slová často vyskytujú v blízkosti seba a vytvárajú frázy, prirodzene sa zhlukujú do skupín. Práve preto je Dorogovtsev-Mendes model vhodný na analýzu jazykových dát, keďže dokáže simulovať podobné vlastnosti, štruktúru ako majú slovné siete vytvorené na reálnych jazykových dátach.

1.3 Slovné siete

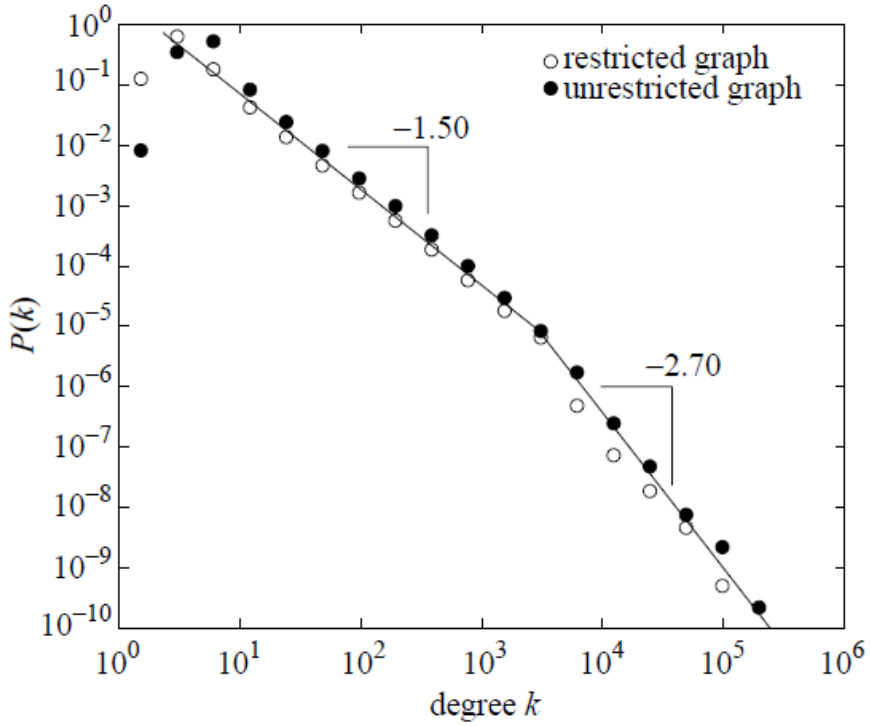
Slovné siete predstavujú efektívny spôsob reprezentácie a analýzy jazyka a jeho štruktúry za pomoci teórie grafov. V takýchto sieťach sú jednotlivé slová reprezentované ako uzly a vzťahy medzi slovami, ktoré sa navzájom ovplyvňujú alebo súvisia tvoria hrany [9] . Podľa metód reprezentácie vzťahov medzi slovami dokážeme slovné siete rozlišovať. V sémantických slovných sieťach vznikajú prepojenia medzi slovami na základe ich významovej podobnosti. Narozdiel od sémantických slovných sietí, v syntaktických slovných sieťach sú interakcie medzi slovami založené na slovnej štruktúre a gramatických pravidlách.

Špecifický typ syntaktickej slovnej siete, ktorej sa budeme venovať v tejto práci sa nazýva pozičná slovná sieť. Táto sieť je zameraná na štruktúru a pozície slov v texte. V pozičnej slovnej sieti uzly tvoria jednotlivé slová a hrany sú vytvorené na základe toho, či sa slová nachádzajú vedľa seba v texte. Ukážku takejto siete vidíme na obrázku č. 1.5 .

Tieto siete majú niekoľko zaujímavých vlastností. Koeficient zhlukovania, ktorý určuje, do akej miery slová tvoria skupiny alebo často používané frázy. Distribúcia stupňov uzlov je typu mocninného zákona, čo znamená, že väčšina slov má veľmi nízky stupeň, zatiaľ čo malý podiel slov má veľmi vysoký stupeň [10] . Toto rozdelenie je typické pre bezškálové siete. Na obrázku č. 1.6 je zobrazená distribúcia stupňov uzlov s dvomi rôznymi priemernými exponentami $y = -1.5$ a $y = -2.7$. Exponent $y = -2.7$ sa blíži k exponentu pre Barabási-Albertov model, ktorý je $y_{BA} = -3$ [11] .



Obr. 1.5: Pozičná slovná sieť z viet: „Auto je rýchle. Auto je čierne. Majiteľ je Zdenko.“.



Obr. 1.6: Dorogovtsev Mendes model, distribúcia stupňov vrcholov pre dve siete, body zoskupené mocninou dvoch [11] .

1.4 Interpunkcia

Interpunkcia je základný prvok jazyka, používa sa na štruktúrovanie a organizovanie textu, vyjadrenie významových vzťahov. Pri správnom použití interpunkcia zlepšuje čitateľnosť a porozumenie textu, zatiaľ čo nesprávne použitie môže viesť k nejednoznačnosti a nedorozumeniu [12].

Tvorí ju súbor znakov, kde každý znak má svoj osobitý význam a funkciu, ako napríklad pauza, dôraz, ukončenie vety, vzťahy medzi vetnými členmi a podobne. Pravidlá písania interpunkcie sú kodifikované v gramatických a jazykových príručkách. Taktiež sa pravidlá písania interpunkcie líšia v závislosti od jazyka, v ktorom je text napísaný.

Najbežnejšie používané interpunkčné znamienka možno rozdeliť do niekoľkých skupín podľa ich funkcie: ukončovacie (bodka, otáznik, výkričník), oddeľovacie (čiarka, bodkočiarka, dvojbodka), typografické (úvodzovky, zátvorky, pomlčky), prípadne aj iné znaky (trojbodka, lomka, atď.).

Z gramatického hľadiska interpunkcia rozdeľuje text, vyznačuje vetnú štruktúru, oddeľuje vetné členy, hlavné a vedľajšie vety. Napríklad umiestnenie čiarky pred spojkami často naznačuje, že ide o vedľajšiu vetu, zatiaľ čo bodka, otáznik, výkričník označujú koniec vetného celku.

Z fonetického hľadiska interpunkcia určuje intonáciu, pauzy pri čítaní textu, dôraz, rytmus a tón. Napríklad bodka naznačuje koniec vety a vyžaduje prestávku, kým čiarka naznačuje, že čitateľ by mal pokračovať v čítaní s malou pauzou. Takýmto spôsobom interpunkcia dokáže preniesť aspekty reči do písaného textu, čo môže ovplyvniť jeho význam a interpretáciu.

Zo štylistického hľadiska interpunkcia ovplyvňuje štýl a tón textu. Napríklad použitie výkričníka môže naznačovať dôraz, vážnosť alebo nadšenie, zatiaľ čo otáznik môže naznačovať pochybnosť. Taktiež môže ovplyvniť rytmus a plynulosť textu. Napríklad dlhé vety s množstvom čiarkami môžu spôsobiť, že text bude pôsobiť ťažkopádne a zložito, zatiaľ čo krátke vety s minimálnym použitím interpunkcie môžu pôsobiť rýchlo a dynamicky.

Kapitola 2

Použité technológie

2.1 NetworkX

NetworkX je open-source knižnica pre Python, ktorá sa zaoberá tvorbou, manipuláciou a analýzou komplexných sietí a grafov. Táto knižnica podporuje všetky druhy grafov, vrátane orientovaných a neorientovaných grafov, vážených a nevážených grafov, ako aj multigrafov [13].

V tejto práci je knižnica NetworkX využitá na základnú grafovú analýzu. Na analýzu sietí sú použité rôzne zabudované funkcie, ktoré umožňujú efektívne vypočítať rôzne metriky, ako napríklad stupeň uzlov, koeficient zhlukovania, priemernú dĺžku najkratšej cesty, hustotu a priemer siete.

Okrem toho je knižnica využitá aj na generovanie rôznych typov bezškálových sietí, ako sú Barabási-Albertov model a Dorogovtsev-Mendesov model. Pre generovanie daných modelov boli použité zabudované funkcie, $nx.barabasi\_albert\_graph()$ a $nx.dorogovtsev\_goltsev\_mendes\_graph()$, ktoré umožňujú rýchle, efektívne vytvorenie týchto sietí.

Knižnica NetworkX bola vybraná pre jednoduchosť implementácie, rozsiahlu dokumentáciu a podporu integrácie s inými knižnicami v Pythone, ako sú NumPy a Matplotlib.

Na ukážke 2.1 je zobrazený kód, ktorý generuje Dorogovtsev-Mendesov model za pomoci funkcie $nx.dorogovtsev\_goltsev\_mendes\_graph()$ a následne využije funkcie $G.number\_of\_nodes()$ pre výpočet počtu vrcholov, $G.number\_of\_edges()$ pre výpočet počtu hrán, $nx.average\_clustering(G)$ pre výpočet koeficientu zhlukovania a $nx.average\_shortest\_path\_length(G)$ pre výpočet priemernej dĺžky najkratšej cesty v grafe.

```
import networkx as nx

G = nx.dorogovtsev_goltsev_mendes_graph(5)

numNodes = G.number_of_nodes()
numEdges = G.number_of_edges()
avgClustering = nx.average_clustering(G)
avgShortestPath = nx.average_shortest_path_length(G)

print(f"Number of nodes: {numNodes}")
print(f"Number of edges: {numEdges}")
print(f"Average clustering coefficient: {avgClustering}")
print(f"Average shortest path length: {avgShortestPath}")
```

Kód 2.1: Použitie NetworkX pre generovanie DGM modelu a grafovú analýzu.

Kapitola 3

Tvorba aplikácie

V tejto kapitole sa budeme venovať postupu riešenia zadanej problematiky. Oboznámime sa s implementáciou a využitím najdôležitejších častí aplikácie. Popíšeme priebeh testovania počas vývoja a na záver sa zameriame na možnosti rozšírenia do budúcnosti, ktoré by mohli zlepšiť funkcionalitu aplikácie.

3.1 Prístup k problematike

Aplikácia je implementovaná v programovacom jazyku Python. Využíva externé knižnice, ako sú Matplotlib pre grafickú vizualizáciu charakteristík, re pre spracovanie textových dát, NumPy pre numerické výpočty a prácu s dátami, wxPython pre vytvorenie grafického používateľského rozhrania, NetworkX pre prácu s pozičnou slovnou sieťou a výpočet analýzy.

Prvým krokom bolo získanie textových dát pre analýzu. Na tento účel bola využitá stránka Project Gutenberg [14] , kde je možné nájsť množstvo verejne dostupných literárnych diel od rôznych autorov preložených do rôznych svetových jazykov.

Následne bolo nevyhnutné zadefinovať spôsob, akým sa vytvorí pozičná slovná sieť. Každé slovo v texte sa považuje za vrchol, tvar slova sa významovo nemení, iba sa normalizuje celé slovo na malé písmená. Dva vrcholy sú spojené neorientovanou neohodnotenou hranou v prípade, že im príslušné slová nasledujú v texte bezprostredne za sebou. Vytvorená sieť je teda neorientovaná a neohodnotená.

Ak je pri tvorbe siete zvolená množina interpunkčných znakov, tieto sa považujú za samostatné slová, ostatné sa zanedbávajú a sieť sa tvorí rovnakým postupom.

Pri vizualizácii sa zameriame na zobrazenie distribúcie stupňov vrcholov, ktorá je jednou z najdôležitejších charakteristík pozičnej slovnej siete. Taktiež zobrazíme závislosť veľkosti siete (t.j. počtu vrcholov) od hodnoty exponentu γ mocninového zákona.

Výpočet analýzy siete je rozdelený do dvoch oblastí. Prvou je grafová analýza,

ktorá sa zaoberá výpočtom základných veličín siete, ako je počet vrcholov; počet hrán; maximálny, minimálny, priemerný stupeň vrcholov; hustota siete; priemerný koeficient zhlukovania; priemerná najkratšia cesta; priemer siete a korelácie medzi jednotlivými centralitami. Druhou oblasťou je jazyková analýza, ktorá zahŕňa výpočet jazykových charakteristík, ako je počet slov; maximálna, minimálna, priemerná dĺžka slova; počet viet; maximálna, minimálna, priemerná dĺžka vety; počet dvojíc a trojíc slov (frázy), ich frekvencia (maximálna, minimálna, priemerná) a počet interpunkčných znakov vrátane ich najčastejšieho a najmenej častého výskytu.

Výsledky uvedených grafických a výpočtových analýz budú porovnávané pre dve verzie siete - bez interpunkcie a s interpunkciou. Následne budeme skúmať ich odlišnosti a podobnosti, taktiež budeme porovnávať výsledky pre rôzne jazyky a texty.

3.2 Grafické používateľské rozhranie

Grafické používateľské rozhranie (GUI) je vytvorené pomocou knižnice wxPython. Jeho úlohou je poskytnúť používateľovi jednoduchý a prehľadný spôsob interakcie s aplikáciou. Obsahuje jednu hlavnú obrazovku, ktorá je rozdelená na niekoľko komponentov, kde každý z nich má svoju špecifickú funkciu.

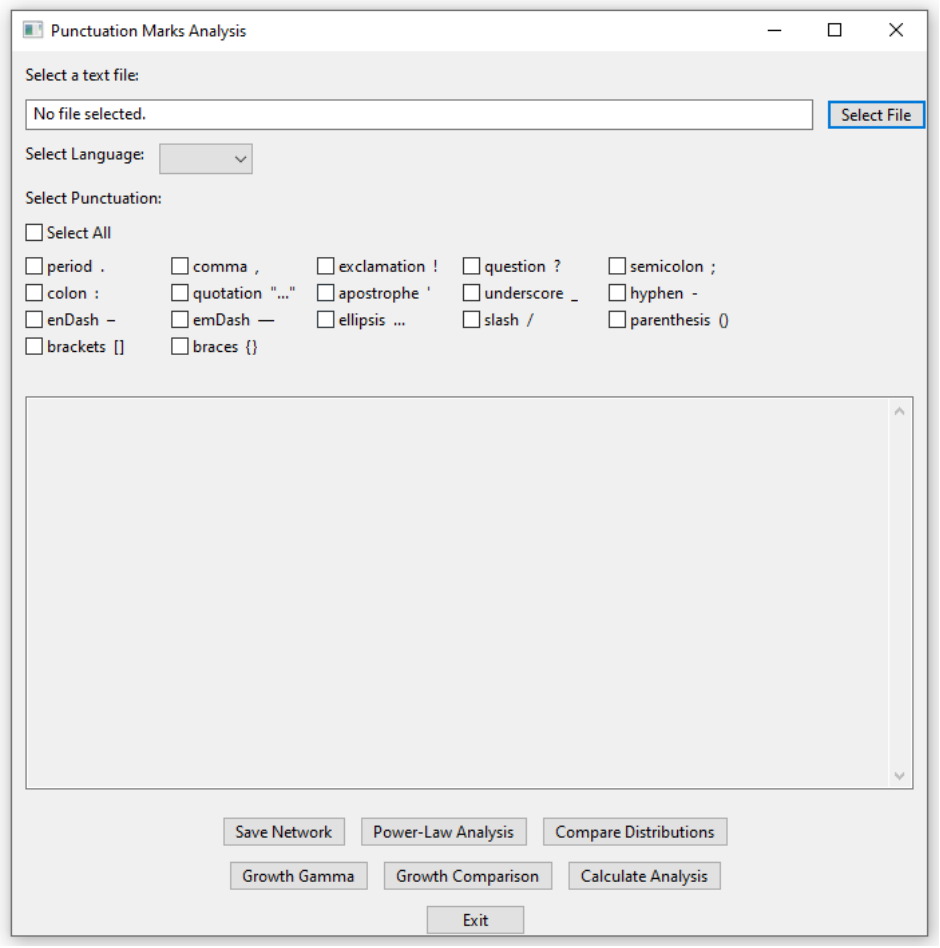
Prvý komponent v hornej časti obrazovky je textové pole spolu s tlačidlom, ktoré slúži na načítanie textového súboru. Po stlačení tlačidla sa otvorí dialógové okno, v ktorom si používateľ môže vybrať textový súbor, ktorý chce analyzovať. Ako náhle je súbor vybraný, jeho názov sa zobrazí v textovom poli.

Nasleduje rozbaľovacia ponuka, ktorá slúži na výber jazyka textu. Používateľ si môže vybrať z niekoľkých jazykov, ktoré aplikácia podporuje. Na základe výberu jazyka sa prispôsobia ďalšie nastavenia aplikácie.

Ďalší komponent obsahuje zaškrtávacie políčka na výber interpunkčných znakov, ktoré sa majú zohľadniť pri tvorbe siete. Obsahuje možnosť označiť všetky interpunkčné znaky naraz alebo ich všetky zrušiť.

V strednej časti obrazovky sa nachádza dialógové pole, ktoré zobrazuje aktuálny stav aplikácie. Tu sa zobrazujú informácie o načítanom súbore, jazyku, vybraných interpunkčných znakoch a po výpočte analýzy aj výsledky.

Ako posledné obsahuje súbor tlačidiel jednotlivých funkcií aplikácie. Používateľ si môže vybrať, či chce vypočítať analýzu siete, zobraziť distribúciu stupňov vrcholov, rastovú analýzu siete alebo porovnanie slovnej siete bez interpunkcie so zvolenou sieťou. Jednou z možností je aj uloženie zvolenej siete do súboru. Ukážku grafického používateľského rozhrania možno vidieť na obrázku č. 3.1.



Obr. 3.1: Grafické používateľské rozhranie aplikácie.

3.3 Tvorba slovnej siete

Pre vytvorenie slovnej siete je nevyhnutné najprv načítaný textový súbor predspracovať. Cieľom predspracovanie je vytvorenie normalizovaného zoznamu slov, ktoré budú predstavovať uzly siete. Na tento účel sa využíva knižnica re, pretože umožňuje efektívne spracovanie textu pomocou regulárnych výrazov. Vďaka nej možno jednoducho definovať, ktoré interpunkčné znaky sa majú zohľadniť pri vytváraní siete, a ktoré sa majú ignorovať. Metódu predspracovania textu možno vidieť na ukážke č. 3.1. Vstupný parameter tejto metódy je množina interpunkčných znakov, zvolených používateľom prostredníctvom GUI v podobe slovníka, ktorý obsahuje názov interpunkčného znaku ako kľúč a regulárny výraz ako hodnotu.

```
1     def processTextFile(self, selectedPunctuation={}):
2         if self.selectedLanguage == 'English':
3             regexDict = self.regexDictEng
4         if self.selectedLanguage == 'German':
5             regexDict = self.regexDictGer
6         if not selectedPunctuation:
7             regexPattern = regexDict['wordsNumbers']
8         else:
9             usePunctuation = [regexDict[key] for key in
                  selectedPunctuation.keys()]
10            regexPattern = '|'.join([regexDict['wordsNumbers']] +
                  usePunctuation)
11        data = re.findall(regexPattern,
12                          self.readTextFile(self.inputTextFile))
13        if not data:
14            return []
15        return [word.lower() for word in data]
```

Kód 3.1: Metóda predspracovania textu.

Po úspešnom predspracovaní textu sa vytvorí slovník susedností a slovník početností pomocou metódy `createGraphData`, ktorú možno vidieť na ukážke č. 3.2. Na reprezentáciu pozičnej slovnej siete sa využíva knižnica NetworkX, špecificky trieda Graph. Implementáciu je možné vidieť na ukážke č. 3.3. Naďalej sa pracuje s touto inštanciou triedy Graph, pretože poskytuje množstvo užitočných funkcií a metód na analýzu a vizualizáciu grafov.

```
    def createGraphData(self, data):
        if self.selectedLanguage == 'English':
            regexDict = self.regexDictEng
        if self.selectedLanguage == 'German':
            regexDict = self.regexDictGer
        graphDataDict = {}
        nodeCounter = Counter()
        previousWord = None
        for element in data:
            if element in regexDict['quotation']:
                element = '"'
            if element in regexDict['apostrophe']:
                element = '\''
            if element not in graphDataDict:
                graphDataDict[element] = []
            if previousWord is not None:
                graphDataDict[previousWord].append(element)
                graphDataDict[element].append(previousWord)
            previousWord = element
            nodeCounter[element] += 1
        return graphDataDict, nodeCounter
```

Kód 3.2: Metóda pre vytvorenie pozičnej slovnej siete z predspracovaného textu.

```
    graphData, occurrenceData = self.createGraphData(
            self.processTextFile(self.selectedPunctuation))
        G = nx.Graph()
        for node, neighbors in graphData.items():
            for neighbor in neighbors:
                G.add_edge(node, neighbor)
```

Kód 3.3: Vytvorenie inštancie triedy NetworkX.Graph.

3.4 Zobrazenie výstupných charakteristík

Na zobrazenie výstupných charakteristík sa využíva knižnica Matplotlib. Táto knižnica poskytuje množstvo funkcií a metód na vytváranie grafov a vizualizáciu dát. V tejto aplikácii sa budeme zameriavať na zobrazenie distribúcie stupňov vrcholov, rastovú závislosť siete od hodnoty exponentu γ mocninového zákona.

3.4.1 Logaritmické zhlukovanie

Pri vizualizácii distribúcie stupňov vrcholov sa využíva metóda logaritmického zhlukovania, ktorá je implementovaná v metóde `calculateLogBin`, ukážka č. 3.4. Táto metóda zhlukuje dáta do logaritmických intervalov, čo umožňuje lepšie zobrazenie distribúcie stupňov vrcholov v prípade, že sú dáta rozptýlené na širokom intervale hodnôt.

Funkcia najprv vypočíta rozsah stupňov vrcholov, následne vytvorí logaritmicky rozdelené intervaly pomocou funkcie `np.logspace`, potom vypočíta hustotu pravdepodobnosti pre každý interval pomocou funkcie `np.histogram`. Výsledkom sú stredy intervalov a hustoty pravdepodobnosti, ktoré sú následne vrátené ako výstup metódy.

```
1     def calculateLogBin(self, degrees, binCount):
2         minDegree = max(1, min(degrees))
3         maxDegree = degrees.max()
4         bins = np.logspace(np.log10(minDegree),
5                            np.log10(maxDegree), num=binCount)
6         hist, binEdges = np.histogram(degrees, bins=bins,
7                                       density=True)
8         binCenters = (binEdges[:-1] + binEdges[1:]) / 2
9         nonzero = hist > 0
10        return binCenters[nonzero], hist[nonzero]
```

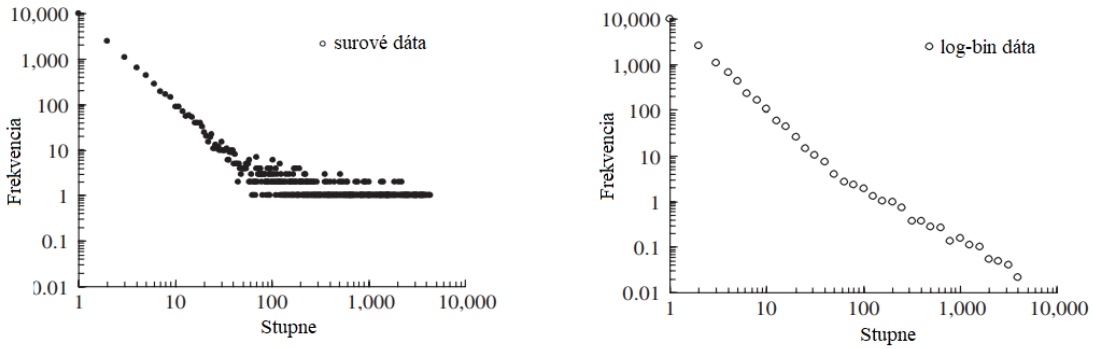
Kód 3.4: Metóda pre výpočet logaritmického zhlukovania.

Výsledné dáta sú zobrazované v log-log grafe, ktorý umožnuje overiť, či dáta nasledujú mocninový zákon a vypočítať hodnotu exponentu γ.

3.4.2 Distribúcia stupňov vrcholov

Distribúcia stupňov vrcholov udáva, ako sú stupne vrcholov rozložené v sieti. Najskôr sa vypočíta tzv. surová distribúcia stupňov, ktorá predstavuje relatívny výskyt každého stupňa v sieti. Pri slovnej sieti býva táto distribúcia zvyčajne veľmi rozptýlená, preto je ťažké získať prehľad o tom, ako sú stupne vrcholov rozložené. Z toho dôvodu sa používa logaritmické zhlukovanie 3.4.1, ktoré zoskupuje dáta do logaritmických intervalov. Porovnanie surovej distribúcie stupňov vrcholov s logaritmicky zhlukovanou distribúciou stupňov vrcholov je zobrazené na obrázku č. 3.2.

Práve logaritmicky zhlukovaná distribúcia stupňov vrcholov je veľmi dôležitá pre analýzu siete, pretože sa využíva na porovnanie s teoretickým mocninovým rozdelením. Pomocou tejto metódy vizualizácie je možné vypočítať hodnotu exponentu γ mocninového zákona, ktorý udáva, ako sú stupne vrcholov rozložené v sieti.



Obr. 3.2: Distribúcia stupňov vrcholov bez a s pomocou logaritmického zhlukovania [15].

3.4.3 Rastová analýza siete

Rastová analýza siete je ďalšou dôležitou charakteristikou, ktorá sa zaoberá tým, ako sa mení štruktúra siete pri zmene veľkosti textu. V tejto analýze sa skúma, ako sa vyvíja hodnota exponentu γ mocninového zákona v závislosti od veľkosti siete (počtu vrcholov). Takáto analýza nám umožňuje určiť, pri akej veľkosti textu sa hodnota exponentu γ stabilizuje, tým pádom vieme určiť, akú veľkosť textu je potrebné použiť na získanie reprezentatívnych výsledkov.

Postup spočíva v rozdelení textu na rovnomerné časti, z ktorých sa vytvára sieť. Pre každú z týchto sietí sa vypočíta logaritmicky zhlukovaná distribúcia stupňov vrcholov a následne sa určí hodnota exponentu γ. Výsledkom je graf, ktorý zobrazuje hodnotu exponentu γ v závislosti od počtu vrcholov v sieti, zobrazený na obrázku č. 3.3.

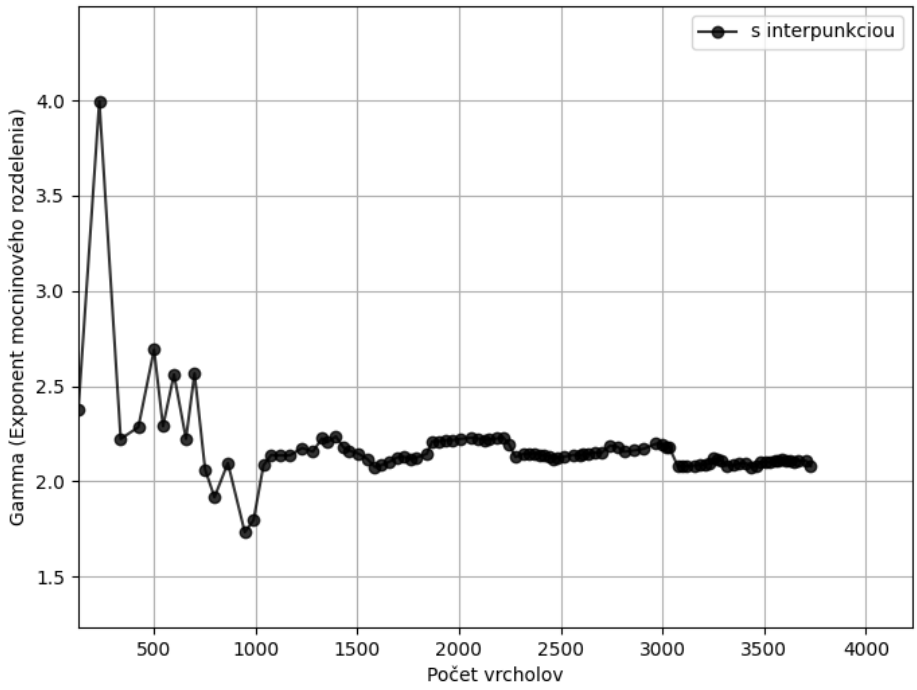
3.5 Výpočet analýzy siete

Výpočet analýzy siete je rozdelený do dvoch oblastí. Prvá, grafická analýza je implementovaná pomocou zadefinovaných metód v knižníci NetworkX. Druhá, jazyková analýza je implementovaná pomocou vlastných metód, ktoré sú špecifické pre danú aplikáciu.

3.6 Uloženie siete do súboru

Jednou z funkcionalít aplikácie je uloženie vytvorenej slovnej siete do súboru. Na tento účel sa využíva knižnica NetworkX, ktorá poskytuje metódy na exportovanie grafov do rôznych formátov. V tejto aplikácii sa využíva formát GraphML, ktorý je vhodný pre externé aplikácie na analýzu a vizualizáciu grafov, ako Cytoscape [16] alebo Gephi [17].

Uloženie siete do súboru je implementované v metóde `saveNetwork`, ktorú možno vidieť na ukážke č. 3.5. Táto metóda sa volá po stlačení tlačidla "Save Network"v GUI,



Obr. 3.3: Rastová charakteristika, hodnota exponentu γ v závislosti od počtu vrcholov v sieti.

ktontroluje či je vybraná interpunkcia a následne uloží sieť do súboru s názvom, ktorý je odvodený od názvu vstupného textového súboru a vypíše miesto uloženia do dialógového okna.

```
1    def saveNetwork(self, event):
2        self.logMessage(self.collectInputDataInfo())
3        self.logMessage('Saving network...\n')
4
5        N = nx.Graph(self.createGraphData(
6                self.processTextFile(self.selectedPunctuation))[0])
7        if self.selectedPunctuation:
8            graphName = f'{self.labelFileSelectPath.GetValue().
                split('/')[-1].split('.')[0]}_graphYesPunct.graphml'
9            self.logMessage(f'Saving graph to {graphName}')
10           nx.write_graphml(N, f'{graphName}')
11       else:
12           graphName = f'{self.labelFileSelectPath.GetValue().
                split('/')[-1].split('.')[0]}_graphNoPunct.graphml'
13           self.logMessage(f'Saving graph to {graphName}')
14           nx.write_graphml(N, graphName)
```

Kód 3.5: Uloženie siete do súboru.

3.7 Testovanie

Testovanie aplikácie bolo vykonané v niekoľkych fázach počas vývoja. Cieľom bolo zabezpečiť správnu funkcionalitu jednotlivých komponentov a výstupov aplikácie. Testovanie prebiehalo prevažne manuálne s využitím jednoduchých testovacích skriptov a simulácií rôznych vstupných dát.

3.7.1 Testovanie predspracovania textu a vytvorenia slovnej siete

Prvou fázou testovania bolo testovanie korektnosti a funkčnosti predspracovania textu a vytvorenia slovnej siete. Pomocou jednoduchých testovacích skriptov a simulácie rôznych vstupných dát sa overovalo:

- Správnosť predspracovania textu (rozdelenie textu na slová)
- Správnosť zohľadnenia interpunkčných znakov pri predspracovaní textu (ak boli zvolené interpunkčné znaky)
- Správnosť vytvorenia slovnej siete
- Správnosť počtu vrcholov a hrán v sieti
- Správnosť reprezentácie siete pomocou knižnice NetworkX

3.7.2 Testovanie grafického používateľského rozhrania

Funkcionalita grafického používateľského rozhrania bola testovaná manuálne. Overovala sa:

- Funkčnosť tlačidiel (načítanie textového súboru, uloženie siete, zobrazovanie a výpočet analýz)
- Funkčnosť rozbaľovacej ponuky pre výber jazyka
- Funkčnosť zaškrtávacích okien pre výber interpunkčných znakov
- Funkčnosť dialógového okna (výpis informácií o aktuálnom stave aplikácie)

3.7.3 Testovanie analýzy siete

Testovanie vizualizácie charakteristík a výpočtovej analýzy siete bolo vykonané manúalne a na rôznych textoch a jazykoch. Overovalo sa:

- Správnosť výpočtu základných grafových veličín siete (overenie správnosti výstupov pomocou externého nástroja `Cytoscape` [16])
- Správnosť výpočtu jazykových charakteristík siete (overenie správnosti výstupov na simulovaných textoch)
- Správnosť zobrazenia distribúcie stupňov vrcholov
- Správnosť zobrazenia rastovej analýzy siete
- Konzistentnosť výsledkov pri rôznych vstupoch
- Odozva pri rozdielnych veľkostiach vstupných textov

3.8 Možnosti budúceho rozšírenia

V tejto časti sa zameriame na potenciálne možnosti vylepšenia a rozšírenia aplikácie, ktoré by mohli zlepšiť funkcionalitu a použiteľnosť aplikácie pri analýze textových dát. Tieto vylepšenia by mohli zahŕňať:

- Podporovanie viacerých jazykov a textových formátov
- Načítanie slovnej siete zo súboru
- Možnosť manuálneho nastavenia parametrov pri charakteristikách, ako je veľkosť výseku vstupného textu alebo dĺžka najdlhšej klesajúcej sekvencie vrcholov
- Manuálne nastavenie parametrov pre logaritmické zhlukovanie, aby používateľ mohol získať presnejšie výsledky pre konkrétne texty
- Výber veličín, ktoré sa majú vypočítať a zobraziť pri analýze siete
- Možnosť porovnávať viacero rôznych sietí naraz a zobraziť ich výsledky v jednom grafe
- Možnosť uloženia siete, grafov v rôznych formátoch

Kapitola 4

Analýza textov

Výskum sa zameriava na literárne diela dvoch autorov, Charlesa Dickensa a Oscara Wildea v dvoch jazykových variáciách: anglická a nemecká. Cieľom je preskúmať vplyv interpunkcie na pozičnú slovnú sieť a jej mocninové rozdelenie. Kapitola je rozdelená do štyroch hlavných častí: vývoj slovnej siete, distribúcia stupňov vrcholov, grafová analýza a jazyková analýza.

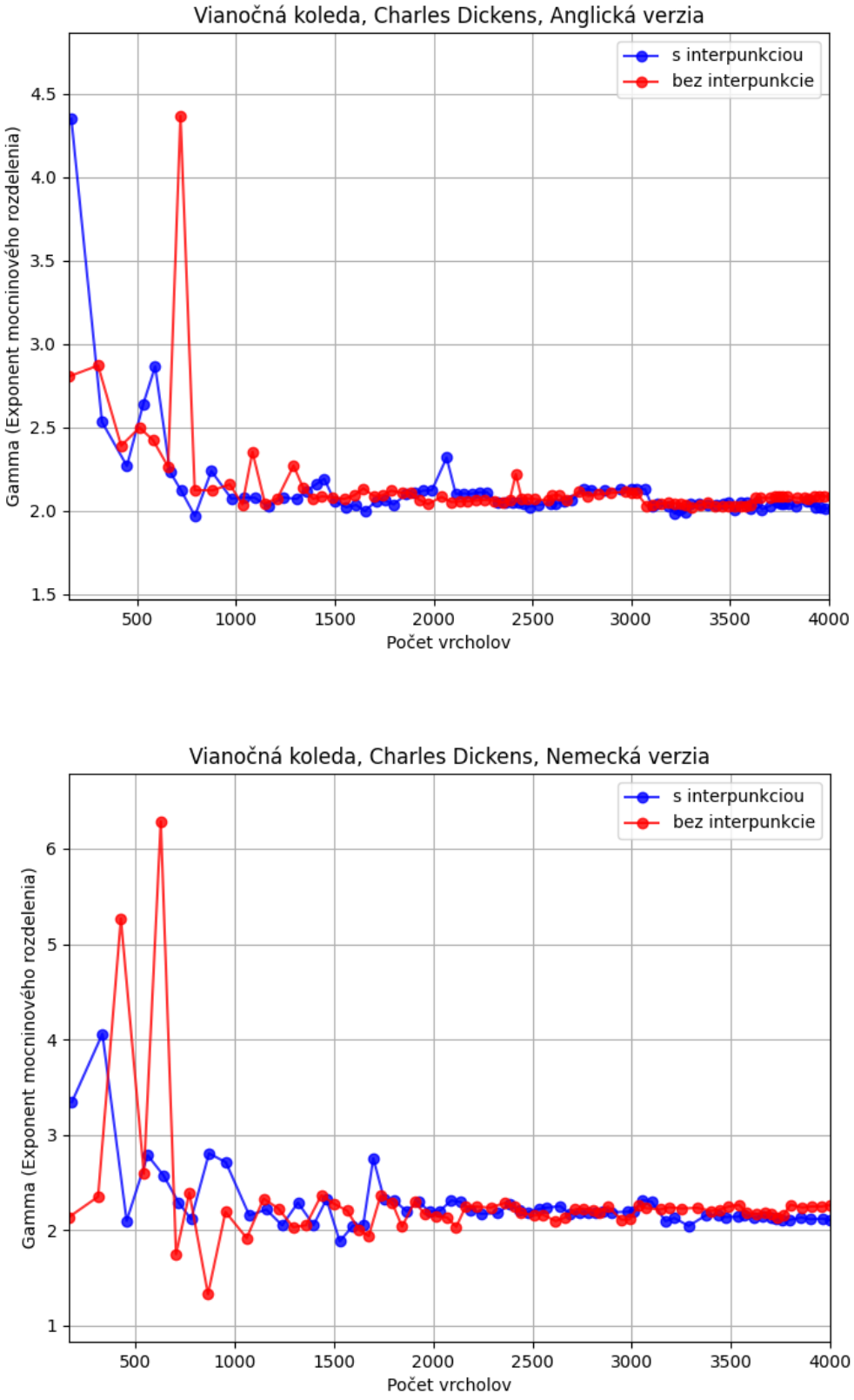
4.1 Vývoj slovnej siete

Ako je možné vidieť na uvedených grafoch obr. č. 4.1, č. 4.2 a č. 4.3, veľkosť exponentu mocninového rozdelenia γ sa mení v závislosti od veľkosti siete. Pozorovanie vývoja slovných sietí vytvorených z literárnych diel v rôznych jazykoch ukázalo, že rozdiel medzi slovnou sieťou bez interpunkcie a slovnou sieťou, ktorá obsahuje interpunkciu je najmä v počiatočnom štádiu vývoja siete.

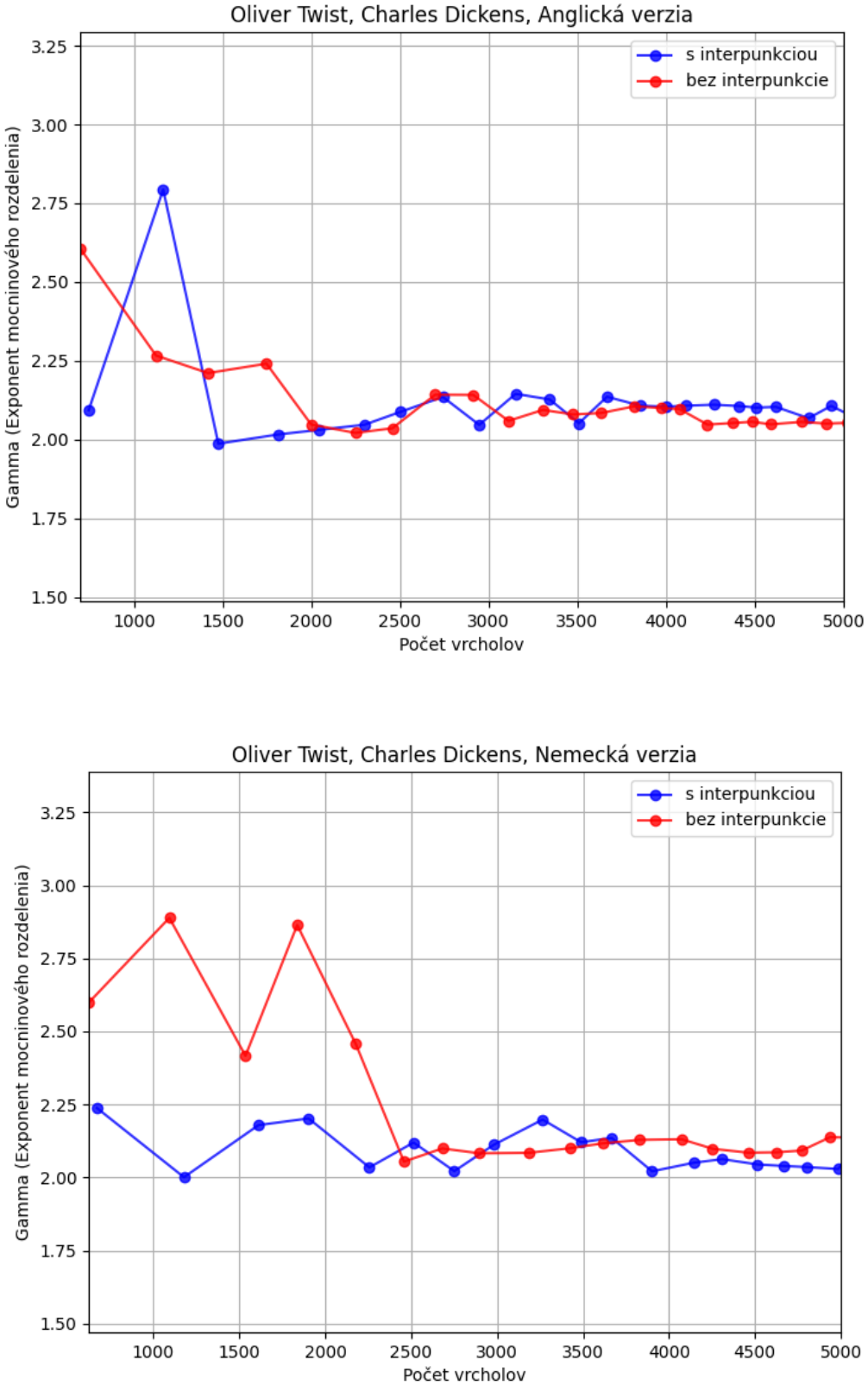
V prípade Charlesa Dickensa sme analyzovali dve literárne diela, Vianočnú koledu a Olivera Twista. V oboch textoch, ako v anglickej, tak aj nemeckej verzii boli najvýraznejšie zmeny exponentu γ pozorované pri raste siete približne do veľkosti 1000 až 2000 vrcholov. Po prekročení tejto hranice sa hodnota exponentu postupne stabilizuje a pohybuje sa v rozpätí 2 až 2.2.

Dielo Portrét Doriana Graya od Oscara Wildea bola analyzovaná rovnako v anglickej a nemeckej verzii. Aj v tomto prípade nastala najväčšia zmena hodnoty exponentu nastala pri raste siete do veľkosti 1500 až 2000 vrcholov. Následne sa hodnota exponentu postupne stabilizovala a pohybovala sa v rovnakom rozpätí 2 až 2.2.

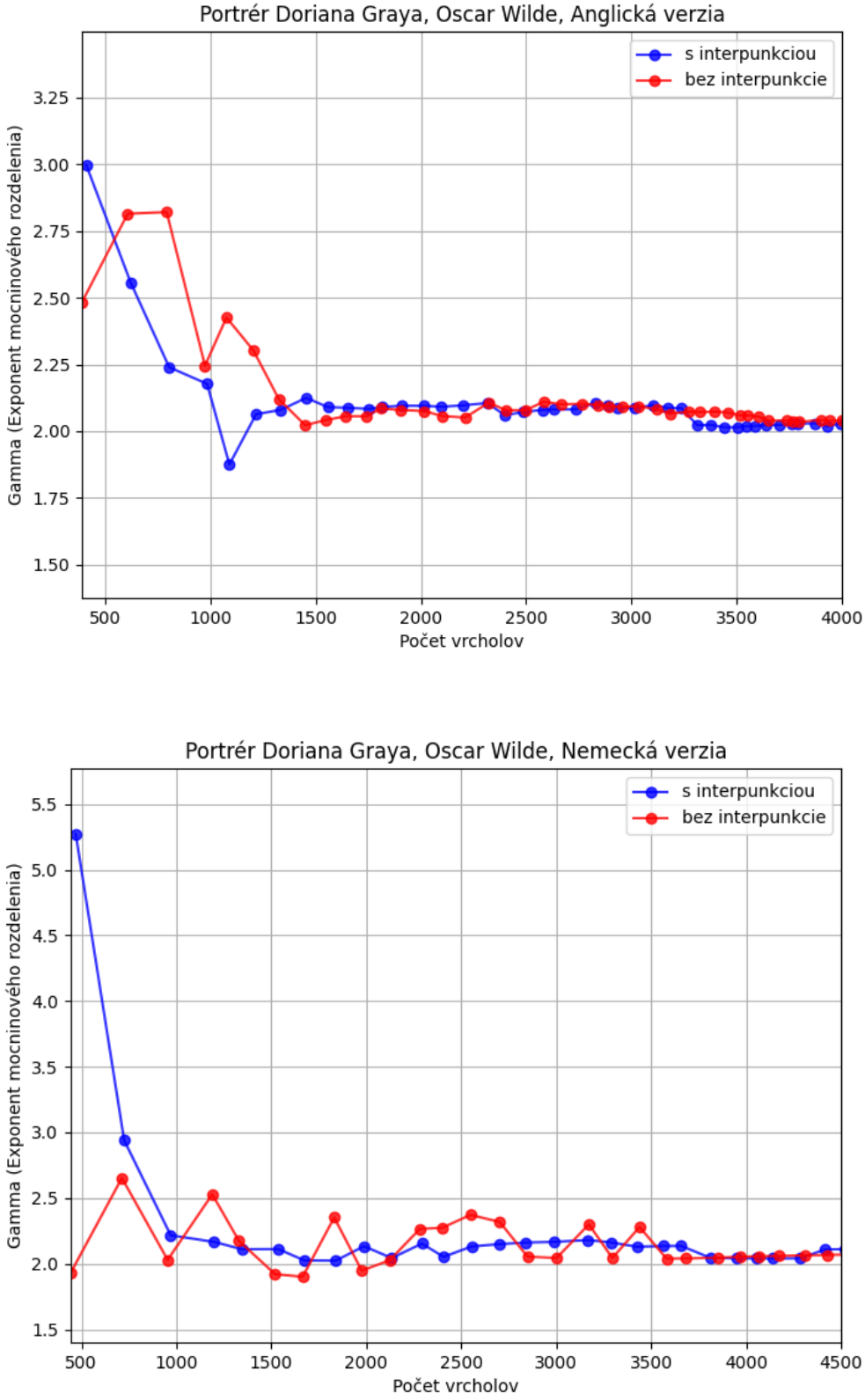
Na základe tejto analýzy boli identifikované dostatočne dlhé úseky textov, ktoré slúžili ako vstupné dáta pre nasledujúce analýzy.



Obr. 4.1: Zmena veľkosti exponentu γ v závisloti od veľkosti siete, text Vianočná koleda od Charlesa Dickensa.



Obr. 4.2: Zmena veľkosti exponentu γ v závisloti od veľkosti siete, text Oliver Twist od Charlesa Dickensa.



Obr. 4.3: Zmena veľkosti exponentu γ v závisloti od veľkosti siete, text Portrét Doriana Graya od Oscara Wildea.

4.2 Grafová analýza

Grafová analýza textu pre slovnú sieť bez interpunkcie a so zohľadnením interpunkcie bola vykonaná na približne rovnako veľkých slovných sieťach. Vo výsledkoch (tabuľka č. 4.1, č. 4.2, č. 4.3 a č. 4.4) jednotlivých analýz môžeme pozorovať niekoľko zmien.

Ako si môžeme všimnúť, rozdiely sa opakovane prejavujú nezávisle od konkrétneho textu a jazyka. Rozdiely medzi slovnými sieťami bez interpunkcie a so zohľadnením interpunkcie majú konzistentný charakter, pričom sa opakovane prejavuje rovnaký trend zmien v sledovaných parametroch.

Počet vrcholov a počet hrán sa v prípade oboch sietí mení len minimálne, zatiaľ čo maximálny stupeň v slovnej sieti s interpunkciou je výrazne vyšší, pretože interpunkčné znamienka sú v porovnaní so slovami veľmi časté a vytvárajú tak väčší počet hrán. Napríklad čiarka alebo bodka sa môže spájať s mnohými slovami, čím vznikajú uzly s vysokým stupňom. Vďaka tomu sa zvyšuje aj priemerný stupeň, ktorý je v prípade slovnej siete s interpunkciou vyšší, okrem textu Portrét Doriana Graya v angličtine, kde sa priemerný stupeň znížil.

Priemerný koeficient zhlukovania je v prípade slovnej siete s interpunkciou vyšší, čo naznačuje, že uzly sú v tejto sieti viac zoskupené do tzv. hubov. To vplýva aj na priemernú najkratšiu cestu, ktorá je v prípade slovnej siete s interpunkciou nižšia.

Ďalšie zaujímavé pozorovania sa týkajú korelácie medzi rôznymi typmi centrality:

- Korelácia medzi stupňom a blízkosťou pohybujúca sa v intervale 0.34 až 0.50, ktorá udáva, ako blízko sú jednotlivé uzly. V prípade slovnej siete bez interpunkcie je vyššia.

- Korelácia medzi stupňom a medziľahlosťou je v oboch prípadoch veľmi vysoká v rozsahu 0.93 až 0.96 s minimálnou zmenou a naznačuje, že uzly s vysokým stupňom sú často aj uzlami, ktoré prepájajú rôzne časti siete.

- Korelácia medzi blízkosťou a medziľahlosťou je v prípade slovnej siete bez interpunkcie opäť vyššia, no vo všetkých prípadoch sa pohybuje v rozmedzí 0.23 až 0.37, čo naznačuje, že uzly majú slabú pozitívnu koreláciu medzi týmito dvoma typmi centrality. Tento výlsedok naznačuje, že uzly, ktoré sú blízko seba, nemusia zohrávať významnú úlohu pri tvorbe najkratších ciest v sieti.

Celkovo grafová analýza ukazuje, že interpunkcia má vplyv na štruktúru slovnej siete, čo sa prejave v rôznych parametroch siete, ako sú stupeň vrcholov, koeficient zhlukovania, priemerná najkratšia cesta a korelácie medzi rôznymi typmi centrality.

Grafová analýza, Anglický jazyk, bez interpunkcie			
Text	Vianočná koleda ∼21000 slov	Oliver Twist ∼22000 slov	Portrét Doriana Graya ∼29000 slov
Počet vrcholov	3725	3842	3751
Počet hrán	14164	14762	17315
Maximálny stupeň	942	961	942
Minimálny stupeň	1	1	1
Priemerný stupeň	7.60483	7.68454	9.23220
Hustota siete	0.00204	0.00200	0.00246
Korelácia centrality (stupeň, blízkosť)	0.43094	0.42376	0.48416
Korelácia centrality (stupeň, medziľahlosť)	0.95007	0.95728	0.93399
Korelácia centrality (blízkosť, medziľahlosť)	0.29077	0.29174	0.31290
Priemerný koeficient zhlukovania	0.35024	0.33708	0.34747
Priemerná najkratšia cesta	2.91347	2.91189	2.86155
Priemer	7	8	7

Tabuľka 4.1: Grafová analýza, Anglický jazyk, bez interpunkcie

Grafová analýza, Anglický jazyk, s interpunkciou			
Text	Vianočná koleda ∼21000 slov	Oliver Twist ∼22000 slov	Portrét Doriana Graya ∼29000 slov
Počet vrcholov	3737	3855	3761
Počet hrán	14389	14820	16934
Maximálny stupeň	1339	1215	1214
Minimálny stupeň	1	1	1
Priemerný stupeň	7.70083	7.68872	9.00505
Hustota siete	0.00206	0.00199	0.00239
Korelácia centrality (stupeň, blízkosť)	0.41137	0.40580	0.42678
Korelácia centrality (stupeň, medziľahlosť)	0.95424	0.96189	0.95068
Korelácia centrality (blízkosť, medziľahlosť)	0.27621	0.28199	0.27787
Priemerný koeficient zhlukovania	0.44544	0.41362	0.44917
Priemerná najkratšia cesta	2.77038	2.80452	2.73230
Priemer	6	6	7

Tabuľka 4.2: Grafová analýza, Anglický jazyk, s interpunkciou

Grafová analýza, Nemecký jazyk, bez interpunkcie			
Text	Vianočná koleda ∼16000 slov	Oliver Twist ∼16000 slov	Portrét Doriana Graya ∼17000 slov
Počet vrcholov	3745	4133	3802
Počet hrán	12037	12287	12631
Maximálny stupeň	771	731	577
Minimálny stupeň	1	1	1
Priemerný stupeň	6.42830	5.94580	6.64440
Hustota siete	0.00172	0.00144	0.00175
Korelácia centrality (stupeň, blízkosť)	0.44845	0.43471	0.50290
Korelácia centrality (stupeň, medziľahlosť)	0.94554	0.95769	0.94727
Korelácia centrality (blízkosť, medziľahlosť)	0.31412	0.31849	0.36585
Priemerný koeficient zhlukovania	0.21771	0.19410	0.22756
Priemerná najkratšia cesta	3.20136	3.25713	3.19821
Priemer	9	9	7

Tabuľka 4.3: Grafová analýza, Nemecký jazyk, bez interpunkcie

Grafová analýza, Nemecký jazyk, s interpunkciou			
Text	Vianočná koleda ∼16000 slov	Oliver Twist ∼16000 slov	Portrét Doriana Graya ∼17000 slov
Počet vrcholov	3757	4145	3811
Počet hrán	12177	12546	12710
Maximálny stupeň	1271	1239	1060
Minimálny stupeň	1	1	1
Priemerný stupeň	6.48230	6.05356	6.67017
Hustota siete	0.00173	0.00146	0.00175
Korelácia centrality (stupeň, blízkosť)	0.34772	0.35362	0.39363
Korelácia centrality (stupeň, medziľahlosť)	0.95683	0.96393	0.95394
Korelácia centrality (blízkosť, medziľahlosť)	0.23112	0.24623	0.26092
Priemerný koeficient zhlukovania	0.34032	0.30007	0.31879
Priemerná najkratšia cesta	2.95941	3.03028	2.99317
Priemer	7	7	7

Tabuľka 4.4: Grafová analýza, Nemecký jazyk, s interpunkciou

4.3 Distribúcia stupňov vrcholov

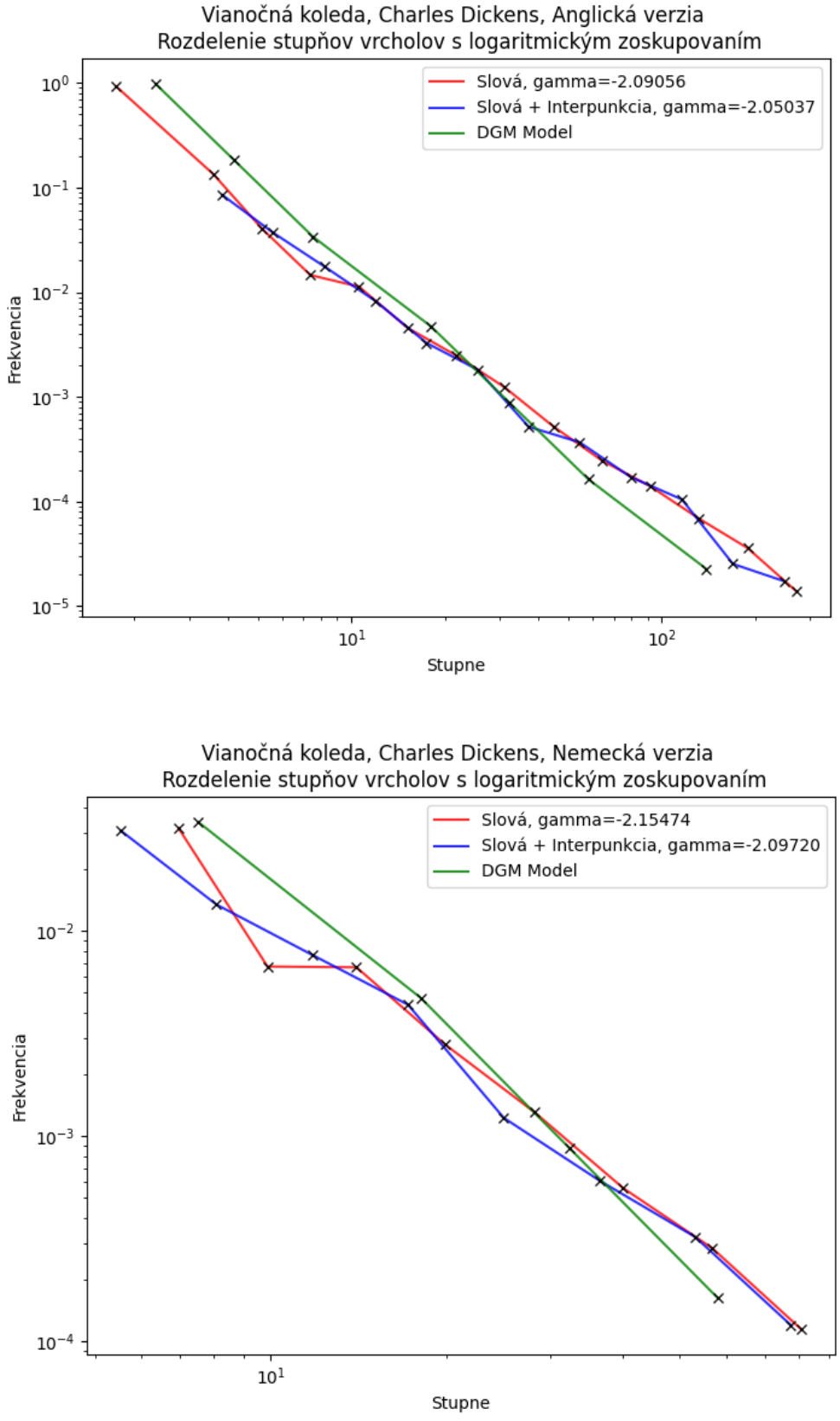
Distribúcia stupňov vrcholov bola analyzovaná pre úryvky textov, ktoré boli vybrané na základe vývoja slovnej siete 4.1. Skúmal sa vplyv zahrnutia interpunkcie do slovnej siete na mocninové rozdelenie stupňov vrcholov, zmenu exponentu γ a podobnosť rozdelenia stupňov vrcholov k teoretickému Dorogovtsev-Mendes modelu.

Výsledky sú zobrazené na obr. č. 4.4, č. 4.5 a č. 4.6 v podobe grafov zobrazujúcich rozdelenie stupňov vrcholov s využitím logaritmického zoskupovania v log-log mierke. Na každom z týchto grafov je osobitne zobrazené rozdelenie pre sieť bez interpunkcie, s interpunkciou a teoretické rozdelenie Dorogovtsev-Mendes modelu. Pre analýzu bolo potrebné odfiltrovať uzly s veľmi nízkym a vysokým stupňom, aby sa znížilo skreslenie výsledkov a vybrať najdlhšiu klesajúcu časť charakteristiky pre výpočet hodnoty exponentu γ mocninové rozdelenia.

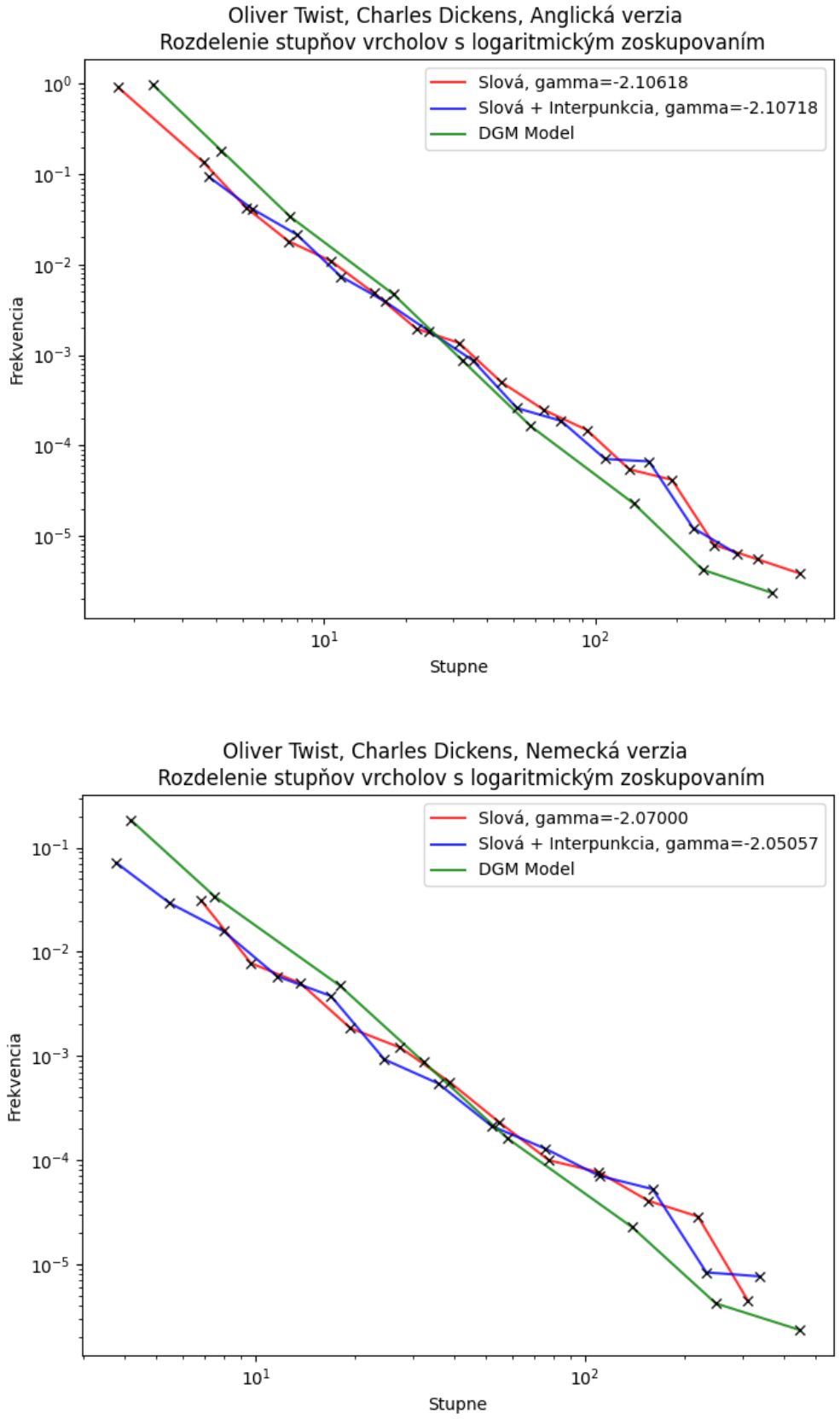
Analýza ukázala, že slovné siete bez interpunkcie majú exponent γ v rozmedzí 2.03 až 2.16 a slovné siete so zohľadnením interpunkcie majú exponent γ v intervale 2.02 až 2.11. Hodnoty γ udávajú, že rozdelenie stupňov vrcholov v oboch prípadoch nasleduje mocninové rozdelenie, pričom hodnoty sa pohybujú v blízkosti 2, čo je typické pre reálne siete. Taktiež dôvod prečo sa hodnoty exponentu γ blížia k 2 je ten, že v prípade slovných sietí vznikajú uzly s vysokým stupňom, zapríčinené častým výskytom spojok alebo predložiek, ktoré sa často spájajú s inými slovami. Môžeme pozorovať, že v prípade slovnej siete s interpunkciou je hodnota exponentu γ nižšia vo všetkých prípadoch, pretože interpunkčné znamienka majú tendenciu vytvárať uzly s vysokým stupňom.

Porovnanie s teoretickým Dorogovtsev-Mendes modelom sa uskutočnilo na základe hodnoty strednej kvadratickej chyby normalizovanej distribúcie stupňov vrcholov, ktorá sa pohybuje v rozmedzí 0.00022 až 0.00028 pre sieť bez interpunkcie a 0.00015 až 0.00021 pre sieť s interpunkciou.

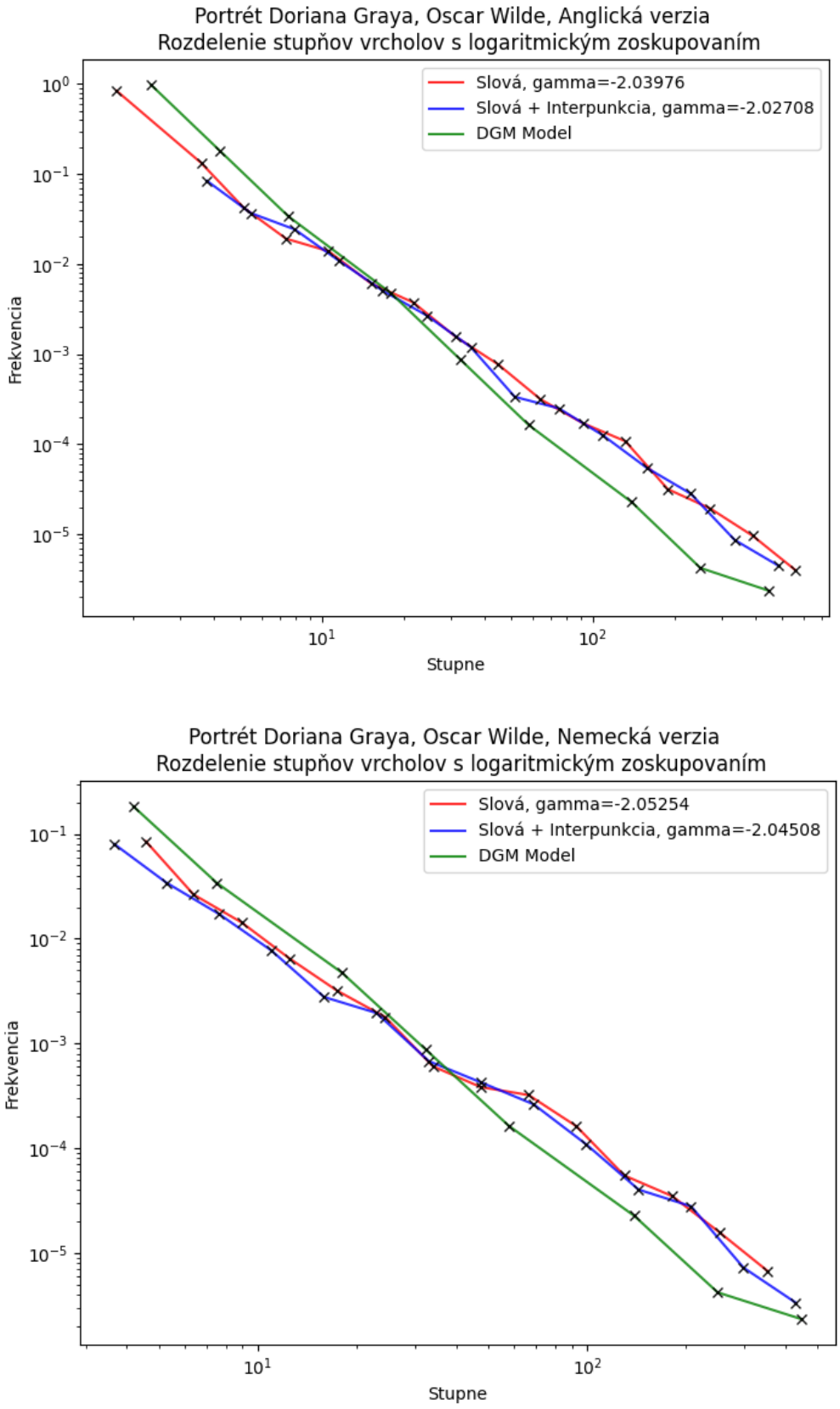
Zistenia potvrdzujú, že rozdelenie stupňov vrcholov v slovných sieťach s interpunkciou aj bez nej vykazuje charakteristiky bezškálových sietí a výraznú podobnosť s teoretickým Dorogovtsev-Mendes modelom.



Obr. 4.4: Distribúcia stupňov vrcholov s využitím logaritmického zoskupovania, text Vianočná koleda od Charlesa Dickensa.



Obr. 4.5: Distribúcia stupňov vrcholov s využitím logaritmického zoskupovania, text Oliver Twist od Charlesa Dickensa.



Obr. 4.6: Distribúcia stupňov vrcholov s využitím logaritmického zoskupovania, text Portrét Doriana Graya od Oscara Wildea.

4.4 Jazyková analýza

Jazyková analýza jednotlivých diel bola vykonaná na slovných sieťach so zohľadnením interpunkcie. V tabuľkách č. 4.5 a č. 4.6 sú uvedené výsledky jazykovej analýzy pre anglický a nemecký jazyk. Možno pozorovať viaceré rozdiely, ktoré sú spôsobené štruktúrou jazykov a štýlom prekladu.

Počet interpunkčných znakov sa v jednotlivých dielach výrazne líši, aj napriek tomu, že analyzované slovné siete sú približne rovnakej veľkosti. Tento rozdiel je zapríčinený tým, že na vytvorenie slovnej siete určitej veľkosti je potrebné odlišné množstvo textu. Množstvo textu je závislé od štruktúry samostatného jazyka a štýlom písania autora či prekladateľa. Ak si vypočítame pomer medzi veľkosťou vzorky textu a počtom interpunkčných znakov, zistíme, že v prípade anglického jazyka sa tento pomer pohybuje v intervale 4.8 až 5.1 a v prípade nemeckého jazyka 5.2 až 5.4. Tento pomer ukazuje, že vo všetkých dielach v oboch jazykoch je počet interpunkčných znakov v priemere približne päťkrát menší, ako počet slov.

Priemerná dĺžka slova je vyššia pre všetky analýzy v nemeckom jazyku (okolo 7.5 až 7.9 znakov), čo je typické pre nemecký jazyk, v ktorom sa často vyskytujú dlhé zložené slová. V anglickom jazyku sa priemerná dĺžka slova pohybuje v rozmedzí 6.4 až 6.7 znakov. Rovnaký trend je pozorovaný aj v prípade maximálnej dĺžky slova, ktorá sa v nemeckom jazyku pohybuje v rozmedzí 23 až 26 znakov, zatiaľ čo v anglickom jazyku sa maximálna dĺžka slova pohybuje v intervale 15 až 16 znakov.

Štatistiky týkajúce sa dĺžky viet vykazujú podobné hodnoty, jediný výrazný rozdiel je v maximálnej dĺžke vety textu Oliver Twist, kde je v anglickom jazyku viac ako dvojnásobná v porovnaní s nemeckým prekladom. Tento rozdiel môže byť spôsobený viacerými faktormi, ako sú štruktúra jazyka či preklad textu.

Taktiež sa skúmal výskyt dvojíc a trojíc slov v textoch, ktoré tvoria časté slovné spojenia. Táto analýza silno závisí od štýlu písania autora, štruktúry a veľkosti textu, ako vidno v anglickej verzii Portrétu Doriana Graya, kde sa vyskytuje najviac dvojíc a trojíc slov. Dôvodom môže byť aj veľkosť textu, ktorý je v porovnaní s ostatnými anglickými textami o takmer 30% väčší. Veľkú úlohu zohráva aj gramatická štruktúra jazyka, ktorá ovplyvňuje výskyt dvojíc a trojíc slov, pretože pravidlá písania napríklad čiarky môže vytvárať dvojice, trojice uzlov s vysokou frekvenciou.

Tieto zistenia naznačujú, že jazyková analýza môže poskytnúť cenné informácie o štruktúre a charakteristikách textu, ako aj o štýle písania autora.

Jazyková analýza, Anglický jazyk			
Veľkosť textu (slová)	Vianočná koleda ∼21000 slov	Oliver Twist ∼22000 slov	Portrét Doriana Graya ∼29000 slov
Počet interpunkčných znamienok	4377	4350	5782
Najväčší výskyt interpunkčného znamienka	2055	2123	2302
Najmenší výskyt interpunkčného znamienka	47	38	16
Počet unikátnych slov	3737	3855	3761
Maximálna dĺžka slova	15	15	16
Minimálna dĺžka slova	1	1	1
Priemerná dĺžka slova	6.45063	6.73281	6.56076
Počet viet	1329	1425	2612
Maximálna dĺžka vety	136	209	124
Minimálna dĺžka vety	1	1	1
Priemerná dĺžka vety	15.99473	15.80561	11.38055
Počet dvojíc slov	15057	15363	17700
Maximálna frekvencia dvojice	400	309	618
Minimálna frekvencia dvojice	1	1	1
Priemerná frekvencia dvojice	1.78409	1.83616	2.09960
Počet trojíc slov	22956	23647	29482
Maximálna frekvencia trojice	73	119	256
Minimálna frekvencia trojice	1	1	1
Priemerná frekvencia trojice	1.17015	1.19288	1.26050

Tabuľka 4.5: Jazyková analýza, Anglický jazyk

Jazyková analýza, Nemecký jazyk			
Veľkosť textu (slová)	Vianočná koleda ∼16000 slov	Oliver Twist ∼16000 slov	Portrét Doriana Graya ∼17000 slov
Počet interpunkčných znamienok	2983	2973	3224
Najväčší výskyt Interpunkčného znamienka	1703	1606	1580
Najmenší výskyt Interpunkčného znamienka	8	24	25
Počet unikátnych slov	3757	4145	3811
Maximálna dĺžka slova	23	26	24
Minimálna dĺžka slova	1	1	1
Priemerná dĺžka slova	7.58957	7.91870	7.89766
Počet viet	1035	1041	1417
Maximálna dĺžka vety	126	90	114
Minimálna dĺžka vety	1	1	1
Priemerná dĺžka vety	15.49758	15.25456	11.93719
Počet dvojíc slov	12703	13039	13229
Maximálna frekvencia dvojice	188	163	204
Minimálna frekvencia dvojice	1	1	1
Priemerná frekvencia dvojice	1.56050	1.50732	1.58644
Počet trojíc slov	17742	17706	18599
Maximálna frekvencia trojice	87	53	155
Minimálna frekvencia trojice	1	1	1
Priemerná frekvencia trojice	1.11724	1.10996	1.12834

Tabuľka 4.6: Jazyková analýza, Nemecký jazyk

Záver

Put your conclusion here.

Literatúra

[1] Reinhard Diestel. *Graph Theory*. Springer, Berlin, Germany, 5th edition, 2017.

[2] Mária Markošová. Dynamika sietí. *Umelá inteligencia a kognitívna veda II*, pages 321–379, 2010.

[3] Albert-László Barabási. *Network Science*. Cambridge University Press, Cambridge, UK, 2016.

[4] A. Rényi P. Erdős. On random graphs. *I. Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[5] Edgar N Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

[6] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[7] Akhlaq Ahmad, Mohamed Ridza Wahiddin, Abdur Rahman, Imad Afyouni, Bilal Sadiq, Faizan Rehman, and Sohaib Ghani. Scale free network analysis of a large crowd through their spatio-temporal activities. 12 2015.

[8] Sergey N Dorogovtsev and José FF Mendes. Evolution of networks. *Advances in Physics*, 51(4):1079–1187, 2002.

[9] Adilson E Motter, Alessandro PS De Moura, Ying-Cheng Lai, and Partha Dasgupta. Topology of the conceptual network of language. *Physical Review E*, 65(6):065102, 2002.

[10] Sergey N Dorogovtsev and José Fernando F Mendes. Language as an evolving word web. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1485):2603–2606, 2001.

[11] Ramon Ferrer I Cancho and Richard V Solé. The small world of human language. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1482):2261–2265, 2001.

[12] Yani Lubis, Fardhan Syahri, Nur Annisa Ameliya, Zaskia Amanda Putri, and Ari Rajasa Tampubolon. Mastering of punctuation marks. *Jurnal Penelitian Ilmiah Multidisiplin*, 9(1), 2025.

[13] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.

[14] Project gutenberg. `https://www.gutenberg.org/`. Navštívené dňa 9.5.2025.

[15] Staša Milojević. Power law distributions in information science: Making the case for logarithmic binning. *Journal of the American Society for Information Science and Technology*, 61(12):2417–2425, 2010.

[16] Cytoscape Consortium. Cytoscape: An open source platform for complex network analysis and visualization. `https://cytoscape.org`. Navštívené dňa 9.5.2025.

[17] Gephi Consortium. Gephi: The open graph viz platform. `https://gephi.org`. Navštívené dňa 9.5.2025.