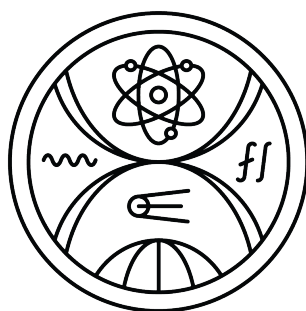


UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

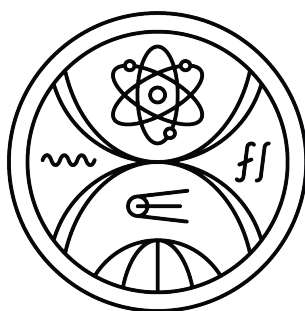


MODEL DISTRIBÚCIE INTERPUNKČNÝCH
ZNAMIENOK V RÔZNYCH TEXTOCH
DIPLOMOVÁ PRÁCA

2025

BC. MARTIN HOŠEK

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



MODEL DISTRIBÚCIE INTERPUNKČNÝCH ZNAMIENOK V RÔZNYCH TEXTOCH

DIPLOMOVÁ PRÁCA

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

Bratislava, 2025
Bc. Martin hošek



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Martin Hošek
Študijný program: aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Model distribúcie interpunkčných znamienok v rôznych textoch
Model of punctuation mark distributions in various texts

Anotácia: Študent naprogramuje, alebo použije už naprogramovanú aplikáciu na hľadanie distribúcie interpunkčných znamienok v textoch v rôznych jazykoch. Ak študent použije hotovú aplikáciu, musí k nej doprogramovať v šetko potrebné na analýzu distribúcie a na tvorbu jej modelu. Študent zanalyzuje distribúcie z textov toho istého autora v rôznych časových obdobiach aby zistil, ako sa táto vlastnosť textu mení a nakoľko ostáva stabilná. Takisto zanalyzuje tieto distribúcie v textoch ľudí s afáziou. Navrhne matematický model tejto distribúcie.

Cieľ: = naprogramovať aplikáciu, alebo doprogramovať potrebné nástroje do aplikácie na analýzu distribúcií interpunkčných znamienok v textoch a vytvoriť matematický model opisujúci získané výsledky

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: 23.02.2024

Dátum schválenia: 21.11.2024
prof. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Pod'akovanie: Tu môžete poďakovať školiteľovi, prípadne ďalším osobám, ktoré vám s prácou nejako pomohli, poradili, poskytli dáta a podobne.

Abstrakt

V tejto diplomovej práci analyzujeme jazyk ako komplexnú sieť, pričom texty reprezentujeme ako grafy, kde vrcholy zodpovedajú entitám (tokenom) a hrany vyjadrujú susednosti týchto tokenov v texte. Zameriavame sa na význam interpunkčných znamienok, ktoré zohrávajú nezanedbateľnú úlohu pri štruktúrovaní a interpretácii textu. Študujeme distribúciu interpunkčných znamienok v rôznych typoch textov, predovšetkým v naratívnych textoch. Vytvárame modely týchto distribúcií a analyzujeme štruktúru grafov vrátane počtu vrcholov, hrán a stupňov vrcholov. Ďalej porovnávame jednotlivé texty so základnými zákonmi jazyka, ako sú Zipfov a Heapsov zákon, a s vybranými existujúcimi modelmi. Nakoniec skúmame rozdiely medzi textami s interpunkčnými znamienkami a bez nich, pričom identifikujeme ich vplyv na štruktúru a vlastnosti jazykovej siete.

Kľúčové slová: distribúcia interpunkčných znamienok, frekvencia slov, interpunkčné znamienka, zipov zákon

Abstract

In this thesis, we analyze language as a complex network, representing texts as graphs where nodes correspond to entities (tokens) and edges reflect the adjacency of these tokens within the text. We focus on the role of punctuation marks, which play a significant role in structuring and interpreting texts. We study the distribution of punctuation marks across different types of texts, primarily narrative texts. We create models of these distributions and analyze graph structures, including the number of nodes, edges, and node degrees. Furthermore, we compare individual texts with fundamental language laws, such as Zipf's and Heaps' laws, and with selected existing models. Finally, we examine the differences between texts with and without punctuation marks, identifying their impact on the structure and properties of the language network.

Keywords: distribution of punctuation marks, punctuation marks, word frequency, zipf's law

Obsah

Úvod	1
1 Dynamika sietí	3
1.1 Pojmy a definície	3
1.2 Matematické modely sietí	3
1.3 Spôsoby rastu sietí	3
1.3.1 Náhodné pripájanie	3
1.3.2 Preferenčné pripájanie	3
1.3.3 Zmiešané pripájanie	4
2 Interpunkčné znamienka v naratívnych textoch	5
2.1 Základné modely distribúcie	7
2.1.1 Zipfov zákon	7
2.1.2 Heapsov zákon	7
3 Jazyk ako graf	9
4 Analýza textov	11
4.1 Analýza a model frekvencie v textoch od autora Erich Maria Remarque: Na Západe nič nové	11
4.1.1 Analýza frekvencie tokenov v texte	11
4.1.2 Analýza susednosti slov v texte	12
4.2 Analýza a model frekvencie v textoch od autora Erich Maria Remarque: Cesta späť	12
4.2.1 Zhrnutie analýzy	12
4.3 Interpretácia výsledkov analýzy	13
5 Implementácia programu	15
5.1 Načítanie PDF súboru a extrakcia textu po znakoch	15
5.2 Analýza získaného textu a rozobratie textu na jednotlivé tokeny	16
5.3 Vyrábanie potrebných štruktúr	17
5.4 Interpretácia (grafových) štruktúr	18

5.5	Vizualizácia dát pomocou knižnice XChart	18
5.6	Skúmanie susednosti grafov pomocou knižnice Gephi	19
Záver		21

Úvod

V tejto práci budeme skúmať distribúcie interpunkčných znamienok v rôznych textoch. Naším cieľom je zistiť, aké sú rozdiely v používaní týchto znamienok v rôznych typoch textov a aké faktory môžu ovplyvniť ich výskyt.

Budeme využívať literatúru [?] a [?].

Kapitola 1

Dynamika sietí

V tejto kapitole budeme popisovať dynamiku sietí (konkrétne modely, ktoré popisujú, ako sa siete vyvíjajú a menia v čase). Dynamika sietí je kľúčová pre pochopenie mnohých reálnych systémov, ako napríklad systém jazyka, ktorému sa budeme v tejto práci venovať. Budeme vychádzať z literatúry [?].

1.1 Pojmy a definície

Matematicky model siete je *graf*. Graf G je definovaný ako množina uzlov (vrcholov) V a množina hrán (spojení) E , teda $G = (V, E)$. Uzly môžu predstavovať rôzne entity, napríklad slová v jazyku, a hrany reprezentujú vzťahy medzi týmito entitami, napríklad *spolu-výskyt* slov v texte (ktoré slová sú vedľa seba).

Uzly (objekty, vrcholy) a vzťahy medzi nimi (hrany) sa v čase prirodzene menia, pričom ich počet spravidla narastá, zatiaľ čo ich zánik je len minimálny.

1.2 Matematické modely sietí

1.3 Spôsoby rastu sietí

1.3.1 Náhodné pripájanie

Pri náhodnom pripájaní sa nové uzly pripájajú k existujúcim uzlom s rovnakou pravdepodobnosťou, nezávisle od ich stupňa.

1.3.2 Preferenčné pripájanie

Preferenčné pripájanie (angl. preferential attachment) je mechanizmus, pri ktorom sa nové uzly s väčšou pravdepodobnosťou pripájajú k uzlom s vyšším stupňom.

1.3.3 Zmiešané pripájanie

Zmiešané pripájanie kombinuje prvky náhodného a preferenčného pripájania.

Kapitola 2

Interpunkčné znamienka v naratívnych textoch

Prirodzený jazyk je jedným z najvýraznejších príkladov zložitých systémov. V prirodzenom jazyku relatívne malý počet elementárnych prvkov (*foném*¹ a *písmen*²) umožňuje vytvárať zložitejšie jednotky: *slová*. Tie odkazujú na všetko, čo môže človek pomenovať a popísať. Slová však samé o sebe netvoria celú podstatu jazyka, tu je potrebná ďalšia zložená entita: *veta*.

Vetná štruktúra je štandardnou vlastnosťou takmer všetkých písaných jazykov. Práve na tejto úrovni sa vytvára *sémantika* (význam) v celej jej bohatosti a s rôznymi nositeľmi: slová, syntax, frázy, vedľajšie vety a interpunkcia v písanom jazyku.

Štatistické analýzy jazykových vzoriek, ktoré sa vykonávajú viac ako storočie [6, 22], potvrdili existenciu zákonov opisujúcich jazyk kvantitatívne. Klasické štatistické štúdie zahŕňajú *empirické rozdelenie*³ frekvencie slov porovnávané s mocninovým zákonom známym ako Zipfov zákon [23] a funkčný vzťah medzi dĺžkou textu a počtom jedinečných slov, modelovaný Heapsovým zákonom [8, 10, 11]. Relatívne nový prístup predstavuje popis jazyka pomocou sieťového formalizmu [4, 7, 9, 18, 19]. Tento prístup umožňuje analyzovať jazyk ako sieť, v ktorej je súbor uzlov a vzťahov medzi nimi. Uzly predstavujú slová alebo pojmy a hrany reprezentujú napríklad ich spolu-výskyt v texte (teda či su v texte hneď za sebou) alebo významovú súvislosť. Ukázalo sa, že niektoré siete, ktoré reprezentujú lexikálnu štruktúru textov (napr. siete založené na spolu-výskyte slov), patria do triedy scale-free sietí, podobne ako sémantické siete, ktoré sa konštruujú na základe významu slov [1, 2, 16]. To znamená, že väčšina slov má len niekoľko spojení, zatiaľ čo niektoré slová (napr. veľmi časté alebo významovo centrálné) majú spojení veľa, čo je typické pre mocninový vzťah.

Písanie textov vyžaduje použitie *interpunkcie*, inak by niektoré výrazy mohli byť

¹fonéma je najmenšia zvuková jednotka jazyka, ktorá rozlišuje význam slov

²písmeno je základná grafická jednotka jazyka, ktorá reprezentuje fonému

³rozdelenie na základe skúmania, teda bez predpovedania podľa teórie

nejasné a zavádzajúce. Interpunkcia tiež umožňuje vyznačiť oddelené logické jednotky, na ktoré môže byť akákoľvek zložitá informácia v texte rozdelená a ľahšie pochopiteľná. Z tohoto pohľadu nie sú *interpunkčné znamienka* len technickými znakmi uľahčujúcimi čítanie textu. Keď sú umiestnené medzi slovami, získavajú aj vlastný význam a stávajú sa zmysluplnými entitami rovnako ako niektoré slová, ktoré plnia predovšetkým gramatickú funkciu, napríklad spojky či články.

Napríklad bodky síce nemajú jasné fonetické vyjadrenie, ale určujú dĺžku viet a môžu ovplyvniť subjektívne vnímanie textu čitateľom – napríklad rýchlosť príbehu, množstvo detailov a komplexnosť opisu situácie. Štúdie [5] navyše naznačujú, že interpunkcia v naratívnych textoch vykazuje dlhodobé korelácie, čo poskytuje merateľný dôkaz jej významu pre štruktúru a vnímanie textu.

Podľa analýzy z [?] vyplýva, že interpunkčné znamienka môžu byť považované za plnohodnotné jednotky jazyka, podobne ako bežné slová. Ich zahrnutie do štatistických analýz (napríklad pri skúmaní frekvencie výskytu alebo sietí susednosti slov) poskytuje hodnotné informácie o štruktúre textu.

V [?] sa skúmali tieto vlastnosti v korpusoch, teda vo veľkých súboroch textov, ktoré slúžia na systematickú analýzu jazyka. Porovnanie výsledkov pre interpunkciu a bežné slová ukázalo, že interpunkcia nie je len technický znak uľahčujúci čítanie, ale nesie vlastný význam a štruktúru, ktorá sa dá kvantitatívne zachytiť. Tieto zistenia môžu zlepšiť spoľahlivosť automatických štatistických výpočtov a analýz textov.

V [?] sa využívala na analýzu relatívne najbližiu literárnu formu: naratívny text, teda prózu. Naratívne texty sú charakteristické tým, že rozprávajú príbeh s určitým dejom, postavami a prostredím. Tento typ textu často využíva interpunkciu na vyjadrenie emócií, rytmu a štruktúry príbehu. V tomto článku sa autori zameriavali na analýzu interpunkčných znamienok v známych románoch, rôznych jazykoch potriacich do rôznych jazykových skupín:

- germánskej (angličtina a nemčina)
- románskej (francúzskej a talianskej)
- slovanskej (poľskej a ruskej)

Rovnako teda budeme analyzovať interpunkčné znamienka v naratívnych textoch aj my (nie vo všetkých jazykoch, lebo sa ťažko hľadajú voľné texty na stiahnutie, ale teda v jazyku slovenčina/čeština, ďalej angličtina/nemčina). Texty sa cielene vyberajú dosť dlhé, aby bolo možné vykonať štatistickú analýzu.

V kapitole 5 popíšeme implementáciu programu, ktorý sme vytvorili na analýzu interpunkčných znamienok v naratívnych textoch, následne popíšeme analýzu vykonanú na vybraných textoch (kapitola 4) a nakoniec zhrnieme získané výsledky a závery.

2.1 Základné modely distribúcie

2.1.1 Zipfov zákon

Zipfov zákon je empirickým zákonom popisujúcim frekvenciu výskytu slov v prirodzenom jazyku. Vyjadruje sa vzťahom:

$$f(r) \propto \frac{1}{r^\alpha} \quad (2.1)$$

kde $f(r)$ je frekvencia slova s poradím r (zoradené zostupne podľa najčastejšieho výskytu) a α je exponent, typicky blízko hodnote 1. To znamená, že druhé najčastejšie slovo sa vyskytuje približne s polovičnou frekvenciou ako prvé slovo, tretie s tretinovou frekvenciou, atď.

2.1.2 Heapsov zákon

Heapsov zákon opisuje vzťah medzi veľkosťou textového korpusu a počtom jedinečných (rozdielnych) slov v ňom. Matematicky sa vyjadruje ako:

$$V(N) = K \cdot N^\beta \quad (2.2)$$

kde $V(N)$ je počet jedinečných slov pri veľkosti textu N slov, K a β sú empirické konštanty (typicky $0 < \beta < 1$, zvyčajne okolo 0,4-0,6). Zákon predpovedá, že počet nových slov objavujúcich sa v texte rastie logaritmicky s dĺžkou textu.

Kapitola 3

Jazyk ako graf

V tejto kapitole popíšeme, ako môžeme reprezentovať jazyk ako grafovú štruktúru. Budeme sa zameriavať na to, ako slová v jazyku súvisia navzájom a ako tieto vzťahy môžu byť modelované pomocou grafov. Pozrieme sa do histórie, kto spracovával jazyk ako graf prvý.

Kapitola 4

Analýza textov

V tejto kapitole popíšeme zistenia ohľadom frekvencie interpunkčných znamienok vybranej literatúry. Analýzu budeme robiť (a porovnávať) podľa autorov a podľa jazykov, v ktorých sú diela napísané.

4.1 Analýza a model frekvencie v textoch od autora

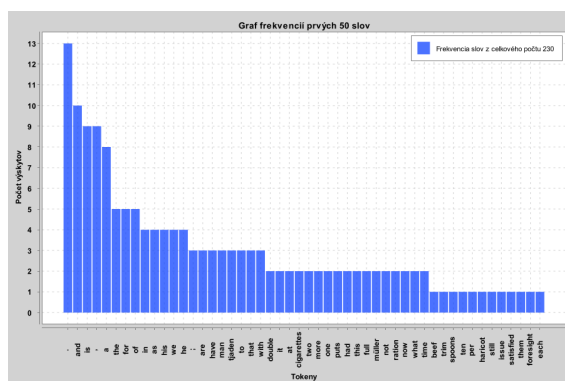
Erich Maria Remarque: Na Západe nič nové

Analyzovať budeme text v anglickom jazyku (All Quiet on the Western Front [?]) a v nemeckom jazyku (Im Westen nichts Neues [?]).

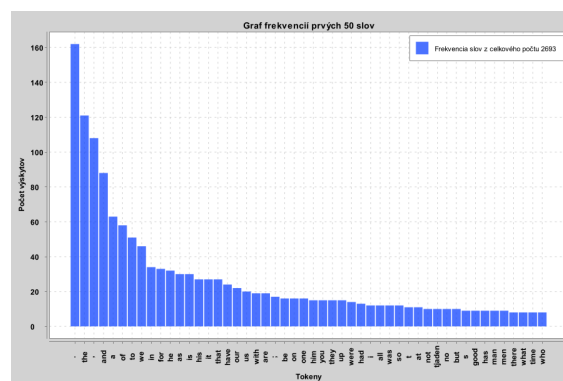
V programe som nastavil, že sa má čítať až od strany 9, aby sa preskočil obsah a úvodné slová. Ďalej som nastavil, aby sa ignoroval názov diela, ktorý bol na každej strane, ako aj meno autora.

4.1.1 Analýza frekvencie tokenov v texte

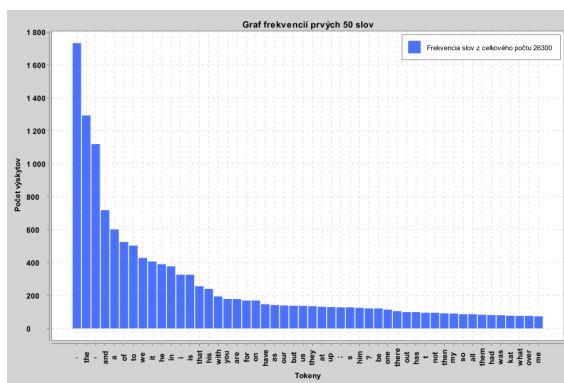
Náš program vygeneroval nasledovné štatistiky pre toto dielo:



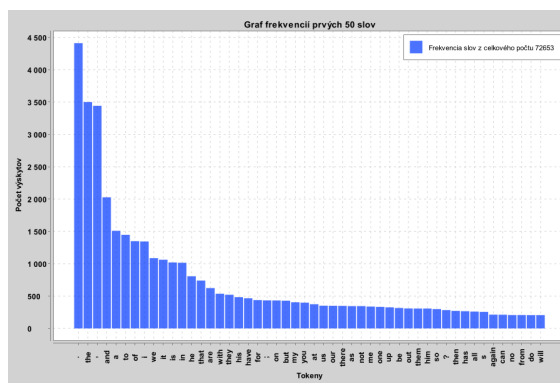
Obr. 4.1: Frekvencia tokenov v texte (1 strana)



Obr. 4.2: Frekvencia tokenov v texte (10 strán)



Obr. 4.3: Frekvencia tokenov v texte (100 strán)



Obr. 4.4: Frekvencia tokenov v texte (celý text)

Tabuľka 4.1: Počet unikátnych slov v texte

Počet strán	1	2	10	20	100	200	280
Počet unikátnych slov	126	238	869	1313	3683	5343	6247

4.1.2 Analýza susednosti slov v texte

Analyzujeme postupne pre 1 stranu, 5, 10, 50, 100, 200 strán a nakoniec pre celý text. (prípadne potom preškalujem testovanie)

Tabuľka 4.2: Počet unikátnych hrán v texte

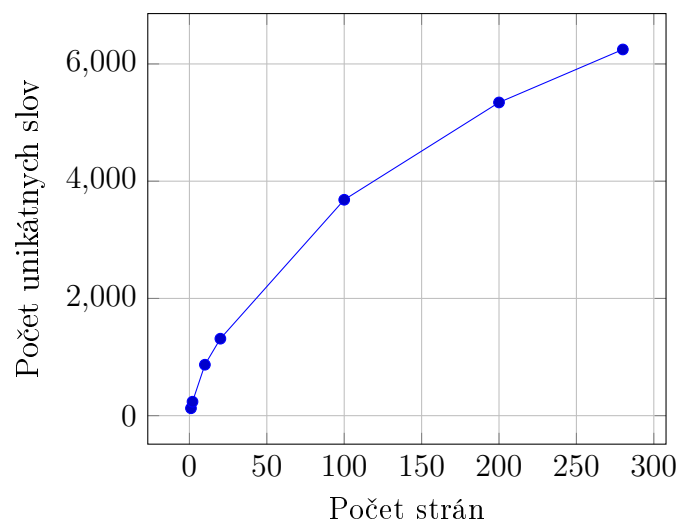
Počet strán	1	2	10	20	100	200	280
Počet unikátnych hrán	218	467	2211	4019	15831	27612	34998

4.2 Analýza a model frekvencie v textoch od autora Erich Maria Remarque: Cesta späť

Analyzovať budeme text v anglickom jazyku (title [?]) a v nemeckom jazyku (Der Weg Zuruck [?]).

4.2.1 Zhrnutie analýzy

Podľa grafu 4.5 môžeme vidieť, že počet unikátnych slov rastie s počtom strán, ale tempo rastu sa postupne spomaľuje. Podľa tohoto grafu vidno, že Heapsov zákon zhruba platí.



Obr. 4.5: Počet unikátnych slov v texte

4.3 Interpretácia výsledkov analýzy

porovnanie so základnými zákonmi (Zipfov zákon, Heapsov zákon), aplikácia modelov na grafy (napr. Drogov)...

Kapitola 5

Implementácia programu

V tejto kapitole popíšeme ako sme využili knižnicu [?] na analýzu a interpretáciu textu zo vstupného súboru formátu PDF.

5.1 Načítanie PDF súboru a extrakcia textu po znakoch

Pri načítavaní využívame knižničné súbory `fontbox-3.0.4.jar` a `pdfbox-app-3.0.4.jar`. Načítavanie celého pdf súboru má na starosti trieda `GetRawTextFromPDF`, ktorá (ako už aj samotný jej názov hovorí) načíta text zo vstupného súboru `.pdf` po jednotlivých znakoch. V konštruktore tejto triedy sú nasledovné parametre:

- Parameter `path` typu `String`, ktorý určuje odkiaľ sa súbor číta, resp. priamo súbor typu `File`.
- Parameter `pageFrom` typu `Integer`, ktorý určuje, od ktorej strany sa súbor číta. Tento parameter slúži na preskočenie obsahu a iného sprievodného textu, ktorý je pre analýzu v rámci tejto práce nežiadúci.
- Parameter `ignoreWords` typu `List<String>` (pole stringov), ktorý určuje, ktoré slovo (slová) sa majú preskočiť, pretože sú tiež pre analýzu v rámci tejto práce nežiadúce (ide napríklad o hlavičky a päty strán, teda názov diela/kapitoly, resp. o autora)

V konštruktore sa načíta súbor do objektu `PDDocument`, z ktorého sa následne pomocou triedy `PDFTextStripper` extrahuje text po jednotlivých znakoch. Do `PDFStrippera` sa ešte nastaví, odkiaľ sa má začať súbor čítať (preskočenie obsahu a iného sprievodného textu). Do premennej `text` typu `String` sa následne uloží celý extrahovaný text zo súboru.

Táto trieda obsahuje metódu `getText()`, ktorá vráti extrahovaný text (`String`) zo súboru bez nežiadúcich slov.

5.2 Analýza získaného textu a rozobratie textu na jednotlivé tokeny

Získaný text zo súboru (pomocou metódy `getText()`) sa následne analyzuje v triede `TypeSplitter`, ktorú si tu opíšeme. Táto trieda má v konštruktoze ako argument text (`String`), ktorý sa bude analyzovať. Pre prechádzanie textu sme zvolili `StringBuilder`. Budeme hľadať jednotlivé *tokeny*, základné jednotky textu, ktoré budeme analyzovať. Budeme rozlišovať tieto tokeny:

- `OTHER(0)` - ostatné znaky, ktoré nespádajú do žiadnej z nasledujúcich kategórií (napr. netypické symboly).
- `WORD(1)` - slová zložené z písmen (vrátane diakritiky).
- `NUMBER(2)` - číselné sekvencie (čísllice).
- `SPACE(3)` - medzery, tabulátory a znaky nového riadku.
- `BRACKETS(4)` - zátvorky typu „()“, „[]“, „{}“ a pod.
- `COMMA(5)` - čiarka „,“.
- `SEMICOLON(6)` - bodkočiarka „;“.
- `DOT(7)` - bodka „.“.
- `COLON(8)` - dvojbodka „:“.
- `APOSTROPH(9)` - apostrof „'“ alebo jednoduchá úvodzovka „‘“.
- `QUOTATION_MARK(10)` - úvodzovky (dvojité " alebo slovenské „“).
- `DASH(11)` - pomlčka alebo spojovník (napr. „-“, „-“).
- `ELLIPSIS(12)` - výpustka (tri bodky „...“ alebo ekvivalent).
- `EXCLAMATION_MARK(13)` - výkričník „!“.
- `QUESTION_MARK(14)` - otáznik „?“.

Tieto tokeny sú reprezentované pomocou enum (vymenovanie možností) triedy `TokenType`. Tokeny sa vytvárajú nasledovným spôsobom (s využitím triedy `CreateToken`): postupne sa prechádza celý text a podľa jednotlivých znakov sa vytvárajú tokeny. Napríklad ak sa narazí na písmeno, začne sa vytvárať token typu `WORD`, ktorý bude obsahovať všetky nasledujúce písmená, až kým sa nenarazí na znak, ktorý už nie je písmenom (napr. medzera alebo interpunkčný znak). Podobne sa postupuje pre ostatné typy tokenov.

V triede `TypeSplitter` sa ukladajú 2 štatistiky:

- Počet výskytov jednotlivých tokenov, ktoré sú uložené v zozname `List<Integer> tokenTypesCount`, kde index zodpovedá hodnote enum triedy `TokenType`.
- Počet výskytov jednotlivých slov, ktoré sú uložené v mape `Map<String, Integer> wordCount`.

Ďalšie trieda, ktorú využívame na analýzu viet ako celkov je trieda `Sentence`. Táto trieda má v konštruktoe ako argument zoznam tokenov (typu `List<Token>`), ktoré sa budú analyzovať a poradie v texte (id) typu `Integer`. Táto trieda obsahuje celkový počet tokenov ako aj počet interpunkčných tokenov. V tejto triede je ďalej naprogramovaný aj komparátor na zoradovanie viet podľa ich dĺžky (počtu tokenov vo vete), ak je zhoda tak podľa počtu interpunkčných tokenov vo vete.

5.3 Vyrábanie potrebných štruktúr

Trieda `TypeSplitter` poskytuje nasledovné metódy:

- Metódu `printTokensCountSorted()`, ktorá vráti mapu so tokenmi a ich počtom výskytov, zoradenú podľa počtu výskytov zostupne.
- Metódu `printWhiteTokensCountSorted()`, ktorá vráti mapu so bielymi¹ tokenmi a ich počtom výskytov, zoradenú podľa počtu výskytov zostupne.
- Metódu `printTokensTypesCountSorted()`, ktorá vráti mapu so výskytmi jednotlivých typov tokenov, zoradenú podľa počtu výskytov zostupne.
- Metódu `printNeighborCountSortedByValue()`, ktorá vráti mapu s dvojicami slov a s počtom výskytov týchto dvojíc, zoradenú podľa počtu výskytov zostupne. Táto metóda teda slúži na analýzu opakovania postupnosti istých slov, napr. po akom slove sa používa slovo „and“, resp. čo nasleduje po ňom. Metóda ako parameter očakáva zoznam tokenov (slov) z pôvodného textu, ktoré sa majú analyzovať.

¹Medzery, tabulátory, znaky nového riadku, ... ak ich nasleduje viac za sebou, tak sa spoja do jedného tokenu

- Metódu `printSentenceStatistic()`, ktorá vráti štatistiky o jednotlivých vetách. Vety sú zoradené zostupne podľa ich dĺžky (počtu slov (tokenov) vo vete), ak je zhoda tak ktorá veta má menej interpunkčných tokenov.

5.4 Interpretácia (grafových) štruktúr

V priečinku `litertúra` sa nachádzajú podpriečinky podľa autorov, v nich sú podpriečinky podľa diel a v podpriečinkoch diel sa nachádzajú konečné podpriečinky podľa jazykov, v ktorých sú diela napísané. V týchto podpriečinkoch sa nachádzajú samotné PDF súbory s textami diel a súbory `.txt` ktoré vytvorí program, ktorý spustíme na danom PDF súbore. Cesta k samotnému priečinku s dielom a jeho analýzov má teda tvar: „`/literatura/<autor>/<dielo>/<jazyk>/<dielo v .PDF a súbory, ktoré vygeneroval program>`“. Pri každom diele sa vygenerujú 4 súbory, ktoré budeme neskôr využívať pri analýze:

- `<jazyk>_sentences.txt` - údaje o dĺžke jednotlivých viet.
- `<jazyk>_tokenCount.txt` - počet výskytov jednotlivých tokenov.
- `<jazyk>_tokenCountWhite.txt` - počet výskytov jednotlivých bielych tokenov.
- `<jazyk>_tokenTypesCount.txt` - počet výskytov jednotlivých typov tokenov.
- `<jazyk>_tokenNeighbors.txt` - počet výskytov jednotlivých dvojíc slov (susedov).

5.5 Vizualizácia dát pomocou knižnice XChart

Na vizualizáciu získaných štatistík z analýzy textov využívame knižnicu [?]. Táto knižnica poskytuje jednoduchý a flexibilný spôsob tvorby rôznych typov grafov a diagramov bez potreby zložitej konfigurácie.

V našej implementácii sme knižnicu XChart využili na grafické znázornenie počtu výskytov prvých 50 slov (tokenov) podľa ich frekvencie, ako aj na zobrazenie ďalších štatistík získaných z analýzy textov. Konkrétne sme vytvorili nasledujúce typy grafov:

- Stĺpcové grafy znázorňujúce počet výskytov jednotlivých tokenov a ich typov.

Tieto grafy sú generované ako obrázkové súbory formátu `.png` a ukládané do príslušných adresárov diel pre ďalšiu analýzu a porovnávanie textov z rôznych autorov a jazykov.

5.6 Skúmanie susednosti grafov pomocou knižnice Gephi

Na skúmanie a analýzu grafov susedností sme využili grafickú knižnicu Gephi [?], ktorá je určená na vizualizáciu a analýzu veľkých sietí a grafov. Gephi poskytuje množstvo nástrojov na prácu s grafmi, vrátane rôznych algoritmov na analýzu štruktúry grafov, vizualizačných techník a možností exportu výsledkov.

Záver

V tejto kapitole zhrniem, akých autorov som analyzoval, aké knihy som analyzoval a v akom jazyku som ich analyzoval.

Zhrniem tiež celkovú prácu (program) a pod.

