Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-5220-49434

Bc. Mário Csaplár

DETEKCIA NEGÁCIE V TEXTE PROSTREDNÍCTVOM SYNTAKTICKÝCH ZÁVISLOSTÍ

Diplomová práca

Študijný program: Softvérové inžinierstvo
Študijný odbor: 9.2.5 Softvérové inžinierstvo
Miesto vypracovania: Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU
Vedúci diplomovej práce: Ing. Ivan Polášek, PhD.

máj 2017

Poďakovanie

Na tomto mieste by som sa rád poďakoval všetkým, ktorí mi pri vypracovávaní tejto diplomovej práce pomohli alebo ma usmernili. Obzvlášť však ďakujem Ing. Ivanovi Poláškovi, PhD. za odbornú pomoc, rady a trpezlivosť.

ANOTÁCIA

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
Študijný program: Softvérové inžinierstvo

Autor: Bc. Mário Csaplár
Diplomová práca: Detekcia negácie v texte prostredníctvom syntaktických závislostí
Vedúci diplomovej práce: Ing. Ivan Polášek, PhD.

máj 2017

Táto práca sa venuje negácii ako jazykovému prostriedku v slovenskom a anglickom jazyku. Predmetom výskumu sú možnosti automatizovanej detekcie negácie v neštruktúrovanom texte. Analytická časť opisuje súčasný stav v oblasti predspracovania textu, pričom sa zameriava na overené riešenia. Samostatnú časť analýzy tvorí lingvistický opis negácie.

Detekcia negácie ako proces je rozdelený na dve časti, a to detekcia negátora a detekcia rozsahu negácie. Pre detekciu ohybných negátorov v slovenskom jazyku je navrhnutá slovníková metóda, zatiaľ čo negátory v anglickom jazyku sú detegované pomocou hodnôt sentimentu. Jadrom návrhu tejto práce je vlastná metóda na detekciu rozsahu negácie založená na analýze syntaktických závislostí medzi slovami. Táto metóda bola následne adaptovaná na vybrané druhy negácie v anglickom jazyku.

Výsledkom je návrh systému, ktorý dokáže detegovať negáciu v slovenskom aj anglickom jazyku. Veľká časť práce bola venovaná podrobnému overeniu navrhnutých metód. Pre slovenský jazyk bol vytvorený vlastný korpus manuálne značkovaných dokumentov, ktoré pri experimentoch slúžili ako dátová množina. Postupy pre anglický jazyk boli overené experimentami na anglickom značkovanom korpuse BioScope.

ANNOTATION

Slovak University of Technology in Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES
Degree Course: Software Engineering

Author: Bc. Mário Csaplár
Master's Thesis: Negation detection in documents using syntactic dependencies
Supervisor: Ing. Ivan Polášek, PhD.

2017, May

This Master's thesis deals with negation as a language phenomenon in both Slovak and English languages. The focus of research lies on automatized detection of negation in non-structured documents. A significant part of analysis is dedicated to the current state of the art for the various phases of preprocessing while providing an overview of existing solutions. The core of the analytical chapter describes negation from linguistic perspective.

Process of negation detection is divided into two parts, namely detection of negation markers and detection of negation scope. While there are proposals on improvement in detection of certain types of negation markers, this work proposes a method for negation scope detection in Slovak language using dependency parsing. The same method is then applied to selected types of negation in English language.

The outcome of the thesis is a proposed system capable of detecting negation in both Slovak and English languages. A major part of the thesis is devoted to detailed evaluation of proposed methods. For the purposes of evaluation in Slovak language, a custom corpus with manually tagged negations was created. This corpus was later used in a series of experiments evaluating the detection of both negators and negation scope. The evaluation of methods for English language was performed on BioScope, a standard English corpus with marked negations.

Obsah

1 Úvod **1**
- 1.1 Použité pojmy . . . 2
- 1.2 Použité skratky . . . 3

2 Analýza predspracovania textu **5**
- 2.1 Extrakcia čistého textu . . . 5
- 2.2 Tokenizácia . . . 5
- 2.3 Základný tvar slov . . . 6
  - 2.3.1 Stemovanie . . . 6
  - 2.3.2 Lematizácia . . . 7
- 2.4 Morfologická analýza . . . 8
- 2.5 Syntaktická analýza . . . 9
  - 2.5.1 Závislostná analýza . . . 10
- 2.6 Balíky jazykových nástrojov . . . 11
- 2.7 Word2vec . . . 12
- 2.8 WordNet a SentiWordNet . . . 12
- 2.9 Zhodnotenie . . . 13

3 Analýza negácie vo vete **15**
- 3.1 Negátor a rozsah negácie . . . 16
- 3.2 Techické prostriedky na detekciu negácie . . . 16
  - 3.2.1 Pravidlové systémy . . . 17
  - 3.2.2 Neurónová sieť a strojové účenie . . . 18
  - 3.2.3 Bezkontextová gramatika . . . 18
  - 3.2.4 Analýza syntaktických závislostí . . . 19
- 3.3 Negácia v slovenskom jazyku . . . 19
  - 3.3.1 Negátory so záporným prefixom . . . 20
  - 3.3.2 Negátor *nie* . . . 20
  - 3.3.3 Negátor *bez* . . . 21
  - 3.3.4 Iné negátory . . . 21
  - 3.3.5 Sentiment ako polarizátor . . . 21
  - 3.3.6 Rozsah negácie a viacnásobná zmena polarity . . . 22
- 3.4 Negácia v anglickom jazyku . . . 22
  - 3.4.1 Negátor *not* . . . 22

3.4.2 Negátor *n-* . . . 23
3.4.3 Prepozičné negátory . . . 23
3.4.4 Adjektívne negátory . . . 23
3.4.5 Analýza sentimentu pri detekcii negátora . . . 24
3.4.6 Dvojitá negácia . . . 24
3.5 Textové korpusy na vyhodnotenie . . . 24
3.5.1 Slovenský národný korpus . . . 24
3.5.2 Pražský závislostný korpus . . . 25
3.5.3 BioScope . . . 25
3.5.4 Conan Doyleneg . . . 26

4 Analýza uplatnenia detekcie negácie 27
4.1 Extrakcie z textu a aplikácia negácie . . . 28
4.1.1 TF-IDF . . . 28
4.1.2 SDE . . . 30
4.1.3 N-gramy . . . 31
4.1.4 SN-gramy . . . 31
4.2 Uplatnenie pri spracovaní softvérových artefaktov . . . 32

5 Analýza existujúceho riešenia pre slovenský jazyk 35
5.1 Prvá generácia riešenia . . . 35
5.2 Druhá generácia riešenia . . . 36
5.3 Možné vylepšenia . . . 37
5.4 Zhodnotenie . . . 38

6 Návrh detekcie negácie pre slovenský jazyk 39
6.1 Detekcia negátora so záporným prefixom . . . 39
6.1.1 Slovníkový prístup . . . 39
6.1.2 Prístup cez sémantické vzťahy *word2vec* . . . 40
6.2 Detekcia rozsahu negácie prechodom závislostného stromu . . . 41
6.2.1 Negácia typu *genitívna predložka* . . . 41
6.2.2 Negácia typu *predikát* . . . 43
6.2.3 Negácia typu *atribút* . . . 44
6.2.4 Negácia typu *odčlenený atribút subjektu* . . . 45
6.2.5 Negácia typu *nie* . . . 45
6.2.6 Dvojitá negácia . . . 47
6.2.7 Presah jednoduchej vety . . . 48

7 Návrh detekcie negácie pre anglický jazyk 49
7.1 Detekcia negátora cez analýzu sentimentu . . . 49
7.2 Detekcia rozsahu negácie prechodom závislostného stromu . . . 49
7.2.1 Negácia typu *predložka* . . . 50
7.2.2 Negácia typu *predikát* . . . 50
7.2.3 Negácia typu *not* a *no* . . . 51
7.2.4 Negácia typu *zdvojené spojky* . . . 52

7.2.5 Dvojitá negácia 53

8 Návrh systému na detekciu negácie **55**
8.1 Špecifikácia požiadaviek 55
8.1.1 Požiadavky na modul detekcie negácie 55
8.1.2 Požiadavky na nadstavbové moduly 56
8.2 Architektúra systému 57
8.3 Procesy v module detekcie negácie 58
8.3.1 Načítanie vstupu 59
8.3.2 Inicializácia parsera a načítanie slovníkov 59
8.3.3 Reprezentácia dokumentu 59
8.3.4 Detekcia negátorov 60
8.3.5 Detekcia rozsahu negácie 60
8.3.6 Uloženie do štruktúrovanej formy 61
8.4 Implementácia prototypu 61
8.4.1 Jazyk a vývojové prostredie 61
8.4.2 Použité knižnice 61

9 Evaluácia detekcie negácie **63**
9.1 Slovenský evaluačný korpus 63
9.1.1 Beletria 64
9.1.2 Šport 65
9.1.3 Recenzie fotoaparátov 66
9.1.4 Tímové projekty 66
9.1.5 Slovenský národný korpus 67
9.1.6 Vlastné vety 67
9.1.7 Celková charakteristika korpusu 68
9.2 Experimenty pre slovenský jazyk 69
9.2.1 Detekcia ohybných negátorov slovníkovou metódou 69
9.2.2 Absolútne počty negácií v korpuse podľa druhu negácie 72
9.2.3 Detekcia rozsahu negácie 73
9.2.4 Optimalizácie detekcie rozsahu predikátovej negácie 74
9.2.5 Výsledná úspešnosť detekcie rozsahu negácie 77
9.2.6 Porovnanie s predošlou generáciou riešenia 79
9.3 Experimenty pre anglický jazyk 80
9.3.1 Charakteristika korpusu BioScope a zastúpenie negátorov 81
9.3.2 Detekcia negátorov 82
9.3.3 Detekcia rozsahu negácie 83
9.4 Zhrnutie evaluácie 84

10 Zhodnotenie **85**
10.1 Ďalšia práca 87

Literatúra **89**

Prílohy

A Technická dokumentácia **A-1**
A.1 Načítanie a ukladanie dát A-1
A.2 Spracovanie textu A-2

B Inštalačná príručka **B-1**
B.1 Systémové požiadavky B-2

C Metodika značkovania korpusu **C-1**
C.1 Druhy negácie C-1
C.2 DTD C-3

D Plán diplomového projektu I **D-1**

E Plán diplomového projektu II **E-1**

F Plán diplomovej práce **F-1**

G Obsah priloženého média **G-1**

1 Úvod

Jazykový fenomén negácie bol diskutovaný už za čias starovekého Grécka, kedy k nemu zaujímali postoj rôzne filozofické školy. Zo svetových jazykov sa nevytratil ani v priebehu storočí a dnes predstavuje oblasť, ktorá vstupuje do popredia v oblasti spracovania prirodzeného jazyka.

Význam negácie pri spracovaní textu sa dá hľadať v dodávaní sémantiky extrahovaným údajom. Podľa [3] až 3 % všetkých viet v medicínskych záznamoch obsahujú v nejakej forme negáciu. To znamená, že v týchto dokumentoch sa nachádzajú pojmy, ktorých súvis so zvyškom textu je explicitne vylúčený.

Ako príklad slúži veta: *„Zvýšená horúčka indikuje, že zdrojom infekcie nie je vírus.“* Hoci veta obsahuje pojmy *„infekcia“* a *„vírus“*, čitateľovi je z kontextu zrejmé, že ich nesmie spájať. Čitateľ totiž na základe prečítaného implicitne *deteguje negáciu*.

Pokiaľ algoritmus na extrakciu kľúčových slov narazí na podobnú vetu, typicky by priradil spomenutým dvom pojmom istú úroveň dôležitosti. Ak by však dokázal rozpoznať, že tieto slová sa vo vete nachádzajú v zápornom význame, úroveň ich dôležitosti by mohol patričným spôsobom znížiť.

Negácia však nie je prvkom, ktorý by mal byť špecifický pre odborné texty. Negované slová tvoria pevnú súčasť jadra slovnej zásoby a negácia ako taká patrí medzi základné jazykové operácie. Stretnúť sa s ňou dá v texte ľubovoľného žánru, či už ide o beletriu alebo technické dokumenty.

Základným cieľom tejto práce je zanalyzovať negáciu v slovenskej a anglickej vete z jazykovedného hľadiska. Na základe toho navrhnúť prístup, ktorý rozšíri možnosti detekcie negácie v slovenskom jazyku, a ten potom adaptovať pre anglický jazyk. Napriek tomu, že tejto téme bola už v minulosti venovaná pozornosť, žiadne z riešení pre slovenský jazyk nebolo v dostatočnej miere overené.

Práca je štruktúrovaná tak, aby postupnými krokmi stanovené ciele splnila. Kapitola analýzy sa venuje prostriedkom predspracovania dokumentov, porovnaniu existujúcich riešení a samostatná časť je venovaná popisu negácie v slovenskom a anglickom jazyku.

Z hľadiska návrhu je pre oba jazyky predstavená ucelená metóda na detekciu negácie,

ktorá sa skladá z dvoch krokov. Prvým je detekcia príznaku negácie, ktorá je závislá od spracovaného jazyka a využíva slovníkový prístup. Druhým, významnejším krokom je detekcia rozsahu negácie prostredníctvom skúmania syntaktických závislostí medzi slovami.

Veľká pozornosť je venovaná evaluácii navrhovaných metód. Práca predstavuje vlastnú množinu dokumentov rôznych žánrov, v ktorých bola manuálne vyznačená negácia. Nad touto množinou vykonáva sériu experimentov spúšťaných z vlastného prototypu na detekciu negácie v slovenskom a anglickom jazyku.

1.1 Použité pojmy

- Afirmácia - kladné tvrdenie. Antonymum pojmu *negácia*.
- Korpus - skupina dokumentov.
- Lexika - slovná zásoba.
- Morfológia - tvaroslovie.
- Morfologické značkovanie - určovanie slovných druhov a gramatických kategórií.
- Morféma - najmenšia časť slova plniaca významovú alebo štruktúrnu funkciu.
- Negácia - proces získania zápornej podoby východiskového slova. Formálno-sémantická operácia.
- Negátor - zápor, teda operátor negácie. Prostredníctvom neho je vykonaná negácia.
- Objekt - predmet.
- Predikát - prísudok.
- Presnosť - evaluačná metrika z anglického *precision*. Vyjadruje pomer správne určených kladných prvkov voči všetkým kladne určeným prvkom.
- Prepozícia - predložka.
- Subjekt - podmet.
- Správnosť - evaluačná metrika z anglického *accuracy*. Vyjadruje pomer správne určených kladných a záporných prvkov voči všetkým prvkom.
- Syntax - skladba vety.
- Syntaktické značkovanie - určovanie vetných členov a vzťahov medzi slovami.
- Úplnosť - evaluačná metrika z anglického *recall*. Vyjadruje pomer správne určených kladných prvkov voči všetkým prvkom, ktoré mali byť kladne určené.

1.2 Použité skratky

- BNF - angl. *Backus-Naur form*. Technika notácie bezkontextovej gramatiky.
- FN - angl. *false negative*. Falošne negatívny prvok. Prvok, ktorý spĺňa hľadanú vlastnosť, ale pri evaluácii bolo nesprávne určené, že túto vlastnosť nespĺňa.
- FP - angl. *false positive*. Falošne pozitívny prvok. Prvok, ktorý nespĺňa hľadanú vlastnosť, ale pri evaluácii bolo nesprávne určené, že túto vlastnosť spĺňa.
- NLP - angl. *natural language processing*. Spracovanie prirodzeného jazyka.
- POS - angl. *part of speech*. Určovanie slovných druhov a gramatických kategórií.
- POSe - angl. *part of sentence*. Určovanie vetných členov.
- SDE - angl. *single document extraction*. Technika extrakcie kľúčových slov opísaná v 4.1.2.
- TF-IDF - angl. *term frequency - inverse document frequency*. Technika extrakcie kľúčových slov opísaná v 4.1.1.
- TN - angl. *true negative*. Správne negatívny prvok. Prvok, ktorý nespĺňa hľadanú vlastnosť a pri evaluácii bolo správne určené, že túto vlastnosť nespĺňa.
- TP - angl. *true positive*. Správne pozitívny prvok. Prvok, ktorý spĺňa hľadanú vlastnosť a pri evaluácii bolo správne určené, že túto vlastnosť spĺňa.
- XML - angl. *extended markup language*. Textovo založený štruktúrovaný formát.

2 Analýza predspracovania textu

Táto kapitola prináša podrobný opis celého procesu transformácie korpusu vstupných dokumentov na vhodne označkované lexikálne jednotky. Úvodná podkapitola 2.1 diskutuje extrakciu čistého textu ako prvý, nevyhnutný krok pri spracovaní dokumentov. Následne sa venuje tokenizácii, úprave slov na základný tvar v závislosti od jazyka, morfologickej a syntaktickej analýze.

Cieľom kapitoly je priniesť prehľad súčasného stavu dostupných nástrojov. Sústredí sa na opis voľne dostupných alternatív, porovnáva ich a hodnotí ich použiteľnosť pre tento projekt. Na záver je zdôvodnený výber konkrétnych balíkov nástrojov 2.6, ktoré prichádzajú do úvahy pre fázy návrhu a implementácie.

2.1 Extrakcia čistého textu

Extrakcia predstavuje proces konverzie zdrojového textu do podoby vhodnej na ďalšie spracovanie. Typicky sa skladá z viacerých krokov, pričom prvý krok vyplýva zo všeobecne rozličnej povahy zdrojového textu.

Medzi štandardné formáty na ukladanie textových dát patria **.txt*, **.doc*, **.docx* či **.pdf*. Je zrejmé, že ak je významný len textový obsah, prirodzeným krokom je vyhladiť rozdiely medzi týmito formátmi extrakciou čistého textu.

V závislosti od programovacieho jazyka sú pre tento cieľ dostupné rôzne alternatívy. Štandardným riešením pre jazyk Java je *Apache Tika*, všeobecný parser disponujúci automatickou detekciou zdrojového textového formátu s podporou takmer všetkých typických formátov [11]. Pre jazyk C# sa dá použiť framework *Toxy*, ktorý takisto dokáže extrahovať text z väčšiny štandardných textových formátov [35].

2.2 Tokenizácia

Zatiaľ čo niektoré algoritmy spracovania textu vyžadujú vstup na úrovni viet, iné si postačia so samostatne stojacimi slovami. Vtedy je nevyhnutné vykonať tokenizáciu, teda

rozdelenie textu na *tokeny*. Podľa [15] je token jazykovou jednotkou, ktorá predstavuje nadmnožinu voči pojmu *slovo*.

Zatiaľ čo slovo je jazykovou jednotkou s určitou formou a významom, tokenom môže byť aj interpunkčné znamienko. Z formálneho hľadiska sa teda čistý text rozdeľuje na tokeny a biele znaky, pod ktorými sa myslia medzery, tabulátory, znaky nového riadku či návratu vozíka.

V niektorých prípadoch môže byť takéto členenie problematické. V slovenčine je medzera štandardnou súčasťou formátovania číslic, oddeľuje číslice od jednotiek v sústave SI či obaľuje pomlčku v slovných spojeniach. Kontext tokenizácie je však úzko spätý s algoritmom, ktorý tokeny vyžaduje, preto nie je možné hovoriť o žiadnom univerzálnom prístupe.

2.3 Základný tvar slov

V rámci predspracovania textu je často nevyhnutnou operáciou transformácia súvislých viet na množinu samostatne stojacich slov v základnom tvare. Jednak také slová môžu ľahšie zohrávať rolu kľúčových slov, ale zároveň zvyšujú šancu nájdenia zhody pri dopytovaní sa. Ako príklad môže slúžiť formulovanie dopytu na slovo *mäso*, kedy sa dá očakávať, že dopytujúci bude vyžadovať zhodu aj s vyskloňovanými podobami.

2.3.1 Stemovanie

Proces hľadania základného tvaru slova je typicky závislý od jazyka, pretože v každom jazyku sú iné pravidlá na gramatické zmeny slov pri použití v rôznych jazykových konštrukciách. Základnou metódou platnou najmä pre anglický jazyk je takzvané *stemovanie*. Ide o systematické redukovanie slov až na úroveň stemovacieho základu, ktorý nemusí byť zhodný s morfologickým základom.

Príkladom je anglický stemovací základ *„print“*, na ktorý sa odstránením sufixu namapujú slová *„printer“*, *„printing“*, *„printable“*, *„printed“* a iné. Pri pohľade na slovo *„ensue“* však vzniká problém, kedy sa toto slovo spolu s odvodenými slovami *„ensued“* a *„ensuing“* namapujú na morfologicky neplatný základ *„ensu“*.

V závislosti od ďalšieho postupu spracovania to môže a nemusí byť problém, ktorý je nutné riešiť. Ak je cieľom porovnávať slová podľa ich základu na zistenie ich ekvivalencie, potom je takýto prístup postačujúci.

Závažnejšou otázkou sú odvodené slová, ktoré nevznikajú pridaním prefixu či sufixu, ale štruktúrnou zmenou na gramatickej úrovni. Typickým príkladom opísaného scenára je sloveso *„grow“* a jeho jednoduchý minulý čas *„grew“*.

V roku 1980 prezentoval Martin Porter prácu [42], ktorá sa stala štandardom v oblasti stemovacích algoritmov pre anglický jazyk. Najprv sa pre každé slovo vypočíta počet dvojíc VC, kde V je ľubovoľný, nenulový počet po sebe idúcich samohlások a C ľubovoľný, nenulový počet po sebe idúcich spoluhlások. Následne každé slovo prejde 5 fázami algoritmu, ktorý sa naň pokúša aplikovať niektoré zo zadefinovaných pravidiel sufixovej redukcie.

Na oficiálnej stránke autora[1] je možné nájsť referenčné implementácie vo viacerých programovacích jazykoch vrátane C, C# či Javy. Autor zároveň odporúča sledovať vývoj druhej generácie algoritmu[2], ktorý je zatiaľ k dispozícii ako pracovná verzia v jazyku C a má pripravené pravidlá pre viacero jazykov.

Hoci je podobný postup možné aplikovať pre akýkoľvek jazyk, problémom je definovanie transformačných pravidiel. Charakteristikou *flektívneho* jazyka je, že viacero gramatických kategórií môže byť vyjadrených jedným a tým istým sufixom, pričom možných sufixov je mnoho. Slovenský jazyk patrí medzi prevažne flektívne jazyky s izolačnými a aglutinačnými prvkami.

V *izolačnom* jazyku sú odvodené tvary úplne odlišné od základného tvaru, príkladom je „mäso“ a jeho genitív plurálu „mias“. Pri *aglutinačných* jazykoch je istá gramatická kategória vždy vyjadrená jediným sufixom, príkladom v slovenskom jazyku je inštrumentál podstatných mien a prípona *-mi*.

Napriek tomu existujú pokusy o uplatnenie stemovania v slovenčine. Rigorózna práca [29] predstavuje prístup založený na Myhillovej-Nerodovej vete opisujúcej regulárne jazyky. Prezentovaný algoritmus je aplikovateľný na ľubovoľný flektívny jazyk, pričom autor použil na evaluáciu slovenčinu.

Iným riešením je dodefinovanie pravidiel pre slovenčinu do druhej generácie Porterovho stemovacieho algoritmu, ktorá nesie názov *Snowball*. Hoci implementácií je viacero, voľne dostupný je, napríklad, *Slovenský stemmer*[3].

2.3.2 Lematizácia

Úplne iný prístup predstavuje *lematizácia*. Ak uvažujeme definíciu tokenu podľa [15], potom *lema* je slovníkový tvar tokenu. Z toho vyplýva, že cieľom lematizácie je získať základný tvar vstupného slova a eliminovať rozdiely medzi jazykovými jednotkami na úrovni gramatickej štruktúry.

Na rozdiel od stemovania, pri lematizácii nejde o systematické odstraňovanie prefixov a sufixov. Rozlišujú sa dva základné prístupy, a to šablónový a slovníkový. Predstaviteľom

[1] http://tartarus.org/martin/PorterStemmer/
[2] http://snowball.tartarus.org/algorithms/english/stemmer.html
[3] http://vi.ikt.ui.sav.sk/Projekty/Projekty_2008//2009/Hana_Pifková_-_Stemer

šablónového prístupu je služba Tvaroslovník[4] vyvinutá na Univerzite Pavla Jozefa Šafárika v Košiciach, ktorá považuje slovotvornú príponu za základný prostriedok ohýbania.

Na základe rôznych sufixov definuje Tvaroslovník pravidlá, ku ktorým má priradené referenčné slová. Vstupné slovo potom ohýba rovnakým spôsobom, ako by ohýbanie prebiehalo v prípade referenčného slova.

Slovníkový prístup je založený na databáze existujúcich lem vytvorenej Jazykovedným ústavom Ľudovíta Štúra. Vstupné slovo je zakaždým kontrolované voči pripraveným lemám a v prípade zhody vráti korektný základný tvar. Vďaka rozsiahlosti databázy je úspešnosť lematizácie takmer stopercentná, výnimku tvoria gramaticky nesprávne slová, neologizmy alebo skratky, ktoré z pohľadu slovníkov nepatria do slovenského jazyka.

V súčasnosti sa šablónový prístup dá považovať za komplementárny ku slovníkovému, pretože v prípade šablón je veľkou nevýhodou nedostatočná úspešnosť výberu vhodného pravidla. Podľa evaluácie vykonanej v bakalárskej práci [17] má Tvaroslovník o 12 percentuálnych bodov nižšiu priemernú úspešnosť než slovníkový prístup.

2.4 Morfologická analýza

Pri pokročilom spracovaní textu môže byť vnímanie tokenov ako samostatných jazykových jednotiek nedostatočné. Základným nositeľom významu v jazyku nie sú len samotné slová, ale zároveň ich prepojenie.

Morfologická analýza (angl. *part of speech analysis*) pridáva ku každému slovu dodatočné údaje o jeho gramatických vlastnostiach. Systém slovenského jazyka rozoznáva desať slovných druhov, pričom každý má vlastné lexikálne, sémantické, morfologické a syntaktické vlastnosti.

Pri ohybných slovných druhoch má zmysel určovať gramatické kategórie. Pri podstatných menách, prídavných menách, zámenách, číslovkách a menných tvaroch slovies sa určujú takzvané *menné kategórie*, ktorými sú rod, číslo, pád, vzor. Pri slovesách sa zasa určujú *slovesné kategórie*, teda osoba, číslo, čas a spôsob.

Cieľom automatickej morfologickej analýzy je určiť slovný druh a gramatické kategórie pre všetky slová vo vstupnej vete. Najväčším problémom je významová nejednoznačnosť, kedy jedno slovo môže spadať do viacerých slovných druhov. Príkladom je slovo *mať*, ktoré môže byť slovesom, ale aj podstatným menom.

V minulosti sa tento problém riešil váhovaním na základe pravdepodobnosti frekvencie výskytu uvažovaného slova v podobe daného slovného druhu [21]. Modernejšie automatizované prostriedky berú pri morfologickej analýze do úvahy kontext, čím vo výraznej miere

[4] http://nazou.fiit.stuba.sk/home/?page=morphonary

Slovný druh	Morfologické vlastnosti	Lexikálne vlastnosti	Vetnočlenská platnosť
Podstatné mená	Ohybný druh	Plnovýznamový druh	Áno
Prídavné mená	Ohybný druh	Plnovýznamový druh	Áno
Zámená	Ohybný druh	Plnovýznamový druh	Áno
Číslovky	Ohybný druh	Plnovýznamový druh	Áno
Slovesá	Ohybný druh	Plnovýznamový druh	Áno
Príslovky	Neohybný druh	Plnovýznamový druh	Áno
Predložky	Neohybný druh	Neplnovýznamový druh	Nie
Spojky	Neohybný druh	Neplnovýznamový druh	Nie
Častice	Neohybný druh	Neplnovýznamový druh	Nie
Citoslovcia	Neohybný druh	Neplnovýznamový druh	Nie

Tabuľka 2.1: Prehľad vlastností slovných druhov v slovenskom jazyku.

dochádza k eliminácii nesprávnych rozhodnutí.

Naivný pravidlový korpusový POS tagger[5] je webovou službou transformujúcou vstupný text na tokeny, pričom každému tokenu priradí úplnú morfologickú informáciu vo formáte *XML*. V prípade viacerých možností sa uplatnia pravidlá zohľadňujúce gramatický kontext vety, teda podobu okolitých slov. Napríklad, ak za slovom *mať* nasleduje podstatné meno v akuzatíve, určite ide o podstatné meno.

Štatistický POS tagger[6] sa pokúša určiť slovný druh a gramatické kategórie na základe pravdepodobnostného modelu. Ide o samostatnú aplikáciu implementovanú v programovacom jazyku C++ vyžadujúcu natrénované modely vo forme katalógu slov.

2.5 Syntaktická analýza

Syntaktická analýza vyjadruje proces určovania vetných členov (angl. *part of sentence analysis*), na ktorý sa dá pozerať ako na nadstavbu morfologickej analýzy. Pri prirodzenom určovaní vetných členov musí človek najprv porozumieť vete po významovej stránke, a potom v nej identifikovať slovné druhy. Až vtedy je schopný určiť prisudzovací sklad (podmet a prísudok) položením otázky, kto je nositeľom deja a čo je predmetom deja.

Ukážkou takéhoto prepojenia je schopnosť určiť prisudzovací sklad. Podmet totiž môže byť vyjadrený podstatným menom (*Lekár* píše.), zámenom (*On* píše.) alebo zamlčaný (Píše.). Prísudok sa rozoznáva jednoduchý slovesný (Lekár *píše.*), zložený slovesný (Lekár *musí písať.*) a menný (Lekár *je schopný* písať.). Pre úplnosť je nutné dodať, že oba vetné členy

[5] http://morpholyzer.fiit.stuba.sk:8080/PosTagger/
[6] https://github.com/Denrasill/SkCrfPosTagger

môžu byť viacnásobné.

Ďalšími vetnými členmi sú predmet, prívlastok a príslovkové určenie. V niektorých vetných konštrukciách je ich vzájomné rozlíšenie náročnou úlohou vyžadujúcou dôslednú významovú analýzu vety. V iných prípadoch je bez ďalších znalostí problematické dokonca aj určenie prisudzovacieho skladu, príkladom je veta *„Myši žerú mačky.“* Poradie slovných druhov je totiž v slovenčine variabilné v závislosti od emocionálneho prifarbenia.

Syntaktický analyzátor pre slovenský jazyk ešte donedávna bol len v procese skúmania a vývoja. Pre český jazyk existujú dva nástroje a mnoho navrhovaných metód, ktoré dosahujú úspešnosť analýzy 60 % až 90 % [25]. Úspešnosť je teda významne nižšia ako pri morfologickej analýze.

Centrum spracovania prirodzeného jazyka Masarykovej univerzity vyvíja vlastný, zložkový prístup ku syntaktickej analýze. Vetu nerozčleňuje na slová, ale na takzvané konštituenty, pri ktorých skúma vzájomné vzťahy. Text spracúva podľa definovanej gramatiky a množiny pravidiel, pričom takisto vyžaduje morfologickú anotáciu. Prehlasovaná úspešnosť je na úrovni 90 %.

2.5.1 Závislostná analýza

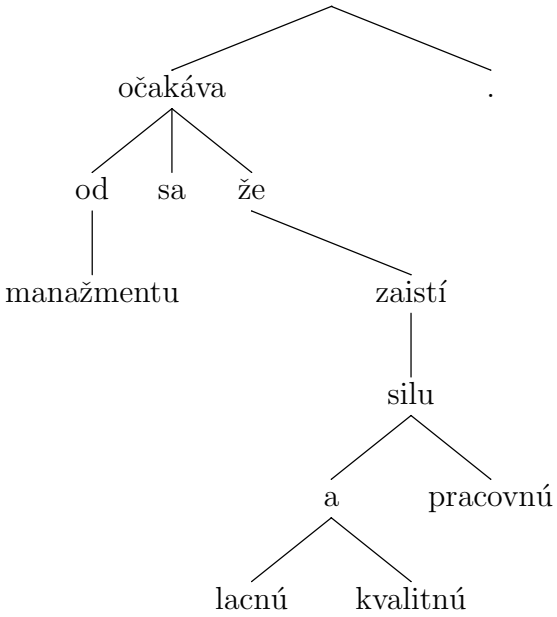
Okolo roku 2006 sa zintenzívnil prístup ku syntaktickej analýze prostredníctvom hľadania závislostí. Ukázalo sa, že pre mnohé problémy spracovania prirodzeného jazyka je postačujúce poznať vzťahy medzi jednotlivými slovami na úrovni nadradenosti a podradenosti. Prostredníctvom závislostnej analýzy je následne možné vykonať ďalšie operácie, medzi ktorými je aj určenie vetných členov.

V Ústave formálnej a aplikovanej lingvistiky Univerzity Karlovej v Prahe je predmetom skúmania závislostný prístup ku vetnočlenskej analýze. Na evaluáciu tohto prístupu bol vytvorený ručne anotovaný závislostný korpus s 1,5 miliónom syntaktických anotácií[7]. Výsledkom je 86,3% zhoda automatickej anotácie vetných členov voči referenčnému korpusu dosiahnutá kombináciou štyroch rozličných metód.

Závislostná analýza ako taká je významne rýchlejšia než analýza konštituentov [8]. Pre samotné potreby detekcie negácie sa takisto ukazuje ako významnejšie určiť vzťahy medzi slovami na úrovni nadradenosti a podradenosti, než presne určiť, o aký vetný člen ide.

V rámci diplomovej práce [28] bol v máji 2016 uskutočnený výskum úspešnosti závislostnej analýzy v slovenskom jazyku. Základom je model natrénovaný na Slovenskom závislostnom korpuse, ktorý je následne porovnaný pri využití v rôznych nástrojoch. Bolo ukázané, že v laboratórnych podmienkach úspešnosť závislostnej analýzy prevyšuje 90 %.

[7] http://ufal.mff.cuni.cz/czech-parsing



Obr. 2.1: Závislostná analýza vety *„Od manažmentu sa očakáva, že zaistí lacnú a kvalitnú pracovnú silu.“*

2.6 Balíky jazykových nástrojov

V súčasnosti (december 2016) existujú ucelené riešenia pre slovenský aj anglický jazyk, ktoré sú schopné zvládnuť väčšinu opísaných úloh. Pre anglický jazyk je jedným z popredných balíkov nástrojov v oblasti spracovania prirodzeného jazyka Stanford CoreNLP[8].

V rámci jediného komplexného súboru nástrojov dokáže čistý text anotovať na viacerých úrovniach. Vykonáva tokenizáciu, lematizáciu, určovanie slovných druhov, analýzu konštituentov, závislostnú analýzu a rozpoznávanie pomenovaných entít. Poskytuje referenčnú implementáciu v jazyku Java a adaptéry pre mnohé iné jazyky.

Alternatívou súčasne tvorenou pre anglický a nemecký jazyk sú Mate Tools z Inštitútu NLP Stuttgartskej univerzity[9]. Poskytujú podobnú množinu operácií ako Stanford CoreNLP s referenčnou implementáciou v jazyku Java. Postavený je na nich slovenský analyzátor Synpar [28] vytvorený ako webové rozhranie aplikácie v Jave.

[8] http://stanfordnlp.github.io/CoreNLP/parse.html
[9] http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/matetools.en.html

2.7 Word2vec

Word2vec je neurónová sieť, ktorá je schopná rozpoznať vzťahy medzi slovami a priradiť im také vektorové reprezentácie, aby vo výslednom mnohorozmernom vektorovom priestore boli podobné slová umiestnené blízko seba [32].

Vďaka výskumu v oblasti spracovania prirodzeného jazyka vznikli na našej fakulte slovenské modely, ktoré sa líšia počtom rozmerov vektorového priestoru, a je možné ich použiť v rámci štandardných nástrojov, akým je *deeplearning4j*[10] pre jazyk Java.

Z hľadiska používania *word2vec* existujú tri základné operácie. Prvou je hľadanie kosínusovej podobnosti medzi dvoma vektormi reprezentujúcimi skúmané slová. Pre model v angličtine je príkladom kosínusová podobnosť medzi slovami *deň* (angl. *day*) a *noc* (angl. *night*) rovná $0,7704$.

Kosínusová podobnosť pre dva vstupné vektory $\boldsymbol{x}$ a $\boldsymbol{y}$ je definovaná ako $cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{||\boldsymbol{x}|| \cdot ||\boldsymbol{y}||}$. Funkčnou hodnotou je reálne číslo v rozsahu $< 0, 1 >$, kde 0 vyjadruje nulovú podobnosť a 1 úplnú zhodu.

Ďalšou operáciou je nájdenie n podobných vektorov. Opäť pre model v angličtine 8 najpodobnejších vektorov pre slovo *deň* (angl. *day*) je *noc, týždeň, rok, zápas, ročné obdobie, počas, kancelária, do* (angl. *night, week, year, game, season, during, office, until*).

Poslednou množinou operácií je jednoduchá vektorová algebra. Na modeli pre anglický jazyk bolo ukázané, že berúc do úvahy vektorovú reprezentáciu slov, je možné dospieť ku sémanticky správnym výsledkom v prípadoch ako *Rím - Taliansko + Čína = Peking* alebo *Koleno - Noha + Ruka = Lakeť*.

2.8 WordNet a SentiWordNet

WordNet je lexikálna databáza anglického jazyka, ktorá vznikla na Princetonskej univerzite. Zoskupuje slová do synonymických skupín, ktoré nazýva *synsety* [33]. Pre túto prácu je však významnejším komplementárny projekt *SentiWordNet* [2].

Cieľom projektu *SentiWordNet*[11] je priradiť každému synsetu z projektu *WordNet* tri koeficienty sentimentu, a to koeficient kladnej výpovede, zápornej výpovede a objektívnej výpovede v rozsahu 0 až 1. Príkladom je anglické slovo *undeniable* (slov. *nepopierateľný*), ktorému *SentiWordNet* priradil koeficient kladnej výpovede 0 a koeficient zápornej výpovede 0,85.

[10] https://deeplearning4j.org/word2vec
[11] http://sentiwordnet.isti.cnr.it/

2.9 Zhodnotenie

Nástroje opísané v časti extrakcia čistého textu môžu tvoriť prvý stavebný blok navrhovaného systému. Z praktického hľadiska sa však ukázalo byť efektívnejšie využiť namiesto mnohých samostatných nástrojov na ďalšie predspracovanie práve súhrnné balíky nástrojov Stanford CoreNLP pre anglický jazyk a Synpar pre slovenský jazyk. Vďaka tomu sa ťažisko práce môže preniesť na využitie korektne anotovaných slov namiesto pokusov o úpravu či prepájanie samostatných nástrojov.

Dvojica podporných nástrojov *word2vec* a *SentiWordNet* môže poslúžiť pri špecifických úlohách spojených s procesom detekcie negácie. Ich využitie je opísané v príslušných častiach práce.

3 Analýza negácie vo vete

Všeobecné poznatky o detekcii negácie sú aplikovateľné pre oba zohľadňované jazyky, teda slovenský aj anglický. V slovenčine sa táto práca opiera o výskum v oblasti negácie, ktorý od 80. rokov minulého storočia prezentuje doc. Pavlovič menovite v publikáciách [37], [39] a [38].

Podľa jeho definície negácia vyjadruje proces transformácie východiskového slova do podoby, v ktorej má opačnú významovú hodnotu. Zásadne ide o operáciu, ku ktorej sú pridružené dva aspekty, a to formálny a sémantický.

Aby bolo možné vykonať sémantickú zmenu, je potrebná formálna zmena pôvodného slova na úrovni štruktúry. Typicky ide o pridanie buď morfémy *ne-*, alebo vlastného negátora. Ako príklad poslúži sloveso *dovoliť*, ktoré má pozitívny význam. Po aplikovaní negácie sa toto sloveso sémanticky priblíži ku antonymickému slovesu *zakázať*. Aby to bolo možné uskutočniť, sloveso *dovoliť* musí byť vo formálnej rovine upravené pridaním morfémy *ne*, čím získa podobu *nedovoliť*.

Ak sa negácia vykoná na samostatnom slove úpravou jeho lexikálnej podoby, samotné slovo sa môže dostať do roly negátora iného slova. Medzi negáciou ako algoritmickým procesom a negátorom ako operátorom negácie existuje významný rozdiel.

Práve na tomto rozdelení sú postavené ďalšie členenia negácie. Pre úspešnú detekciu negácie je potrebné vo vete detegovať obe fázy, teda vyhľadať negátor (angl. *negation marker*) a určiť jeho rozsah negácie (angl. *negation scope*).

Problémom je však existencia negácií, ktoré nie sú explicitné, prípadne sa nedá jednoznačne určiť ich rozsah. Typickým príkladom je implikatívna negácia, ktorá využíva sarkazmus, iróniu a iné vyššie jazykové prostriedky, či konštrukcie špecifické pre danú doménu.

Vo vete *„Ak je toto zelená farba, potom som ja pápež.“* je negátorom celá podradená veta. Čitateľ musí najprv vyhodnotiť, že autor tvrdenia v skutočnosti nie je pápežom, čím podradená veta získava zápornú pravdivostnú hodnotu. Použitím logickej operácie *implikácia* potom vyhodnotí *rozsah negácie* na nadradenú vetu.

3.1 Negátor a rozsah negácie

Pod pojmom negátor alebo *platnosť negácie* sa rozumie rozhodnutie o kladnom alebo zápornom význame každého slova vo vete. Takéto rozhodnutie je možné vykonať na základe analýzy samotného slovného tokenu, väčšinou bez nutnosti poznania kontextu.

Ako bolo spomenuté, nie vždy sa pri negácii vyskytuje explicitný negátor. Niekedy je potrebné využiť širšiu bázu znalostí na vyhodnotenie vyšších jazykových prostriedkov, aby bolo možné usúdiť, že isté slovné spojenie zohráva rolu zamlčaného negátora.

Je zjavné, že ak sa detekcia negátorov obmedzí len na tokeny, implicitné negátory nie je možné odhaliť. V skutočnosti nie je zložité vytvoriť vetnú konštrukciu, ktorá by bola vyhodnotiteľná iba doménovým expertom. Stačí ako podraďovaciu vetu použiť odborné tvrdenie z ľubovoľnej oblasti, ktorá si môže vyžadovať vysoko špecifické doménové znalosti.

Ďalším rozmerom problému sú jazykové prostriedky ako sarkazmus či hyperbola. V tvrdení *„To bola ale zábava!“* nie je možné určiť prítomnosť implicitného negátora bez poznania širšieho kontextu, v ktorom autor tento výrok vyslovil. Spojka *ale* mohla byť použitá ako intenzifikačný prostriedok, ale zároveň aj ako negátor. Opísané prípady sú dôvodom, prečo sú hlavnou oblasťou skúmania platnosti negácie práve explicitné negátory.

Významným problémom pri detekcii negácie je detekcia rozsahu negácie. Po odhalení negátora vo vete je potrebné rozhodnúť, ktoré slovo alebo skupinu slov neguje. Jeden z postupov pre anglický jazyk je prezentovaný v práci [16].

3.2 Techické prostriedky na detekciu negácie

Automatizovaná detekcia negácie je samostatnou oblasťou skúmania, pričom pri algoritmických riešeniach sa rozlišuje niekoľko konceptuálne odlišných spôsobov spracovania textu. Algoritmy založené na pravidlových systémoch opísaných v časti 3.2.1 obsahujú množinu podmienok vo forme *IF - THEN*.

Neurónové siete opísané v časti 3.2.2 poskytujú väčšiu flexibilitu z hľadiska možností detekcie negácie, pretože kopíruje spôsob, akým človek spracúva text. Nevýhodou je vyššia výpočtová zložitosť v porovnaní s pravidlovými systémami a nutnosť cvičných dátových množín. Podľa publikácie [43] dosahujú pravidlové systémy lepšie výsledky pri zohľadňovaní rozsahu negácie než strojové učenie.

Ďalším prístupom je formalizácia súboru pravidiel prostredníctvom definovania bezkontextovej gramatiky. Pomocou nich je možné detegovať negátory, ale aj rozsah negácie. Nevýhodou je nízka flexibilita, ktorá sa prejavuje najmä v jazykoch, ktoré nemajú pevný slovosled.

Záver tejto podkapitoly 3.2.4 sa venuje závislostnej analýze ako relatívne novému prístupu ku detekcii rozsahu negácie. Posledná časť je venovaná modelu *word2vec* 2.7, ktorá prostredníctvom modelovania vzťahov medzi slovami môže dopôcť ku flexibilnejšej detekcii negátorov.

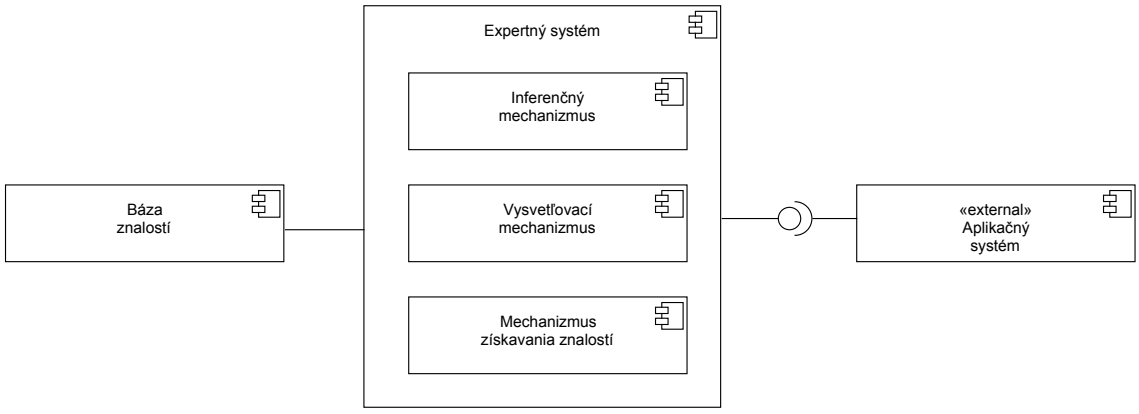
3.2.1 Pravidlové systémy

Princípom *expertných systémov* je prevziať rolu doménového odborníka pri riešení netriviálnych problémov s využitím vysoko špecifických doménových znalostí. Cieľom je simulovať rozhodovanie človeka na takej úrovni, aby kvalita výsledkov rozhodovania systému zodpovedala skutočnému ľudskému expertovi. Medzi hlavné rysy sa radia [10]:

- Separácia doménových znalostí od ich aplikácie.
- Schopnosť uskutočniť rozhodnutie pri rovnosti znalostí.
- Schopnosť vysvetliť a odôvodniť rozhodovací proces.

Ako znázorňuje diagram 3.1, typicky sa expertný systém skladá zo štyroch hlavných častí. *Bázu znalostí* tvorí zoznam všetkých doménovo špecifických znalostí, ktoré sú v prípade *pravidlového systému* reprezentované pravidlami vo forme $IF \rightarrow THEN$. Ľavá časť pravidla je nazývaná podmienková časť, pravá časť obsahuje vyplývajúce závery.

Inferenčný mechanizmus je zoskupenie doménovo nezávislých algoritmov, ktoré dokážu riešiť problémy len na základe obsahu bázy znalostí. Nejde teda o aplikačnú logiku, ktorá by obsahovala dodatočnú rozhodovaciu logiku, ide o všeobecnú implementáciu jadra expertného systému. Inferenčný mechanizmus typicky dokáže odvodzovať nové poznatky z existujúcich znalostí a vyhľadávať vhodnú znalosť na použitie v kontexte aktuálneho problému.



Obr. 3.1: Diagram kompozitnej štruktúry znázorňujúci súčiastky expertného systému.

Na podobnom systéme bol postavený algoritmus na detekciu negácie v anglickom jazyku pod názvom NegEx [6] a jeho rozšírenie SynNeg [48].

3.2.2 Neurónová sieť a strojové účenie

Neurónová sieť je prístup k inteligentnému spracovaniu informácií, ktorý vychádza z biologickej podstaty fungovania mozgu [26]. Človek je schopný identifikovať súvis medzi dvoma rozličnými vecami aj napriek tomu, že do detailov tomu súvisu nerozumie.

Hlavnou funkcionálnou jednotkou neurónovej siete je neurón, ktorý je podľa predpokladov zodpovedný za vykonávanie operácií. Jednotlivé neuróny v ľudskom mozgu sú prepojené synapsiami, ktoré takýmto spôsobom prepájajú bázy ľudských vedomostí.

Strojové účenie je, na druhej strane, spôsob definovania určitých čŕt, ktoré si algoritmus v procese trénovania všíma. Následne dokáže v testovacom režime na základe odpozorovaných skutočností odvodiť vlastné uzávery.

Tému tejto diplomovej práce paralelne spracúva Bc. Jozef Gáborík, ktorého predmetom výskumu sú práve metódy strojového učenia. Vo svojej práci porovnáva viacero prístupov, pomocou ktorých je možné detegovať negátor a rozsah negácie v slovenskom jazyku.

3.2.3 Bezkontextová gramatika

V oblasti spracovania prirodzeného jazyka je bežnou alternatívou použiť na jednoduché spracovanie vzorov vo vstupnom jazyku bezkontextovú gramatiku. Ide o štvoricu $G = (N, T, P, S)$, kde N je množina neterminálnych symbolov, T množina terminálnych symbolov, P množina transformačných pravidiel a S počiatočný symbol z množiny neterminálnych symbolov.

Všetky pravidlá majú tvar $\alpha \rightarrow \beta$, kde α je práve jeden neterminál β je ľubovoľne dlhý reťazec zložený z terminálov a neterminálov. Potom jazyk generovaný touto gramatikou je množina všetkých reťazcov, ktoré je možné odvodiť z počiatočného symbolu pomocou transformačných pravidiel.

Pomocou pravidiel bezkontextovej gramatiky je možné definovať množinu vzorov jednak pre negátory, ale zároveň aj pre vety s obsahom vybraných druhov negácie, ako to urobil Kvitkovič v [27]. Tento prístup však so sebou prináša isté obmedzenia.

Nech je daná bezkontextová gramatika G_N s pravidlami na detekciu negácie pre slovenský jazyk. Problém tohto prístupu nastáva v počte opísateľných vzorov. Keďže slovenský jazyk disponuje voľným slovosledom, je potrebné zahrnúť tento fakt do tvorby pravidiel. Ďalším problémom je kompozícia vety, kedy rozšírenie pravidiel z jednoduchej vety na súvetie môže opäť významne zvýšiť počet potrebných pravidiel.

Pri takto definovanom probléme je otázkou, či veta zo vstupného jazyka patrí do jazyka definovaného gramatikou. Pomocou nástrojov ako sú kompilátory kompilátorov je síce takéto spracovanie jednoduché implementovať, ale na určenie rozsahu negácie je nutné ďalšie spracovanie. Pomocou ďalších pravidiel je totiž nutné určiť negované slová na základe toho, aká množina pravidiel gramatiky bola použitá pri generovaní.

3.2.4 Analýza syntaktických závislostí

Závislostná analýza je prístup, ktorý je vhodný na určenie rozsahu negácie, a bol použitý vo viacerých prácach. Určenie rozsahu negácie vzhľadom na pomenované entity v anglickom jazyku sa venuje práca [47]. Hybridný prístup kombinujúci závislostnú analýzu s algoritmom NexEx pre anglický jazyk prezentuje práca [31]. Syntaktické závislosti boli dokonca využité pri detekcii negácie v španielskom jazyku [7].

Všeobecne analýza syntaktických závislostí poskytuje flexibilitu oproti bezkontextovej gramatike, pretože namiesto pevne stanoveného slovosledu sa pozerá na vetu ako na závislostný strom. To je mimoriadne užitočné najmä vo flektívnych jazykoch s voľným slovosedom. Pravidlá na určenie rozsahu negácie sú potom definované ako prechody od negátora ku vopred definovanému druhu uzla v strome.

Prehľadávanie stromu môže takýmto spôsobom byť realizované aj na úrovni súvetí. Závislostný strom pre súvetie je totiž vybudovaný ako dvojica podstromov opisujúca jednoduché vety, pričom koreňom stromu je spojka, ktorá tieto jednoduché vety spája.

Ďalšou výhodou je jednoduchá manipulácia s uzlami v strome. Pri samotnom prechádzaní stromu je možné nastaviť cieľovým uzlom príznak hovoriaci o tom, že daný uzol je cieľom negácie iného uzla. Z uvedených dôvodov bude na určovanie rozsahu negácie v oboch jazykoch použitá práve táto metóda.

3.3 Negácia v slovenskom jazyku

Ako opisuje časť 3.1, negácia ako proces sa skladá z dvoch prepojených entít, ktorými sú *negátor* ako iniciátor procesu a *rozsah negácie* ako množina jednotiek zasiahnutá týmto procesom. Na detekciu negácie je nevyhnutné úspešne detegovať obidve entity.

Zatiaľ čo detekcia explicitných negátorov sa zaoberá hľadaním vzorov na úrovni samostatných slov, na detekciu rozsahu negácie je potrebné využiť syntax celej vety. Metóda na detekciu rozsahu negácie prezentovaná v časti 6 tejto práce využíva závislostný strom, ktorý pre vybrané dvojice slov modeluje vzťah nadradenosti a podradenosti.

Táto kapitola formuluje jednotlivé druhy negátorov a príklady ich uplatnenia. Dáva si za

cieľ sformulovať javy, ktoré vstupujú do procesu detekcie negátorov a zároveň opísať spôsob kooperácie viacerých negátorov. Celá kapitola sa opiera o lingvistické závery publikácie [39], ktorá kategorizuje negátory podľa typu viet, v ktorých môžu byť zastúpené.

3.3.1 Negátory so záporným prefixom

Typickou črtou negátora v slovenskom jazyku je formálny prefix *ne-*, ktorý otáča sémantickú polaritu slova. Takýmto slovom môže byť prídavné meno (*známy* → *neznámy*), príslovka (*vidno* → *nevidno*), sloveso (*mať* → *nemať*) a zriedkavo aj podstatné meno.

Je potrebné brať na zreteľ, že nie každé slovo začínajúce morfémou *ne-* je v skutočnosti negátorom. Príkladmi sú podstatné mená *neandertálec* či *Nemec*, prídavné mená *nemotorný* či *nenávistný*, sloveso *nechať* a mnohé ďalšie.

Okrem morfémy *ne-*, ktorá záporné hodnotenie totálne vyjadruje, existujú ďalšie prefixy s rôznou úrovňou explicity negácie. Medzi domáce totálne negátory patrí *proti-* (*protirečiaci*), *pa* (*paveda*), *bez-* (*bezodkladný*). Domáce kontextovo závislé negátory sú *roz-, polo-, taktiež-, akoby-, trochu-, truc-, mimo-*.

Medzi zahraničné potom radí *pseudo* (*pseudohistorický*), *a-* (*agramatický*), *dis-* (*disfunkcia*), *i-* (*iregulárny*), *anti-* (*antisemitský*), *de-/dez-* (*dezinformujúci*), *kontra-* (*kontraproduktívny*). Pavlovič ďalej v tejto kategórii definuje zriedkavé prefixy *in-, an-, non-, ex-, extra-, kvázi-, hypo-, non-*.

Prístup, ktorý bol v slovenskom jazyku zvolený v minulosti, bol postavený na zoznamoch. Buď si autor vytvoril zoznam slov, ktoré s istotou považuje za negátor (angl. *whitelist*), alebo zoznam výnimiek (angl. *blacklist*). Zjavným nedostatkom tohto prístupu je škálovateľnosť, keď pre čoraz väčšie množstvo spracovaného textu by takéto zoznamy bolo potrebné neustále dopĺňať.

Táto práca navrhuje dva spôsoby, ako automaticky detegovať negátory so záporným prefixom. Konceptuálne ide o slovníkový prístup 6.1.1 a prístup založený na vektorovej reprezentácií slov 6.1.2.

3.3.2 Negátor *nie*

Slovo *nie* je väčšinou realizátorom členskej formy negácie, pretože vo väčšine prípadov neguje nepredikatívne vetné členy (ľubovoľný okrem prísudku, teda podmet alebo rozvíjacie vetné členy). Príkladmi sú vety: *„Nie každý je Slovák.“* či *„Urobili to nie z lenivosti.“*

Napriek tomu však existuje výnimka, kedy je slovo *nie* súčasťou predikátu. Ide o prípady, kedy je naviazané na sloveso *byť*. Príkladmi sú vety: *„Doska nie je drevená.“* alebo *„Každý nie je dokonalý.“*

Nie bolo zámerne označované ako slovo, pretože môže vystupovať ako spojka, ale aj ako častica. Vtedy z hľadiska rozsahu negácie plní rôzne funkcie. Ako častica sa nachádza práve pri členskej forme negácie, teda *„Nie neprávom sa hnevá.“* Ako priraďovacia spojka vystupuje v spojeniach *„Nie že by sa polepšil.“* alebo *„Keď už nie maľovať, tak aspoň kresliť.“*

3.3.3 Negátor *bez*

Slovu *bez* je medzi negátormi prikladaná mimoriadna váha, čomu nasvedčuje aj článok venovaný výhradne jeho postaveniu [36]. Zaraďuje sa medzi prostriedky gramatickej negácie s troma rozličnými funkčnými hľadiskami, ktoré však pre túto prácu nie sú významné, pretože v závislostnom strome vždy vytvoria rovnakú štruktúru podstromu. V tejto práci bude rovnakým spôsobom ako ku predložke *bez* pristupované ku takzvaným funktorom nepriamej reštrikcie, ktorými sú predložky *mimo, okrem, namiesto, nehľadiac* a iné.

3.3.4 Iné negátory

Pavlovič ďalej definuje ako samostatnú kategóriu popieracie modálne slovesá, avšak pre túto prácu budú modálne slovesá patriť do kategórie *negačný prefix*. Významnú množinu negátorov tvoria poperacie častice typu *figu*, ktoré však v technických či publicistických textoch nehrajú veľkú rolu. Prínos môžu mať, ak je cieľom spracovať beletriu alebo text citujúci voľnú reč.

O niečo formálnejšie sú popieracie príslovky *nemožno, neradno, nedajbože, neslobodno, netreba, nehodno, nevidno, nevedno*. Lexikálne však nie sú odlišné od ostatných negátorov so záporným prefixom, pretože takmer pre všetky z nich mnohé z existuje kladná dvojica, hoci nie často používaná. Príkladom sú existujúce pozitívne slová *radno, dajbože* či *hodno*.

3.3.5 Sentiment ako polarizátor

Úplne iný pohľad na detekciu negátorov však prináša sémantická rovina každého slova. Obzvlášť veľký význam má sémantika slovies, ktorá môže modifikovať polaritu celej výpovede. Ako príklad poslúžia vety: *„Peter priznal vinu.“* a *„Peter poprel vinu.“* Je zjavné, že sloveso *poprieť* nejaví po lexikálnej stránke žiadne príznaky negácie. Avšak v sémantickej rovine dodáva dvojici slov *Peter* a *vina* vylučovací vzťah.

Charakter tohto javu sa dá pripísať sentimentu alebo citovosti slova. Výskum v oblasti sentimentu pre anglický jazyk dosiahol významné výsledky, avšak v slovenčine v súčasnej dobe (december 2016) neexistuje spôsob, ako odhaliť, či sloveso vyjadruje kladný alebo

záporný jav. Slovesá typu *poprieť, zanedbať, vynechať* a mnohé iné je preto možné zohľadniť len na úrovni zoznamu výnimiek. Pavlovič ich klasifikuje ako funktory z bázy lexikálneho významu a radí pod implikatívnu negáciu.

3.3.6 Rozsah negácie a viacnásobná zmena polarity

Každý negátor sa vzťahuje na jedno alebo viacero slov vo vete, pre ktorá otáča sémantickú polaritu. Nie je neobvyklé, ak je negátorov vo vete viac, a dokonca ani to, ak menia polaritu toho istého slova. Ako príklad slúži veta: *„Bez neho by sme neodišli.“*, ktorá má význam *„Odišli sme aj s ním.“* Cieľ negácie oboch negátorov *bez* a *neodišli* je z kontextu zjavný, teda ide o zámeno *on*.

Iným spôsobom viacnásobnej negácie je využitie združených prostriedkov. Namiesto *„Všetci neprišli.“* sa používa *„Nikto neprišiel.“* Zámená *nikto, nič, nikdy, nijako, nijaký* sa nazývajú totalizátory, avšak Pavlovič im nepripisuje funkciu plnohodnotných negátorov. Spravidla sa vo vete nachádzajú len ako spoluoperátory negácie.

3.4 Negácia v anglickom jazyku

Podobne ako v slovenskom jazyku, aj pre anglický jazyk je potrebné rozoznávať obe fázy procesu negácie, teda negátor a rozsah negácie. Táto kapitola má za cieľ opísať možné podoby negátorov v anglickom jazyku spolu s príkladmi ich uplatnenia, pričom vychádza z publikácie [18]. Záver je opäť venovaný spôsobu kooperácie viacerých negátorov.

3.4.1 Negátor *not*

Základným prostriedkom negácie je slovo *not*, ktoré môže stáť samostatne alebo byť pridružené ku modálnemu slovesu ako *n't*. Príkladmi sú dvojice *does not* → *doesn't*, *will not* → *won't*. Pri iných ako modálnych slovesách lexikálne pridruženie vo forme sufixu nie je možné, negácia je vykonávaná v kompozícii s modálnym slovesom.

Podobne ako slovenské *nie*, aj *not* sa môže viazať na rôzne vetné členy. Zatiaľ čo vo vete *„He does not write.“* ide o vzťah ku predikátu, vo vete *„Not one of them knew what to do.“* zasa o vzťah ku podmetu. Môže stáť aj pri rozvíjacom vetnom člene, ako v prípade atribútu vo vete *„This coffee was not good.“*

Negačný variant zdvojených spojok v angličtine je *neither - nor*, pričom obidve spojky zohrávajú rolu negátora vo svojej časti súvetia. Príkladom je *„He neither looks like a gentleman, nor speaks proper English.“*

3.4.2 Negátor *n-*

Anglický jazyk hojne využíva polarizované zámená. V pozitívnych oznamovacích a zvolacích vetách sa používajú slová *some, someone, something, somewhere, sometimes*. V opytovacích vetách sa menia do podoby *any, anyone, anything, anywhere, ever*.

Napokon v negatívnych vetách ide o *no, noone, nobody, nothing, none, nowhere, never*. Narozdiel od slovenčiny, negácia je tvorená výlučne týmito prostriedkami, bez prítomnosti ďalšieho modifikátora výpovede.

Na porovnanie poslúži slovenská veta *„Nikto nevedel, čo sa stalo.“* Hoci je prítomné slovo *nikto*, hrá len rolu intenzifikačného prostriedku voči negátoru *nevedel*. Oproti tomu v anglickej vete *„Nobody knew what had happened.“* je *nobody* jediným modifikátorom výpovede.

3.4.3 Prepozičné negátory

V slovenskom jazyku ide najmä o predložku *bez*, ktorá má rovnako významnú rolu aj v anglickom jazyku ako *without*. Typicky modifikuje polaritu najbližšieho podstatného mena alebo zámena. Príkladom je veta: *„He's walking without a stick.“*

Ak je predložka *without* použitá v podraďovacom súvetí, takmer vždy obsahuje podradená veta ďalší negátor, napríklad: *Without his help, we wouldn't be here.* V takom prípade je však podradená časť vety *his help* braná pozitívne s významom: *„We are here thanks to his help.“*

V niektorých prípadoch predložkové negátory naberajú svoju funkciu iba vo dvojiciach, akými sú *apart from, outside of* a iné. Ďalším prípadom je prepojenie predložky so slovom *no*, teda *for no, with no* a podobne, tu sa však slovo *no* typicky nachádza v podraďovacom vzťahu voči cieľu negácie.

3.4.4 Adjektívne negátory

Lexikálna negácia prídavných mien prebieha prostredníctvom pridania jedného negačných prefixov alebo sufixov, ktorými sú typicky *un-, dis-, im-, in-, a-, -less*. Príklady tvoria slová *uncooperative, displeased, improper, inconsistent, asymmetrical, senseless*.

V prípade anglického jazyka však nie sú so všetkými týmito vzormi také výrazné problémy ako so slovenskými negačnými prefixami. Sufix *less* v prídavnom mene zaručene znamená zápor. Avšak stále existujú slová ako *unitary, adjacent* či *initial*, ktoré prefixové vzory narúšajú.

3.4.5 Analýza sentimentu pri detekcii negátora

Podobne ako v slovenskom jazyku, aj v anglickom je významná báza implikatívnych slovesných negácií. Medzi tie sa radia *deny, fail, hate* a mnohé ďalšie. Detekcia sentimentu pre anglický jazyk však prináša možnosti opísané v časti 2.8, ako takéto slová odhaliť a následne ich zaradiť medzi negátory.

„Peter denies guilt.“ je významovo ekvivalentné s tvrdením *„Peter does not accept guilt.“* Azda najčastejšie citované negované tvrdenie z korpusu BioScope 3.5.3 je *„Patient denies chest pain.“*, pričom takéto vzory sú v medicínskych textoch časté. Analýza sentimentu pomocou dostupných prostriedkov teda môže dopomôcť ku vybudovaniu efektívnejšieho algoritmu detekcie negácie pre anglický jazyk.

3.4.6 Dvojitá negácia

Dvojitá negácia v anglickom jazyku vzniká jedine spojením vetnej a členskej negácie, pretože inak to gramatické pravidlá nepovoľujú. Výpoveď *„Peter does not deny guilt.“* však nadobúda význam nejednoznačnosti. Obdobne je to vo vete: *„She was not unattractive.“*

Inu možnosťou je, keď výpoveď zdôrazňuje negatívny dopad nevykonania nejakej činnosti. Príkladom je veta: *„We cannot not go to sleep.“*, ktorá vypovedá o tom, že neísť spať nie je akceptovateľné. So zachovaním sémantiky by sa dala prepísať na *„We have to go to sleep.“* Všeobecné pravidlo pre anglický jazyk znie, že prítomnosť dvoch alebo viacerých negátorov zachováva kladnú polaritu výpovede.

3.5 Textové korpusy na vyhodnotenie

Aby bolo možné ľubovoľnú metódu evaluovať, je potrebná rozsiahla kolekcia dokumentov. Táto sekcia prináša prehľad korpusov, ktoré už v minulosti boli použité na overenie riešení z oblasti spracovania prirodzeného jazyka alebo špecificky negácie.

3.5.1 Slovenský národný korpus

Slovenský národný korpus (skr. *SNK*) je najväčšia slovenská elektronická kolekcia dokumentov, ktorá obsahuje jazykové anotácie. Štýlovo ide o odborné texty, ale aj beletriu od rôznych autorov počínajúc rokom 1955[1].

Každý token je v SNK morfologicky anotovaný a ak to pre daný token má zmysel, je k nemu pripojená aj lema. Zároveň obsahuje pomocné informácie o tom, či je token správne

[1] http://korpus.juls.savba.sk/

alebo chybne zapísaný, či ide o skratku, citát, vlastné meno, číslicu, interpunkciu a rôzne iné.

V prípade negovaných slovies je lema uvádzaná v pozitívnom význame, teda slovo *„nerobil“* je transformované na *„robiť“*. V istej časti korpusu sa pri slovesách uvádza ich význam, znak + znamená kladný a - záporný tvar.

Takáto anotácia slovies však vychádza iba z prítomnosti negačného prefixu, a teda nevyjadruje polaritu slovesa na základe sentimentu. Príkladom je, že sloveso *poprieť* je definované ako kladné, zatiaľ čo antonymické sloveso *nepoprieť* je anotované ako záporné.

Podstránka morfologickej analýzy[2] obsahuje podrobný opis morfologických značiek pre každý druh tokenu. Príkladom je morfologická značka pre podstatné meno, ktorá má vždy dĺžku 5 znakov. Na prvej pozícii je informácia o slovnom druhu, na druhej o paradigme, na tretej o rode, na štvrtej o čísle a na piatej o páde. Hodnoty na daných pozíciách sú vymenovaným typom, pričom všetky možnosti sú opísané v dokumentácii.

Komunikácia s korpusom prebieha prostredníctvom webového rozhrania alebo SSH. Celkovo databáza obsahuje 1 250 382 876 tokenov. Bohužiaľ, anotácie pre negáciu sú zahrnuté iba pre slovesá a ani tie nie sú vhodné pre účely tejto práce, preto Slovenský národný korpus v súčasnej podobe nie je možné priamo využiť na evaluáciu detekcie negácie.

3.5.2 Pražský závislostný korpus

Pražský závislostný korpus[3] obsahuje 2 milióny morfologicky anotovaných slovných jednotiek, z toho 1,5 milióna je aj syntakticky anotovaných. Slúži ako hlavný evaluačný prostriedok pre závislostné algoritmy, ktoré sú jedným z dvoch funkčných prístupov pre syntaktickú analýzu v českom jazyku, ako opisuje sekcia 2.5.

Obsah korpusu je možné anonymne prezerať cez webové rozhranie, ktoré poskytuje základnú funkcionalitu. Pre úplný prístup cez nástroj PML Tree Query je potrebná registrácia. Ani tento korpus však neobsahuje anotácie súvisiace s negáciami.

3.5.3 BioScope

Pracovná skupina maďarských autorov vytvorila anotácie neurčitosti a negácie v kolekcii anglických dokumentov z medicínskeho prostredia[4]. Korpus BioScope tvorí rozsiahla skupina klinických textov, abstraktov a vedeckých článkov. Slúžil ako evaluačný prostriedok pre algoritmus detekcie negácie v anglickom jazyku.

[2] http://korpus.juls.savba.sk/morpho.html
[3] https://ufal.mff.cuni.cz/pdt3.0
[4] http://rgai.inf.u-szeged.hu/index.php?lang=en&page=bioscope

Autori si dávali za cieľ vyznačiť neurčitosť a zápor s ohľadom na pojmy z oblasti medicíny. Z toho dôvodu nie sú v korpuse vyznačené všetky bežné druhy negácií, iba také, ktorých rozsah pokrýva iné slová. To znamená, že v korpuse by nemali byť vyznačené negácie prídavných mien, ktoré v anglickom jazyku majú typicky lokálny rozsah, teda negujú samé seba.

3.5.4 Conan Doyleneg

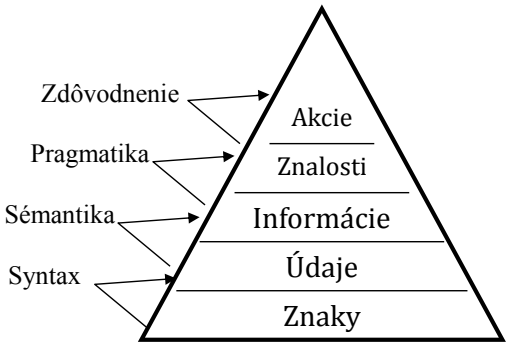
V roku 2012 vyšiel korpus v anglickom jazyku s vyznačenými negáciami a rozsahom negácie [34]. Použité boli príbehy Conan Doyla *Pes baskervillský* a *Vila Vistária*. Korpus je voľne dostupný[5]. V tomto prípade ide žánrovo o beletristické texty, v ktorých by mali byť vyznačené všetky druhy negácií. Pre tento účel boli definované kompozitné slová, teda slová zložené z viacerých častí (predpona a základ slova).

[5]http://www.clips.ua.ac.be/BiographTA/corpora.html

4 Analýza uplatnenia detekcie negácie

Spracovanie prirodzeného jazyka (angl. *natural language processing*, skr. *NPL*) má širokú škálu využitia naprieč celým spektrom informačných domén. Či už ide o formulovanie hlasových dopytov pre vyhľadávač, štatistické textové operácie alebo spracovanie textových dokumentov za účelom ich lepšieho pochopenia.

Bergmann vo svojej publikácii [4] definuje viacero úrovní porozumenia textovým hodnotám, ako to znázorňuje diagram 4.1. Základnou úrovňou je prúd znakov, ktoré s využitím tokenizácie prechádzajú v celistvé údaje. Vďaka sémantickej rovine sa údaje menia na informácie. Ak sa hovorí o čísle 5 ako o údaji, vďaka pridanej sémantickej rovine vieme tento údaj interpretovať ako informáciu o veku, teplote alebo počte prstov.



Obr. 4.1: Úrovne pochopenia textu podľa Bergmanna. Diagram adaptovaný z [24].

Detekcia negácie hierarchicky patrí na úroveň transformácie údajov na znalosti. Vďaka označeniu, či danú informáciu treba interpretovať ako kladnú alebo zápornú, môže dôjsť ku lepšiemu pochopeniu textu ako celku. Jednou z domén uplatnenia detekcie negácie je extrakcia kľúčových slov, pojmov či fráz z textu.

Podkapitola 4.1 analyzuje niekoľko vybraných algoritmov extrakcie a uvádza spôsob, ako do nich zakomponovať detekciu negácie. Výsledky extrakcií sú ďalej využiteľné pri snahe o lepšie pochopenie softvéru, ako to diskutuje podkapitola 4.2.

4.1 Extrakcie z textu a aplikácia negácie

História extrakcie kľúčových slov z dokumentov siaha do druhej polovice 20. storočia. V práci [46] bol predstavený štatistický model váhovania slov na základe frekvencie ich výskytu. Následne bolo možné porovnať výsledky s frekvenciami jednotlivých slov v referenčnom korpuse.

Základnou mierkou, ktorá je adaptáciou tohto princípu, je TF-IDF opísaná v sekcii 4.1.1. Všetky odvíjajúce sa mierky sú postavené na otázke, kedy je slovo dôležité. V dokumente sa totiž môžu nachádzať slová s vyšším výskytom, ktoré však preň nemajú veľký význam. Namiesto toho môžu byť dôležitými práve slová s nízkou frekvenciou.

Nevýhodou všetkých štatistických modelov je ich závislosť od referenčného korpusu. Ďalšou nevýhodou je nutnosť orientovať extrakciu na jednotlivé slová namiesto celých pojmov, čo môže v niektorých prípadoch viesť ku nežiadúcim výsledkom.

Súčasnou skúmanou oblasťou sú algoritmy schopné extrahovať kľúčové slová z jednotlivých dokumentov bez nutnosti referenčného korpusu. Algoritmus založený na Helmholtzovom princípe bol predstavený v publikácii [5] a opisuje ho sekcia 4.1.2.

V kontexte detekcie negácie sa však orientácia na jednotlivé pojmy javí ako nedostatočná. Preto je v časti 4.1.3 opísaná metóda orientujúca sa na veľmi jednoduchú extrakciu fráz zvoleného rozmeru, teda takzvaných *n-gramov*. Jej ďalším vylepšením je extrakcia syntaktických n-gramov alebo *sn-gramov*.

4.1.1 TF-IDF

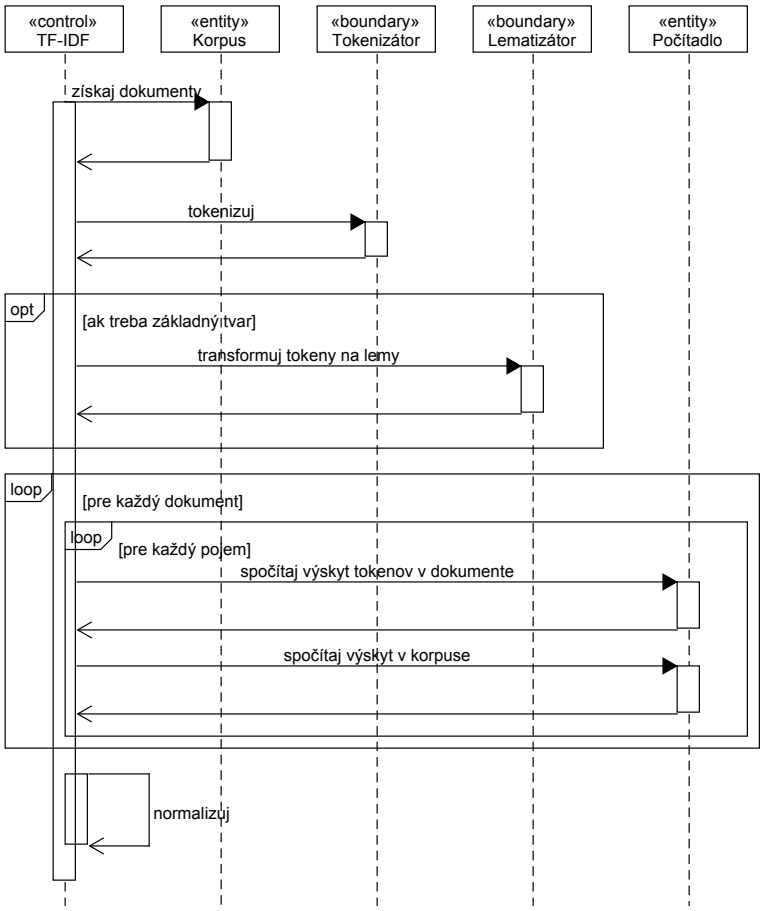
TF-IDF (angl. *term frequency - inverse document frequency*) je štatistický model, ktorý opisuje dôležitosť každého pojmu z vybraného dokumentu vzhľadom na väčšiu množinu alebo kolekciu dokumentov. Zložka TF vyjadruje frekvenciu výskytu daného slova vo vybranom dokumente, zatiaľ čo IDF definuje počet dokumentov, ktoré toto slovo obsahujú aspoň raz.

Hlavnou myšlienkou takéhoto rozdelenia na dve zložky je, že spojky, predložky a iné neplnovýznamové slová majú síce vysokú frekvenciu výskytu, zároveň sú však obsiahnuté vo veľkom množstve referenčných dokumentov. Čím je vybrané slovo častejšie použité v dokumente, tým sa zvyšuje jeho dôležitosť. Ak je však často použité vo všetkých dokumentoch, potom ide o bežnú súčasť lexiky a jeho dôležitosť je nízka.

Z uvedenej definície vyplýva poznatok, že TD-IDF je alternatívne možné použiť na odstránenie takzvaných *stop slov*. Ide najmä o slová nenesúce výpovednú hodnotu, napríklad *a*, *ale*, *preto* a ďalšie.

Vstupom algoritmu je tokenizovaný text, pričom tokeny v závislosti od domény použitia

môžu byť transformované na základný tvar, ale aj nemusia. Diagram 4.2 znázorňuje sled volaní jednotlivých systémových komponentov, ktorých výsledkom je normovaná váha.



Obr. 4.2: Kroky algoritmu TF-IDF.

Dôležitým krokom pred spojením zložiek TF a IDF je ich normalizácia. Pre zložku TF to znamená transformovať absolútne počty výskytu pojmov na relatívne hodnoty s maximálnou hodnotou 1. To znamená, že ak by mal dokument tri slová, každé by získalo zložku TF rovnú $0,33$. Týmto sa eliminujú rozdiely v dĺžke jednotlivých dokumentov, ktoré prirodzene nemusia byť zhodné.

Váhovanie zložiek a úpravy v procese normalizácie sú predmetom výskumu už desaťročia. Príkladom je práca [9] z júla 2015, ktorá porovnáva váhovacie schémy z ostatných rokov a zároveň prezentuje vlastnú normalizáciu.

Za jednu zo štandardných váhovacích schém sa považuje logaritmická. Princípom je, že ak je jedno slovo v dokumente zastúpené desaťkrát a v inom stokrát, nemá pre ten druhý dokument desaťkrát väčšiu dôležitosť.

Vzorec na výpočet pomocou logaritmickej váhovacej schémy je $tf - idf_{t,d} = (1 + \log tf_{t,d}) \cdot \log \frac{N}{df_t}$. Pri úvahe o detekcii negácie sa javí ako vhodné riešenie upraviť mierku *TF* tak, aby zohľadňovala počet výskytov slova v dokumente v kladnom a zápornom významne.

Upravený vzorec by mohol mať podobu $tf - neg = \frac{tf_{positive}}{1+tf_{negative}}$. Ak sa v dokumente nachádza slovo trikrát, z toho v zápornom význame ani raz, mierka TF zostáva nezmenená. Ak sa však nachádza jedenkrát v kladnom a jedenkrát v zápornom význame, jeho dôležitosť je významne nižšia. Takýto prístup sa konceptuálne javí ako lepší než jednoducho vziať do úvahy len pozitívne výskyty slova, pretože explicitná indikácia záporného významu v dokumente zvyčajné má zdôvodnenie.

Z hľadiska implementácie ide o pomerne nenáročný algoritmus. Existujú však aj pokusy o jeho nadstavbu v podobe strojového učenia. Jedným z predstaviteľov je práca [49], ktorá prezentuje algoritmus strojového učenia založený na *TF-IDF* a zdrojových dokumentoch, ku ktorým boli kľúčové slová vybrané manuálne. Výcvikovú množinu tvorilo 20 dokumentov s vybranými kľúčovými slovami a referenčný korpus pre frekvenciu výskytu slov obsahujúci len 10 dokumentov, ktoré stačili na vytvorenie algoritmu kvantitatívne konkurujúcemu štandardnej mierke TF-IDF.

4.1.2 SDE

Japonskí autori prezentovali prácu [30], podľa ktorej je možné extrahovať kľúčové slová z jediného dokumentu bez nutnosti referenčného korpusu. Z tohto faktu pramení aj označenie algoritmu ako SDE (angl. *single document extraction*), v preklade extrakcia z jediného dokumentu.

Namiesto porovnávania často sa vyskytujúcich slov s ich frekvenciou v korpuse, tento prístup sa zameriava na počítanie usporiadaných n-tíc slov. Ak teda nejaká dvojica, trojica či väčšie zoskupenie slov stojí viackrát vedľa seba, ide z pohľadu tohto algoritmu o dôležité slová.

Postup algoritmu:

1. Vstupom algoritmu sú stemované tokeny, ktoré sa usporiadajú podľa frekvencie výskytu a ponechá sa horných 30 % označených ako N_{total}.

2. Zhlukujú sa dvojice pojmov, ktorých Jensenova-Shannonova divergencia [12] je väčšia než $0.95 * log2$. Získané zhluky sa označia ako C.

3. Vypočíta sa očakáva pravdepodobnosť výskytu pojmu. Pre každý pojem $c \in C$ sa vezme počet jeho spoluvýskytov s ostatnými pojmami n_c a vydelí sa počtom ponechaných pojmov, teda $p_c = \frac{n_c}{N_{total}}$.

4. Výpočet rozdelenia χ^2 podľa doteraz získaných hodnôt. Schéma je obsahom prezentovaného článku.

5. Pojmy s najvyššími hodnotami χ^2 sú kľúčovými slovami.

Podľa evaluácie vykonanej na dátovej množine s 20 000 pojmami sú odchýlky presnosti voči klasickému TF-IDF rádovo v jednotkách percent. Hodnoty χ^2 by opäť mohli byť vynásobené koeficientom zápornosti $\frac{tf_{positive}}{1+tf_{negative}}$, čo by zmenilo zoradenie výslednej množiny kľúčových slov s prihliadnutím na negáciu.

4.1.3 N-gramy

N-gram sa dá definovať ako spojitá postupnosť n položiek v danej sekvencii. Môže ísť o rôznu úroveň textovej abstrakcie, napríklad o znaky alebo o slová. Pre potreby extrakcie fráz z dokumentov ide typicky o sekvencie slov, ktoré sa v texte nachádzajú vedľa seba.

Spracovanie textu na úrovni slov je v podstate špeciálnym prípadom n-gramov, kde n je rovné 1, teda ide o *unigramy*. *Bigramy* majú potom dĺžku 2, *trigramy* dĺžku 3. Pre vybrané úlohy spracovania prirodzeného jazyka sú n-gramy dôležité vo forme modelov. Na základe nich sa dá, napríklad, určiť najpravdepodobnejšie slovo, ak je úlohou vytvoriť z bigramu trigram [22].

Pre potreby zohľadnenia prítomnosti negácie pri opise dokumentov to však nie je potrebné. Tradičný prístup ku extrakcii n-gramov spočíva v tom, že pre danú dĺžku n sa berú $n-tice$ slov stojace vedľa seba. Trigramy z vety *„Peter nerád hovorí o politike.“* by boli *„Peter nerád hovorí“*, *„nerád hovorí o“*, *„hovorí o politike“*.

Jednoduchou metódou na zohľadnenie výskytu n-gramov v korpuse dokumentov je použiť TF-IDF na každé slovo tvoriace n-gram a spojiť ich pomocou geometrického priemeru. Reálne sa však používajú skôr štatistické metódy [23]. Odfiltrovaním fráz s výskytom negátora a slov v jeho rozsahu by bolo možné dospieť len ku takým častiam textu, ktoré sú uvádzané v pozitívnom význame.

4.1.4 SN-gramy

Vzhľadom na detekciu negácie prostredníctvom syntaktických závislostí boli preskúmané aj možnosti extrakcie n-gramov prostredníctvom závislostného stromu. V práci [45] bolo ukázané, že syntaktické n-gramy alebo *sn-gramy* dosahujú vyššiu korektnosť výsledkov, pretože zohľadňujú vzťahy medzi slovami. To, že slová stoja vedľa seba, najmä v jazykoch s voľným slovosledom nemusí mať vysokú relevanciu.

Namiesto n slov stojacich vedľa seba sú preto brané do úvahy prechody v syntaktickom strome s dĺžkou n. Výhodou je, že pri tejto metóde sa dá jednoducho aplikovať poznatok o prítomnosti negácie. Jednoduchým spôsobom by bolo pre všetky slová tvoriace n-gram preskúmať, či a koľkokrát sú rozsahom negácie. Otáčaním polarity by potom bolo možné dospieť k rozhodnutiu, či n-gram na základe obsiahnutých slov extrahovať alebo nie.

4.2 Uplatnenie pri spracovaní softvérových artefaktov

IEEE v článku [1] definuje softvér ako *zbierku programov, procedúr, pravidiel a súvisiacej dokumentácie patriacu ku systému na spracovanie informácií*. Softvérové inižinierstvo je následne v rovnakom článku definované ako *systematická aplikácia vedeckých a technologických znalostí, metód, skúseností pri návrhu, implementácii, testovaní a dokumentovaní softvéru*. V priebehu životného cyklu potom vznikajú softvérové artefakty, ktorými môžu byť diagramy, modely, návrhové dokumenty, plány a podporné materiály, ktoré sa dajú považovať za súčasť softvérovej dokumentácie.

Softvérové systémy sa v priebehu desaťročí presunuli zo súborov jednoduchých príkazov do formy komplexných ekosystémov, ktoré majú byť nielen byť strojovo vykonateľné, ale zároveň vybudované okolo sémantickej roviny. Práve sémantika v kontexte softvérových systémov výrazne vystupuje do popredia a venujú sa jej mnohé práce, ktorých predstaviteľom je článok Iaakova Exmana [19].

Pochopenie sémantickej roviny softvéru však nie je triviálna úloha. Pri snahe o nájdenie podstaty softvéru sa často siaha ku softvérovým artefaktom, na ktoré sa aplikujú rôzne úlohy spracovania prirodzeného jazyka. Na základe ich výstupov je potom možné definovať účel softvéru, znalosti obsiahnuté v softvére a dokonca vybudovať infraštruktúru na ich inteligentné znovupoužitie.

Práca [41] sa venuje extrakcii, identifikácii a vizualizácii obsahu, používateľov a autorov v softvérových projektoch. Na tieto účely používa mnoho krokov a prostriedkov spomenutých v tejto práci, či už ide o extrakciu čistého textu, predspracovanie či využitie synonymických vzťahov WordNet. Po dokončení základného spracovania deteguje v artefaktoch témy prostredníctvom metriky TF-IDF. Ako uvádza časť 4.1.1, jednoduchou úpravou vzorca pre výpočet tejto mierky je možné zahrnúť do procesu negáciu, čím by došlo ku spresneniu detekcie tém.

V ďalšej práci [40] sa kolektív autorov venuje extrakcii pomenovaných entít priamo zo zdrojového kódu, na čo je opäť použitý štandardný proces spracovania prirodzeného jazyka. Pomenované entity sú následne zoskupené do syntaktických stromov na základe autorstva. Následne pridáva ku blokom kódu vlastné značenie a komentáre, v ktorých sa neskôr

snaží vyhľadávať. Detekcia negácie v tejto časti práce by opäť dopomohla ku lepšiemu vyhľadávaniu v značení a komentároch, kedy by vyfiltrovanie výsledkov so záporne uvedenými skutočnosťami (napr. *Trieda nepatrí ku funkcionalite.*) mohlo zvýšiť efektívnosť práce a potenciálnu znovupoužiteľnosť.

5 Analýza existujúceho riešenia pre slovenský jazyk

V roku 2011 bol prvýkrát prezentovaný výskum v oblasti extrakcie kľúćových slov z dokumentov, ktorého špecifikom bolo uplatnenie negačných pravidiel pre slovenský jazyk [20]. Autor Ing. Martin Jaborník si dal za cieľ vylúčiť zo zoznamu článkov o digitálnych fotoaparátoch také, ktoré neobsahovali požadované kľúčové slovo, respektíve ho obsahovali v zápornom význame.

Táto kapitola prináša analýzu pôvodného návrhu a jeho druhej generácie, ktorá bola publikovaná v roku 2015 autorom Ing. Matejom Kvitkovičom [27]. Záver kapitoly sa venuje vymenovaniu nutných úprav systému, ktoré vyplývajú z analýzy.

5.1 Prvá generácia riešenia

Predložený systém je koncipovaný do piatich fáz, ktoré zodpovedajú postupom pre extrakciu kľúčových slov a detekciu negácie. Každá fáza je v tejto podkapitole opísaná spolu s technologickými prostriedkami, ktoré boli na jej realizáciu použité.

Príprava dokumentov spočíva v použití shareware nástroja ConvertDoc[1], ktorý dokáže vyňať čistý text zo súborov vo formáte Microsoft Word 2007. Tokenizácia je vykonaná programovo, pričom za oddeľovač boli brané biele znaky.

Rozdeľovanie súvislého textu na vety bolo takisto vykonané programovo zadefinovaním ukončovacích znakov *.!?:* a gramatických pravidiel písania viet. Tento prístup sa takto stáva závislým od gramatickej korektnosti vstupu. Autor opisuje problém identifikácie súvetí, ktorý vyplynul z nejednoznačnosti spojky *„a“* a gramatických pravidiel spojených s písaním čiarky pred spojkou *a*.

Lematizácia je vykonaná prostredníctvom databázy lem *form2lemma.cdb* vytvorenej Radovanom Garabíkom z JÚĽŠ. Autor navrhuje alternatívny prístup ku lematizácii pomo-

[1] http://www.softinterface.com/Convert-Doc/Convert-Doc.htm

cou preloženia slova do angličtiny, aplikácie Porterovho stemovacieho algoritmu a spätného preloženia do slovenčiny, ktorý ale neimplementoval. Ďalším krokom je odstránenie stop slov podľa manuálne vytvoreného zoznamu.

Modul pre extrakciu kľúčových slov implementuje metodiku TF-IDF opísanú v 4.1.1 aj algoritmus SDE opísaný v 4.1.2. Krokom navyše je zhlukovanie kľúčových slov do kategórií, ktoré využíva techniku *k-priemerov*.

Modul pre detekciu negácie pracuje na základe 6 definovaných pravidiel. Niektoré sa v texte vyhľadávajú cez regulárne výrazy, iné sú implementované podmienkami v rámci programového kódu. Evaluácia bola vykonaná na článkoch o digitálnych fotoaparátoch a dokumentáciách ku tímovým projektom.

Autor nepopisuje metodiku testovania, avšak tvrdí, že anotoval vybraných 30 dokumentov, v ktorých našiel 251 negácií. Implementovaný algoritmus správne detegoval 197 negovaných foriem a nesprávne identifikoval 13 foriem, z čoho vyšla presnosť 93 % a pokrytie 78 %.

Záver práce sa venuje nedostatkom, medzi ktorými sa nachádza nedostatočný počet pravidiel pre negáciu, problémy s nesprávnou detekciou negácie pri slovách začínajúcich na predpony *ne-*, *nie-* a náročnosť práce so súvetiami.

5.2 Druhá generácia riešenia

Druhá generácia projektu si dala viacero cieľov, pričom tým hlavným bolo skvalitnenie detekcie negácie v slovenskom jazyku. Táto podkapitola opäť opisuje technologické prostriedky, respektíve zmeny či vylepšenia v jednotlivých fázach procesu.

Príprava dokumentov bola ponechaná na externom shareware nástroji ConvertDoc. Zmenou oproti predošlej generácii je zavedenie formalizmu do reprezentácie textu, ktorý mal dopomôcť ku lepšej aplikovateľnosti pravidiel, ktoré sa predtým nachádzali v programovom kóde.

Bol pridaný kompilátor kompilátorov ANTLR, pomocou ktorého je možné tokenizovať dokument a vygenerovať syntaktický strom. Celá fáza predspracovania a detekcie negácie bola realizovaná priamo v tomto nástroji prostredníctvom tvorby gramatiky.

Hoci jedným z formulovaných cieľov bola tvorba vlastného POS značkovača, použitá bola externá webová služba Morpholyzer[2], ktorá v čase písania tohto dokumentu už nie je dostupná. Problém nejednoznačnosti opísaný v 2.3 rieši váhovaním podľa pravdepodobnosti výskytu slovných druhov.

[2]http://morpholyzer.fiit.stuba.sk:8080/PosTagger/xml/tag/all

Modul extrakcie kľúčových slov bol prebraný z prvej generácie a nebol nijako upravovaný. Modul detekcie negácie bol významne rozšírený, navrhnuté boli 3 formálne modely, z ktorých vyplynulo celkovo 10 pravidiel na implementáciu. Prototyp bol okrem toho čiastočne refaktorovaný, avšak ku oddeleniu aplikačnej logiky od prezentačnej nedošlo, preto sú zásahy doň naďalej komplikované.

Na evaluáciu boli použité rovnaké korpusy ako v prípade prvej generácie. Manuálne bolo označkovaných 50 dokumentov, v ktorých z 298 negovaných foriem bolo správne detegovaných 257 a nesprávne 46. Percentuálne vyjadrenia sú 84,82 % presnosť a 86,24 % pokrytie negácie. Rovnako ako v prípade prvej generácie, ani tu nie sú označkované dokumenty k dispozícii.

5.3 Možné vylepšenia

Prípravu dokumentov je vhodné nahradiť niektorým z univerzálnych nástrojov opísaných v sekcii 2.1, pričom ideálne sa na túto úlohu javí Apache Tika. Zmizla by tak závislosť na externom shareware programe, ktorý pri každom spustení vypisuje chybovú hlášku.

Vďaka pokroku v oblasti predspracovania slovenských textov je možné nahradiť lematizáciu aj morfologickú analýzu modernejšími prostriedkami. Lematizácia prostredníctvom *form2lemma.cdb* už nie je nutná, namiesto toho sa dá použiť webová služba[3] kombinujúca výhody slovníkového aj štatistického prístupu.

Na základe analýzy stavu morfologických analyzátorov v sekcii 2.4 je taktiež možné nahradiť neexistujúci Morpholyzer modernejšou službou, ktorá berie do úvahy vetný kontext na úspešnú elimináciu nejednoznačností. Z toho vyplýva, že štatistické váhovanie, ktoré bolo zdrojom nepresností, už taktiež nie je potrebné uvažovať.

Najvhodnejším riešením by však bolo nahradiť celé predspracovanie dokumentu balíkom nástrojov na spracovanie prirodzeného jazyka Synpar, ktorý bol opísaný v sekcii 2.6. Tým by opadla zbytočná réžia pri presúvaní dát z jedného externého nástroja do iného.

Obe generácie sa spoliehajú na manuálne vytvorené zoznamy stop slov a výnimiek. Alternatívnou je komplementárne riešenie, a teda ignorovať všetky neplnovýznamové slová, čo je možné vďaka presnejšiemu POS značkovaniu. Následne by sa vytvoril zoznam výnimiek so slovami so záporným významom cielene vytvorený pre potreby detekcie negácie.

Obaja autori hlásili problém so slovami začínajúcimi na predpony *ne-*, *nie-* a inými, teda s detekciou negátorov. Zároveň hlásia problémy s spracovaním súvetí. Tieto problémy rieši navrhovaná metóda v kapitole 6.

[3] http://text.fiit.stuba.sk/lemmatizer/

5.4 Zhodnotenie

Po vyhodnotení druhej generácie prototypu sa dospelo ku rozhodnutiu, že v rámci tejto práce sa doňho nebude zasahovať. Aplikačná logika je stále úzko zviazaná s prezentačnou logikou, autori sa venujú predovšetkým detekcii negátorov pomocou zoznamov výnimiek a rozsah negácie je úzko previazaný s extrakciou kľúčových slov.

6 Návrh detekcie negácie pre slovenský jazyk

Táto kapitola opisuje návrh vlastnej metódy na detekciu negácie v slovenskom jazyku. Metóda je rozdelená na dve samostatné časti, ktorými sú detekcia negátora a detekcia rozsahu negácie. Problematika negácie bola diskutovaná v časti 3, špecificky pre slovenský jazyk v časti 3.3.

Začiatok kapitoly sa zaoberá detekciou negátorov a prináša dva prístupy na detekciu negátorov so záporným prefixom. Obe metódy majú svoje výhody aj nevýhody, ktoré sú v príslušných častiach opísané. Väčšina kapitoly je venovaná detekcii rozsahu negácie prostredníctvom prechodov cez vetvy závislostného stromu.

6.1 Detekcia negátora so záporným prefixom

Na úspešné realizovanie systému na detekciu negácie je potrebné uvažovať všetky negátory, ktoré boli opísané v sekcii 3.3. Väčšinu z nich je možné detegovať priamo, pretože ide o neohybné slovné druhy (negátory *bez, mimo, okrem, nie* a iné. Problematika záporného prefixu bola podrobne opísaná v sekcii 3.3.1, z ktorej vychádza nasledujúci návrh dvoch rozličných prístupov.

6.1.1 Slovníkový prístup

Jednoduchý prístup, ako detegovať prítomnosť negátora so záporným prefixom, je skontrolovať každé slovo na prítomnosť tohto prefixu. V minulosti však autori [20] [27] hlásili problémy so slovami, ktoré sa síce záporným prefixom začínajú, ale nemajú záporný význam. Príkladom sú slová *nejaký, nemehlo* či *nevesta*.

Tento problém sa dá čiastočne vyriešiť odstránením negačného prefixu z lemy skúmaného slova, čím vznikne predpokladaný kladný tvar slova. Slovenský národný korpus vytvoril

Morfologickú databázu, ktorá obsahuje približne 99 000 unikátnych lem[1].

Po odstránení prefixu je teda možné skontrolovať v databáze lem, či sa v nej takéto slovo nachádza alebo nie. Týmto sa odfiltrujú slová *jaký, andertálec, mec* vzniknuté z pôvodných tvarov *nejaký, neandertálec, Nemec*. Nastáva však problém náhodnej zhody s inými slovami, ktoré sa lexikálne líšia iba negačným prefixom, ale sémanticky majú úplne odlišný význam.

Príkladom sú dvojice slov *nečudo* → *čudo*, *nevesta* → *vesta* a mnohé ďalšie. Keďže Morfologická databáza obsahuje aj morfologickú anotáciu každého slova, prvý prípad (*nečudo* → *čudo*) sa dá odfiltrovať zohľadnením slovného druhu, ktorý musí byť rovnaký pre obe slová z dvojice. Ak sú však obidve slová rovnakého slovného druhu (*nevesta* → *vesta*), sémantický rozdiel medzi nimi nie je možné detegovať.

Klasickou výhodou slovníkových prístupov je jednoduchá implementácia a rýchlosť. Vzhľadom na vývoj jazyka a spôsob konštruovania slovníkov je však potrebné predpokladať, že nie všetky slová sa budú v slovníku nachádzať. Ďalším problémom, ako už bolo spomenuté, je nejednoznačnosť prípadov, pretože môžu existovať dvojice líšiace sa iba negačným prefixom, ktoré sú rovnakého slovného druhu, ale sémanticky k sebe nepatria.

6.1.2 Prístup cez sémantické vzťahy *word2vec*

Alternatívnym prístupom, ktorý má potenciál vyššej robustnosti, je využiť vektorovú reprezentáciu slov prostredníctvom modelov *word2vec*. Všeobecný opis funkcionality *word2vec* priniesla časť 2.7.

Keďže model dokáže zachytiť sémantický vzťah medzi slovami, bolo by takýmto spôsobom možné vyhodnotiť aj slová so záporným prefixom. Hľadaný vzťah medzi slovami sa dal formulovať ako *Záporné slovo - Negácia* ≈ *Kladné slovo*. Problémom v tejto rovnici je však vektor reprezentujúci negáciu. K tomu je možné dospieť nasledujúcim procesom:

1. Nájdu sa dvojice slov (záporný prefix, bez záporného prefixu), ktoré k sebe sémanticky patria.
2. Pomocou *word2vec* sa nájde vektorová reprezentácia ich rozdielu.
3. Vektory rozdielu sa znormalizujú na jednotnú dĺžku.
4. Vytvorí sa priemer z týchto vektorov, ktorý sa prehlási za vektor reprezentujúci negáciu.

Výhodou tohto prístupu by mala byť vyššia robustnosť, kedy by boli správne zachytené aj vzťahy medzi slovami ako *vesta* a *nevesta*. Nevýhodou je, že na zachytenie vzťahov musí systém počas trénovania naraziť v trénovacej množine na každé slovo, ideálne viackrát. Dá

[1] http://korpus.juls.savba.sk/morphology_database.html

sa preto predpokladať, že úspešnosť tohto prístupu bude závislá od kvality a rozsiahlosti trénovacej množiny, z ktorej bol vytvorený model.

6.2 Detekcia rozsahu negácie prechodom závislostného stromu

Jadrom navrhovanej metódy je detekcia rozsahu negácie prostredníctvom vzťahov nadradenosti a podradenosti medzi vybranými dvojicami slov. Tieto vzťahy sú získané zo závislostnej analýzy, ktorej sa venovali časti 2.5.1 a o niečo technickejšie 3.2.4.

Hlavnou myšlienkou je, že závislostný strom zachováva vzťahy medzi slovami vo vete. Z hľadiska spracovania vety je potom bezpredmetné, či slovosled sleduje bežné pravidlá syntaxe alebo bol upravený prostredníctvom citovosti, ako je to bežné v jazykoch s voľným slovosledom.

Vstupom algoritmu je závislostný strom s vrcholmi, v ktorých sú vyznačené negátory a ich morfologické a syntaktické vlastnosti. Rozsah negácie je potom určený na základe prechodov, ktoré sú definované v nasledujúcich častiach.

Výhodou je najmä flexibilita tohto prístupu. Na demonštráciu každého prechodu bolo vybraných niekoľko príkladov, avšak prechody nie sú od stavby zvyšku stromu nijako závislé. Vždy je dôležitý iba spôsob prechodu od negátora ku cieľovému uzlu. Takto je možné detegovať v jednej vete viacero súbežných negácií, prípadne sa abstrahovať od toho, či ide o jednoduchú vetu alebo súvetie.

Ako ukazuje časť 6.2.6, zjednodušuje sa takto aj proces detekcie viacnásobnej negácie. Jedno slovo totiž môže patriť do rozsahu viacerých negátorov, a tým meniť svoju polaritu. Iným prípadom je, ak sa v rámci cesty v strome vyskytne viacero negátorov. Vtedy sa dá opäť uvažovať o obrátení polarity pre vybrané vrcholy vo vetve.

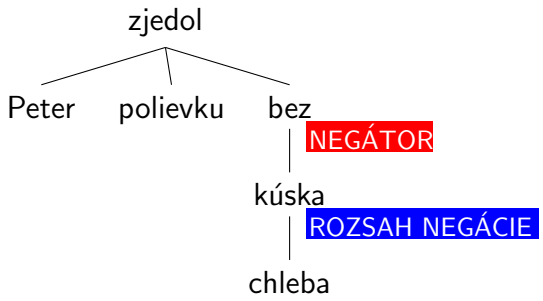
6.2.1 Negácia typu *genitívna predložka*

Základom je negátor typu predložky *bez* 3.3.3, ktorý sa spája s podstatným menom alebo zámenom v genitíve. Ako bolo spomenuté, z hľadiska skladby tento negátor realizuje tri funkcie, avšak závislostný podstrom je vždy rovnaký.

Tento prechod sa použije v prípade, že skúmaným vrcholom je negátor typu predložky *bez*, kam patria aj predložky *mimo, okrem* a iné. Rozsahom negácie je podradené podstatné meno alebo zámeno, teda najbližšie podstatné meno alebo zámeno v závislostných podstromoch od skúmaného vrcholu.

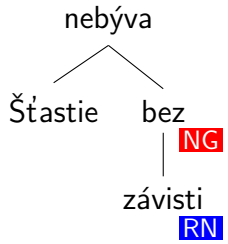
Vzdialenosť dvoch vrcholov je štandardne vyjadrená počtom hrán medzi nimi. Ak sa v najväčšej blízkosti nachádza viacero podstatných mien, do rozsahu negácie patria všetky.

„Peter zjedol polievku bez kúska chleba.“



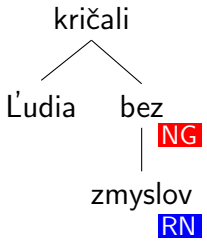
Obr. 6.1: *Bez* ako súčasť rozvíjacej pozície

„Šťastie bez závisti nebýva.“



Obr. 6.2: *Bez* ako súčasť mennej zložky prísudku.

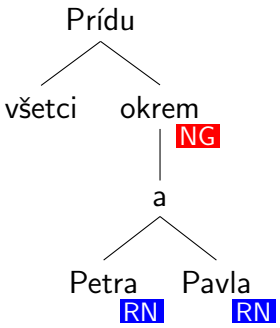
„Ľudia kričali bez zmyslov.“



Obr. 6.3: *Bez* ako súčasť príslovkového určenia.

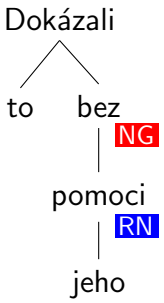
Nasledujúci príklad bol vybraný na demonštráciu prípadu, kedy sa v rovnakej vzdialenosti od skúmaného vrcholu nachádza viacero podstatných mien. Ide o negáciu viacnásobného vetného člena, kedy sa popierajú všetky z nich. Posledný prípad sa potom dotýka spôsobu, ako závislostný strom rieši prípadné nesprávne detegované negácie, ak sa ako rozsah negácie zohľadňuje aj zámeno.

„Prídu všetci okrem Petra a Pavla.“



Obr. 6.4: Predložka *okrem* negujúca viacnásobný vetný člen.

„Dokázali to bez jeho pomoci.“



Obr. 6.5: Predložka *bez* negujúca najbližšie podstatné meno alebo zámeno.

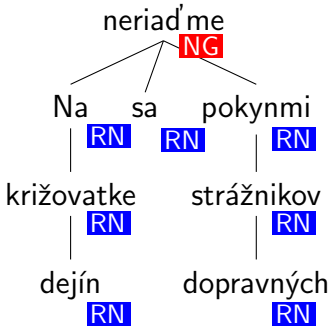
Vo vete *„Dokázali to bez neho.“* by sa najbližšie nachádzalo podradené zámeno *„neho“*, ktoré by správne bolo označené ako rozsah negácie. V prípade spojenia *„jeho pomoc“* však závislostná analýza správne identifikuje vzťah nadradenosti a podradenosti.

6.2.2 Negácia typu *predikát*

Negácia typu predikát nastáva, keď je ako negátor určené niektoré zo slov tvoriace prísudok. Pavlovič v takomto prípade hovorí o vetnej forme negácie, pričom sémanticky ide o popretie platnosti výpovede celej vety. Prechod je použitý, ak je skúmaným vrcholom negovaný prísudok.

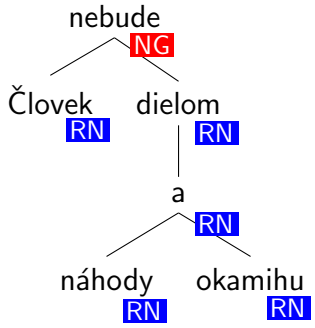
Rozsah negácie je v takom prípade každé slovo jednoduchej vety. Ak ide o súvetie, označí sa jednoduchá veta, v ktorej je prítomný negátor. V prípade podraďovacieho súvetia sa označí nielen nadradená veta s negátorom, ale aj podradená veta v prípade, že do nej má negátor sémantický presah. Ak sa negátor nachádza v podradenej vete, do nadradenej vety negátor nikdy nepresahuje.

„Na križovatke dejín sa neriaďme pokynmi dopravných strážnikov.“



Obr. 6.6: Negácia typu *predikát* so zamlčaným podmetom. Rozsahom negácie je celá veta.

„Človek nebude dielom náhody a okamihu.“

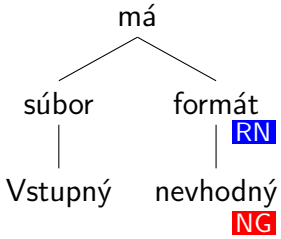


Obr. 6.7: Negácia typu *predikát*. Rozsahom negácie sú všetci potomkovia negátora.

6.2.3 Negácia typu *atribút*

Počiatočným vrcholom je uzol s negovaným atribútom, ktorý je takmer výlučne negovaný záporným prefixom. Atribútový vzťah je jedným zo základných, ktoré závislostná analýza spoľahlivo deteguje. Rozsah negácie je nadradené podstatné meno v zhode v páde.

„Vstupný súbor má nevhodný formát.“



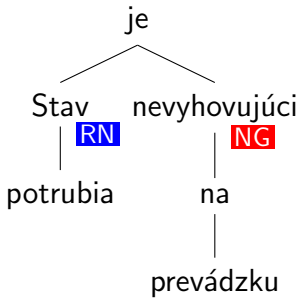
Obr. 6.8: Negácia typu *atribút*.

6.2.4 Negácia typu *odčlenený atribút subjektu*

Keďže slovenčina umožňuje voľný slovosled, poradie rozvíjacích vetných členov podmetu môže byť rôzne. Negácia tohto typu opäť berie ako počiatočný bod vrchol s negovaným atribútom. Ak sa nenájde predchádzajúci prechod, potom je vhodné použiť tento.

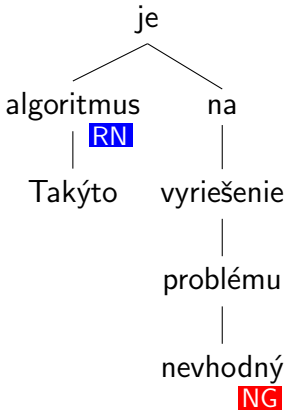
Rozsahom negácie je v takomto prípade podmet. Rozdiel medzi dvoma definovanými atribútovými prechodmi je prítomnosť podstatného mena v zhode v páde cestou od negovaného atribútu po koreň stromu.

„Stav potrubia je na prevádzku nevyhovujúci.“



Obr. 6.9: Negácia typu *odčlenený atribút subjektu*. Rozsahom negácie je podmet.

„Takýto algoritmus je na vyriešenie problému nevhodný.“



Obr. 6.10: Negácia typu *odčlenený atribút subjektu*. Rozsahom negácie je podmet.

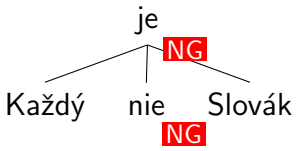
6.2.5 Negácia typu *nie*

Ako opisuje časť 3.3.2, negátor *nie* môže byť použitý v dvoch polohách. Prvou je negácia slovesa *byť*, kedy sa stáva súčasťou predikátu, druhou je členská negácia, kedy je rozsahom

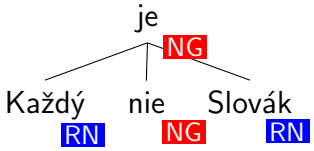
najbližšie podstatného meno alebo zámeno. Aby tieto dva spôsoby použitia bolo možné odlíšiť, je potrebné sledovať poradie vetiev vychádzajúcich z koreňa (pod)stromu.

Počiatočným bodom použitia prechodu je vždy vrchol so slovom *nie*. V prípade, že sa má stať súčasťou predikátu, je potrebné označiť aj ten ako negátor. Vtedy je poradové číslo anotovaného slova *nie* o 1 menšie než poradové číslo slovesa *byť* vo vete.

Keďže v spojení *nie je* vždy musia stáť pri sebe v takomto poradí, sémantické rozdelenie je možné vykonať už počas predspracovania. Prechod typu *nie* teda musí byť vykonaný skôr než prechod typu *predikát*, aby následne bolo možné aplikovať aj ten podľa 6.2.2.

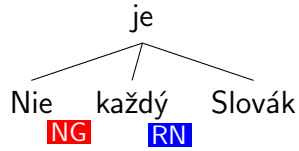


Obr. 6.11: Začlenenie slova *nie* do predikátu. Po prvom kroku je predikát označený tiež ako negátor.



Obr. 6.12: Uskutočnenie prechodu typu *predikát*.

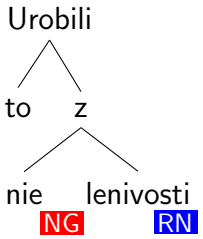
Druhý variant opisuje členskú negáciu. V takom prípade je rozsahom negácie vždy prvé slovo z prvého susedného podstromu, ktorý sa nachádza napravo od aktuálneho. Ide o vyjadrenie faktu, že *nie* neguje vždy najbližšie plnovýznamové slovo vo vete stojace napravo od neho.



Obr. 6.13: Členská negácia. Rozsah negácie sa určí prechodom do rodičovského uzla a vojdením do prvého pravého podstromu.

Posledný príklad ukazuje, ako si závislostná analýza poradí s vetou, v ktorej prvé slovo napravo od slova *nie* nie je plnovýznamové. Na porovnanie, práca [27] nezohľadňovala plnovýznamovosť prvého slova vpravo, preto by rozsah negácie nesprávne určila na neplnovýznamovú predložku *z*.

„Urobili to nie z lenivosti.“

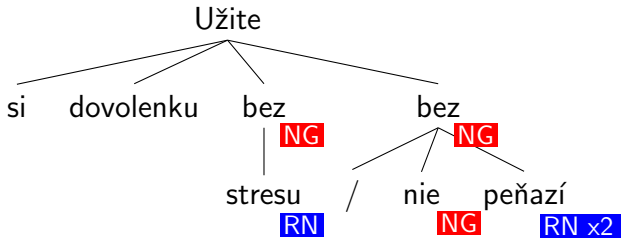


Obr. 6.14: Členská negácia slovom *nie*.

6.2.6 Dvojitá negácia

Nivelizácia negácie alebo oslabená afirmácia nastáva pri spojení vetného a lexikálneho záporu. Druhým spôsobom je spojenie slova *nie* s predložkou typu *bez*. Ak slovo vo vete dvakrát patrí do rozsahu negácie, jeho polarita sa stáva kladnou. Oba druhy dvojitej negácie navrhnutá metóda dokáže zachytiť bez špeciálnych upráv.

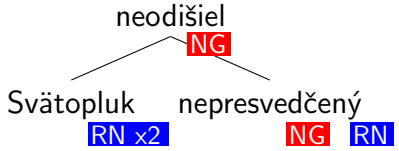
„Užite si dovolenku bez stresu, nie bez peňazí.“



Obr. 6.15: Dvojitá negácia spojením negátora *nie* a negátora *bez*.

Ako bolo spomenuté, predikatívna negácia sa deteguje ako posledná. Najprv sa aplikuje prechod zo slova *nepresvedčený*, ktoré vystupuje v roli odčleneného atribútového negátora. Následne sa deteguje negácia z prísudku *neodišiel*, ktorá podľa definovaného prechodu zasiahne podmet aj rozvíjací vetný člen.

„Svätopluk neodišiel nepresvedčený.“

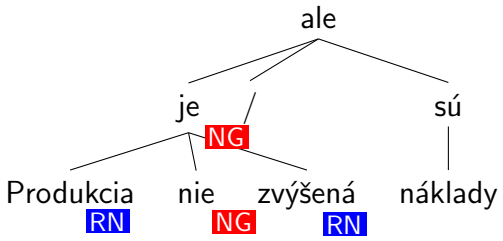


Obr. 6.16: Dvojitá negácia spojením vetnej a členskej negácie.

6.2.7 Presah jednoduchej vety

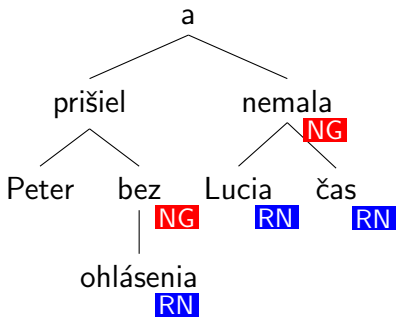
Ako bolo spomenuté, závislostná analýza dokáže súvetie rozložiť na samostatne stojace jednoduché vety. Preto nehrá rolu ani počet negátorov, ani dĺžka vety. Navrhované prechody sú opäť aplikovateľné bez nutnosti úprav, pretože sa orientujú na negované vrcholy vo vnútri stromu. Celková komplexita stromu nijakým spôsobom nevstupuje do aplikácie prechodov.

„Produkcia nie je zvýšená, ale náklady sú.“



Obr. 6.17: Detekcia negácie v podraďovacom súvetí.

„Peter prišiel bez ohlásenia a Lucia nemala čas.“



Obr. 6.18: Detekcia negácie v priraďovacom súvetí.

7 Návrh detekcie negácie pre anglický jazyk

Táto kapitola opisuje pokus o aplikáciu metódy 6 pre anglický jazyk, pričom vychádza z teoretických poznatkov opísaných v sekcii 3.4. Na začiatku sa venuje úlohe špecifickej pre anglický jazyk, a tou je analýza sentimentu s ohľadom na určenie negátorov. Dôležitosť tohto kroku opisuje časť 3.4.5. Zvyšok kapitoly sa opäť venuje definovaniu prechodov pre vybrané druhy negácií.

7.1 Detekcia negátora cez analýzu sentimentu

Analýza sentimentu vychádza z databázy SentiWordNet 2.8 pre anglický jazyk. Pre každú lemu je stanovené percentuálne vyjadrenie kladného a záporného emocionálneho prifarbenia. Takýmto spôsobom je možné overiť polaritu slovies, ale zároveň aj slov s negačným prefixom alebo sufixom.

Slovesá ako *deny, prevent, restrict* majú koeficient záporného prifarbenia 0,75 a kladného prifarbenia 0. Vďaka tomu je možné lepšie zohľadniť ich postavenie vo vete, kde vystupujú ako negátory. Podobné hodnoty majú aj slová *useless, improper, undoubtedly.*

Keďže ide o slovníkový prístup, platia preň rovnaké obmedzenia, ako boli opísané pri Morfologickej databáze. Práca s týmito dátami je veľmi rýchla a jednoduchá na implementáciu, avšak problémom je rozsah slovníka.

7.2 Detekcia rozsahu negácie prechodom závislostného stromu

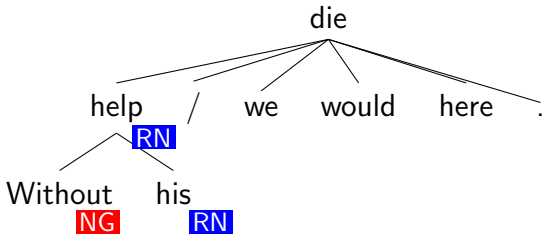
V tejto fáze projektu boli definované prechody len pre vybrané druhy negácií. Tie boli zvolené na základe toho, že na ich detekciu postačuje jednoslovný negátor. Ako bolo opísané pri jednotliých negátoroch v časti 3.4, v anglickom jazyku je mnoho negátorov zložených

z dvoch a viacerých slov, čoho príkladom sú spojenia *not one, apart from, for no* a ďalšie. Takéto druhy negácií metóda nezohľadňuje.

7.2.1 Negácia typu *predložka*

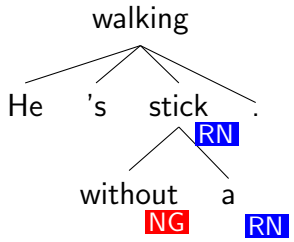
Počiatočným bodom je vrchol s jednoslovnou zápornou predložkou *without*. Rozsahom negácie je najbližšie nadradené podstatné meno alebo zámeno a jeho potomkovia.

„Without his help, we would die here.“



Obr. 7.1: Dvojitá negácia spojením vetnej a členskej negácie.

„He's walking without a stick.“



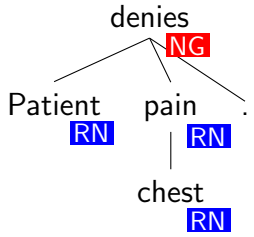
Obr. 7.2: Dvojitá negácia spojením vetnej a členskej negácie.

7.2.2 Negácia typu *predikát*

Negácia predikátom sa typicky vzťahuje opäť na všetky slová vo vete. Počiatočným bodom je záporný prísudok, rozsahom negácie sú všetci potomkovia negátora. Stromy použité na demonštráciu sú priamym výstupom použitého analyzátora. Tento prechod je zároveň jedným z miest, kde sa využívajú negátory získané cez analýzu sentimentu opísanú v časti 7.1.

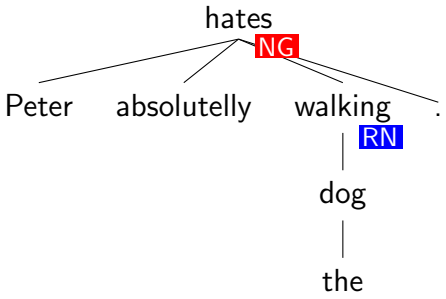
Pri tomto druhu negácie sú negátorom výlučne sémanticky detegované negátory bez negačných morfém, pretože iným spôsobom je jedine negácia tvorená modálnym slovesom *not*. Pri tej sa však uvádza takisto rozsah celej vety, preto rozdiel vo výslednom určení rozsahu medzi týmito druhmi negácii nie je.

„Patient denies chest pain.“



Obr. 7.3: Negácia typu predikát s rozsahom predmetu.

„Peter absolutelly hates walking the dog.“



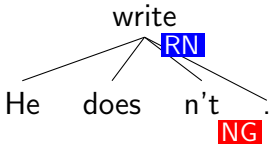
Obr. 7.4: Negácia typu predikát s rozsahom prechodníka.

7.2.3 Negácia typu *not* a *no*

Slovo *not* alebo *n't* môže plniť rozličné úlohy, avšak vo všetkých z nich je rozsah negácie rovnaký. To platí aj pre slovo *no*. Vstupným bodom je teda vrchol v závislostnom strome so slovom *not*, *n't* alebo *no*.

V prvom prípade ide o súčasť modálneho slovesa. To zahŕňa prípady *does not, doesn't, did not, didn't, will not, won't, could not, couldn't* a ďalšie pre všetky modálne slovesá a všetky časy. V takom prípade je rozsahom negácie nadradené slovo a praví susedia.

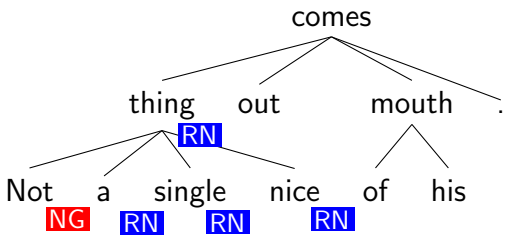
„He doesn't write.“



Obr. 7.5: Negátor *not* ako súčasť modálneho slovesa.

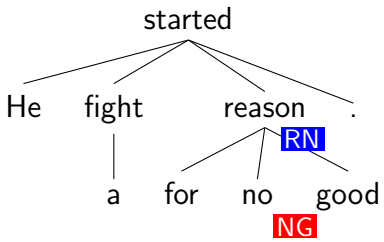
V ďalšom prípade môže ísť o členskú negáciu, podobne ako v prípade slovenského *nie*. V tomto prípade je opäť rozsahom negácie nadradené slovo a praví susedia. Ako bolo spomenuté, viacslovné spojenia *not that*, *not one* a iné momentálne nie sú zohľadňované.

„Not a single nice thing comes out of his mouth.“



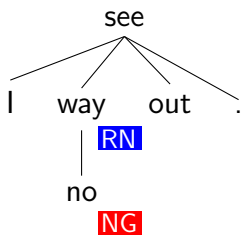
Obr. 7.6: Negátor *not* realizujúci členskú negáciu s vyznačením pravých susedov.

V prípade slova *no* je opäť rozsahom negácie nadradené slovo.
„He started a fight for no good reason.“



Obr. 7.7: Členská negácia slovom *no*.

„I see no way out.“



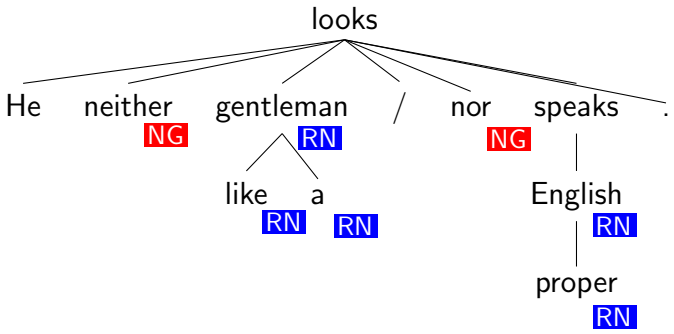
Obr. 7.8: Členská negácia slovom *no*.

7.2.4 Negácia typu *zdvojené spojky*

Zdvojené spojky *neither* - *nor* na rozdiel od slovenského ekvivalentu *ani* - *ani* vyjadrujú totálnu negáciu. Zatiaľ čo slovenské *ani* predstavuje okrem tohto konkrétneho prípadu skôr intenzifikačný prostriedok, slová *neither* a *nor* sa používajú iba spoločne.

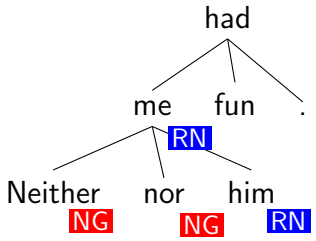
Pre každú spojku je nutné definovať samostatný prechod. Pre spojku *neither* je rozsahom negácie nadradené slovo a praví susedia až po *nor*. Pre spojku *nor* je sú to len praví susedia. Tento vzor bol získaný na základe analýzy viacerých formiem použitia týchto spojok. Demonštrujú to dva z nich, ktoré ukazujú predikátovú platnosť a členskú platnosť.

„He neither looks like a gentleman, nor speaks proper English.“



Obr. 7.9: Zdvojené spojky s negáciou predikátu.

„Neither me nor him had fun.“



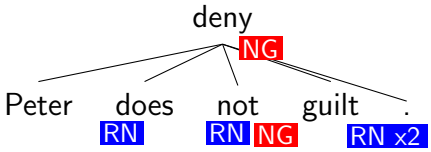
Obr. 7.10: Zdvojené spojky s členskou negáciou.

7.2.5 Dvojitá negácia

Cieľom tejto sekcie je opäť ukázať, ako prechody z navrhnutej metódy dokážu pracovať s dvojitou negáciou bez explicitného prispôsobenia. Podobne ako v prípade metódy pre slovenský jazyk platí poradie hľadania prechodov tak, aby predikátová negácia bola hľadaná ako posledná. Poznatky pre dvojitú negáciu v angličtine boli zdokumentované v sekcii 3.4.6.

V prvom prípade slovo *not* neguje podľa prechodu 7.2.3. Platnosť negácie predikátu sa tým pádom anuluje.

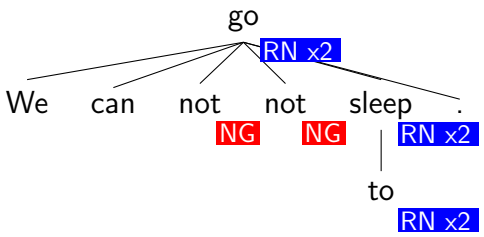
„Peter does not deny guilt.“



Obr. 7.11: Dvojitá negácia spojením vetnej a členskej negácie.

V druhom prípade autor výpovede zdôrazňuje, že je nevhodné neísť spať. Podľa prechodu typu *not* je rozsahom negácie nadradené slovo a praví susedia. Predikát je preto negovaný dvakrát a veta nadobúda kladný význam.

„We cannot not go to sleep.“



Obr. 7.12: Dvojitá negácia spojením vetnej a členskej negácie.

8 Návrh systému na detekciu negácie

Cieľom tejto kapitoly je opísať konceptuálny návrh systému na detekciu negáciu. Ako prvé sú v sekcii 8.1 špecifikované požiadavky, ktoré musí systém spĺňať. V nasledujúcej časti 8.2 sa nachádza opis architektúry a v časti 8.3 sú vysvetlené dôležité procesy.

Ako bolo uvedené v 5.3, systém nebude rozširovať predošlú generáciu riešenia, ale bude postavený od základov. Dôvodom je nedostatočná organizácia kódu predošlej generácie riešenia, výrazné prepojenie aplikačnej logiky s prezentačnou a nutnosť nahradiť nielen všetky nástroje pri predspracovaní, ale aj dodať vlastný spôsob detekcie negácie. Technická dokumentácia ku samotnej implementácii sa nachádza v prílohe A.

8.1 Špecifikácia požiadaviek

Špecifikácia požiadaviek sa dotýka funkcionálnych, ale aj nefunkcionálnych nárokov na tvorený systém. Vychádza z analýzy existujúceho riešenia poskytnutej v sekcii 5.3, ale aj z metód navrhnutých v kapitolách 6 a 7.

Základnou požiadavkou na systém je striktné, modulárne oddelenie funkcionálneho jadra od akejkoľvek nadstavby. Systém na detekciu negácie by mal byť tvorený ako samostatne stojaci modul alebo balík, ktorý bude používateľom poskytovať základnú vybranú funkcionalitu, ale zároveň aj možnosť vhodným prepojením alebo rozšírením túto funkcionalitu doplniť. Nadstavbu tohto modulu z pohľadu predkladanej práce predstavuje grafické rozhranie, respektíve evaluačný modul, ktorý by vystupoval ako jeho používateľ.

8.1.1 Požiadavky na modul detekcie negácie

Výsledný modul by mal byť schopný spracovať vstup ľubovoľného štandardného textového formátu s ľubovoľným kódovaním znakov. Používateľ by nemal byť nútený manuálne vyberať typ otváraného súboru, modul by ho mal rozpoznať automaticky. Pre pohodlnú prácu s kolekciami dokumentov by malo byť postačujúce určiť ako vstup priečinok obsahujúci dokumenty v rôznych formátoch.

Modul by nemal byť závislý od vnútornej štruktúry extrahovaného čistého textu. Nemal by očakávať žiadne pomocné značkovanie, či už ide o fázu morfologickej a syntaktickej analýzy alebo detekciu negácie. Malo by ísť o súvislý text v slovenskom alebo anglickom jazyku, ktorý je možné skopírovať z ľubovoľného zdroja. Modul by mal byť schopný automaticky rozoznať jazyk korpusu načítaných dokumentov.

Významnou požiadavkou je schopnosť inicializovať externý parser podľa detegovaného jazyka dokumentu. Modul by mal podporovať slovenský a anglický jazyk a byť postavený tak, aby bol v budúcnosti jednoducho rozšíriteľný o prípadné ďalšie jazyky. Vnútorná stavba modulu by mala zohľadňovať, že korpus dokumentov sa typicky skladá z viet, ktorých slová je potrebné označkovať morfologickými a syntaktickými anotáciami. Tie by mali byť opäť získané automaticky, bez nutnosti zásahu. Slovenský jazyk by mal byť anotovaný pomocou nástroja SynPar, anglický pomocou balíka Stanford CoreNLP.

Je nevyhnutné, aby modul dokázal v načítanom korpuse detegovať samostatne negátory a samostatne rozsah negácie. Komponenty plniace tieto funkciu by mali byť vymeniteľné, respektíve jednoducho rozšíriteľné. To sa týka detekcie negátorov, kde sa uvažujú rôzne prístupy, ale aj rozsahu negácie, kde má každý druh negácie odlišný rozsah. Modul by mal byť celkovo organizovaný tak, aby bol v budúcnosti jednoducho udržovateľný a zásahy doňho boli nenáročné aj pre prípadných iných autorov. To zahŕňa vhodnú objektovú štruktúru a možnosť jednoduchého zakomponovania do iných systémov.

Z hľadiska detekcie negátorov by modul mal byť schopný využiť slovníkové prístupy, teda morfologickú databázu pre slovenský jazyk a SentiWordNet pre anglický jazyk. Po implementácii prototypu a evaluácii výsledkov detekcie negácie bude zvážená možnosť rozšíriť detekciu negátorov o začlenenie sémantických vzťahov word2vec. Modul by mal teda aj pri detekcii negátorov byť postavený tak, aby bolo možné bez veľkých zásahov do programového kódu doimplementovať prácu s inou externou službou a zameniť ju za aktuálnu službu.

Stavba modulu by mala počítať s tým, že okrem neštruktúrovaného textu by malo byť možné pracovať aj s vlastným, pevne zadefinovaným štruktúrovaným formátom. Ten by slúžil najmä na serializáciu dokumentov s vyznačenou negáciou. Takýmto spôsobom by bolo možné uložiť si výsledky značkovania negácie do súboru, manuálne ich skontrolovať alebo upraviť a neskôr opäť načítať, čo by výrazne uľahčilo tvorbu vlastného korpusu a evaluáciu. Načítavanie a ukladanie vo všeobecnosti by malo byť realizované rozšíriteľným spôsobom.

8.1.2 Požiadavky na nadstavbové moduly

Detekcia negácie by mala byť vizualizovaným, nie skrytým procesom. Systém by mal byť schopný vo vhodnej grafickej podobe zobrazovať jednak anotované dokumenty na úrovni viet

a slov, ale zároveň aj syntaktický strom s uvedením negátora a rozsahu negácie.

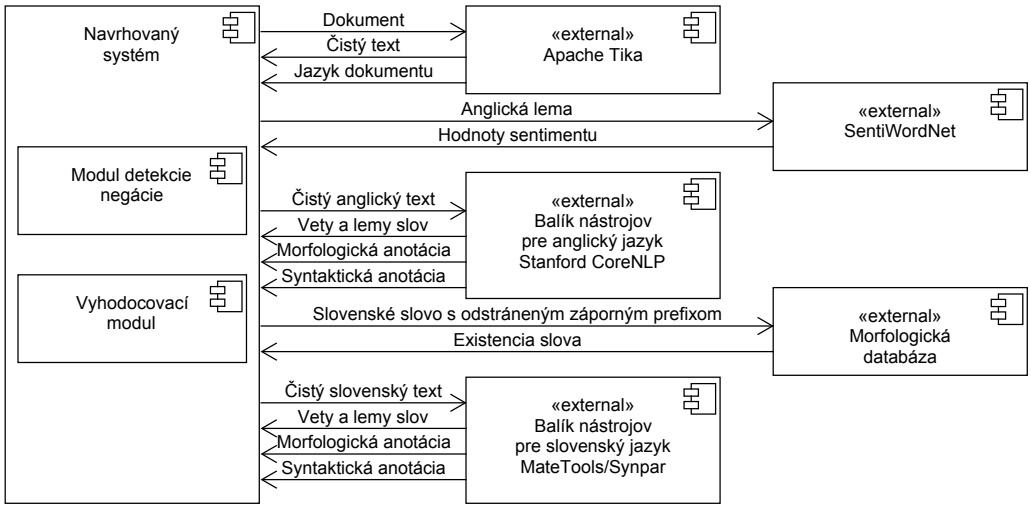
Systém by mal okrem možnosti načítať korpus dokumentov obsahovať aj možnosť vykresliť závislostný strom a detegovať negáciu v používateľsky zadaných vetách. To by uľahčilo overovanie existujúcich prechodov a prípadné skúmanie nových druhov prechodov aj pre používateľov systému, ktorí s ním nemajú veľké skúsenosti.

Z používateľského hľadiska sa ako vhodné požiadavky radia indikácia o aktuálnom stave procesu a intuitívne rozvrhnutie grafického prostredia. Zatiaľ čo druhá požiadavka je subjektívna, tá prvá bola citeľne chýbajúca najmä v predošlej generácii prototypu, kde program beží na jedinej niti. To znamená, že po spustení procesu grafické prostredie prestalo reagovať až do dokončenia výpočtov.

Aby bolo možné modul detekcie negácie overiť, mal byť zahrnutý modul na evaluáciu. Ten by mal byť schopný spravovať zoznam evaluačných dát, poslať ich do modulu na detekciu negácie a vyhodnotiť úspešnosť prostredníctvom pripravených experimentov.

8.2 Architektúra systému

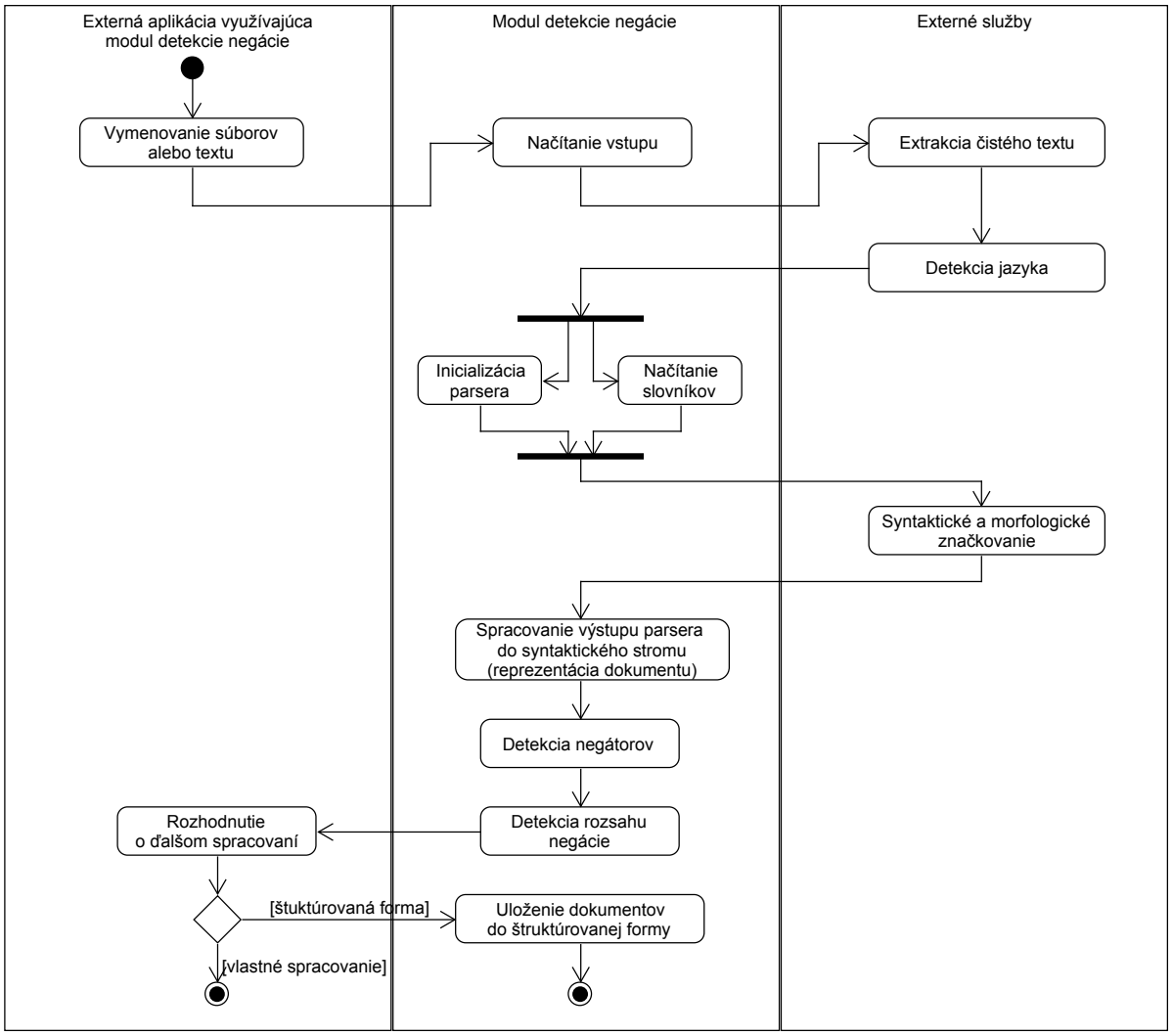
Kontextový diagram 8.1 znázorňuje architektonický informačný pohľad na jednotlivé komponenty systému podľa Woodsa a Rozanského. Navrhovaný systém je zo svojho charakteru nutne modulárny, pričom sa opiera o viaceré externé služby. Konkrétny výber jednotlivých externých systémov bol opísaný v sekcii 2.9. Na diagrame sú znázornené dátové toky medzi systémami, z ktorých je zrejmé, ktoré externé služby sa v skutočnosti použijú, aké údaje systém im posiela a aké od nich prijíma.



Obr. 8.1: Kontextový diagram informačného pohľadu podľa Woodsa a Rozanského [44].

8.3 Procesy v module detekcie negácie

Pohľad na činnosť modulu detekcie negácie prináša diagram aktivít 8.2. V ňom modul detekcie negácie vystupuje ako samostatná entita, ktorá je kontaktovaná externou aplikáciou. Tou môže byť nadstavba v podobe grafického rozhrania alebo evaluačného modulu navrhovaného v tejto práci, alebo ľubovoľný iný systém, ktorý vyžaduje funkcionalitu detekcie negácie.



Obr. 8.2: Proces detekcie negácie z pohľadu navrhovaného modulu a externých systémov.

Vstupom je teda zoznam súborov s neštruktúrovanými dokumentami, zoznam súborov so štuktúrovanými dokumentami alebo čistý text. Aktivity spadajúce pod modul detekcie negácie sa vo výslednej implementácii pretavia v konkrétne služby, respektíve funkcionalitu

modulu. Každá z nich preto bude v nasledovných sekciách opísaná.

8.3.1 Načítanie vstupu

Základnou operáciou je načítanie a príprava korpusu dokumentov na ďalšie spracovanie. Dokumenty sú typicky uložené v priečinku na pevnom disku, v rôznych formátoch, kódovaniach a podpriečinkoch. Načítavací komponent vtedy rekurzívne prehľadá zadaný priečinok a vytvorí zoznam súborov. Každý súbor je následne poslaný na vstup externého systému *Apache Tika*, ktorý z neho získa čistý text a deteguje jazyk dokumentu.

Alternatívou je načítavanie štruktúrovaného vstupu vo formáte XML. Vtedy je opäť nutné rekurzívne prehľadať zadaný priečinok, avšak tentokrát sa očakáva formát uvedený v prílohe C. V ňom sa nachádza už referenčne spracovaný text rozdelený na vety a slová, pričom slová nesú informáciu o správnom značkovaní negácie.

Objektový návrh zohľadňuje možnosť doplnenia ďalších druhov načítavaní, ktorými môže byť načítanie korpusu BioScope, korpusu ConnanDoyle-neg alebo iného. Základnou funkcionalitou zdieľanou medzi všetkými druhmi načítavania je schopnosť rekurzívne prehľadať priečinok a kontaktovať službu Apache Tika.

8.3.2 Inicializácia parsera a načítanie slovníkov

Na základe detegovaného jazyka modul inicializuje príslušný parser. Pre anglický jazyk sa použije balík nástrojov Stanford CoreNLP, pre slovenský jazyk balík nástrojov Synpar založený na Mate Tools. Vzhľadom na to, že inicializácia Stanford CoreNLP trvá na štvorjadrovom procesore s frekvenciou 3,6 GHz vyše 20 sekúnd a inicializácia nástrojov Synpar dokonca okolo 90 sekúnd, sú realizované ako *singleton*, aby sa po prvotnom načítaní dali v systéme ďalej používať.

Rovnako v závislosti od jazyka modul načíta slovníky. Pre anglický jazyk sa použije SentiWordNet, ktorý dokáže vrátiť údaj o polarite slova na základe hodnôt kladného a záporného sentimentu. Pre slovenský jazyk sa použije morfologická databáza, ktorá obsahuje okolo 99 000 lem. Vzhľadom na nutnosť spracovania rozsiahlych textových databáz je opäť vhodná realizácia ako *singleton*.

8.3.3 Reprezentácia dokumentu

Čistý text sa odošle do inicializovaného parsera, ktorý vráti zoznam viet. Pre každú vetu sú ďalej uvedené jednotlivé slová s morfologickými a syntaktickými značkami vo formáte *ConLL*. Vzhľadom na to, že slovenský parser využíva rozšírený formát *ConLL09* a Stan-

ford CoreNLP zasa *CoNLL-X*, spracovanie výsledkov parsovania do slovných entít je opäť jazykovo závislé.

Výsledné slovné entity však zdieľajú všetky atribúty, ako lema, poradie slova vo vete, závislosť na nadradenom slove, morfologická značka, syntaktická značka a ďalšie. Objektový návrh toto zohľadňuje formou spoločného rodiča. Po prevode výstupu parsera do slovných entít môže vzniknúť vetná entita, ktorá slová ukladá jednak ako usporiadaný zoznam podľa poradia slov vo vete, ale zároveň ako stromovú štruktúru. Vety sú napokon priradené do dokumentu.

8.3.4 Detekcia negátorov

Detekcia negátora spočíva v iterácii cez všetky slová každej vety a kontrole, či ide o negátor. Neohybné negátory v slovenskom jazyku sú *bez, okrem, mimo, namiesto, nie*. Ohybné negátory sú detegované prostredníctvom slovníkovej metódy. To znamená, že lema každého slova sa rozdelí na negačný prefix *a, ab, an, anti, bez, de, dez, dis, dys, i, im, kontra, ne, proti, mimo* a zvyšok lemy. Ak sa časť lemy bez tohto prefixu nachádza v morfologickej databáze, slovníková metóda ho prehlási za negátor vybraného druhu. Druh je v slovenskom jazyku určený slovným druhom.

V anglickom jazyku ide o detekciu *no, not, none, never, neither, nor, lack, without* a viacslovných negátorov *rather than, instead of, apart from*. Ostatné slová sú kontrolované pomocou slovníka SentiWordNet, ktorý ku každému obsiahnutému slovu uvádza skóre kladného sentimentu v rozsahu ¡0,1¿ a záporného sentimentu v rozsahu ¡0,1¿. Ako záporné je určené také slovo, ktorého rozdiel záporného a kladného sentimentu je vyšší než 0,485. Druh negácie je opäť určený podľa slovného druhu.

Pokiaľ sa jedna alebo druhá metóda ukáže v procese evaluácie ako nedostatočná, alternatívou je doplniť sémantické vzťahy *word2vec* ako komplementárny prostriedok. Objektový návrh preto počíta s vymeniteľnosťou komponentu na detekciu negátora.

8.3.5 Detekcia rozsahu negácie

Detekcia rozsahu negácie je v oboch jazykoch závislá od druhu negátora. Opäť sa preskúmajú všetky vety z dokumentov. Ak sa vo vete nachádza negátor, spustí sa detekcia rozsahu negácie pre príslušný druh negácie. Keďže jedno slovo môže byť rozsahom viacerých negátorov, pre každé slovo v rozsahu sa uchováva zoznam negátorov.

Objektový návrh je postavený tak, aby sa jednotlivé metódy detekcie rozsahu dali jednoducho upravovať, pridávať či odoberať. Pri detekcii rozsahu negácie sa využívajú prechody

syntaktickým stromom. Všetky prechody formulované v 6 a 7 využívajú nasledujúcu množinu operácií:

- Nájdenie priameho rodiča daného slova.
- Nájdenie podstromu pre dané slovo.
- Nájdenie podstromu daného morfologickou alebo syntaktickou vlastnosťou.
- Nájdenie vrcholu daného morfologickou alebo syntaktickou vlastnosťou.
- Nájdenie pravých susedov.

8.3.6 Uloženie do štruktúrovanej formy

Po vyznačení korpusu dokumentov pomocou morfologických a syntaktických značiek, vyznačení negátorov a vyznačení rozsahu negácie je možné korpus uložiť do štruktúrovanej formy. Formát XML podrobne opisuje C. Z hľadiska prehľadnosti je dôležité, aby výsledné názvy súborov zodpovedali zdrojovým názvom súborov. Ak bol zdrojom čistý text, názov sa vygeneruje automaticky.

8.4 Implementácia prototypu

Táto sekcia prináša prehľad základných údajov ku implementácii prototypu. Na úvod je opísané implementačné prostredie a jazyk, sekcia 8.4.2 prináša prehľad použitých knižníc a prípadné nutné úpravy použitých systémov. Podrobný pohľad na implementáciu modulu detekcie negácie prináša technická dokumentácia v prílohe A.

8.4.1 Jazyk a vývojové prostredie

Vzhľadom na to, že všetky zvažované nástroje, teda extraktor textu Apache Tika, balík nástrojov pre anglický jazyk Stanford CoreNLP a najmä slovenský analyzátor Synpar boli vytvorené na platforme Java, na implementáciu tohto systému bol použitý Java SDK 8. Vývoj prebiehal v prostredí Eclipse Neon 4.6 pod operačným systémom Windows 7.

8.4.2 Použité knižnice

Na manažovanie závislostí je použitý nástroj Maven. Väčšina závislostí bola dostupná ako internetový repozitár, takže pri prenose vývojovej verzie systému ich Maven dokáže

automaticky nájsť a prevziať. Niektoré závislosti sú však pridané lokálne ako súbory **.jar*, keďže centrálny reporizár k nim neexistuje.

V rámci spracovania textu boli bez úprav použité závislosti na Apache Tika 1.13, Apache LangDetect 1.13, Stanford NLP 3.6, MateParser 1.0 a MateTools 3.6. Na vizualizáciu syntaktických stromov bola použitá knižnica Abego TreeLayout 1.0.3.

Slovenský závislostný parser Synpar bol však dostupný len ako webová služba. Z pôvodných zdrojových súborov projektu, ktorý bol vytvorený podľa architektúry MVC, bol preto prevzatý model a upravený tak, aby namiesto s webovou službou dokázal pracovať lokálne. Upravené zdrojové súbory modelu tohto projektu sú súčasťou tohto projektu v balíku *fiit.nlp.Synpar*. Zároveň boli všetky odkazy na registre systému Windows, do ktorých sa mali pridávať cesty ku modelom, nahradené konfiguračným súborom.

9 Evaluácia detekcie negácie

Táto kapitola sa venuje evaluácii návrhu detekcie negácie pre slovenský a anglický jazyk. Veľká pozornosť bola venovaná slovenskému jazyku, pre ktorý bol vytvorený vlastný korpus s vyznačenými negáciami, ktorý je pre tento jazyk zároveň prvým svojho druhu. Jeho tvorbu, obsah a význačné charakteristiky prináša podkapitola 9.1. Následne bola pomocou neho vykonaná séria experimentov podrobne opísaná v podkapitole 9.2, ktorých cieľom bolo vyhodnotiť detekciu negátorov slovníkovým prístupom a detekciu rozsahu negácie prostredníctvom syntaktických závislostí.

Keďže práca sa venuje aj detekcii negácie v anglickom jazyku, vykonané bolo overenie na štandardnom anglickom korpuse BioScope predstavenom v časti 3.5.3. Experimenty opísané v podkapitole 9.3 sa samostatne venujú detekcii negátorov a detekcii rozsahu negácie. Záver tejto kapitoly prináša zhrnutie výsledkov evaluácie a porovnáva úspešnosť navrhovanej metódy s inými prístupmi.

9.1 Slovenský evaluačný korpus

Ako bolo spomenuté v časti 3.5.1, najväčšou kolekciou slovenských viet s jazykovými anotáciami je Slovenský národný korpus. Z hľadiska negácie však obsahuje len označenie, či sloveso začína alebo nezačína záporným prefixom *ne-*. Pre účely vyhodnotenia navrhnutých metód je takáto anotácia nedostatočná, pretože pri slovesách nezohľadňuje ich sémantickú rovinu, iné druhy negátorov vyznačené nie sú a takisto nie je vyznačený rozsah.

Z tohto dôvodu bol pre účely evaluácie pripravený vlastný korpus, ktorý bol tvorený podľa požiadaviek vychádzajúcich z analýzy negácie v slovenskom jazyku uvedenej v časti 3.3 a navrhovanej metódy na detekciu negácie z časti 6.2. Metodiku značkovania spolu s prislúchajúcou technickou realizáciou korpusu opisuje príloha C. Základné požiadavky na korpus sa však dajú zhrnúť do nasledovných bodov:

1. Značkovanie negácie musí byť rozdelené na značkovanie negátora a značkovanie rozsahu negácie.

2. Značkovanie negátora musí byť rozšírené o informáciu o druhu negátora.
3. Značkovanie rozsahu negácie musí niesť informáciu o konkrétnom negátore, pod ktorý dané slovo spadá.
4. Negátor je vždy jednoslovný.
5. Jedno slovo môže byť najviac jedným druhom negátora.
6. Negátor môže negovať viacero slov. Týmto sa zabezpečí jednak negácia viacnásobného vetného člena pri členskej negácii, ale zároveň negácia s vetným rozsahom.
7. Jedno slovo môže patriť do rozsahu viacerých negátorov. Týmto sa pokryjú dvojité, respektíve viacnásobné negácie.

Korpus bol tvorený tak, aby bol žánrovo rozmanitý. Existuje totiž predpoklad, že použitie rozličných jazykových štýlov bude spojené s potlačením alebo vyzdvihnutím istých jazykových prostriedkov, ktoré môžu mať vplyv na úspešnosť spracovania textu. Medzi ne sa radia, napríklad, interpunkcia, subštandardné slová či zložené súvetia, teda súvetia poskladané z troch alebo viacerých jednoduchých viet.

Každá žánrovo odlišná časť korpusu je preto samostatne opísaná v nasledovných častiach spoločne so základnými charakteristikami, na základe ktorých je možné formulovať ďalšie predpoklady. Základné kvantitatívne ukazovatele ako počet viet a počet slov sú doplnené ukazovateľmi spojenými s negáciami.

Pre každú časť korpusu bol určený počet negácií, počet dvojitých negácií a počet negovaných viet. Negácií typicky môže byť v jednej vete nezávisle od seba viacero. Za negovanú vetu sa považuje taká, v ktorej sa nachádza aspoň jedna negácia.

Na ďalšie dokreslenie vlastností korpusu je pre každú časť uvedené aj množstvo interpunkcie prislúchajúce na jednu vetu. Interpunkcia pridáva vetám komplexitu, či už v podobe čiarok oddeľujúcich podraďovacie súvetie či prístavok, alebo ako úvodzovky vyznačujúce priamu reč. Dá sa predpokladať, že čím väčší je podiel interpunkcie, tým nižšia je šanca na korektné určenie syntaktických závislostí [28].

9.1.1 Beletria

Beletria je umelecká literatúra, pod ktorú sa radí próza aj poézia. Zdrojom tejto časti korpusu bol Zlatý fond, ktorý poskytuje voľný prístup ku významným dielam najmä slovenských autorov z obdobia posledných storočí. Do negačného korpusu boli zaradené rozprávky, poviedky, ale aj niekoľko básní. Menovite ide o diela:

- Jozef Gregor Tajovský: Mamka Pôstková, Maco Mlieč

- Pavol Dobšinský: Zakliata hora
- Ľubomír Feldek: Rozprávka o Perinbabe, Smrť v ružovom
- Ján Botto: Duma (báseň)
- Ferko Urbánek: Moja vlasť (zbierka básní)

V procese značkovania sa ukázalo, že tieto texty obsahujú množstvo dlhých viet, priamej reči, interpunkcie a subštandardných slov. Zároveň však obsahujú mnoho veľmi krátkych viet s nevyjadrenými časťami prisudzovacieho skladu. Tieto ukazovatele kvantitatívne zachytáva tabuľka 9.2.

Viet	**Slov**	**Negovaných viet**	**Negácií**	**Negácií na 1 vetu**	**Dvojitých negácií**
770	12468	212	261	0,34	4

Tabuľka 9.1: Charakteristika beletrie v korpuse z pohľadu negácie.

Interpunkcie	**Int. na 1 vetu**	**1 - 5 slov**	**6 - 19 slov**	**20 - 49 slov**	**50+ slov**
2525	3,28	105	441	206	18

Tabuľka 9.2: Charakteristika beletrie v korpuse z pohľadu jazykových prostriedkov.

9.1.2 Šport

Športové články boli vybrané ako zástupca publicistického štýlu. Celkovo bolo značkovaných 35 úplných článkov z troch slovenských športových portálov[1]. Obsahovo ide o sumáre hokejových a futbalových zápasov spolu s vyjadreniami trénerov, informácie o kluboch, športových podujatiach alebo hráčskych prestupoch, ale aj rozhovory so športovcami.

Proces značkovania ukázal, že zastúpenie zložených súvetí je menšie než u beletristických článkov. Zároveň sa vo vetách nachádza menej interpunkcie a celkovo sú vety kratšie, čo dokladuje tabuľka 9.8. Z tohto dôvodu by úspešnosť spracovania tejto časti korpusu mala prevýšiť úspešnosť spracovania beletrie.

Viet	**Slov**	**Negovaných viet**	**Negácií**	**Negácií na 1 vetu**	**Dvojitých negácií**
1118	18064	279	342	0,31	11

Tabuľka 9.3: Charakteristika športových článkov v korpuse z pohľadu negácie.

[1] HokejPortal, ProFutbal, Šport SME

Interpunkcie	Int. na 1 vetu	1 - 5 slov	6 - 19 slov	20 - 49 slov	50+ slov
2705	2,42	52	761	300	5

Tabuľka 9.4: Charakteristika športových článkov v korpuse z pohľadu jazykových prostriedkov.

9.1.3 Recenzie fotoaparátov

Recenzie fotoaparátov predstavujú časť pôvodného korpusu dokumentov, ktorý bol vytvorený pre účely diplomových prác Ing. Martina Jaborníka [20] a Ing. Mateja Kvitkoviča [27]. Hoci bol použitý na evaluáciu, nikdy nebol označkovaný. V rámci prípravy korpusu pre túto prácu bolo prevzatých a označkovaných 35 článkov o fotoaparátoch. Recenzia síce žánrovo opäť patrí pod publicistický štýl, avšak tentokrát ide o analytickú vetvu.

Tieto články neobsahujú žiadnu priamu reč, čo sa prejavilo v zatiaľ najnižší pomer interpunkcie voči jednej vete, ako ukazuje tabuľka 9.6. Použité je jadro slovnej zásoby rozšírené o odborné technické výrazy a zároveň o značky, skratky a skratkové slová.

Viet	Slov	Negovaných viet	Negácií	Negácií na 1 vetu	Dvojitých negácií
924	14872	167	229	0,25	4

Tabuľka 9.5: Charakteristika recenzií fotoaparátov v korpuse z pohľadu negácie.

Interpunkcie	Int. na 1 vetu	1 - 5 slov	6 - 19 slov	20 - 49 slov	50+ slov
1783	1,93	61	589	274	0

Tabuľka 9.6: Charakteristika recenzií fotoaparátov v korpuse z pohľadu jazykových prostriedkov.

9.1.4 Tímové projekty

Ďalšou striktne technickou zložkou korpusu sú vybrané časti dokumentácií z tímových projektov vedených na Fakulte informatiky a informačných technológií STU. Dokumentácie inžinierskeho diela typicky obsahujú úvod do problematiky, analýzu, zber požiadaviek, návrh riešenia, plánovanie šprintov, sumarizáciu výsledkov a technickú dokumentáciu. Z jazykového hľadiska by malo ísť o výkladový štýl, ktorý sa prejavuje vecnou formuláciou s množstvom odborných termínov.

Na zachovanie pomeru bolo opäť zvolených približne tisíc viet, ktoré boli vyberané z rôznych kapitol. Väčšinu tejto časti korpusu preto tvoria kapitoly analýzy a špecifikácie požiadaviek, ale okrem nich aj návrh komponentov systému či používateľská príručka. Je dôležité podotknúť, že do korpusu sa nedostali časti dokumentácií, ktoré obsahovali zdrojový kód, tabuľky alebo štruktúrované zoznamy vo forme odrážok. Takéto entity sú z hľadiska vyhodnotenia nepoužiteľné, pretože by nebolo možné určiť vetné jednotky.

Viet	Slov	Negovaných viet	Negácií	Negácií na 1 vetu	Dvojitých negácií
1048	16576	93	119	0,11	1

Tabuľka 9.7: Charakteristika tímových projektov v korpuse z pohľadu negácie.

Interpunkcie	Int. na 1 vetu	1 - 5 slov	6 - 19 slov	20 - 49 slov	50+ slov
2501	2,38	146	589	306	7

Tabuľka 9.8: Charakteristika tímových projektov v korpuse z pohľadu jazykových prostriedkov.

9.1.5 Slovenský národný korpus

Slovenský národný korpus uvedený v 3.5.1 obsahuje morfologicky anotovanú kolekciu viet rôznych žánrov. Bc. Jozef Gáborík, paralelne riešiaci diplomovú prácu pod názvom *Rozpoznávanie negácie v texte pomocou strojového učenia*, prispel ku tvorbe korpusu označkovaním vybraných 235 viet obsahujúcich negáciu [13]. Pri svojom výbere sa zameral najmä na dlhšie vety s množstvom interpunkcie, pričom obsahovo ide o rozmanitú zbierku zloženú z rozprávok, rozhovorov či publicistiky.

Viet	Slov	Negovaných viet	Negácií	Negácií na 1 vetu	Dvojitých negácií
235	6048	229	322	1,37	10

Tabuľka 9.9: Charakteristika Slovenského národného korpusu v korpuse z pohľadu negácie.

Interpunkcie	Int. na 1 vetu	1 - 5 slov	6 - 19 slov	20 - 49 slov	50+ slov
895	3,81	1	91	126	17

Tabuľka 9.10: Charakteristika Slovenského národného korpusu v korpuse z pohľadu jazykových prostriedkov.

9.1.6 Vlastné vety

V procese tvorby a iteratívneho vyhodnocovania navrhnutej metódy boli pripravené vety obsahujúce vybrané druhy negácií. Pomocou nich bola metóda priebežne testovaná a ladená. Tieto vety sú prevažne jednoduché, s malým množstvom interpunkcie a stavané tak, aby detekcia negácie nezlyhala na nekorektnej tvorbe syntaktického stromu. Zdrojom týchto viet sú ukážky použité v časti 6.2 tejto práce a zároveň ukážky pripravené v publikácii doc. Pavloviča [39].

Viet	Slov	Negovaných viet	Negácií	Negácií na 1 vetu	Dvojitých negácií
51	412	51	65	1,27	6

Tabuľka 9.11: Charakteristika vlastných viet v korpuse z pohľadu negácie.

Interpunkcie	Int. na 1 vetu	1 - 5 slov	6 - 19 slov	20 - 49 slov	50+ slov
71	1,39	9	42	0	0

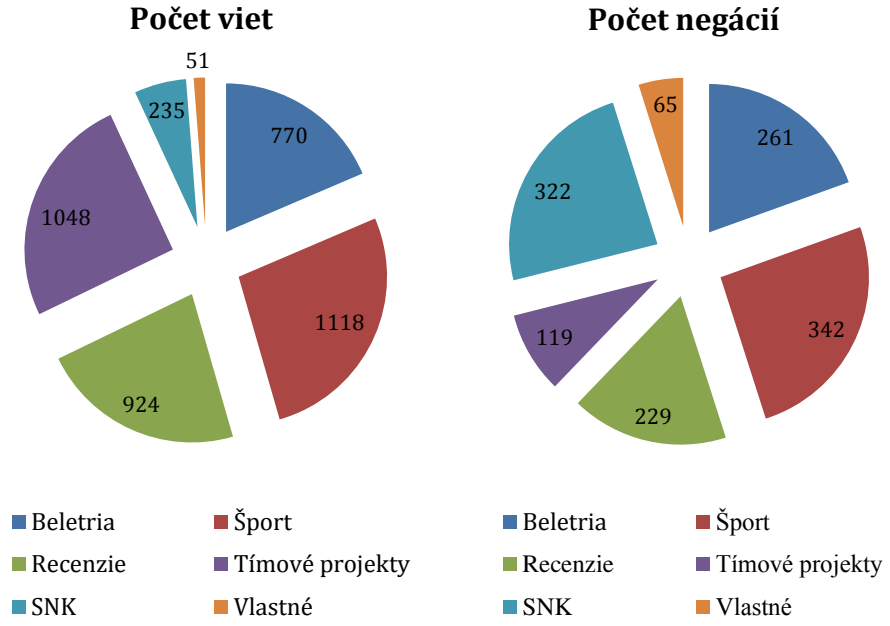
Tabuľka 9.12: Charakteristika vlastných viet v korpuse z pohľadu jazykových prostriedkov.

9.1.7 Celková charakteristika korpusu

Ako ukazujú čiastkové charakteristiky, negácia je zastúpená približne v štvrtine viet. Základný cieľ, teda vytvoriť žánrovo a štýlovo rozmanitú množinu dokumentov s vyznačenými negáciami, sa podarilo splniť. Podrobné celkové štatistiky sú uvedené v tabuľke 9.13, vybrané ukazovatele sú vizuálne znázornené na grafe 9.1.

Ukazovateľ	Hodnota
Počet viet	4146
Počet viet s 1 - 5 slovami	374
Počet viet s 6 - 19 slovami	2513
Počet viet s 20 - 49 slovami	1212
Počet viet s 50 a viac slovami	47
Počet negovaných viet	1031
Počet negácií	1338
Počet dvojitých negácií	34
Priemer negácií na 1 vetu	0,32

Tabuľka 9.13: Celková charakteristika korpusu.



Obr. 9.1: Porovnanie počtu viet a počtu negácií v korpuse.

9.2 Experimenty pre slovenský jazyk

Táto podkapitola obsahuje opis experimentov, ktorých cieľom bolo overiť detekciu negácie v slovenskom jazyku ako proces zložený z dvoch samostatných krokov, menovite detekcie negátorov slovníkovou metódou a detekcie rozsahu negácie prostredníctvom syntaktických stromov. Experimenty sú stavané tak, aby čo najdôslednejšie zachytili viacero aspektov tohto procesu, ale zároveň tak, aby boli schopné vykázať celkovú úspešnosť týchto metód.

Všetky experimenty boli vykonané na vlastnom evaluačnom korpuse, ktorý bol opísaný v predchádzajúcej podkapitole 9.1. Úvodom každého experimentu je formulácia istého predpokladu či konkrétneho cieľa. Jadrom experimentu je využitie navrhnutého prototypu na dosiahnutie výsledku, ktorý buď potvrdí či zamietne formulovaný predpoklad, alebo kvantitatívne naplní požiadavky daného cieľa.

Úvodné experimenty 9.2.1 a 9.2.2 približujú proces detekcie negátorov. Bez toho, aby bol úspešne detegovaný negátor, nie je možné ani určiť rozsah negácie, preto je dôležité poznať úspešnosť už tohto prvého, základného kroku. Nasledovné tri experimenty sa venujú overeniu úspešnosti detekcie rozsahu negácie, optimalizácie algoritmu oproti pôvodnému návrhu a určenia celkovej, výslednej úspešnosti detekcie rozsahu negácie. Posledná časť tejto podkapitoly 9.2.6 je venovaná kvantitatívnemu aj kvalitatívnemu porovnaniu navrhovaného riešenia s pôvodnou prácou Ing. Mateja Kvitkoviča.

9.2.1 Detekcia ohybných negátorov slovníkovou metódou

Detekcia negátora je základným krokom pri detekcii negácie ako takej. Ako bolo opísané v časti 6.2, v slovenskom jazyku existujú ohybné a neohybné negátory. Detekcia úspešnosti neohybných negátorov sa predpokladá na úrovni 100 %, pretože ide o jednoduché porovnávanie slov. Ohybné negátory zvyčajne začínajú negačným prefixom, ktorý môže byť buď domáci (*ne-*), alebo niektorý z cudzokrajných (*a-, anti-, dis-* a iné). Na základe vlastnej skúsenosti z tvorby negačného korpusu existuje predpoklad, že najviac ohybných negátorov bude mať domáci prefix *ne-*.

Cieľom tohto experimentu je overiť, či je v korpuse skutočne väčšie zastúpenie domácich ohybných negátorov než zahraničných. Ďalším, nemenej významným cieľom je overiť úspešnosť detekcie oboch druhov negátorov slovníkovou metódou. Ohybné negátory sú preto v tomto experimente kategorizované do dvoch celkov podľa prefixu, a to *ne-* a *iné negačné prefixy*. Napokon, výsledky úspešnosti detekcie ohybných negátorov sa môžu líšiť podľa žánru, pričom pri beletrii sa predpokladá najnižšia úspešnosť v dôsledku použitia subštandardných alebo archaických slov.

Postup vykonania experimentu:

1. Načítanie evaluačného korpusu podľa jednotlivých druhov textu.
2. Spustenie značkovania negátorov slovníkovou metódou.
3. Každé slovo začínajúce na záporný prefix bolo automaticky overené, či sa ho pomocou slovníkovej metódy podarilo korektne vyznačiť.
4. Výsledky boli zoskupené podľa žánrov a podľa význačných druhov prefixov.

Výsledky sú spracované v tabuľke 9.14. Ako sa ukázalo, v korpuse sa nachádzalo 1243 slov začínajúcich na negačný prefix *ne-*, z ktorých 795 bolo v skutočnosti negátorom. Slovníkovou metódou sa podarilo správne odhaliť 728 z nich, čo udáva úplnosť na úrovni 91,57 %. Rovných 10 slov začínajúcich na *ne-* bolo slovníkovou metódou nesprávne označených ako negátor, čoho výsledkom je presnosť 98,64 %.

Z hľadiska žánrov sa potvrdilo, že druhú najnižšiu úspešnosť z pohľadu metriky F_1 dosahuje beletria. O niečo nižšiu úspešnosť dosiahli texty zo Slovenského národného korpusu, ktoré sa ale takisto skladajú prevažne z beletristických úryvkov. Vecná časť korpusu dosiahla úspešnosť okolo 97 %.

Druh	**Ne-**				**Iné negačné prefixy**			
	Počet	Presnosť	Úplnosť	F_1	Počet	Presnosť	Úplnosť	F_1
Beletria	327	96,17 % ($\frac{226}{235}$)	93,38 % ($\frac{226}{242}$)	94,75 %	539	-	-	-
Šport	328	100 % ($\frac{74}{74}$)	94,88 % ($\frac{74}{87}$)	97,37 %	801	27,27 % ($\frac{3}{11}$)	100 % ($\frac{3}{3}$)	42,87 %
Recenzie	167	100 % ($\frac{119}{119}$)	94,44 % ($\frac{119}{126}$)	97,14 %	766	60 % ($\frac{3}{4}$)	75 % ($\frac{3}{5}$)	66,67 %
Projekty	109	100 % ($\frac{60}{60}$)	93,75 % ($\frac{60}{64}$)	96,77 %	1071	53,84 % ($\frac{7}{13}$)	100 % ($\frac{7}{7}$)	70 %
SNK	273	99,53 % ($\frac{216}{217}$)	88,89 % ($\frac{216}{243}$)	93,91 %	363	87,5 % ($\frac{7}{8}$)	87,5 % ($\frac{7}{8}$)	87,5 %
Vlastné	39	100 % ($\frac{33}{33}$)	100 % ($\frac{33}{33}$)	100 %	15	-	-	-
Celkovo	1243	98,64 % ($\frac{728}{738}$)	91,57 % ($\frac{728}{795}$)	**94,97 %**	3555	55,56 % ($\frac{20}{36}$)	86,96 % ($\frac{20}{23}$)	**67,8 %**

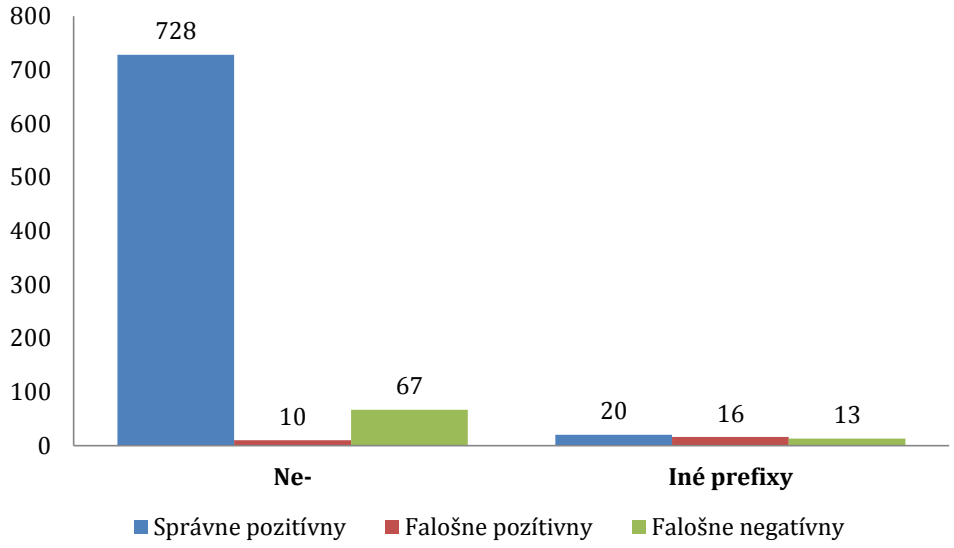
Tabuľka 9.14: Úspešnosť detekcie ohybných negátorov slovníkovou metódou.

Z pohľadu cudzokrajných záporných prefixov je možné pozorovať, že ich výskyt v korpuse je veľmi zriedkavý. Z 3555 kandidátov na negátor s cudzokrajným prefixom sa iba 23 ukázalo byť naozaj negátorom. Slovníková metóda z nich celkovo úspešne detegovala 20 (úplnosť 86,96 %), ale pridala 16 nesprávnych označení (presnosť 55,56 %). V beletristických textoch sa takýto negátor dokonca vôbec nenachádzal.

Väčšinu ohybných slov, ktoré nie sú negátormi, ale boli za ne nesprávne prehlásené, tvoria náhodné dvojice. Vezmúc do úvahy len cudzokrajné negačné prefixy *a-* a *ab-*, slovníková metóda nájde mnoho párov, akými sú *žúr - ažúr, ažúr - abažúr, legát - ablegát, rabský - arabský, rázny - abrázny, dresový - adresový, gitovať - agitovať*. Takéto dvojice existujú aj pre domáci prefix *ne-*, napríklad, *vesta - nevesta*.

Ak slovo v skutočnosti bolo negátorom, ale slovníková metóda ho takto nedetegovala, dôvod väčšinou spočíval v jeho gramatickom tvare. V žiadnom súčasnom slovníku sa nenachádzajú prechodníky (*nerobiaci, nepíšucich*), preto zlyhá už krok lematizácie, kedy sa parseru nepodarí nájsť jeho základný tvar. Druhým dôvodom bolo, že išlo o subštandardné slovo, teda slovo mimo jadra slovnej zásoby, ktoré sa v slovníku nenachádzalo.

Na základe týchto výsledkov sa dá tvrdiť, že v bežnom písomnom prejave je z hľadiska ohybných negátorov drvivá prevaha negátorov s domácom prefixom *ne-*. Výsledkom je detekcia domácich ohybných negátorov na úrovni 94,97 %, cudzokrajných negátorov s nízkym zastúpením v korpuse 67,8 %, čo vizuálne porovnáva graf 9.2.



Obr. 9.2: Porovnanie výsledkov detekcie negátorov podľa prefixu.

Detekcia ohybných negátorov slovníkovou metódou vykazuje v pripravenom evaluačnom korpuse veľmi vysokú celkovú mierku F_1 na úrovni **93,96 %**, ktorá by nemala byť limitujúcim faktorom pri detekcii rozsahu negácie.

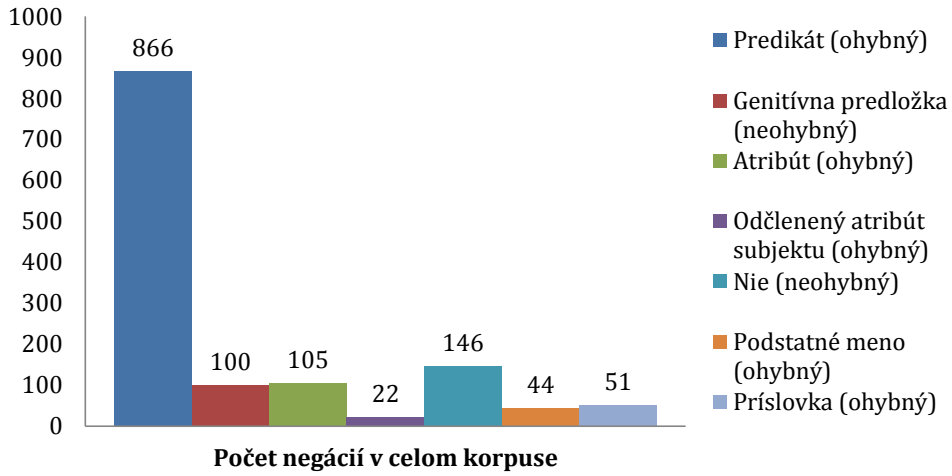
9.2.2 Absolútne počty negácií v korpuse podľa druhu negácie

Ešte pred samotným vyhodnotením úspešnosti detekcie rozsahu negácie bol vykonaný experiment, ktorého cieľom bolo ďalej vyprofiloval korpus z hľadiska druhov zastúpených negácií. Na základe skúsenosti s tvorbou korpusu existuje predpoklad, že väčšina negácií v slovenskom jazyku je predikátová. Výstupom tohto experimentu je porovnanie počtov jednotlivých druhov negácií v korpuse nielen celkovo, ale opäť aj podľa žánrov textu.

1. Načítanie evaluačného korpusu podľa jednotlivých druhov textu.
2. Vypočítala sa štatistika zastúpenia druhov negátorov tak, ako boli v korpuse manuálne označené.
3. Výsledky boli zoskupené podľa žánrov. Podrobne sú uvedené v tabuľke 9.15, celkové výsledky vizuálne porovnáva graf 9.3.

Druh	**pre**	**gen**	**atr**	**sub**	**nie**	**sbs**	**adv**
Beletria	211	3	19	0	12	9	6
Šport	241	25	20	4	35	5	10
Recenzie	123	33	14	5	35	15	4
Projekty	56	11	10	2	19	5	16
SNK	206	8	39	6	37	10	15
Vlastné	29	20	3	5	8	0	0
Celkovo	866	100	105	22	146	44	51

Tabuľka 9.15: Podrobné porovnanie absolútnych počtov negácií v korpuse podľa žánrov.



Obr. 9.3: Celkové porovnanie druhov negácií v korpuse.

Výsledky experimentu potvrdzujú, že vo všetkých častiach korpusu dominuje predikátová negácia, ktorá z celkového hľadiska tvorí takmer dve tretiny negácií v korpuse. Práve na predikátovú negáciu by teda mala smerovať väčšina úsilia z hľadiska detekcie rozsahu negácie.

9.2.3 Detekcia rozsahu negácie

Po vyhodnotení úspešnosti detekcie negátorov bolo možné pristúpiť ku vyhodnoteniu úspešnosti detekcie rozsahu negácie prostredníctvom syntaktických závislostí. Tento experiment priamo vyhodnocuje úspešnosť metódy opísanej v časti 6.2 tak, ako bola navrhnutá pomocou grafov závislostných stromov.

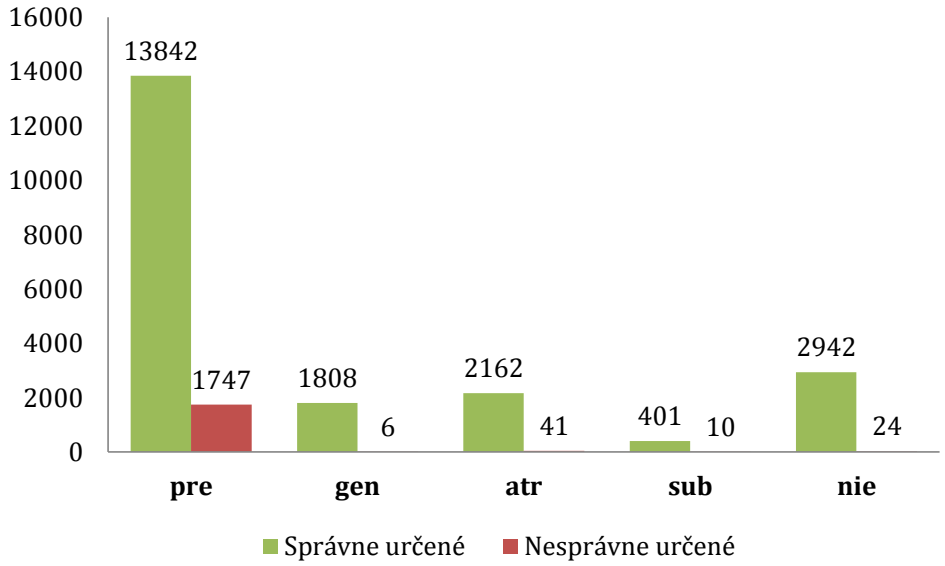
Cieľom experimentu je kvantitatívne vyhodnotiť úspešnosť detekcie rozsahu v celom korpuse. Vzhľadom na to, že predikátová negácia má ako jediná vetný rozsah, očakáva sa pri nej nižšia úspešnosť ako pri iných druhoch negácie s členským rozsahom. Zároveň existuje predpoklad, že negácie typov *genitívna predložka* a *nie* budú mať najvyššiu úspešnosť detekcie rozsahu, pretože tieto pri pozorovaní výsledkov syntaktickej analýzy ukazovali najvyššiu stabilitu.

Postup vykonania experimentu:

1. Načítanie celého evaluačného korpusu.
2. Spustenie značkovania negátorov slovníkovou metódou.
3. Spustenie značkovania negácie prostredníctvom syntaktických závislostí.
4. Pre každú vetu boli nájdené očakávané negátory podľa manuálneho značkovania v korpuse. K nim boli pridané prípadné ďalšie negátory, ktoré boli nesprávne vyznačené prototypom.
5. Pre každý negátor z predošlého kroku boli preskúmané všetky slová vo vete a bola určená zhoda medzi manuálnym a automatickým značkovaním rozsahu negácie.
6. Výsledky boli zoskupené podľa druhov negácie do tabuľky 9.16 a grafu 9.4.

Druh	**TP**	**TN**	**FP**	**FN**	**Presnosť**	**Úplnosť**	**Správnosť**	**F_1**
pre	4281	9561	1318	429	76,46 %	90,89 %	88,79 %	83,05 %
gen	101	1707	3	3	97,11 %	97,11 %	99,97 %	97,11 %
atr	55	2107	8	33	87,30 %	62,5 %	98,13 %	72,85 %
sub	14	387	4	6	77,78 %	70 %	97,56 %	73,68 %
nie	117	2825	6	18	95,12 %	86,67 %	99,19 %	90,70 %

Tabuľka 9.16: Podrobné porovnanie detekcie rozsahu negácie v celom korpuse.



Obr. 9.4: Správnosť detekcie negácie v korpuse.

Podľa očakávania, negácie typov *genitívna predložka* a *nie* dosiahli pri detekcii rozsahu najvyššiu úspešnosť. Na základe skúmania výsledkov atribútovej a subjektovej negácie sa dá povedať, že práve u nich sa najviac prejavili chyby pri detekcii negátora.

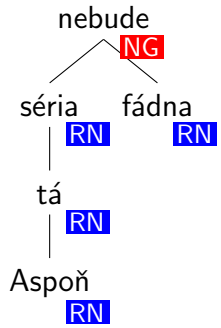
Nízka úplnosť je zapríčinená tým, že atribút sa často nachádza v tvare činného príčastia (napr. *nepíšucich*), ku ktorému lematizátor nedokáže nájsť správny slovný základný. Chyby presnosti spôsobené negátorom opäť spočívali v už opísanom fenoméne, kedy došlo ku náhodnej lexikálnej zhode sémanticky nesúvisiach slov (napr. *rabský - arabský*).

Menšiu časť neúspešnosti pri týchto dvoch druhoch negácie ďalej spôsobuje nepravidelnosť spracovania vety do syntaktického stromu. Model zachytáva najčastejší vzťah medzi podradeným atribútom a nadradeným podstatným menom či zámenom, ale v špecifických prípadoch vetných konštrukcií sa môže tento vzťah narušiť. Príkladmi sú nesprávne vloženie interpunkčného znamienka alebo atypický slovosled.

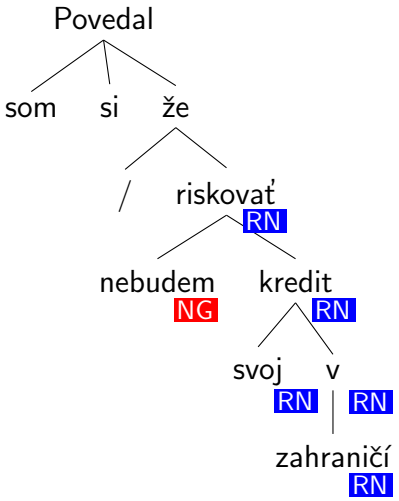
9.2.4 Optimalizácie detekcie rozsahu predikátovej negácie

Vzhľadom na výsledky experimentu 9.2.2, ktorý ukázal dvojtretinové zastúpenie predikátovej negácie, bola vynaložená snaha o zlepšenie skóre F_1 z pôvodnej úrovne 83,05 %. Počas značkovania korpusu vyšli na povrch dva javy, ktoré pri tvorbe modelu neboli zohľadnené.

Optimalizácia 1. Prvým je negácia zabudovaná do modálnych slovies. Pri klasickej predikátovej negácii sa ako rozsah označí celý podstrom, ako to znázorňuje syntaktický strom na obrázku 9.5. Ak je však negovaným slovom modálne sloveso, to je typicky vo vzťahu podradenosti ku skutočnému predikátu. Rozšírený model teda počíta aj s takými syntaktickými



Obr. 9.5: Značkovanie podstromu pri *„nebude“* ako predikáte.



Obr. 9.6: Doplnené značkovanie ako optimalizácia pri *„nebude“* ako modálnom slovese.

stromami, aké znázorňuje obrázok 9.6.

Optimalizácia 2. Ako hovorí časť 6.2, predikátová negácia má vetný rozsah. Pojem veta je však nejednoznačný, pretože môže označovať jednoduchú vetu, ale aj zloženú vetu. Zo skúmania súvetí vzišiel záver, že nie všetky spojky prenášajú negatívny význam do svojich častí súvetia. Tento záver podporuje aj publikácia [14], ktorá vymenúva vybrané spojky ako *vysvetľujúce* a radí medzi ne *ale, takže, pretože, nech, keby, keďže, keď, aby, pokiaľ, nakoľko, tak*. Podstromy týchto spojok boli pri detekcii rozsahu explicitne vylúčené.

Postup vykonania experimentu:

1. Načítanie celého evaluačného korpusu.
2. Spustenie značkovania negátorov slovníkovou metódou.
3. Spustenie značkovania predikátovej negácie prostredníctvom syntaktických závislostí.
   (a) Bez optimalizácií (*pre*).
   (b) S optimalizáciou modálnych slovies (*pre+OPT1*).

(c) S optimalizáciou vysvetľovacích spojok (*pre+OPT2*).

(d) S oboma optimalizáciami (*pre+OPT1+2*).

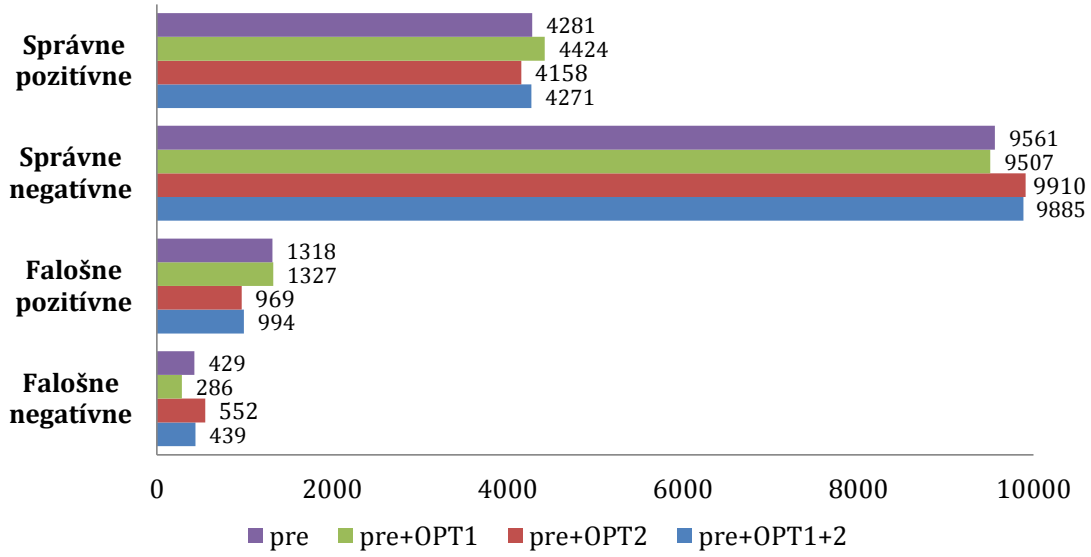
4. Pre každú vetu boli nájdené očakávané negátory podľa manuálneho značkovania v korpuse. K nim boli pridané prípadné ďalšie negátory, ktoré boli nesprávne vyznačené prototypom.

5. Pre každý negátor z predošlého kroku boli preskúmané všetky slová vo vete a bola určená zhoda medzi manuálnym a automatickým značkovaním rozsahu negácie.

6. Výsledky boli zoskupené podľa fáz optimalizácie do tabuľky 9.17 a grafu 9.7.

Druh	**TP**	**TN**	**FP**	**FN**	**Presnosť**	**Úplnosť**	**Správnosť**	**F_1**
pre	4281	9561	1318	429	76,46 %	90,89 %	88,79 %	83,05 %
pre+OPT1	4424	9507	1327	286	76,32 %	93,93 %	89,36 %	84,21 %
pre+OPT2	4158	9910	969	552	81,10 %	88,28 %	90,24 %	84,54 %
pre+OPT1+2	4271	9885	994	439	81,12 %	90,67 %	90,80 %	85,63 %

Tabuľka 9.17: Podrobné porovnanie optimalizácie detekcie rozsahu predikátovej negácie v celom korpuse.



Obr. 9.7: Porovnanie výsledkov optimalizácií predikátovej negácie.

Optimalizácia modálnych slovies zvýšila počet správne pozitívnych výsledkov o 143, o rovnaký počet sa zároveň znížil ukazovateľ falošne negatívnych výsledkov, čo vyústilo do zvýšenia úplnosti o 3 %. Optimalizácia spojok zvýšila o 349 slov ukazovateľ správne negatívnych výsledkov a zhruba o rovnaký počet znížila výskyt falošne pozitívnych výsledkov. Obidve optimalizácie so sebou ale priniesli aj mierny negatívny dopad.

Po optimalizácii modálnych slovesách sa zväčšil počet skúmaných slov vo vete, čo prirodzene vyústilo do mierneho zvýšenia falošne pozitívnych výsledkov. Po optimalizácii spojok sa zasa zvýšil počet falošne negatívnych výsledkov, čo sa dá zdôvodniť dvojakosťou vybraných spojok, ktoré v určitých prípadoch môžu v skutočnosti negáciu prenášať do svojich podstromov.

Po spojení oboch optimalizácií došlo ku zvýšeniu presnosti o 4,5 %, úplnosť zostala približne na pôvodnej hodnote, čo v celkovom výsledku zvyšuje mierku F_1 z pôvodných 83,05 % na výsledných 85,63 %. Týmto výsledkom sú optimalizácie považované za úspešné, i keď len s miernym dopadom, a obidve zostávajú zahrnuté aj do samotnej implementácie.

9.2.5 Výsledná úspešnosť detekcie rozsahu negácie

Záverečný experiment cielený na metódu detekcie rozsahu negácie sa zameriava na porovnanie úspešnosti detekcie podľa žánru textu z hľadiska metriky F_1. Zároveň si dáva za cieľ stanoviť súhrnnú úspešnosť detekcie rozsahu negácie pre všetky druhy negácie.

Postup vykonania experimentu:

1. Načítanie evaluačného korpusu podľa jednotlivých druhov textu.
2. Spustenie značkovania negátorov slovníkovou metódou.
3. Spustenie značkovania negácie prostredníctvom syntaktických závislostí.
4. Pre každú vetu boli nájdené očakávané negátory podľa manuálneho značkovania v korpuse. K nim boli pridané prípadné ďalšie negátory, ktoré boli nesprávne vyznačené prototypom.
5. Pre každý negátor z predošlého kroku boli preskúmané všetky slová vo vete a bola určená zhoda medzi manuálnym a automatickým značkovaním rozsahu negácie.
6. Výsledky boli zoskupené podľa žánrov a podľa druhov negácie do tabuľky 9.18 a grafu 9.8.
7. Výsledky boli spočítané a celkovú úspešnosť uvádza tabuľka 9.19.

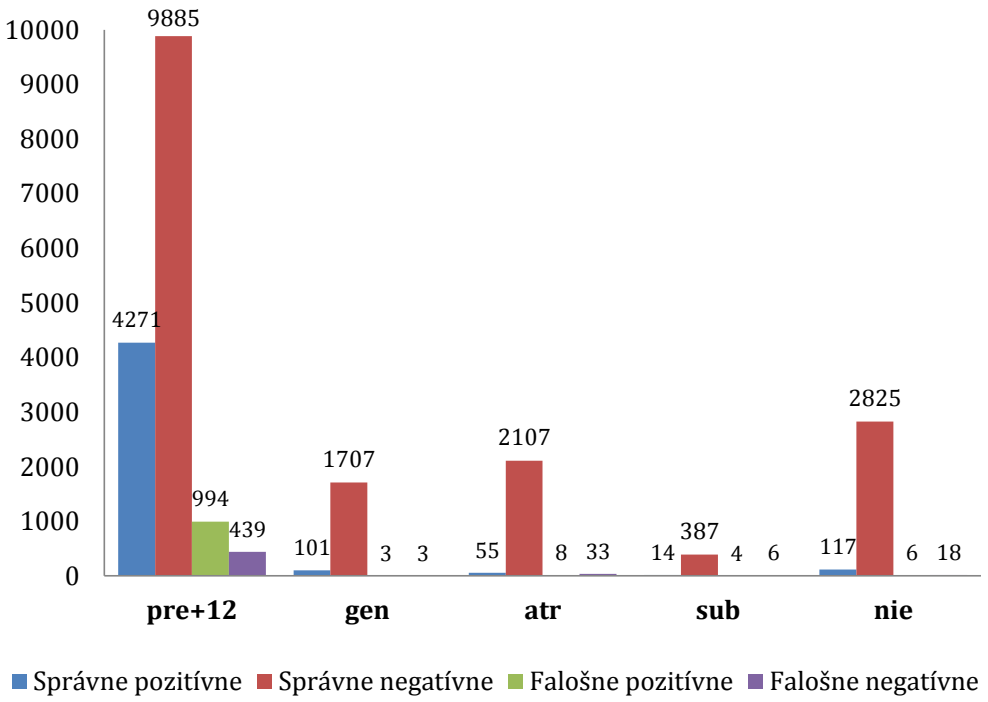
Tieto čísla sa dajú považovať za konečné vyhodnotenie úspešnosti navrhnutej metódy na detekciu rozsahu negácie prostredníctvom syntaktických závislostí. Z hľadiska predikátovej negácie sa potvrdil pôvodný predpoklad, že umelecké texty budú mať nižšiu úspešnosť spracovania ako technické texty. Negácie s neohybným negátorom dosahujú priemerne najvyššiu úspešnosť detekcie rozsahu.

Hoci sú hodnoty F_1 v čiastkových výsledkoch jednotlivých druhov negácií často nízke, treba vziať do úvahy početnosť jednotlivých komponentov. Na dokreslenie tohto aspektu

slúži graf 9.8 vyjadrujúci absolútne početnosti. Celková úspešnosť detekcie rozsahu negácie prostredníctvom syntaktických závislostí bola stanovená na **85,89 %**.

Druh	**pre+12**	**gen**	**atr**	**sub**	**nie**
Beletria	81,37 %	100 %	82,35 %	-	66,67 %
Šport	89,18 %	96,15 %	74,28 %	50 %	93,55 %
Recenzie	85,55 %	100 %	82,35 %	60 %	95,52 %
Projekty	94,69 %	90 %	87,5 %	66,67 %	100 %
SNK	81,95 %	88,88 %	62,30 %	66,66 %	80,60 %
Vlastné	95,19 %	100 %	80 %	100 %	93,33 %
Celkovo	85,63 %	97,12 %	72,85 %	73,68 %	90,70 %

Tabuľka 9.18: Porovnanie skóre F_1 pre rozsah negácie podľa druhov negácie.



Obr. 9.8: Porovnanie absolútnych počtov jednotlivých komponentov metrík úspešnosti pre rozsah negácie.

TP	**TN**	**FP**	**FN**	**Presnosť**	**Úplnosť**	**Správnosť**	**F_1**
4558	16911	1013	484	81,82 %	90,40 %	93,48 %	85,89 %

Tabuľka 9.19: Výsledné ukazovatele detekcie rozsahu negácie prostredníctvom syntaktických závislostí.

9.2.6 Porovnanie s predošlou generáciou riešenia

Ciele detekcie negácie navrhnutej v tejto práci boli stanovené na základe analýzy predošlých generácií riešenia opísaných v časti 5.1. Ako bolo spomenuté už tam, detekcii negácie sa venovali Ing. Martin Jaborník a Ing. Matej Kvitkovič, pričom druhý menovaný vo svojej práci uvádza kvantitatívne porovnanie úspešnosti. Toto porovnanie bolo uskutočnené na recenziách fotoaparátov, ktoré boli zahrnuté aj do evaluačného korpusu opísaného v časti 9.1.

Ako uvádza Ing. Kvitkovič [27], jeho metóda založená na EBNF gramatike určuje, či daná jednoduchá veta obsahuje alebo neobsahuje nejakú negáciu. Pod negáciou sa v tomto prípade rozumie prítomnosť ľubovoľného negátora. Metodika jeho testovania nebola podrobne uvedená a takisto neboli uvedené žiadne značkované dáta. Priebeh jeho testovania bol nasledovný:

1. Načítanie korpusu recenzií fotoaparátov.
2. Text bol rozdelený na jednoduché vety, teda rozdelené boli aj súvetia.
3. Pomocou EBNF gramatiky bolo rozhodnuté, či daná jednoduchá veta spadá do niektorého z definovaných druhov negovaných viet.
4. Toto rozhodnutie bolo manuálne skontrolované.

Inak povedané, predmetom evaluácie jeho riešenia bola iba detekcia negátorov. Podobnú evaluáciu negátorov na rovnakom korpuse vykonal aj pôvodný autor Ing. Martin Jaborník. Na porovnanie bola na tom istom korpuse recenzií fotoaparátov spustená detekcia negátorov slovníkovou metódou, ktorú navrhuje táto práca. Výsledky porovnáva tabuľka 9.20.

Autor	**Presnosť**	**Úplnosť**	$\mathbf{F}_1$
Ing. Martin Jaborník [20]	93 %	71 %	80,52 %
Ing. Matej Kvitkovič [27]	84,82 %	86,24 %	85,52 %
Slovníková metóda z tejto práce	99,11 %	97,38 %	98,23 %

Tabuľka 9.20: Porovnanie troch generácii systému na detekciu negácie.

Takéto výsledky sú vzhľadom na analýzu ich riešení uvedenú v časti 5.1 očakávateľné. Obaja autori hlásili vo svojich prácach problém s detekciou negátorov so záporným prefixom, ktorý obaja riešili udržiavaním zoznamu výnimiek. Táto práca nepoužíva zoznam výnimiek, ale slovníkové overenie, ktoré na technických textoch vykazuje vysokú úspešnosť.

Ani jeden z autorov neoveroval úspešnosť detekcie rozsahu. Metóda založená na syntaktickej analýze uvedená v tejto práci dosahuje výsledné skóre F_1 na úrovni 85,89 %, čo je vyššia hodnota než úspešnosť detekcie negátorov u predošlých autorov. Táto práca nevykonáva porovnanie úspešnosti detekcie rozsahu negácie u týchto autorov z nasledovných dôvodov:

- Ako bolo uvedené už v analýze, zásahy do programového kódu oboch riešení sú mimoriadné náročné.
- V obidvoch riešeniach je detekcia rozsahu negácie úzko zviazaná s extrakciou kľúčových slov a ich oddelenie by si vyžadovalo veľa úsilia.
- Bolo by potrebné navrhnúť formát, do ktorého by sa uložili výsledky detekcie rozsahu, a takéto ukladanie do oboch riešení implementovať.
- Zároveň by bolo potrebné mapovať ich vetné jednotky na vetné jednotky z vytvoreného korpusu, keďže každý definuje tokenizáciu iným spôsobom.
- Autori prehlasujú úspešnosť detekcie negátorov nižšiu ako je úspešnosť detekcie rozsahu negácie v tejto práci. Keďže detekcia negátorov je základným, východiskovým krokom, bez ktorého nie je možné určovať rozsah negácie, úspešnosť detekcie rozsahu negácie v ich riešeniach musí byť zákonite nižšia.

Z hľadiska kvalitatívneho porovnania sa dá skonštatovať, že predkladaný modul na detekciu negácie splnil väčšinu cieľov, ktoré si táto práca vytýčila, a zároveň mnohé z cieľov, ktoré navrhli samotní predošlí autori ako prácu do budúcna. Medzi splnené ciele patrí:

- Oddeliť detekciu negátora od detekcie rozsahu negácie.
- Zvýšiť presnosť detekcie negátorov so záporným prefixom.
- Navrhnúť samostatnú metódu na detekciu rozsahu negácie, ktorá by dokázala pracovať s viacnásobnou negáciou a negáciou v súvetiach.
- Pripraviť dostatočne veľký evaluačný korpus s jednoznačnou metodikou značkovania.
- Dôsledne overiť všetky aspekty detekcie negátorov a detekcie rozsahu negácie.
- Implementovať systém modulárne, aby ho mohol ľubovoľný externý používateľ jednoducho pripojiť ku svojej aplikácii bez nutnosti programových zásahov.

9.3 Experimenty pre anglický jazyk

Táto podkapitola obsahuje opis experimentov, ktoré boli vykonané na štandardnom anglickom korpuse BioScope opísanom v časti 3.5.3. Experimenty sú opäť stavané tak, aby zachytili obe fázy procesu detekcie negácie, a zároveň tak, aby dostatočne vyprofilovali nielen korpus, ale aj samotný proces detekcie.

Úvodný experiment 9.3.1 je cielený na odhalenie, aké druhy negátorov sú v korpuse BioScope najčastejšie zastúpené. Nasledujúci experiment 9.3.2 prináša úspešnosť detekcie týchto negátorov s využitím sémantickej databázy SentiWordNet opísanej v časti 2.8. Posledným

experimentom 9.3.3 je zistenie úspešnosti detekcie rozsahu negácie prostredníctvom metódy syntaktických závislostí navrhnutej v časti 7.2.

9.3.1 Charakteristika korpusu BioScope a zastúpenie negátorov

Korpus BioScope bol tvorený tak, aby v ňom bola vyznačená negácia a neistota s ohľadom na medicínske pojmy. Cieľom prvého experimentu je pripraviť charakteristiku korpusu, potvrdiť neprítomnosť adjektívnych negátorov a zistiť, aké negátory sú najčastejšie zastúpené. Použitý bol iba súbor *full_papers.xml*, ktorý obsahuje vety vybrané z medicínskych článkov. Súhrnnú charakteristiku tohto súboru prináša tabuľka 9.21.

Viet	**Slov**	**Negovaných viet**	**Negácií**	**Negácií na 1 vetu**	**Dvojitých negácií**
2432	66386	325	411	0,17	40

Tabuľka 9.21: Charakteristika korpusu BioScope z pohľadu negácie.

Na rozdiel od slovenského jazyka, v ktorom je každý negátor jednoslovný, obsahuje anglický jazyk mnoho viacslovných negátorov. Podrobnú analýzu tejto problematiky prináša sekcia 3.4. Z tohto dôvodu je zaujímavým ukazovateľom zastúpenie jednotlivých negátorov v korpuse BioScope. Cieľom tohto experimentu je teda uviesť početnosť všetkých negátorov, ktoré sú v korpuse zastúpené aspoň štyrikrát.

Negátor	**Počet**
not	204
no	49
without	26
lack	19
cannot	13
rather than	13
none	11
fail	10
neither	6
absence	6
instead of	4
iné	27

Tabuľka 9.22: Porovnanie zastúpenia negátorov v korpuse BioScope.

Postup vykonania experimentu:

1. Načítanie dokumentov korpusu BioScope.

2. V každej vete boli identifikované očakávané negátory.
3. Výsledky boli číselne zoskupené do tabuľky 9.23.

Z tabuľky je zjavné, že BioScope skutočne neobsahuje značkovanie adjektívnych negátorov, pretože rozsahom adjektívnych negátorov sú samotné negátory. Z hľadiska viacslovných negátorov vystupuje do popredia dvojica *rather than* a *instead of*. Ako bolo očakávané, najčastejším negátorom je *not*.

9.3.2 Detekcia negátorov

Detekcia negátora je aj v anglickom jazyku základným krokom pri detekcii negácie, od ktorého sa bude odvíjať úspešnosť detekcie rozsahu negácie. Na rozdiel od slovenského jazyka, kde bola samostatná kapitola venovaná ohybným negátorom, tento experiment je stavaný na detekciu všetkých negátorov v korpuse dokumentov.

Postup vykonania experimentu:

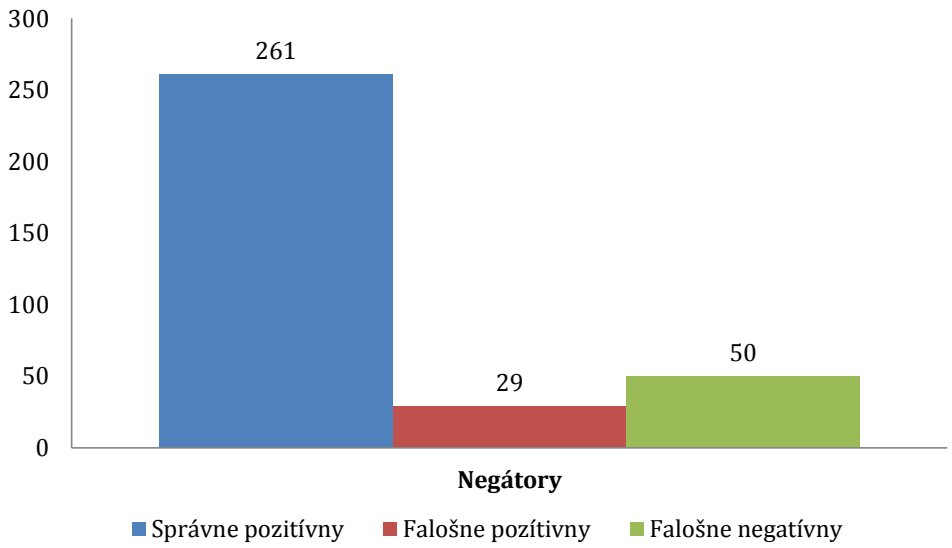
1. Načítanie dokumentov korpusu BioScope.
2. V každej vete boli identifikované očakávané negátory.
3. Spustenie značkovania negátorov s využitím slovníka polarity SentiWordNet.
4. Každé slovo bolo automaticky overené, či sa ho podarilo korektne vyznačiť.
5. Výsledky boli číselne zoskupené do tabuľky 9.23 a grafu 9.9.

TP	**TN**	**FP**	**FN**	**Presnosť**	**Úplnosť**	**F_1**
361	64705	29	50	92,56 %	87,83 %	90,13 %

Tabuľka 9.23: Úspešnosť detekcie negátorov v korpuse BioScope.

Dôvod väčšiny falošne negatívnych slov spočíva vo výstupe parsera Stanford CoreNLP, ktorý slová ako *cannot* a *can't* rozdelí na dvojicu slov *can, not*. Ďalším dôvodom je rozmanitosť anglického jazyka, teda slová nepokryté v množine zahrnutých negátorov. Falošne pozitívne slová sú väčšinou zasa také, ktoré SentiWordNet vyhodnotí ako skôr negatívne, ale anotátori BioScopu ich za negatory neoznačili.

Celkové výsledky sa však pohybujú na dobrej úrovni 90,13 %, ktorá by mala pokladať dobrú východiskovú pozíciu do fázy detekcie rozsahu negácie. Zároveň však treba povedať, že zatiaľ čo slovenský korpus bol viacžánrový, v BioScope sa nachádzajú iba medicínske texty. Hodnotu úspešnosti na textoch iného žánru nemožno predpokladať.



Obr. 9.9: Úspešnosť detekcie negátorov v korpuse BioScope.

9.3.3 Detekcia rozsahu negácie

Po vyhodnotení úspešnosti detekcie negátorov bolo možné pristúpiť ku vyhodnoteniu úspešnosti detekcie rozsahu negácie prostredníctvom syntaktických závislostí. Tento experiment priamo vyhodnocuje úspešnosť metódy opísanej v časti 7.2 tak, ako bola navrhnutá pomocou grafov závislostných stromov. Cieľom experimentu je kvantitatívne vyhodnotiť úspešnosť detekcie rozsahu v korpuse BioScope.

Postup vykonania experimentu:

1. Načítanie dokumentov korpusu BioScope.
2. Spustenie značkovania negátorov s využitím slovníka polarity SentiWordNet.
3. Spustenie značkovania negácie prostredníctvom syntaktických závislostí.
4. Pre každú vetu boli nájdené očakávané negátory podľa manuálneho značkovania v korpuse. K nim boli pridané prípadné ďalšie negátory, ktoré boli nesprávne vyznačené prototypom.
5. Pre každý negátor z predošlého kroku boli preskúmané všetky slová vo vete a bola určená zhoda medzi manuálnym a automatickým značkovaním rozsahu negácie.
6. Výsledky boli vynesené do tabuľky 9.24.

Výsledky ukazujú úspešnosť detekcie rozsahu negácie na úrovni 53,38 %. Toto číslo je pripísané nepravidelnostiam vo výsledkoch syntaktického analyzátora Stanford CoreNLP

TP	**TN**	**FP**	**FN**	**Presnosť**	**Úplnosť**	**Správnosť**	**F_1**
1662	10630	1707	1169	49,33 %	58,15 %	80,89 %	53,38 %

Tabuľka 9.24: Výsledné ukazovatele detekcie rozsahu negácie v korpuse BioScope.

a spôsobu, akým je v anglickom jazyku definovaný rozsah negácie. Anglický jazyk sa vyznačuje striktným slovosledom, pričom syntaktická analýza sa ukazuje byť užitočnejšou v jazykoch s voľnejším slovosledom.

Mnohé systémy na detekciu rozsahu negácie v anglickom jazyku opísané v analýze sa pozerajú na rozsah negácie ako na *„niekoľko slov vľavo alebo vpravo od negátora“*. Na takéto spracovanie je potom vhodnejšia analýza konštituentov a nie závislostí.

Hoci bola vyvinutá snaha o optimalizáciu navrhnutého modelu, pri analýze značkovania korpusu BioScope bolo náročné identifikovať vzory, ktoré by boli všeobecne aplikovateľné a pozdvihli by hodnotu mierky F_1. Druhým aspektom je, aké výsledky by metóda dosiahla na korpuse iného textového žánru. Tvorba vlastného korpusu pre anglický jazyk je však mimo záberu tejto diplomovej práce a alternatívny korpus ConnanDoyle-neg sa nepodarilo v rámci technických možností plnohodnotne sprístupniť a využiť.

9.4 Zhrnutie evaluácie

Táto kapitola sa venovala podrobnému overeniu úspešnosti procesu detekcie negácie v slovenskom jazyku navrhnutému v kapitole 6.2 a anglickom jazyku navrhnutom v kapitole 7.2. Na účely evaluácie bol vytvorený vlastný korpus zložený z rôznych textových žánrov, ktoré mali simulovať reálne podmienky nasadenia modulu do prevádzky.

V slovenskom jazyku sa podarilo dosiahnuť pomocou slovníkového prístupu 93,96% úspešnosť detekcie ohybných negátorov, pričom drvivá väčšina týchto negátorov je v korpuse tvorená prefixom *ne-*. So zahrnutím neohybných negátorov sa presnosť nachádza na úrovni 97,45 %, úplnosť na úrovni 93,42 %, a teda celková úspešnosť v podobe mierky F_1 na úrovni 95,39 %. Úspešnosť detekcie rozsahu negácie dosahuje hodnotu 85,89 %.

V anglickom jazyku, evaluovanom na korpuse BioScope, dosiahla detekcia negátorov úspešnosť 90,13 %. Detekcia rozsahu negácie prostredníctvom syntaktických stromov vykazuje úspešnosť 53,38 %. Každý vykonaný experiment bol podrobne opísaný a po prezentácii výsledkov obsahuje diskusiu, v ktorej sa uvádzajú zdôvodnenia týchto výsledkov.

10 Zhodnotenie

Táto diplomová práca sa zameriava na fenomén negácie v slovenskom a anglickom jazyku z pohľadu spracovania prirodzeného jazyka. Analýza diskutuje celý proces predspracovania dokumentu. Každá fáza predspracovania bola opísaná z konceptuálneho a teoretického hľadiska a zároveň boli zanalyzované moderné technologické prostriedky, ktoré je možné použiť na jej realizáciu.

Pozornosť bola venovaná najmä takým riešeniam, ktoré podliehajú dvom hlavným kritériám: súčasnosť a overenosť. Väčšina služieb a prostriedkov, ktoré boli v rámci analýzy opísané, sú v čase písania tohto textu (máj 2017) staré najviac tri roky.

Záverom analýzy predspracovania bol výber štandardného nástroja na extrakciu čistého textu Apache Tika, štandardného balíka nástrojov na spracovanie anglického jazyka Stanford CoreNLP a balíka nástrojov Synpar pre slovenský jazyk. Ten bolo zásahom do programového kódu potrebné upraviť z webovej služby na lokálnu aplikáciu.

Výhodou týchto nástrojov je prítomnosť závislostnej analýzy ako moderného prístupu ku spracovaniu prirodzeného jazyka. Výstupom sú vzťahy dvojíc slov na úrovni nadradenosti a podradenosti, ktoré sa javia ako vhodné východisko pre reprezentáciu vety.

Samostatná časť analýzy je venovaná negácii ako jazykovému prostriedku v slovenskom aj anglickom jazyku. Rozčleňuje ju na dve fázy, ktorými sú detekcia negátora a detekcia rozsahu negácie. Okrem toho spomína význačné vlastnosti negácie, akými sú dvojitá negácia alebo presah jednoduchej vety.

Pôvodným zámerom tejto práce bolo nadviazať na dva prechádzajúce projekty venujúce sa tejto problematike v slovenskom jazyku. Na základe ich analýzy bol však vynesený záver, že sa v nich nebude pokračovať, namiesto toho práca prinesie vlastný prístup ku detekcii negácie. K tomu pomohlo začlenenie nástroja na syntaktickú analýzu slovenských viet, ktorý bol zverejnený až v priebehu riešenia tohto projektu.

Návrhová časť práce sa venuje detekcii negátorov a detekcii rozsahu negácie v slovenskom a anglickom jazyku. Pre slovenský jazyk práca navrhuje dva spôsoby na identifikáciu ohybných negátorov so záporným prefixom, menovite slovníkový prístup a využitie sémantických vzťahov *word2vec*. V anglickom jazyku bola pozornosť venovaná slovesám

bez záporného prefixu alebo sufixu, ktoré majú napriek tomu zápornú výpovednú hodnotu, a sú detegovateľné prostredníctvom analýzy sentimentu.

Lingvistická analýza negácie vyústila do návrhu vlastnej metódy na detekciu rozsahu negácie v slovenskom jazyku prostredníctvom skúmania syntaktických závislostí slov, ktorá je jadrom tejto práce. Vybrané druhy negácií definuje na úrovni prechodov syntaktického stromu. V ďalšej kapitole bol formulovaný pokus o aplikáciu rovnakej metódy na anglický jazyk, v ktorom už podobné metódy boli predtým navrhnuté.

Na základe uvedených metód na detekciu negácie bol vytvorený návrh vlastného systému, ktorého cieľom je detegovať negátory slovníkovým prístupom a rozsah negácie prechodmi cez syntaktický strom v slovenskom aj anglickom jazyku. Systém bol rozdelený na viacero samostatne stojacich modulov s dôsledne odčlenenou aplikačnou a prezentačnou logikou. Detekcia negácie je realizovaná ako samostatný balík v programovacom jazyku Java. Takýmto spôsobom môže byť navrhnutý modul začlenený do ľubovoľného externého systému na spracovanie textu.

Hlavným cieľom práce bolo dospieť ku evaluácii navrhnutých metód. Pre tento účel bola vytvorená vlastná množina dokumentov zložená z 5 rôznych zameraní, menovite beletrie, športu, technických recenzií, tímových projektov a viet zo Slovenského národného korpusu. Prvé štyri boli značkované v rámci tejto práce, posledný menovaný druh viet bol prebratý zo súbežne riešenej práce [13]. Výsledný vytvorený korpus obsahuje 4146 viet, v ktorých bolo manuálne vyznačených 1338 negácií.

Nad týmto korpusom bola prostredníctvom implementovaného modulu na detekciu negácie spustená séria šiestich experimentov overujúcich úspešnosť navrhnutých metód pre slovenský jazyk. Výsledkom je úspešnosť detekcie ohybných negátorov slovníkovou metódou na úrovni 93,96 %. Ak sa k tomu pripočíta detekcia neohybných negátorov, celková úspešnosť detekcie negátorov v slovenskom jazyku pre uvádzaný systém je na úrovni 95,39 %. Experimenty cielené na detekciu rozsahu negácie prostredníctvom syntaktických závislostí vykázali veľmi dobrú celkovú úspešnosť metódy 85,89 %.

Ďalším krokom bola evaluácia adaptácie metód pre anglický jazyk, ktorá bola spustená na štandardnom anglickom korpuse BioScope. Cieľom týchto experimentov nebolo dosiahnuť finálnu úspešnosť, skôr zhodnotiť využiteľnosť a potenciál adaptácie metódy na detekciu rozsahu negácie pre anglický jazyk.

Úspešnosť detekcie negátorov sa pohybovala na úrovni 90,13 %, čo tvorilo dobrú východiskovú pozíciu pre algoritmus detekcie rozsahu. Ten však dosiahol úspešnosť len na úrovni 53,38 %. Dôvod spočíva najmä v odlišnosti syntaxe anglického jazyka, pre ktorú sa javí ako perspektívnejšie riešenie spracúvať text nie na úrovni závislostí, ale konštituentov. Navrhnutá metóda na detekciu rozsahu je vhodnejšia skôr pre flektívne jazyky s voľným slovosledom.

10.1 Ďalšia práca

Ako každá práca, ani táto nezostala bez niekoľkých nevyriešených problémov alebo otvorených otázok do budúcnosti. Táto časť poskytuje ich prehľad pre jednoduchú orientáciu.

- Napriek veľkej snahe o použitie čo najmodernejších nástrojov na spracovanie slovenského jazyka, pri evaluácii došlo ku prekvapivému zisteniu, že pri spracovaní niektorých slov zlyháva už fáza lematizácie. Ide najmä o príčastia ako *nepíšuci, nerobiaci*, ktoré sa v skutočných vetách zvyknú vyskytovať v rôznom skloňovaní. Dôvod spočíva v slovníkovom základe lematizácie, pričom príčastia a prechodníky sa v žiadnej štandardnej kodifikačnej príručke nenachádzajú. Je predpoklad, že dokonalejšia lematizácia by dokázala zvýšiť úspešnosť jednak detekcie negátorov, ale aj detekcie rozsahu negácie.

- Ďalšiu časť neúspešne spracovaných viet tvorí nesprávne určený slovný druh, kde sa ako problematické ukazovali opäť najmä prechodníky a príčastia, v niektorých prípadoch pomocné slovesá alebo slová tvoriace menný prísudok. Bez správneho určenia slovného druhu nie je možné správne určiť ani negátor.

- Iným prípadom bol nesprávne určený pád. V drvivej väčšine išlo o zámenu nominatívu a akuzatívu pri slovách, ktoré majú v týchto pádoch rovnaký tvar. V takom prípade zlyháva detekcia atribútovej a subjektovej negácie, ktorá sa priamo spolieha na vyhľadanie plnovýznamového slova, ktoré je v zhode v páde s atribútom.

- Z hľadiska samotnej metódy sa ako vhodný predmet ďalšieho výskumu javia spojky v podraďovacom súvetí. Momentálne sa v systéme nachádza 11 vymenovaných spojok, ktoré v súvetí neprenášajú negáciu do podradenej vety. Tieto boli vybrané ako optimalizácia predikátovej negácie na základe publikácie [14]. Hoci experiment ukázal, že ich začlenením ako výnimiek došlo ku zvýšeniu úspešnosti detekcie negácie, bolo by vhodné dôslednejšie preskúmať zastúpenie spojok podraďovacích súvetiach a sledovať, ktoré spojky v skutočnosti sémanticky prenášajú negáciu do podraďovacej vety.

- Vzhľadom na vysokú úspešnosť detekcie ohybných negátorov v slovenskom jazyku nebol implementovaný prístup cez sémantické vzťahy *word2vec* navrhnutý v časti 2.7. Stálo by za pokus vyskúšať, do akej miery by sa tento postup ukázal byť úspešný vzhľadom na dostupné natrénované sady.

- Experiment ukázal veľmi nízke zastúpenie negátorov s cudzojazyčným záporným prefixom vo vytvorenom korpuse dokumentov. To je spôsobené tým, že takéto negátory sú typické pre odborné texty. Na lepšie posúdenie úspešnosti pri tejto kategórii slov by bolo vhodné označkovať odborné texty, napríklad, z oblasti medicíny alebo biológie a pridať ich do korpusu. Predpokladom je, že prehlasovaná úspešnosť by sa pri väčšom zastúpení takýchto slov zvýšila.

- V rámci experimentov sa podarilo zopakovať metodiku testovania detekcie negácie u predošlých dvoch autorov, Ing. Martina Jaborníka a Ing. Mateja Kvitkoviča, ktorí

evaluovali iba detekciu negátorov. Napriek tomu ich práce prezentujú prístup ku detekcii rozsahu, ktorý v istej forme implementujú. Zásah do ich programového kódu bol však natoľko náročný, že sa nepodarilo dosiahnuť žiaden zrealizovateľný komunikačný alebo transformačný kanál medzi ich výsledkami a označkovaným korpusom. Úspešnosť detekcie rozsahu negácie pomocou ich metód sa preto nepodarilo zistiť ani odhadnúť.

- Mnoho otázok zostalo otvorených pri anglickom jazyku. Implementácia metód bola cielená na korpus BioScope, v ktorom nie sú značkované adjektívne negácie, pretože v anglickom jazyku patrí do rozsahu adjektívneho negátora základ slova bez negačného prefixu. Z toho dôvodu nebol implementovaný detektor záporných prefixov a sufixov, ktorý by pre reálne použitie bolo vhodné pridať.

- Korpus BioScope, hoci najdostupnejší, nie je jediným anglickým značkovaným korpusom. Bola snaha o evaluáciu na alternatívnom korpuse ConnanDoyle-neg, ktorá však nebola vykonaná z dôvodu iného chápania vetných a slovných jednotiek. Tento korpus zavádza kompozitné slová, aby mohol zachytiť jav adjektívnej negácie. Mnoho pojmov zhlukuje do pomenovaných entít a tie neguje ako celok. V rámci technickej realizácie sa nepodarilo nájsť rozumné riešenie, ako tieto odlišnosti v reprezentácii namapovať na reprezentáciu viet v predkladanom systéme. Okrem týchto dvoch existujú aj ďalšie korpusy, ktoré však nie sú voľne dostupné.

- Existuje predpoklad, že dôslednejším skúmaním negovaných viet by sa dalo vylepšiť pravidlá pre detekciu rozsahu negácie v anglickom jazyku. Korpus BioScope bol anotovaný dvoma nezávislými anotátormi a pri manuálnom skúmaní výsledkov sa objavilo mnoho nekonzistencií, ktoré môžu byť spôsobené odlišným vnímaním negácie u anotátorov, ale zároveň môžu byť zdôvodniteľné gramatickými pravidlami alebo javmi, ktoré na prvý pohľad ostávali skryté. Aj z tohto dôvodu by bolo zaujímavé porovnať úspešnosť detekcie rozsahu na inom anglickom korpuse.

Literatúra

[1] Systems and software engineering – vocabulary. *ISO/IEC/IEEE 24765:2010(E)*, pages 1–418, Dec 2010.

[2] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA).

[3] I. Barbantan and R. Potolea. Exploiting word meaning for negation identification in electronic health records. In *Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on*, pages 1–7, May 2014.

[4] Ralph Bergmann. *Experience Management: Foundations, Development Methodology, and Internet-based Applications*. Springer-Verlag, Berlin, Heidelberg, 2002.

[5] T. Bohne, S. Rönnau, and U. Borghoff. Efficient keyword extraction for meaningful document perception. In *Proceedings of the 11th ACM Symposium on Document Engineering*, DocEng '11, pages 185–194, New York, NY, USA, 2011. ACM.

[6] Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 34(5):301 – 310, 2001.

[7] Viviana Cotik, Vanesa Stricker, Jorge Vivaldi, and Horacio Rodriguez. Syntactic methods for negation detection in radiology reports in spanish. 2016.

[8] Cer D., de Marneffe M., Jurafsky D., and Manning Ch. Parsing to stanford dependencies: Trade-offs between speed and accuracy. *International Journal on Artificial Intelligence Tools*, 2010. Online dostupné na `http://nlp.stanford.edu/pubs/lrecstanforddeps_final_final.pdf` (12. december 2016).

[9] G. Domeniconi, G. Moro, R. Pasolini, and C. Sartori. A study on term weighting for text categorization: a novel supervised variant of tf.id. In *Proceedings of 4th International Conference on Data Management Technologies and Applications*, volume 1, July 2015.

[10] J. Dvorak. *Expertní systémy*, January 2004. Online dostupné na `http://www.uai.fme.vutbr.cz/~jdvorak/Opory/ExpertniSystemy.pdf` (12. december 2016).

[11] The Apache Software Foundation. Apache tika supported document formats. Online dostupné na `http://tika.apache.org/1.12/formats.html` (12. december 2016).

[12] B. Fuglede and F. Topsoe. Jensen-shannon divergence and hilbert space embedding. In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, pages 31–, June 2004.

[13] J. Gaborik. Rozpoznávanie negácie v texte pomocou strojového učenia. Master's thesis, Slovenská technická univerzita v Bratislave, Bratislava, Slovakia, 2017.

[14] F. Gaher. On propositional connectives: Logic versus linguistics. pages 23–37, 2001.

[15] R. Garabik, L. Gianitsova, A. Horak, and Simkova M. Tokenizácia, lematizácia a morfologická anotácia Slovenského národného korpusu, May 2004.

[16] A. Hogenboom, P. van Iterson, B. Heerschop, F. Frasincar, and U. Kaymak. Determining negation scope and strength in sentiment analysis. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 2589–2594, Oct 2011.

[17] R. Horvath. Podpora výkladu neznámeho pojmu pri prehľadávaní v slovenčine. Master's thesis, Slovenská technická univerzita v Bratislave, Bratislava, Slovakia, 2011.

[18] Rodney D. Huddleston and Geoffrey K. Pullum. *The Cambridge Grammar of the English Language*. Cambridge University Press, April 2002.

[19] Exman I., Llorens J., Fraga A., and Alvarez-Rodríguez J.M. Skyware: The unavoidable convergence of software towards runnable knowledge. 21(11):1405–1424, nov 2015.

[20] M. Jabornik. Extrakcie z kľúčových slov pre vyhľadávanie znalostí. Master's thesis, Slovenská technická univerzita v Bratislave, Bratislava, Slovakia, 2011.

[21] Cheng Juan. Research and implementation english morphological analysis and part-of-speech tagging. In *E-Health Networking, Digital Ecosystems and Technologies (EDT), 2010 International Conference on*, volume 2, pages 496–499, April 2010.

[22] Noriaki Kawamae. Supervised n-gram topic model. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 473–482, New York, NY, USA, 2014. ACM.

[23] Su Nam Kim, Timothy Baldwin, and Min-Yen Kan. Evaluating n-gram based evaluation metrics for automatic keyphrase extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 572–580, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[24] M. Kompan. Úvod do vyhľadávania informácií, 2016.

[25] V. Kovar. Syntaktická analýza s využitím postupné segmentace věty. Master's thesis, Masarykova univerzita, Praha, Czech Republic, 2008.

[26] V. Kvasnicka, L. Benuskova, and J. Pospichal. *Úvod do teórie neurónových sietí*, January 1997. Online dostupné na `http://www2.fiit.stuba.sk/~kvasnicka/Free%20books/Uvod%20do%20teorie%20neuronovych%20sieti_all.pdf` (12. december 2016).

[27] M. Kvitkovic. Extrakcia kľúčových slov z dokumentov s využitím rozpoznávania negácie. Master's thesis, Slovenská technická univerzita v Bratislave, Bratislava, Slovakia, 2015.

[28] J. Loebl. Automatické spracovanie textu - syntaktická analýza vety. Master's thesis, Slovenská technická univerzita v Bratislave, Bratislava, Slovakia, 2016.

[29] M. Maly. Natural language processing with application to slovak language. Master's thesis, Univerzita Komenského, Bratislava, Slovakia, 2011.

[30] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13:2004, 2004.

[31] Saeed Mehrabi, Anand Krishnan, Sunghwan Sohn, Alexandra M. Roch, Heidi Schmidt, Joe Kesterson, Chris Beesley, Paul Dexter, C. Max Schmidt, Hongfang Liu, and Mathew Palakal. Deepen: A negation detection system for clinical text incorporating dependency relation into negex. *Journal of Biomedical Informatics*, 54:213 – 219, 2015.

[32] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[33] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[34] Roser Morante and Walter Daelemans. Conandoyle-neg: Annotation of negation cues and their scope in conan doyle stories. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *LREC*, pages 1563–1568. European Language Resources Association (ELRA), 2012.

[35] Neuzilla. Toxy documentation. Online dostupné na `http://toxy.codeplex.com/documentation` (12. december 2016).

[36] J. Pavlovic. Predložka bez ako prostriedok gramatickej negácie. In *Slovenská reč*.

[37] J. Pavlovic. Prostriedky negácie v spisovnej slovenčine. In *Slovenská reč*.

[38] J. Pavlovic. Syntax slovenského jazyka ii.

[39] J. Pavlovic. *Negácia v jednoduchej vete*. Slavistický kabinet SAV, 1 edition, 2003.

[40] I. Polasek, I. Ruttkay-Nedecky, Ruttkay-Nedecky P., T. Toth, A. Cernik, and P. Dusek. Information and knowledge retrieval within software projects and their graphical representation for collaborative programming. pages 269–295, 2013.

[41] I. Polasek and M. Uhlar. Extracting, identifying and visualisation of the content, users and authors in software projects. pages 269–295, 2013.

[42] M. F. Porter. Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[43] N. Prollochs, S. Feuerriegel, and D. Neumann. Enhancing sentiment analysis of financial news by detecting negation scopes. In *System Sciences (HICSS), 2015 48th Hawaii International Conference on*, pages 959–968, Jan 2015.

[44] L. Sesera, P. Grec, and P. Navrat. *Architektúra softvérových systémov*. Slovenská technická univerzita v Bratislave, 1 edition, 2013.

[45] Grigori Sidorov, Francisco Velasquez, Efstathios Stamatatos, Alexander Gelbukh, and Liliana Chanona-Hernandez. Syntactic n-grams as machine learning features for natural language processing. *Expert Syst. Appl.*, 41(3):853–860, February 2014.

[46] K. Sparck. Document retrieval systems. chapter A Statistical Interpretation of Term Specificity and Its Application in Retrieval, pages 132–142. Taylor Graham Publishing, London, UK, 1988.

[47] S. Sunghwan, S. Wu, and C. Chute. Dependency parser-based negation detection in clinical narratives.

[48] Hideyuki Tanushi, Hercules Dalianis, Martin Duneld, Maria Kvist, Maria Skeppstedt, and Sumithra Velupillai. Negation scope delimitation in clinical text using three approaches: Negex; pycontextnlp and synneg. In Stephan Oepen, Kristin Hagen, and Janne Bondi Johannessen, editors, *NODALIDA*, volume 85 of *Linköping Electronic Conference Proceedings*, pages 387–397. Linköping University Electronic Press, 2013.

[49] I. Witten, G. Paynter, E. Frank, C. Gutwin, and C. Nevill-Manning. Kea: Practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, DL '99, pages 254–255, New York, NY, USA, 1999. ACM.

PRÍLOHY

A Technická dokumentácia

Jadrom technickej dokumentácie je opis implementácie vybraných častí modulu na detekciu negácie. Odkazuje sa na diagram tried uvedený ako obrázok A.1.

A.1 Načítanie a ukladanie dát

Táto sekcia opisuje implementáciu tried určených na načítavanie vstupného korpusu. Korpus dokumentov sa typicky načítava z pevného disku a môže ísť buď o klasické dokumenty ľubovoľného formátu, štruktúrovaný slovenský evaluačný korpus (XML) alebo anglický korpus BioScope (XML).

- `AbstractCorpusReader`. Abstraktná trieda poskytujúca predpripravenú funkcionalitu na rekurzívne prehľadávanie priečinka a inicializáciu detektora jazyka. Konštruktor prijíma parameter s priečinkom, ktorý prehľadá. Abstraktnými metódami sú:
  - `createCorpus()`. Spustí sa spracovanie súborov nájdených v priečinku.
  - `createDocument(String path)`. Alternatívna funkcia na spracovanie jediného dokumentu určeného cestou.
- `CorpusReaderFileSystem`. Rozšírenie abstraktnej triedy. Používaný pri bežných dokumentoch. Priamo volá nástroj Apache Tika.
- `CorpusReaderXML`. Rozšírenie abstraktnej triedy. Slúži na načítanie vlastného formátu XML opísaného v sekcii C.2. Využíva externú knižnicu na čítanie XML súborov `org.xml.sax`. Po načítaní XML nevolá Apache Tika, ale priamo inicializuje parser a spracúva text do dokumentov.
- `CorpusReaderBioScope`. Rozšírenie abstraktnej triedy. Korpus BioScope obsahuje rekurzívne vnáranie XML elementov typu `<xcope>`, ktoré môžu obsahovať elementy `<cue>` alebo znova `<xcope>`. Na načítanie tejto štruktúry bol použitý vzor *Composite*.

  - `ICompositeScope`. Jednotné rozhranie pre návrhový vzor. Jadrom vzoru sú najmä metódy `getText()` a `getContent()`, ktoré v prípade kompozitnej triedy vracajú vnorený čistý text a negačné elementy, a v prípade listového elementu vlastný text a elementy.
  - `CompositeScope, TextScope, CueScope`. Implementácie rozhrania, pričom listové triedy zodpovedajú XML elementom.

- `CorpusWriterXML`. Trieda na zapísanie do štandardného formátu evaluačného korpusu, ako uvádza príloha C.2. Využíva externú knižnicu na zápis XML súborov `org.xml.sax`.

A.2 Spracovanie textu

Triedy `Document, Sentence, AbstractAnnotatedWord` spoločne tvoriace reprezentáciu dokumentov boli tvorené ako dátové úložiská, ktorých cieľom je udržovať potrebné údaje a vystavovať ich na ďalšie spracovanie. Z toho dôvodu je väčšina zastúpených atribútov verejná. Trieda reprezentujúca slovo je abstraktná, pretože konštruktory konkrétnych tried `AnnotatedWordSlovak`, `AnnotatedWordEnglish` priamo spracúvajú výstup z jazykovo závislého analyzátora.

- `ITextParser`. Jednotné rozhranie pre celé spracovanie textu. Poskytuje metódy:
  - `parse(String text, List<Sentence> sentences)`. Zoznam parametrov je vstupno-výstupný, teda vstupom je neštruktúrovaný súvislý text a výstupom naplnenie inicializované poľa viet.
  - `detectNegators(List<SentenceNKE> sentences)`. Metóda pracujúca so zoznamom vytvorených viet. Účelom je iterovať cez slová zadaných viet a vyznačovať v nich negátory.
  - `detectNegationScope(List<SentenceNKE> sentences)`. Metóda pracujúca so zoznamom vytvorených viet. Účelom je iterovať cez slová zadaných viet a vyznačovať v nich rozsah negácie.

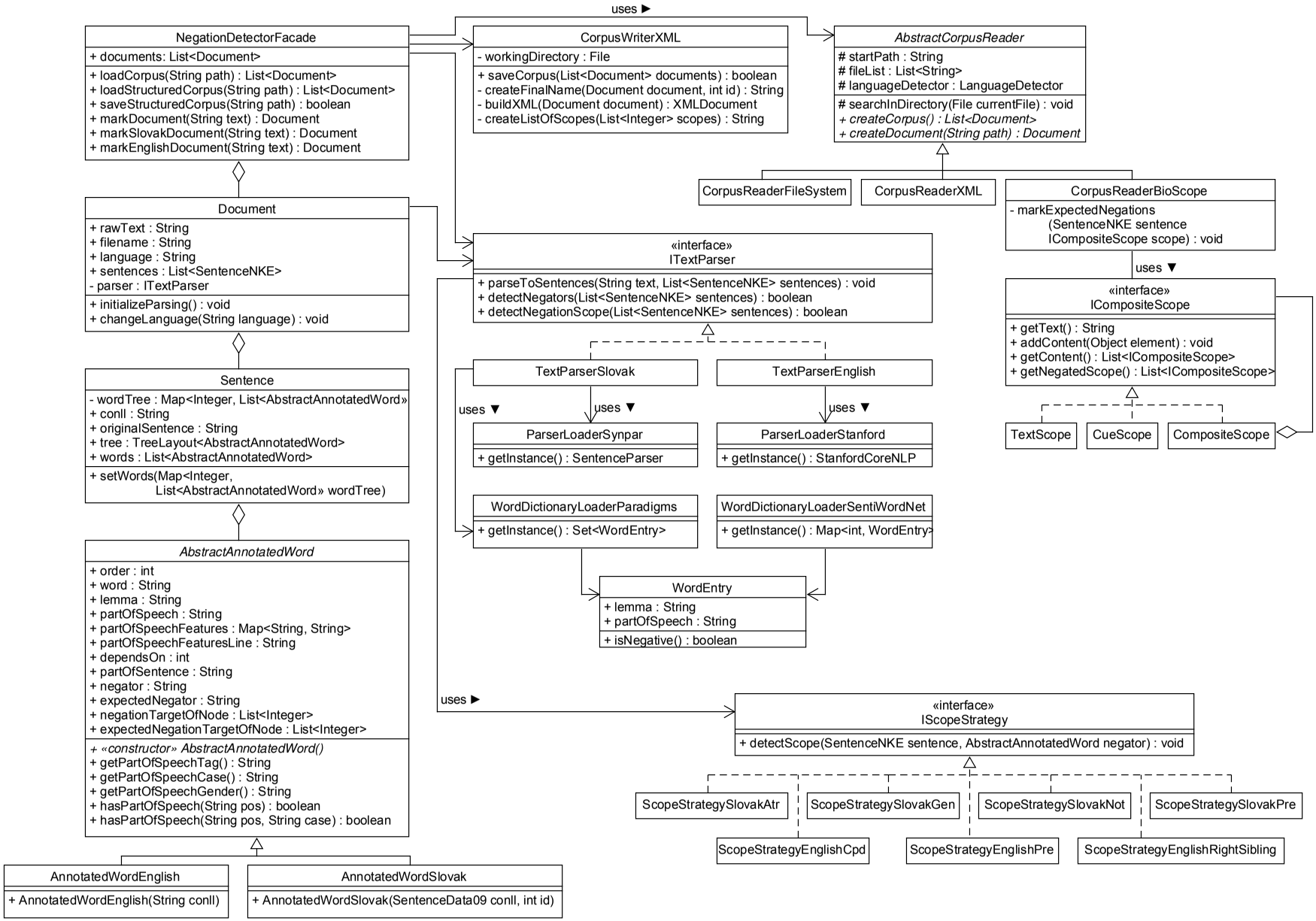
- `TextParserSlovak`. Implementácia rozhrania pre slovenský jazyk.
  - Vo fáze spracovania textu kontaktuje singleton `ParserLoaderSynpar`, ktorý sa stará o to, aby parser Synpar bol inicializovaný iba raz, pretože toto trvá približne 90 sekúnd. Následne pracuje s objektami triedy `SentenceData09`, ktoré poskytujú pole reťazcov vo formáte ConLL09. Vytvára objekty triedy `AnnotatedWordSlovak`, ktoré v konštruktore spracúvajú tento formát do jednotlivých atribútov.

  - Vo fáze detekcie negátorov kontakuje singleton `WordDictionaryLoaderParadigms`, ktorý sa stará o to, aby bola morfologická databáza načítaná iba raz, pretože toto trvá približne 5 sekúnd. Využíva objekt triedy `NegativePrefixSlovakParadigms Strategy` implementujúci rozhranie `INegativePrefixStrategy` slúžiaci na detekciu negátora slovníkovou metódou. Rozhranie je pripravené, ak by sa v budúcnosti implementovala stratégia cez word2vec.
  - Vo fáze detekcie rozsahu negácie vytvára zoznam stratégií na detekciu rozsahu. Každý negátor má priradený trojznakový druh. V tejto metóde je interne vytvorený slovník druhov negácií, ku ktorým podľa vzoru *Strategy* vytvára príslušné inštancie implementácií rozhrania `IScopeStrategy`. Tento proces je demonštrovaný kusom kódu nižšie.

- `TextParserEnglish`. Implementácia rozhrania pre anglický jazyk.
  - Vo fáze spracovania textu kontaktuje singleton `ParserLoaderStanford`, ktorého načítanie trvá približne 25 sekúnd. Následne vytvára objekty triedy`AnnotatedWord English`.
  - Vo fáze detekcie negácie komunikuje so singletonom `WordDictionaryLoaderSenti WordNet`. Na značenie negátorov využíva `NegativePrefixEnglishStrategy`.

```
@Override
public void detectNegationScope(List<SentenceNKE> sentences) {
	Map<String, IScopeStrategy> strategyMap = new HashMap<String,
	    IScopeStrategy>();
	strategyMap.put("gen", new ScopeStrategySlovakGen());
	strategyMap.put("pre", new ScopeStrategySlovakPred());
	strategyMap.put("atr", new ScopeStrategySlovakAttr());
	strategyMap.put("nie", new ScopeStrategySlovakNot());

	for(SentenceNKE sentence : sentences) {
		for(AbstractAnnotatedWord word : sentence.getWords()) {
			IScopeStrategy strategy = strategyMap.get(word.negator);
			if(strategy != null) {
				strategy.detectScope(sentence, word);
			}
		}
	}
}
```



Obr. A.1: Diagram tried vytvorený na základe funkcionálneho pohľadu.

B Inštalačná príručka

Táto príručka opisuje kroky nutné na spustenie projektu. Tie je nevyhnutné vykonať, ak sa modul použije exportovaný ako knižnica *jar*, ale aj pred spustením z vývojového prostredia pre jazyk Java.

Na spustenie aplikácie je nevyhnutné skopírovať priečinok *kniznice* z priloženého optického média na pevný disk. V totmo priečinku sa nachádzajú modely pre parsery a slovníky. Aplikácia si načítava cestu k týmto súborom z konfiguračného súboru *NegationDetector.ini*, ktorý sa musí nachádzať v domovskom priečinku používateľa. Pre systém Windows je príkladom cesta *C:/Users/Meno/NegationDetector.ini*, pre iné systémy, napríklad, */home/meno/NegationDetector.ini*.

Samotný konfiguračný súbor sa nachádza taktiež na priloženom médiu v priečinku *konfiguracia*. Cesty v konfiguračnom súbore môžu obsahovať medzeru. Ukážka obsahu tohto súboru:

```
morphologicalDatabase=F:/Java/lib/ma-2015-02-05.txt
sentiWordNet=F:/Java/lib/SentiWordNet_3.0.0_20130122.txt
segmentator=F:/Java/lib/en-sent.bin
slovakModel=F:/Java/lib/model-pred
taggerModel=F:/Java/lib/slovak2-utf8.par
```

Na úspešné sprevádzkovanie systému je ďalej nevyhnutné vykonať nasledovné kroky:

1. Prevziať interpreter Perl[1] a nainštalovať ho.
2. Prevziať a nainštalovať TreeTagger[2].
3. Prevziať slovenský jazykový súbor[3] a vložiť ho do priečinku *TreeTagger/lib*.
4. Pre systém Windows pridať v *TreeTagger/bin* do premennej prostredia *PATH*.

[1] https://www.activestate.com/activeperl/downloads
[2] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/
[3] http://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/data/slovak-par-linux-3.2-utf8.bin.gz

5. Nastaviť premennú prostredia *TREETAGGER_HOME* do koreňového priečinka programu TreeTagger.

Závislosti projektu sú manažované cez Maven. K projektu je priložený súbor *pom.xml*, pomocou ktorého by mal Maven automaticky prevziať závislosti z internetových repozitárov. Výnimkou je dvojica lokálnych repozitárov *MateParser* a *anna*, ku ktorým je potrebné upraviť cestu v súbore *pom.xml*. Tieto súbory sa takisto nachádzajú v priečinku *kniznice* na sprievodnom médiu.

B.1 Systémové požiadavky

Vzhľadom na použitie jazyka Java je projekt multiplatformový. Minimálne systémové požiadavky sú:

- Java SDK/JRE 1.8
- 2GB RAM

Pre mierne urýchlenie syntaktickej analýzy je možné navýšiť množstvo pamäte príkazom virtuálnemu stroju Javy `-Xmx16G`, ktorý navýši množstvo využiteľnej pamäte na 16GB RAM. Treba počítať s tým, že načítanie modelov do pamäte v prípade slovenského parsera trvá na štvorjadrovom procesore s frekvenciou 3 GHz okolo 90 sekúnd. Načítanie anglických modelov trvá okolo 45 sekúnd. Spracovanie celého korpusu slovenských dokumentov zaberie približne 20 minút. Spracovanie súboru *full_papers.xml* z korpusu BioScope zaberie odhadom 45 minút.

C Metodika značkovania korpusu

Metodika je určená pre anotátorov korpusu dokumentov. Slúži na opis procesu a jednotlivých značiek, aby bola medzi výstupmi rozličných anotátorov dosiahnutá čo najvyššia konzistencia. Touto metodikou sa riadili súčasní autori korpusu, ale je určená aj pre prípad, keď bude korpus rozširovaný inými autormi.

C.1 Druhy negácie

- Genitívna predložka.
  - Negátorom sú neohybné slová *bez, okrem, mimo, namiesto*. Označenie negátora je *gen*.
  - Rozsahom negácie je typicky jedno plnovýznamové slovo v genitíve, typicky podstatné meno, zámeno alebo číslovka. V prípade viacnásobného vetného členu sú označené všetky slová tvoriace viacnásobný vetný člen.
  - Návšteva mala prísť **bez** *Petra* a *Pavla*, preto sme čakali všetkých **okrem** *nich*.
  - Ľudia kráčali **bez** akéhokoľvek *strachu*.
  - Všetky tieto problémy sú **mimo** môjho *chápania*.
- Prísudok (predikát).
  - Negátorom je vždy sloveso, ktoré má typicky záporný prefix. Jedinou výnimkou je sloveso *byť*, ak pred ním stojí častica *nie*. V takom prípade sa označí *je* ako negátor. Označenie negátora je *pre*.
  - Rozsahom negácie je celá veta. Ak ide o priraďovacie súvetie, rozsahom je len jednoduchá veta. Ak ide o podraďovacie súvetie a negátor je v nadradenej vete, je potrebné sémantické pochopenie vety a rozhodnutie, či do rozsahu negácie patrí aj podradená veta. Pomôckou je, že spojky *ale, takže, pretože, nech, keby, keďže, keď, aby, pokiaľ, nakoľko, tak* typicky rozsah negácie neprenášajú. Naopak,

takmer vždy sa rozsah prenáša do prístavku a spojkami *že, ktorý, aký, či*. Spojka sa do rozsahu negácie nezahŕňa. Ak je negátorom sloveso *byť* spojené s časticou *nie*, v tomto prípade sa častica do rozsahu negácie nezahŕňa.

- **Nepamätám**, *že by si mi bol o tom rozprával*.
- Tri krát tri je deväť, a *nikto to* **nemôže** *poprieť*.
- *Mačky momentálne* **nežerú** *myši*.
- Popevok, ktorým *si Ľuboš nikdy* **nespríjemňoval** *cesty*, odrazu znel dolinou.

- Atribút.
  - Negátorom je prídavné meno so záporným prefixom. Označenie negátora je *atr*.
  - Rozsahom negácie je podstatné meno alebo zámeno v zhode v páde, ku ktorému dané prídavné meno vystupuje ako rozvíjací vetný člen.
  - Vstupný súbor má **nevhodný** *formát*.
  - Zvolili ste **nevhodný** a **neodporúčaný** *postup*.

- Odčlenený atribút subjektu.
  - Negátorom je prídavné meno so záporným prefixom. Označenie negátora je *sub*.
  - Rozsahom negácie je predmet, ku ktorému dané prídavné meno vystupuje ako rozvíjací vetný člen.
  - *Stav* potrubia je na prevádzku **nevyhovujúci**.
  - Takýto *algoritmus* je na vyriešenie problému **nevhodný**.

- Nie.
  - Negátorom je slovo *nie*. Označenie negátora je *nie*.
  - V prípade, že ide o súčasť predikátu tvoreného slovesom *byť*, rozsahom negácie je toto sloveso. Vtedy treba označiť sloveso *byť* za nový negátor a pristúpiť ku vyznačeniu predikátovej negácie. Toto riešenie bolo zvolené z toho dôvodu, že kombinácia *nie + byť* je jediným prípadom v slovenskom jazyku, kedy dochádza ku vytvoreniu viacslovného negátora.
  - V prípade, že ide o členskú negáciu, rozsahom negácie je najbližšie podstatné meno alebo zámeno.
  - Pero **nie** *je* nástroj drevený.

  - Chyby vznikali **nie** z *lenivosti*.

- Podstatné meno.
  - Negátorom je podstatné meno so záporným prefixom. Označenie negátora je *sbs* zo slova *substantívum*.
  - Rozsah negácie sa nevyznačuje.
  - Došlo ku **nezhode** názorov.

- Príslovka.
  - Negátorom je príslovka so záporným prefixom. Označenie negáora je *adv* zo slova *adverbium*.
  - Rozsah negácie sa nevyznačuje.
  - Dopadlo to **nedobre**.

C.2 DTD

Nasledovný úryvok kódu predstavuje definíciu platného formátu XML prostredníctvom definície typu dokumentu.

```
<!ELEMENT document (sentence+)>
<!ELEMENT sentence (word+)>
<!ATTLIST sentence text CDATA #REQUIRED>
<!ELEMENT word (#PCDATA)>
<!ATTLIST word id CDATA #REQUIRED>
<!ATTLIST word negator (gen|pre|atr|sub|nie|sbs|adv) #IMPLIED>
<!ATTLIST word scope CDATA #IMPLIED>
<!ATTLIST word lemma CDATA #IMPLIED>
<!ATTLIST word pos CDATA #IMPLIED>
```

D Plán diplomového projektu I

Táto príloha obsahuje rozpis priebežnej práce na diplomovom projekte počas celého letného semestra 2015/2016. Uvedená je náplň práce pre každý týždeň semestra a krátke zhodnotenie toho, do akej miery sa plán podarilo naplniť.

Týždeň	Náplň práce
1.	Oboznámenie sa s náplňou predmetu Diplomový projekt 1, dohodnutie stretnutí.
2.	Oboznámenie sa s témou práce, navrhnutie plánu práce.
3.	Bližšia špecifikácia témy práce.
4.	Získanie existujúceho riešenia.
5.	Analýza závislostí existujúceho riešenia.
6.	Analýza prípravy dokumentov, predspracovania a iných riešení.
7.	Konzultácia v JÚĽŠ na tému syntaktickej analýzy a korpusov.
8.	Analýza korpusov, návrh vlastných vylepšení.
9.	Analýza extrakcie kľúčových slov.
10.	Analýza procesu detekcie negácie.
11.	Základný návrh a opis vlastného riešenia.
12.	Dokončenie dokumentu.

Tabuľka D.1: Prehľad práce v letnom semestri 2015-2016.

Väčšinu plánu sa podarilo splniť. Analýza predspracovania bola vypracovaná takmer v plnom rozsahu, pričom cieľ porovnať existujúce externé riešenia bol splnený. Bol vytvorený hrubý návrh a špecifikácia požiadaviek na výsledný systém. Do hĺbky boli zanalyzované najmä kroky prípravy dokumentov a extrakcie kľúčových slov.

Do detailu boli zanalyzované obidve predošlé generácie riešenia, čo bol ďalší hlavný cieľ tohto semestra. Tento semester sa nepodarilo dokončiť analýzu procesu negácie ani začať s implementáciou prototypu.

E Plán diplomového projektu II

Táto príloha obsahuje plán priebežnej práce na diplomovom projekte počas celého zimného semestra 2016/2017 tak, ako práca prebiehala. Pôvodný plán z letného semestra nebol dodržaný, pretože došlo ku posunu v oblasti témy výskumu. Za týmto posunom stálo objavenie dostupného závislostného analyzátora pre slovenčinu, ktorý bol vydaný v máji 2016, a bolo usúdené, že môže výrazným spôsobom posunúť vopred možnosti detekcie rozsahu negácie.

1.	Nadviazanie na závery analýzy predošlého riešenia, dohodnutie stretnutí.
2.	Analýza možností extrakcie fráz, pokusy so Stanford CoreNLP.
3.	Analýza možností závislostnej analýzy v slovenskom jazyku.
4.	Formulovanie nových cieľov výskumu.
5.	Úprava závislostného analyzátora, integrácia do projektu.
6.	Implementácia prototypu - predspracovanie.
7.	Implementácia, analýza výstupov závislostného analyzátora, skúmanie vzťahov.
8.	Experiment s detekciou rozsahu negácie pre slovenský jazyk.
9.	Formulácia závislostí pri detekcii rozsahu negácie v slovenčine.
10.	Návrh detekcie negátorov pre slovenčinu a angličtinu.
11.	Návrh detekcie rozsahu negácie pre oba jazyky cez závislosti.
12.	Dokončenie dokumentu.

Tabuľka E.1: Prehľad práce v zimnom semestri 2016-2017.

Pôvodným cieľom bolo začať s prípravou anotovaných dát na overenie predošlej generácie riešenia spolu s refaktoringom. V dôsledku objavenia nových možností výskumu sa začalo s tvorbou nového prototypu postaveného na moderných nástrojoch.

Podarilo sa navrhnúť úplný proces detekcie negácie pre slovenský aj anglický jazyk. Ďalej sa podarilo implementovať predspracovanie do takej úrovne, že je možné experimentovať s prechodmi cez závislostný strom a bolo možné zatiaľ síce len čiastočne overiť, že navrhnutá metóda pre slovenský jazyk funguje.

F Plán diplomovej práce

Táto príloha obsahuje plán práce na diplomovom projekte počas letného semestra 2016/2017. Uvedená je predpokladaná náplň práce pre každý týždeň semestra.

Týždeň	Náplň práce
1.	Finalizácia detekcie negácie v prototype, dohodnutie stretnutí.
2. - 4.	Tvorba dátovej sady.
4. - 8.	Evaluácia.
8. - 12.	Finalizácia práce, dokumentu, vynesenie záverov.

Tabuľka F.1: Plán práce v letnom semestri 2016-2017.

Cieľ vynesený na posledný semester, a to evaluovať navrhnuté metódy, sa podarilo splniť. Práca prináša podrobnú evaluáciu navrhnutých metód pre slovenský jazyk na vytvorenej dátovej sade. Experimenty pre anglický jazyk boli vykonané na korpuse BioScope.

Do termínu odovzdania práce sa nepodarilo dokončiť vedecký článok, ktorý je však v príprave a venuje sa prezentácii metód a výsledkom evaluácie. Zároveň sa podarilo pripraviť modul detekcie negácie len ako samostatne stojacu jednotku, pričom nadstavba v podobe ukážkové grafického rozhrania zostala v procese tvorby.

G Obsah priloženého média

Priložené médium má nasledovnú štruktúru:

```
\
\dokument
\kniznice
\konfiguracia
\korpus
\korpus\clanky
\korpus\xml
\projekt
```

- Priečinok **dokument** obsahuje elektronickú formu tohto dokumentu.
- Priečinok **kniznice** obsahuje zoznam lokálnych knižníc, ku ktorým je potrebné nastaviť cestu pre Maven. Zároveň obsahuje zoznam modulov, ku ktorým treba nastaviť cestu v konfiguračnom súbore.
- Priečinok **konfiguracia** obsahuje konfiguračný súbor pre navrhnutý modul, odkaz na aktuálny git repozitár.
- Priečinok **korpus** obsahuje v podpriečinku *clanky* zdrojové formáty článkov a v podpriečinku *xml* ich podobu s vyznačenými negáciami. V hlavnom priečinku sa nachádza dtd súbor pre evaluačné súbory.
- Priečinok **projekt** obsahuje projekt s implementáciou v jazyku Java.