УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ“ – ШТИП

ФАКУЛТЕТ ЗА ИНФОРМАТИКА

Катедра по информациски технологии

Штип

МАРИНА ИВАНОВА

ГРАДЕЊЕ НА МАКЕДОНСКИ ДИГИТАЛЕН ЈАЗИЧЕН КОРПУС

- МАГИСТЕРСКИ ТРУД -

Штип, 2017 г.

Апстракт

Денешните информатички технологии кои се одликуваат со голем број на карактеристики и функционалности, овозможуваат развој на веб сервиси, апликации, системи и сл. кои може да се приспособуваат во сите сфери на денешницата. Она што е заедничко, всушност е дека најголемиот број од овие системи, сервиси и апликации генерално се состојат од форми за внесување или презентирање на резултат од претходно пребарување. Тоа што можеби е занемарено во однос на податоците во облик на текст т.е. говорен јазик е начинот на кој тој е презентиран на крајните корисници. Многу мал број од светските јазици во моментов ги уживаат придобивките на современите јазични технологии како што се препознавање на говорот и машински превод. Малку поголем број (помалку од 100) успеале да ги соберат основните средства кои се потребни како основа за напреднување на технологиите за крајниот корисник: монолингвални и билингвални корпуси, машинско-читливи речници, синоними, морфолошки анализатори, парсери, итн. Остатокот, повеќе од 98% од светските живи јазици ги немаат повеќето алатки, и затоа спаѓаат во групата на јазици дефинирани како недоволно ресурсни.

Проучувањето на јазик кој вклучува поими и примери од секојдневната јазична употреба се нарекува корпус – компјутеризирани бази на податоци создадени за јазично истражување. Корпусот може да содржи текст од еден јазик (еднојазичен корпус) или текст од повеќе јазици (повеќејазичен корпус). Во лингвистиката, корпус (текст корпус) е голем и структуриран збир на текстови кои во денешно време се чуваат и обработуваат електронски. Корпусите се користат за да се направи статистичка анализа, тестирање, проверка или потврдување на јазичните правила во рамките на одредена јазична територија.

Статистичките техники се клучен дел од најмодерните системи за обработка на природниот јазик. За жал, таквите техники бараат постоење на големи тела на текст, а во последно време развивањето на корпуси се покажа доста скапо. Како резултат на тоа, значителни корпуси постојат пред сè за

јазици како англискиот, францускиот, германскиот итн., каде што постои потребата од NLP (Natural Language Processing) алатки.

Во контекст на ова, во овој труд ќе се запознаеме со основните техники и начини кои се користат за креирање на дигитален јазичен корпус. Ќе проучиме кои се технологиите за градење на јазичниот корпус, какви сè алатки постојат за обработка на природните јазици и за вршење на различни анализи и статистики преку примена на македонскиот јазик.

Клучни зборови: корпус, алатки, техники, обработка на природни јазици, препознавање на говор, машински превод.

Abstract

Information technologies that are developed are characterized by a number of features and functionalities that enable the development of Web services, applications, systems and so on. The common thing of these systems, services and applications generally consist input forms or presenting the results of some query. The thing that is neglected is the way of representing the spoken language in some form of text to the end users. Very few of the world's languages currently enjoy the benefits of modern language technologies such as speech recognition and machine translation. Slightly more (less than 100) have managed to collect basic funds needed as a basis for advancing technology for the end user: monolingual and bilingual corpora, machine-readable dictionaries, thesauri, morphological analyzers, parsers, etc… The rest, more than 98% of the world's living languages don't have many tools and therefore belong to the languages that are defined as insufficient resource.

Studying a language that includes terms and examples of everyday language use is called corpus- computerized databases created for linguistic research. Corpus may contain text from one language (monolingual corpus) or text from multiple languages (multilingual corpus). In linguistics, a corpus is a large and structured set of texts that nowadays is stored and processes electronically. The corpus is used to make a statistical analysis, testing, checking or validating linguistic rules within a particular linguistic territory.

The statistical techniques are a key part of the most modern systems for natural language processing. Unfortunately, such techniques require the existence of large bodies of text, and recently developing corpus and that looks very expensive. There is corpus for languages like English, French, German, etc..., where there is a need for Natural Language Processing tools.

In this context, this paper will introduce the basic techniques and methods used to create a digital language corpus. We will study about the technologies for building corpus languages, tools that exists to process natural language and to perform various analyzes and statistics through the use of the Macedonian language.

Key words: corpus, tools, techniques, natural language processing, speech recognition, machine translation.

Содржина

Слики

1. Вовед

Со пристигнувањето на новата компјутеризирана т.е. електронска ера, развиени се многубројни компјутерски алатки кои се користат во сите полиња на човековите активности, вклучувајќи ја и обработката на јазикот. Со примена на ваквите алатки се овозможува испитување и проучување на јазичните манифестации, нивната својственост, правила, прописи, аномалии и сл., кои на лингвистите им овозможуваат нивно следење и евидентирање на можните промени. Ваквиот пристап на анализа на јазикот, доведе до воведување на нова јазична дисциплина која е именувана како **корпус лингвистика**. Иако нејзиното потекло е поврзано со втората половина на дваесеттиот век, корпус лингвистиката имаше маргинализиран статус, и оваа релативно млада дисциплина беше навистина призната во текот на 1980-тите и 1990-тите години на 20 век. Со тек на време, оваа дисциплина стана дел од повеќе лингвистички проучувања, што беше од големо значење за нејзиниот пораст, со што се овозможи корпус лингвистиката да стане една од водечките лингвистички методологии.

Корпусот е дефиниран во поглед на неговата форма и намена. Лингвистите секогаш го користат зборот корпус за да ја опишат колекцијата на природно настанатите форми на јазик, која е составена од неколку реченици, претставува збир на пишани текстови или снимани ленти кои се собрани за јазични студија (Hunston, 2002 година, стр. 2). Она што треба да се спомне е дека денес овој термин се користи повеќе за дефинирање на збирка на текстови (или делови од текст) кои се чуваат и се обработуваат електронски. Фокусот на корпус лингвистиката во главно е текст во форма на имплементиран јазик. Текстовите, всушност, претставуваат репрезентативен примерок на природно настанатиот јазик кои даваат сигурни јазични податоци.

Основната задача на корпус лингвистиката е да утврди како јазичните закони и модули се реализираат во конкретни јазични контексти. Корпусот се користи за посебни јазични цели. Тој се чува на таков начин што може да биде испитуван нелинеарно, како во квантитет така и во квалитет, каде што ваквиот пристап овозможува оддалечување од строгите граматички правила и

универзалната граматика. Во зависност од намената и употребата, корпусите се разликуваат по својата големина и структура. Компјутерските корпуси се кодирани и стандардизирани, оптимизирани за пребарување и анализа, и тие се чуваат во компјутерски бази на податоци.

Корпус лингвистиката претставува релативно ново поле во јазичното истражување и јазичната примена. Пред повеќе од половина век, корпус лингвистиката го започна својот пат како комплементарно поле во општата лингвистика, вештачката интелигенција, компјутерската лингвистика и применетата лингвистика, со директно вклучување на компјутерската технологија во областа на јазичното истражување и примена. Покрај ова, еволуираше како една од најперспективните емпириски области во јазичните студии кои придонесуваат за мултидимензионален раст на лингвистиката и јазичната технологија, генерално.

Изучувањето на јазикот и од емпириски и од интуитивен агол е еден од најстарите трендови во историјата на човековата еволуција. Во историјата, беше ставено акцент на истражување на природата на јазикот и да се разбере како лингвистичко знаење кое одигра важна улога во сознавањето и комуникацијата. Низ вековите, областа на лингвистиката се разви преку долг процес на когнитивни претпријатија за воспоставување концептуална поврзаност со другите гранки на човековото знаење.

На почетокот на новиот милениум, започна да се истражува за тоа како теориите за различни аспекти на човечкиот јазик се потврдени со докази за вистинската употреба на јазикот кој се манифестира на повеќе начини на јазичниот израз на обичните луѓе. Оваа нова насока на истражување на јазикот, додаде дополнителна димензија на дисциплината на традиционалната лингвистика. Сето ова е овозможено благодарение на воведувањето на компјутерската технологија која и помогна на лингвистиката да расте и да се развива со набавка на алатки и техники за да се акумулираат примери на вистинската употреба на јазикот, како и да ги анализираат овие бази на податоци во понови перспективи. Воведувањето на овој пристап има придонес на два основни начини кои се од областа на лингвистиката во целина:

- На лингвистите им се овозможи да проверат дали прастарите теории и претпоставки за употреба на јазикот се вредни за следење;
- Се обезбеди доволно материјал за директна употреба на лингвистички докази и информации во редовните работи и активности на лингвистиката и јазичната технологија.

Така, овој нов тренд на јазичното истражување и примена, претставува еликсир за заживување и опстанок на една прастара дисциплина, која многу години беше целосно занемарена.

Јасно е дека пронајдокот и напредокот на компјутерската технологија додава нова димензија во областа на лингвистиката.

Во последно време, како резултат на оваа иновација, компјутерската лингвистика се разви како важна област на вештачката интелигенција, која има за цел да гледа во јазикот како основен инструмент на човечката комуникација, директно поврзан со човечкото сознание.

Корпус лингвистиката, како важна област на компјутерската лингвистика, игра важна улога. Таа обезбедува големи количини на податоци на емпириският јазик, кои акумулираат, на систематски начин, од различни области на вистинска употреба на јазикот по некои статистички методи и техники на земање примероци од податоци. Исто така, дава некои софистицирани уреди за да ги анализираат овие корпуси за да се извлечат лингвистички податоци, примери и информации кои се неопходни во применетата лингвистика, компјутерска лингвистика и вештачката интелигенција за разбирање на човечкиот јазик во подобар начин, како и за примена на овие податоци и информации во различни области на човековото знаење.

Секогаш постои силна когнитивна и јазична мотивација да се предвиди начинот на кој комуницираме преку јазикот низ времето и просторот. Исто така постои и техничка мотивација да се изгради интелигентен компјутерски систем, кој ќе биде во можност да направи ефикасна јазична интеракција со луѓе. Со овие мотиви, компјутерските научници со лингвистите, заедно се приклучија да

се развијат системи како што се машинското преведување, екстракцијата на податоци, јазичното разбирање и генерација, разбирањето на говорот и генерирање, компјутерски потпомогнат јазик итн., кои придонесуваат за корист и напредок на целото човештво. Сепак, за дизајнирање и развој на таков систем, треба да се разберат емпириските природни јазици заедно со сите свои редовни и ретки обележја. Затоа јазичниот корпус стана неопходен, бидејќи тие имаат добар потенцијал да ги изложат повеќето карактеристики на природниот јазик кој се манифестира во голема колекција на емпириски податоци.

На почетокот, научниците низ целиот свет биле ангажирани за компјутеризирање на информациите од различни видови, од основните цели на компјутерската лингвистика, за да ги карактеризираат, колку што е можно, карактеристиките на природниот јазик во рамките на компјутерската архитектура. Стана неопходно да се спроведе истрага во јазичниот корпус за да се има корист од информациите и сознанијата изнесени во лингвистичката анализа на јазичните бази на податоците складирани во корпуси. На пример, ако сакаме да се направи толкување на едноставна реченица на еден јазик од страна на компјутерот, треба претходно известување на лингвистичката анализа на таквите реченици, која се врши од страна на експерти за зајакнување на системот. Така, описот и анализата на јазичните особини кои се зачувани во корпусот, стануваат значајни инпути и во компјутерската лингвистика и применетата лингвистика. Всушност, информациите добиени од корпусите, не придонесуваат сами за компјутерската лингвистика. Тоа подеднакво обезбедува вредни информации за опис и разбирање на јазикот, кој е важен дел од описот и изучувањето на јазикот.

1.1 Што е корпус?

Терминот корпус произлегува од латинскиот збор корпус кој значи „тело“. Во доменот на современата корпус лингвистика, терминот „корпус“ се однесува на „голема колекција на јазични податоци, или пишани текстови или транскрипција на снимен говор, кој може да се користи како почетна точка на јазичниот опис, или како начин на проверка на хипотези за јазикот“ (Crystal 1995). Така, тоа се однесува на голема колекција на примероци на пишан и

говорен текст, достапни во машински - читлив облик, акумулирани на научен начин да претставуваат одреден вид или употреба на јазикот.

Според научниците, корпусот е збирка на јазични елементи кои се избрани и наредени според некои експлицитни јазични критериуми, утврдени од страна на корисниците со цел да се користи како примерок на јазикот. Тој е методички дизајниран да содржи милиони зборови составени од различни видови на текст во многу демографски разлики, за да ја опфати разновидноста на природниот јазик преку својата повеќеслојна употреба. McEnery и Wilson (1996: 215) го класифицираат корпусот во пофина шема на класификација која се карактеризира со нејзините својствени карактеристики:

- Корпусот се однесува на кој било дел од текстот;
- Најчесто, тоа се однесува во делот на машински – читлив текст;
- Тоа се однесува на колекција на примери на машински – читливи текстови за го претставуваат максимално јазикот или типовите на јазикот.

Во принцип, корпусот е дизајниран за прецизно проучување на јазичните карактеристики и феномените во јазикот. Затоа, се посочува дека систематски составен корпус кој е мал по големина, треба да се придржува на следните критериуми (Dash 2005: 12):

- Корпусот треба подеднакво да ги претставува и двете, заедничките и посебните јазични особини на јазикот од којшто е дизајниран и развиен. Идејата на текстот којшто е застапен во корпусот индиректно се однесува на неговите компоненти (зборови, фрази, реченици итн.) кои се вклучени во него. Меѓутоа, во пракса, од вкупниот број на зборови кои се наоѓаат во корпусот може да се утврди неговата големина, но може да не се успее во придржувањето кон начелото на соодветна застапеност на текстот. Затоа е подобро да се задржат полињата отворени за корпусот, како и да се задржи неограничен број на зборови во корист на јазикот и на корисниците.
- Корпусот треба да биде голем и широк за да опфати текстови од различни дисциплини. Со други зборови, различни видови на употреба

на јазикот се манифестираат во различни дисциплини и области, и затоа треба да има пропорционална застапеност во него. На пример, примери на текст од областа на природните науки, треба да бидат со иста тежина како и оние од естетика, литература, медиуми, инженерство и општествени науки. Така, со рамноправната застапеност на текстовите добиени од сите дисциплини и области на употребата на јазикот ќе се осигури неговата стабилност.

- Корпусот треба да претставува вистинска реплика на физичкиот текст кој е достапен во печатена форма. Притоа, треба да се зачуваат различните форми на зборови, правописните варијации, интерпункциски знаци, како и други различни правописни симболи кои се користат во изворните текстови. Инаку, доколку вистинската слика на еден јазик или на повеќе јазици се изобличи, корпусот ќе ја изгуби својата вредност и автентичност.
- Корпусот треба да биде достапен во електронска форма за полесен пристап од страна на крајните корисници, со цел да им се овозможи на обичните корисници, како и на истражувачите на јазикот да се користи базата на податоци во повеќе задачи поврзани со јазичниот опис и анализа, статистичка анализа, обработка на јазикот, превод итн.

Основната задача на корпус дизајнерите е да соберат голем дел на примероци на репрезентативен текст кои покриваат широки сорти на јазикот кој се користи во разни домени од својата редовна јазична интеракција.

1.2 Главни карактеристики на корпусот

За корпусот се претпоставува дека поседува одредени карактеристични функции. Тоа подразбира дека корпус кој поседува еден или повеќе вредности што не се основни во однос на карактеристичните функции на генералниот корпус, тој може да биде идентификуван како „специјален корпус“ кој ќе има некои отстапувања од општите рамки во однос на општиот корпус. Во продолжение се наведени некои главни карактеристики со кои се одликува корпусот.

Квантитет

Најчестото прашање кое се поставува од страна на нови членови во корпус лингвистиката е: Колку голем корпус треба да се генерира? На ова прашање всушност и не може да се даде конкретен одговор. Сепак, терминот „количество“ значи дека воопшто корпусот содржи голема количина на податоци или јазик во усна или во писмена форма. Всушност, големината на корпусот е практично збирот на големината на неговите компоненти кои се користат за да го сочинуваат неговото тело. Целата поента при составување на корпусот е да се соберат јазични бази на податоци во големи количини.

Квалитет

Почетната вредност при дефинирање на квалитетот на корпусот директно се однесува на неговата автентичност. Тоа значи дека јазичните бази на податоци треба да содржат податоци од нормално зборување до пишани текстови. Основната улога на собирач на податоци е ограничена во рамките на дефиниран закон за стекнување на податоци од нормални текстови за производство на корпусот.

Застапеност

Еден генерален корпус, во принцип треба да вклучува примероци од широк спектар на текстови со цел да се постигне соодветна застапеност на јазикот. Покрај тоа, корпусот треба да биде балансиран, да содржи текст примероци од сите дисциплини за да се претставуваат максималниот број на лингвистички карактеристики во еден јазик. Базите на податоци кои се чуваат во еден корпус треба да бидат автентични во нивното претставување на изворниот текст, бидејќи иднината на лингвистичката анализа и системите за истражување, кои се базирани на бази на податоци, ќе треба да ги верификуваат и да вршат проверка на информациите добиени од корпус што претставува дадениот јазик.

Едноставност

Оваа функција означува дека корпусот треба да содржи текст материјали во едноставен и обичен текст формат, за корисниците на корпусот да имаат

лесен пристап до обичните текстови, без притоа да имаат некакви проблеми во рамките на текстуалните примероци. Моментално постојат неколку корпуси во кои текст примероците се означени во SGML (i.e. Standard Generalized Mark-up Language, ISO 8879: 1986) формат, каде што внимателно се користат за да не се наметнува дополнителен товар на информациите за текст примероците. Нормално, улогата на системот за обележување во однос на застапеноста на текстот, е да се означува во линеарно кодирање, некои од лингвистичките и не-лингвистичките карактеристики, кои во спротивно ќе бидат изгубени во обработката на корпусот. Системот на кодирање на текстот или коментар се смета дека е многу корисен, бидејќи неговото присуство го подобрува лесното пронаоѓање на лингвистичките информации од корпусот.

Еднаквост

Текст примероците кои се собрани во даден корпус треба да бидат со еднаква големина во однос на бројот на зборови кои се зачувани во секој примерок. Сепак, ова не е толку прифатливо, бидејќи тоа не може да донесе насекаде на единствен начин. Големината на примероците на текстот пропорционално ќе се разликуваат - во зависност од потребите на корисниците, како и од достапноста на текст материјалите.

Достапност

Јазичните податоци кои се чуваат во корпусот треба да бидат достапни и лесно употребливи од страна на крајните корисници. Затоа потребно е да се обрне внимание на техниките кои се користат за зачувување на податоците во електронски облик, во компјутер или во дигитална архива за корисници. Сегашната технологија, на корисниците им овозможува да генерираат корпус на персонален компјутер, и на тој начин да се задржи, при што секој ќе може лесно да пребарува информации и податоци кога и да е потребно.

Верификација

Текст примероците собрани од различни извори мора да бидат автентични и сигурни во застапеноста на јазикот. Без можност за емпириска верификација, важноста на корпусот се сведува на нула. Всушност, овој квалитет го прави корпусот да биде достапен за сите видови на емпириско

истражување од страна на корисниците за да се потврдат претходните ставови или да се побијат постојните забелешки. Затоа, оваа функција ја направи корпус лингвистиката многу понапредна во однос на генеративниот модел на јазични студија и истражување.

2. Употреба на корпус лингвистика

Корпус лингвистиката претставува метод за вршење на лингвистичка анализа. Може да се користи за истражување на многу видови на јазични прашања и како што почна да прикажува интересни, фундаментални и нови сознанија за јазикот, корпус лингвистиката стана една од најпознатите методи што се широко распространети во последниве неколку години.

Корпус лингвистиката, всушност врши анализа на природно настанатите јазици врз основа на употреба на компјутерските технологии. Најчесто, анализата се врши со помош на компјутер, односно со употреба на специјализиран софтвер, и ги зема предвид зачестеноста на некои феномени, кои понатаму се користат како основа за анализа.

2.1 Типови на корпора

Повеќето објавени материјали врз основа на употребата на корпуси, ја направи употребата на корпусот многу голема, односно настана генерален корпус на кој многу читатели веруваат дека ова е тип на корпус што може да им биде од корист. Всушност, постојат осум видови на корпуси – генерализиран, стручен, ученички, педагошки, историски, паралелен и споредлив. Кој тип на корпус ќе се одбере за да се користи - зависи од целта на корпусот. Четири вида од овие корпуси се најглавни, односно се најкорисни при применување на корпусот за различни цели.

Генерализиран корпус

Најширок тип на корпус е генерализираниот корпус. Ваквите типови на корпуси најчесто се многу големи, повеќе од 10 милиони зборови, и содржат голем број јазици, а ова води кон тоа дека може да биде малку генерализиран. Иако не постои корпус кој некогаш ќе ги претставува сите можни јазици, генерализираниот корпус на корисниците им дава една претстава за јазикот. British National Corpus (BNC) и American National Corpus (ANC) се примери на големи, генерализирани корпуси. Овие големи, генерализирани корпуси содржат пишани текстови како што се весници и списанија, дела на фикција и документаристика, како и пишување на научни списанија. Овие корпуси, исто

така, содржат и говорни записници како неформални разговори од владини постапки и бизнис состаноци.

Специјализиран корпус

Специјализираниот корпус содржи текстови од одреден тип и има за цел да биде претставник на јазик од ваков тип. Овие корпуси може да бидат и големи и мали и често се креирани за да можат да одговорат на многу конкретни прашања. Примери на специјализирани корпуси вклучуваат Michigan Corpus of Academic Spoken English (MICASE), кој содржи само говорен јазик за универзитетот, CHILDES Corpus (MacWhinney, 1992), кој содржи јазик кој се користи од страна на децата, MICUSP, Michigan Corpus of Upperlevel Student Papers, колекција од трудови од голем број на универзитетски дисциплини, и медицински корпус, кој содржи јазик кој се користи од страна на медицинските сестри и болничкиот персонал.

Корпус за учење

Еден ваков корпус претставува еден вид на специјализиран корпус кој содржи пишани и / или говорни записници од јазикот кој се користи од страна на студентите кои во моментов се стекнуваат со тој јазик. Корпусот за учење често е обележан и може да се испита, на пример, да се видат заеднички грешки кои студентите ги прават. Еден добро познат корпус за учење е International Corpus of Learner English (ICLE) (Granger, 2003), кој содржи есеи напишани од страна на ученици на англиски јазик со 14 различни мајчини јазици.

Педагошки корпус

Педагошкиот корпус е корпус кој содржи јазик кој се користи во училница за учење. Овие корпуси вклучуваат академски учебници, записници од интеракциите во училницата, или кој било друг пишан текст или говорен препис со кој учениците се сретнале во образовниот процес. Педагошките корпуси може да се користат за да се обезбеди дека студентите учат корисен јазик, или како саморефлективна алатка за професионален развој на наставниците.

2.2 Клучни зборови во корпус лингвистиката

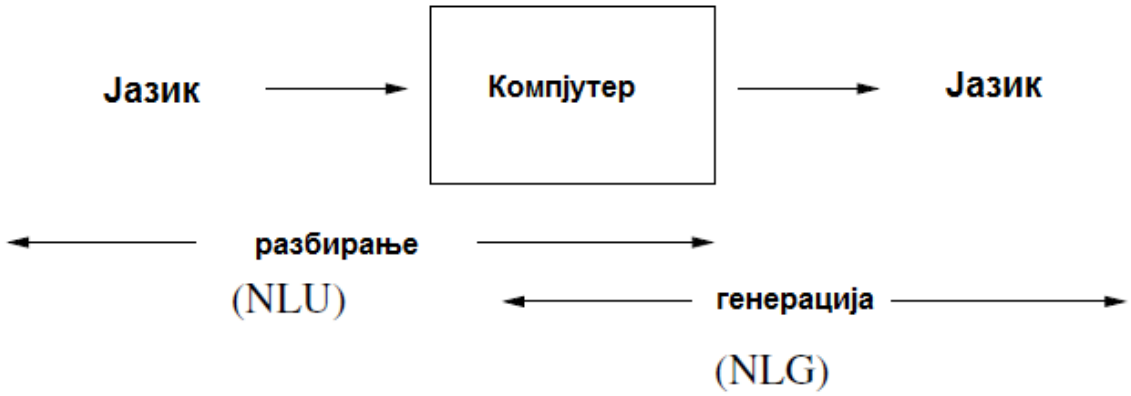
Литературата на корпус лингвистиката има специјална техничка терминологија, која ги вклучува следниве термини:

- **Токен**: Пред каква било обработка која може да се направи на внесениот текст, тој треба да се подели во јазични единици како што се зборови, интерпункциски знаци, бројки или алфанумерички знаци. Овие единици се познати како токени;
- **Реченица**: Подредена низа на токени;
- **Токенизација**: Процесот на разделување на реченицата во нејзините составни токени;
- **Корпус**: Тело од текст, обично содржи голем број на реченици;
- **Дел од говор таг (POS = Part-of-Speech):** Еден збор може да се класифицира во една или повеќе лексички или дел од говор категории како што се именки, глаголи, придавки и членови. POS тагот е симбол кој ја претставува лексичката категорија – NN (именка), VB (глагол), JJ (придавка), AT (член);
- **Дрво на парсирање**: Дрвото се дефинира за одредена реченица, и ја претставува синтаксичката структура на реченицата дефинирана преку формална граматика;

3. Обработка на природните јазици

Обработката на природните јазици (Natural Language Processing) е доста актуелна област која што и не е така едноставна за изучување и има огромно влијание во нашиот секојдневен живот. Таа се базира на сознанија од компјутерските науки, од лингвистиката, но најмногу од веројатност и статистика. Со зголемување на трендот, многу апликации и технологии ги применуваат основните идеи, методологии и процеси од оваа област.

Обработката на природните јазици се однесува на употребата на компјутерите во обработката на човековиот или природните јазици. Она што најчесто се нарекува како разбирање на природниот јазик (Natural Language Understanding), е доделување на одреден текст како влез на компјутер, каде ваквиот текст се обработува за да се опфати нешто корисно со неговата обработка. Од друга страна, имаме нешто што често се нарекува генерација на природниот јазик (Natural Language Generation), односно, онаму каде што компјутерот во некаква смисла произведува природен јазик во комуникацијата со човекот или корисникот.



Слика бр. 2 Користење на природниот јазик

Слика бр. 1. Користење на природниот јазик со компјутер

Една од најстарите апликации и проблем што е од големо значење при обработката на природните јазици е машинското преведување. Ова е проблем на мапирање на реченици од еден јазик во реченициte на друг јазик и претставува многу тешка задача. Друга клучна апликација е сумирање на текст. Проблемот во овој случај е да се земе еден документ или, потенцијално група од неколку документи и да се обиде да се направат како еден краток преглед, што во извесна смисла ќе ги зачувува главните информации во тие документи. Друга апликација е она што се нарекува дијалог системи. Тоа се системи каде што човекот може всушност да комуницира со компјутерот за да се постигне некоја задача.

Како дополнение на овие апликации, ќе се разгледаат некои основни проблеми при обработката на природните јазици кои зависат многу од овие апликации. Еден од нив се нарекува проблем на означување, каде како влез имаме дадена секвенца (пр. низа на букви) и на излез да има обележана низа каде на секоја буква во влез сега има поврзан таг. Друг проблем кој се јавува се нарекува part-of-speech тагирање. Проблемот во овој случај е да се земат реченици како влез и да се изврши тагирање на секој збор од реченициte на влез со неговиот дел од говорот. Ова претставува еден од основните проблеми при обработката на природниот јазик. Доколку обработката се изврши со голема точност, тоа ќе биде од голема корист за голем број на апликации.

Меѓу другото, друг основен проблем кој се јавува кај проблемот на означување се нарекува препознавање на ентитет,во кој под ентитет се подразбира компанија, локација, лица и сл. Овој проблем може да се формулира како проблем на означување, во кој секој збор на влез се означува, дури и во случај да не припаѓа на ниту еден именуван ентитет. Проблемот на означување е всушност проблем на мапирање на низа на елементи на влез, најчесто зборови со дадена низа, каде што секој збор во секвенцата има поврзан таг.

Останати потешкотии кои се јавуваат при обработка на природниот јазик е проблемот на парсирање. Проблемот овде е повторно да се земат одредени реченици на влез и да се мапираат на излез, ова вообичаено се

нарекува дрво за парсирање/анализирање. Ова дрво всушност ни дава хиерархиска декомпозиција на една реченица што одговара на нејзината граматичка структура. Ова е уште еден основен проблем што се јавува при обработка на некој природен јазик. Она што е од големо значење, е преку овие проблеми голем број на апликации да се здобијат со основниот чекор во разбирањето на природниот јазик. Еден од клучните проблеми во парсирањето е суштинското појаснување, односно, изборот помеѓу различни синтаксички структури кои одговараат на различни толкувања. Ова е уште еден степен на двосмисленоста во јазикот, она што може да се појави на семантичко ниво.

Јазичните модели првично биле развиени за проблемот на препознавање на говор, во кој тие се уште ја имаат централната улога во современите системи за препознавање на говор. Тие исто така широко се користат и во други апликации за обработка на природниот јазик. Еден јазичен модел се дефинира на следниов начин. Најпрво се дефинира V, тоа претставува збир од сите зборови во даден јазик или даден корпус. На пример, при изградба на јазичен модел за нашиот јазик, V би можело да изгледа како следниот пример:

V = { Здраво, како, се, си, Марија, викаш...}

Во пракса V може да биде доста големо. Тоа може да содржи неколку илјади, или десетици илјади зборови. Претпоставуваме дека V е ограничен сет од зборови. Една реченица на јазикот е низа од зборови во која...

$$x_1 x_2 \dots x_n$$

каде што n е цел број, така што n ≥ 1, при што $x_i \in \mathcal{V}$ за $i \in \{1 \dots (n-1)\}$, каде се претпоставува дека Xn е посебен симбол. Кај јазичните модели секоја реченица на крај, најчесто завршува со симболот STOP (се претпоставува дека STOP не е член на V). Во продолжение се претставени примери со користење на STOP симболот:

Здраво, како си STOP

Здраво Марија STOP

Како се викаш STOP

Здраво STOP

Здраво Здраво Здраво STOP

STOP

Ќе дефинираме $\mathcal{V}^\dagger$ да биде збир на сите реченици од V, каде $\mathcal{V}^\dagger$ ќе биде бесконечен сет од реченици, бидејќи реченициите овде можат да бидат од кој било должина. Во ваков случај се добива следната дефиниција:

Дефиниција 1 (Јазичен модел): Еден јазичен модел се состои од ограничен сет V, и функција p (X_1, X_2,..... X_n) така што:

1. За секое $\langle x_1 \dots x_n \rangle \in \mathcal{V}^\dagger$, $p(x_1, x_2, \dots x_n) \geq 0$
2. Покрај тоа,

$$\sum_{\langle x_1 \dots x_n \rangle \in \mathcal{V}^\dagger} p(x_1, x_2, \dots x_n) = 1$$

Оттука (X_1, X_2,..... X_n) претставува распределба на веројатноста во реченициите во $\mathcal{V}^\dagger$.

3.1 Парсирање

Парсирањето е една од најважните технологии што се користат во обработката на природните јазици. Проблемот на парсирање е во суштина да се поврзе некој вид на структура, најчесто тоа е таканаречена дрво структура со една реченица. Ова, најчесто се прави со помош на граматика, многу често во контекст на слободна граматика. Може да се случи да има само една таква структура на стебло, во зависност од реченицата и граматиката. Исто така, може да има многу реченици од кои би сакале да ја избереме онаа која има најголема веројатност или е повеќе соодветна. Или исто така, може да има случај во кој нема да има ниту една реченица која нема да може да биде успешно разложена од страна на таа граматика.

Една работа која треба да се има предвид е дека сите граматики се декларативни. Ова важи за сите граматики кои вклучуваат контекст слободна граматика. Значи, тоа значи дека ќе може да се користи граматиката за да се опише реченицата, но не може автоматски да се користат методи кои ја конвертираат реченицата во дрво за парсирање. Меѓутоа, граматиките не се доволни за да се одреди како ќе се гради дрвото за парсирање.

3.2 Апликации за парсирање

Во продолжение ќе разгледаме какви видови апликации за парсирање постојат. Така, првата се нарекува граматичка проверка (grammar checking). Секогаш кога во некој едитор ќе се внесе некаква реченица и ако таа реченица не е формулирана граматички правилно, едиторот како резултат ќе врати дека постојат некои неправилности кои понатаму ќе може да се поправат. Друга форма на парсирање е модел на одговарање на прашања. На овој начин, доколку се постави некакво прашање, парсерот ќе треба да препознае некој запис кој се наоѓа во базата и кој ќе биде соодветен одговор за поставеното прашање. Други форми на парсирање се машинското преведување како и екстракцијата на информации, која има за цел да препознае различни фрази и да препознае како се поврзани едни со други и да види од кој тип се. Постојат многу други апликации, на пример за производство на говорот, за разбирање на говорот, за толкување на реченици итн.

3.3 Контекст слободна граматика (Context free grammar)

Контекст слободна граматика (Context free grammar) претставува четворка од (N,Σ,R,S) каде што:

- N претставуваат нетерминални симболи;
- Σ се состои од терминални симболи (врши расчленување од N);
- R: правило $(A \rightarrow \beta)$, каде што β е стринг од $(\Sigma \cup N)^*$;
- S е почетен симбол од N.

Значи, N е збир на нетерминални симболи. Сигма претставува збир на терминални симболи. Се претпоставува дека сетот од терминални симболи се разликува или расчленува од множеството на нетерминални симболи. Исто така постои и збир на правила, каде што на левата страна имаме

нетерминален симбол A кој е дел од множеството на нетерминални симболи. На десната страна се наоѓа β при што претставува стринг каде што може да ги комбинира симболите од Σ и N. Може да има каков било број од овие симболи, од нула па се до некоја голема бројка. И на крај S е специфичен почетен симбол во N. Кога ќе се интерпретираат цели реченици, се случува S да биде симбол за реченица, но во целина не постои причина зошто контекстот и граматиката не можат да се користат за да се парсираат и некои други реченички конституенти.

```
["the", "child", "ate", "the", "cake", "with", "the", "fork"]

      S -> NP VP
      NP -> DT N | NP PP
      PP -> PRP NP
      VP -> V NP | VP PP
      DT -> 'a' | 'the'
      N -> 'child' | 'cake' | 'fork'
      PRP -> 'with' | 'to'
      V -> 'saw' | 'ate'
```

Слика бр. 2 Пример за контекст слободна граматика (Context free grammar)

Сликата погоре, ни претставува пример која гласи: „ the child ate the cake with the fork“. Граматиката која ја имаме во овој пример е контекст слободна граматика со осум нетерминални симболи. S се користи за реченица, NP, PP, и VP за не фрази, предлошки фрази, глаголски фрази соодветно, DT се однесува на одредници или на статија во овој пример, со N се означуваат глаголите и имаме дадено некои глаголи во минато време. Некои од правилата имаат опции. Така на пример, именската фраза може да биде или одредница проследена со именка, или рекурзивно може да се претвори во именска фраза за би-предлошка фраза. Она што е важно да се спомне, е дека фразите кои завршуваат на P како симболите NP, PP, и VP се смета дека се водечки елементи. Тоа значи дека една од нивните компоненти е поважна од другите и тоа не е изненадувачки за именката од именската фраза, за предлогот од предлошката фраза и за глаголот од глаголската фраза.

4. Проблеми при означување

Во повеќето NLP (Natural Language Processing) проблеми, би сакале да се моделираат парови од секвенци. Part-of-speech (POS) тагирањето е можеби еден од најстариот и најпознатиот пример за ваков тип на проблем. Кај POS тагирањето, главната цел е да се изгради модел каде на влез ќе има дадена реченица, пример:

The dog saw a cat

каде на излез ќе прикаже низа од ознаки од следниот тип

D N V D N

Во овој пример ознаката *D* важи за одредница, *N* за именка и *V за* глагол. Крајниот резултат прикажан на излез ја има истата должина како и влезната реченица, со тоа што секој збор од реченицата е означен со одреден таг на излез (во овој пример *D одговара за The, N за dog, V за saw, D за a и N за cat*). За означување на влез во моделот на означување се користи $x_1....x_n$, и тоа најчесто ќе се однесува како на една реченица. Во нашиот пример, должината на n= 5 и x_1= the, x_2= dog, x_3= saw, x_4= a, x_5= cat. Понатаму, се користи $y_1....y_n$ како формат кој го прикажува резултатот на излез од ваквиот модел. Во нашиот пример, тоа би се прикажало на следниов начин: y_1= D, y_2= N, y_3= V, y_4= D, y_5= N.

Овој тип на проблем, каде што единствена цел е да се изврши мапирање на дадена реченица $x_1....x_n$ во таг секвенца $y_1....y_n$, најчесто се нарекува како проблем на етикетирање или проблем со означување.

INPUT:
Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

OUTPUT:
Profits/N soared/V at/P Boeing/N Co./N ,/, easily/ADV topping/V forecasts/N on/P Wall/N Street/N ,/, as/P their/POSS CEO/N Alan/N Mulally/N announced/V first/ADJ quarter/N results/N ./.

KEY:

N = Noun
V = Verb
P = Preposition
Adv = Adverb
Adj = Adjective
. . .

Слика бр. 3 Part-of-speech (POS) пример за тагирање

Да претпоставиме дека имаме множество од примери, $(x^{(i)}, y^{(i)})$ i= 1…m, каде секое $x^{(i)}$ е реченица $x_1^{(i)}….x_{ni}^{(i)}$, и секое $y^{(i)}$ претставува таг секвенца $y_1^{(i)}…y_{ni}^{(i)}$ (се претпоставува дека i-тиот пример е со должина ni). Оттука, $X_j^{(i)}$ е всушност j-тиот збор во i-тото множество од реченици, и $y_j^{(i)}$ го претставува тагот на излез од тој збор. Она што е најважно, е да се научат функциите кои ги мапираат реченициte во таг секвенци од дадените примери.

4.1 POS тагирање и Named-Entity Recognition

POS тагирањето и Named-Entity Recognition се двата најзначајни примери кај проблемот на означување. Примерот од сликата погоре ни го илустрира Part-of-speech проблемот. На влез имаме дадена реченица. Излезот всушност е таг секвенца, во која секој збор од реченицата е поврзан со неговиот дел од говорот. Нашата цел е да се изгради модел кој ќе ги обновува POS таговите за реченици кои се со висока точност. POS тагирањето е еден од основните проблеми во NLP, и е многу корисен во многу апликации поврзани со природните јазици. Ќе претпоставиме дека имаме множество од примери за проблемот: всушност, имаме множество од реченици спарени со нивните правилни POS таг секвенци.

Еден од главните предизвици во POS тагирањето е двосмисленоста. Голем број на зборови во секој јазик може да бидат двосмислени или во зависност од контекстот на реченицата, во една реченица може да бидат прикажани како именка, а во друга како глагол.

Вториот предизвик е присуството на зборови кои се ретки, особено зборови кои воопшто и не се сретнати во сетот од примери. Дури и да има милиони зборови во базата, ќе има многу зборови кои ќе се сретнат во нови реченици и кои нема да можат да бидат препознаени. Од особена важност би било доколку се развијат методи кои ќе се справат ефикасно со зборови кои не можат да се видат во сетот од реченици.

Во обновувањето на POS таговите, корисно е да се мисли на два различни начина на информации. Прво, индивидуалните зборови имаат статистички параметри за својот дел од говорот: на пример, една четвртина може да биде именка или глагол, но поверојатно би било да биде именка. Второ, контекстот на реченицата има важен ефект за зборот. Некои секвенци на POS таговите се многу поверојатни од некои други. Ако ги земеме предвид POS триграмите, секвенцата D N V кај англискиот јазик е многу честа, со оглед на тоа редоследот D V N е многу помалку веројатен.

Понекогаш овие два извори на докази се во конфликт: пример, во реченицата

The trash can is hard to find

зборот *can* претставува именка, но исто така *can* може да биде и модален глагол, всушност најчесто и се јавува како модален глагол во англискиот јазик.

ВЛЕЗ: Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

ИЗЛЕЗ: Profits soared at [Company Boeing Co.], easily topping forecasts on [LocationWall Street], as their CEO [Person Alan Mulally] announced first quarter results.

Named-Entity Recognition пример. На влез е дадена една реченица. На излез е исто така реченица означена со именувани ентитети кои одговараат на компании, локации, луѓе итн.

Втор важен пример во проблемот со тагирање е проблемот со Named-Entity Recognition. За овој проблем влезот е повторно една реченица. На излез се прикажува реченица со обележани ентитети. Во конкретниов пример, постојат три различни типови на ентитет: PERSON, LOCATION и COMPANY. Излезот во овој пример го идентификува Боинг како компанија, Вол Стрит како локација и Ален Мелали како личност. За препознавање на ентитети како што се луѓе, места и организации има многу апликации, затоа Named-Entity Recognition е широко изучуван во NLP истражувањата.

На прв поглед проблемот со Named-Entity Recognition не се одразува како проблем на означување. Сепак, едноставно е да се мапираат вакви примери со проблемот на означување. Секој збор во реченицата е означен како да не е дел од некој ентитет (тагот NA се користи за оваа намена), доколку е на почетокот на одреден ентитет (тагот SC одговара на зборови кои се првиот збор во една компанија), или доколку означува како продолжување на одреден ентитет се користи тагот CC кој одговара за означување на зборови кои се дел од името но не се првите зборови.

Пример за Named-Entity Recognition

ВЛЕЗ: Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

ИЗЛЕЗ: Profits/NA soared/NA at/NA Boeing/SC Co./CC ,/NA easily/NA topping/NA forecasts/NA on/NA Wall/SL Street/CL ,/NA as/NA their/NA CEO/NA Alan/SP Mulally/CP announced/NA first/NA quarter/NA results/NA ./NA

KEY:
NA = No entity
SC = Start Company
CC = Continue Company
SL = Start Location
CL = Continue Location

4.2 Генеративни модели

Во овој дел ќе разгледаме една важна класа на модел за надгледувано учење: класата на генеративни модели. Поставувањето на надгледуваните проблеми во учењето е следново: Се задаваат одредени примери ($x^{(1)}$, $y^{(1)}$)…($x^{(m)}$, $y^{(m)}$), каде секој пример се состои од влез $x^{(i)}$ во комбинација со излез $y^{(i)}$. Се користи X кој се однесува на сет на можни влезови и Y кој се однесува на сет на можни излези. Наша задача е да знаеме дека функцијата *f : X -> Y* врши мапирање на секој влез x во дадена етикета *f(*x*).*

Многу проблеми во обработката на природните јазици претставуваат и надгледувани проблеми во учењето. На пример, во проблемот со означување секое $x^{(i)}$ претставува секвенца од зборови $x_1^{(i)}$….$x_{ni}^{(i)}$, и секое $y^{(i)}$ претставува секвенца од тагови $y_1^{(i)}$…$y_{ni}^{(i)}$. Затоа, X се однесува на сет од сите секвенци x_1….x_n и *Y* се однесува на сет од сите таг секвенци y_1….y_n. Наша цел е да научиме дека функцијата *f : X -> Y* врши мапирање на реченициите во таг секвенци.

Еден начин за да се дефинира функцијата *f(*x*)* е преку употреба на условен модел. Со ваквиот начин дефинираме модел кој ќе ја дефинира условната веројатност

$$p (y \mid x)$$

за кој било x, y пар. Параметрите на моделот се проценети врз основа на примерите. При доделување на нов тест пример x, излезот од овој модел е

$$f(x) = \arg\max_{y \in \mathcal{Y}} p(y|x)$$

Така, едноставно се зема најдобриот резултат од y како излез од моделот. Доколку овој модел *p (y I x)* прикажува резултати кои се блиску до точната условна распределба на етикети од дадените примери на влез, тогаш функцијата *f(*x*)* ќе биде блиску до оптимална.

Алтернативниот пристап, кој често се користи во машинското учење и обработката на природните јазици, е да се дефинира генеративен модел. Наместо директна проценка на условната распределба *p (y I x)*, во генеративниот модел наместо тоа се моделира заедничка веројатност

p (x, y)

со повеќе *(x, y)* парови. Параметрите на моделот *p (x, y)* повторно се проценуваат од примерите ($x^{(i)}$, $y^{(i)}$) за *i* = 1….n. Во многу случаи, дополнително се разложуваат веројатностите *p (x, y),* на следниов начин:

$$p(x, y) = p(y)p(x|y)$$

и потоа се проценуваат моделите за p (y) p(x I y) одделно.

Во однос на генеративниот модел, може да се користи Баесово правило, да се изведе условната веројатност *p (y I x)* за секој *(x, y)* пар:

$$p(y|x) = \frac{p(y)p(x|y)}{p(x)}$$

Каде што

$$p(x) = \sum_{y \in \mathcal{Y}} p(x, y) = \sum_{y \in \mathcal{Y}} p(y)p(x|y)$$

На тој начин, заедничкиот модел е доста разновиден, така што може да се изведат веројатностите за p (x) и *p (y I x).*

Ние го користиме Баесовото правило директно во примена на заедничкиот модел на нов тестен пример. За даден влез x, излезот од нашиот модел *f(*x*)* може да се изведе како следново:

$$
\begin{aligned}
f(x) &= \arg\max_y p(y|x) \\
&= \arg\max_y \frac{p(y)p(x|y)}{p(x)} \\
&= \arg\max_y p(y)p(x|y)
\end{aligned}
$$

Во оваа равенка имаме примена на Баесово правило. Бидејќи именителот p (x) не зависи од *y,* со тоа не влијае ниту на функцијата arg max. Ова на еден начин е погодно, бидејќи тоа значи дека не е потребно да се пресметува p (x), која може да биде и сложена операција.

Модели кои ја разделуваат заедничката веројатност во услови p(y) и p(x I y) најчесто се нарекуваат noisy-channel модели.

Интуитивно, кога ќе видиме тестен пример x, се претпоставува дека е генериран во два чекори: прво, етикетата y е избрана со веројатност p (y); Второ, дека примерот x е генериран од распределбата p (x I y). Моделот p (x I y) може да се толкува како „канал“ кој ја зема етикета *y* како негов влез, и ја разградува за да се произведе *x* како краен резултат. Наша задача е да се најде повеќе веројатна етикета *y,* со оглед на тоа што ние ја знаеме вредноста на *x*. Наоѓање на излез F (x) за даден влез *x*, често се нарекува декодирање на проблемот.

4.3 Генеративен модел на означување

Дефиниција 1 (Генеративен модел на означување): Да претпоставиме дека имаме конечно множество на зборови V, и конечно множество на тагови K. Дефинираме S да биде збир од сите секвенци / таг – секвенци парови < $X_1 \ldots Xn$, $Y_1 \ldots Yn$> така што n≥ 0, $X_1 \in V$ за $i = 1 \ldots . n$ и $Y_1 \in K$ за $i = 1 \ldots . n.$ Така, генеративниот модел на означување претставува функција која е дефинирана на следниов начин:

1. За секое < $X_1 \ldots Xn$, $Y_1 \ldots Yn$> ∈ S,

$$p(x_1 \ldots x_n, y_1 \ldots y_n) \geq 0$$

2. Покрај тоа,

$$\sum_{\langle x_1 \ldots x_n, y_1 \ldots y_n \rangle \in \mathcal{S}} p(x_1 \ldots x_n, y_1 \ldots y_n) = 1$$

Оттука, p < X_1…Xn, Y_1…Yn> претставува распределба на веројатноста кај повеќето парови во дадената секвенца.

Со оглед на генеративниот модел на означување, функцијата од реченици X_1…Xn во таг секвенци Y_1…Yn се дефинира како

$$f(x_1 \ldots x_n) = \arg \max_{y_1 \ldots y_n} p(x_1 \ldots x_n, y_1 \ldots y_n)$$

Каде што arg max се зема во текот на сите секвенци Y_1…Yn така што $Y_1 \in K$ *за* $i \{1 \ldots n\}$. Така за секој влез X_1…Xn ние ја земаме таг секвенцата со најголема веројатност како краен излез од моделот.

4.4 Марков модел

Марковиот модел претставува низа од случајни променливи кои не се независни. На пример, со овој модел може да се прикажуваат временски извештаи. Извештајот за времето или за температурата за одреден ден зависи од тоа каква била температурата во претходниот ден. Исто така, може да се применува и при даден текст, така што веројатноста од следниот збор или следната буква во ваков случај зависи од претходниот збор или од претходната буква - соодветно.

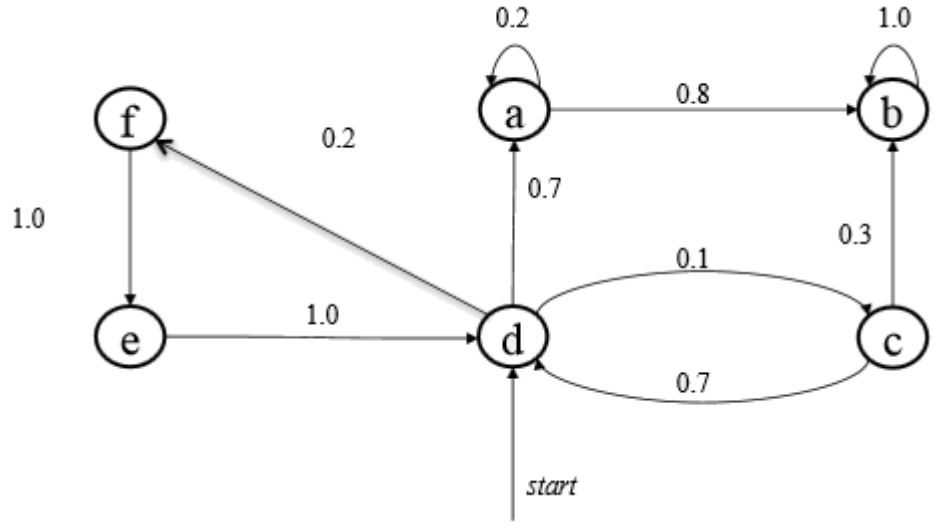
Марковиот модел се одликува според неколку карактеристики.

1. Ограничен хоризонт: Со карактеристиката ограничен хоризонт, веројатноста за набљудување на времето t+1, мора да зависи само од веројатностите на најновата историја, односно, само претходниот збор или претходните неколку зборови.

$$P(X_{t+1} = s_k|X_1,\ldots,X_t) = P(X_{t+1} = s_k|X_t)$$

2. Временски непроменлив, па така веројатноста на гледање на одредена променлива во одредено време не треба да зависи од самото време.

Дефиницијата на Марковиот модел во однос на транзицијата на дадена матрица А, ни кажува колкава е веројатноста на движење од една состојба во друга, како и за почетна состојба со веројатност π. Сликата бр. 4 ни претставува пример со шест состојби, a, b, c, d, e, f.



Слика бр. 4 Пример за Марков модел

Согласно на примерот, можеме да ја дефинираме распределбата на веројатноста за транзициите помеѓу овие состојби. Се започнува од состојбата d, и оттука доколку ги погледнеме сите лаци кои излегуваат од оваа состојба, можеме да видиме дека во состојбата *a* се движи со веројатност 0.7, во состојбата *c* со веројатност 0.1, и оди се до состојбата f со веројатност од 0.2. Доколку сме во состојбата *a*, гледаме дека во истата состојба се враќа со веројатност од 0.2 или се движи кон состојбата *b* со веројатност од 0.8. Истото важи и за останатите четири состојби. Она што е важно да се спомне, е дека со оглед на која било одредена состојба, збирот на веројатностите на сите појдовни транзиции е еднаков на еден.

Доколку сакаме да се пресмета веројатноста за секвенци од состојби, $X_1 \ldots X_t$, се користи следнава формула:

$$P(X_1,\ldots X_T) = P(X_1)\, P(X_2|X_1)\, P(X_3|X_1,X_2) \ldots P(X_T|X_1,\ldots,X_{T-1})$$

$$= P(X_1)\, P(X_2|X_1)\, P(X_3|X_2) \ldots P(X_T|X_{T-1})$$

$$= \pi_{X_1} \prod_{t=1}^{T-1} a_{X_t X_{t+1}}$$

Оваа формула всушност го претставува првото набљудување за првата состојба, која е помножена со веројатноста од втората состојба, со оглед на првата состојба, со времето на веројатноста на третата состојба со оглед на претходните две, и така натаму до последната состојба со оглед на сите состојби претходно. Ова претставува најопштата формула за заедничка дистрибуција $X_y \ldots X_t$. Со користење на ознаката на претпоставката и користењето на биграм моделот, може да се изврши промена со веројатност за набљудување X1 пати, за набљудување на X2 со дадено X1, помножена со веројатноста за набљудување Xt со дадена само X2 состојбата. Во ваквата ситуација, се врши игнорирање на некои состојби од историјата, во конкретниот пример тоа е X1, гледајќи само во двете најнови состојби, и потоа крајниот израз е Xt со дадено Xt-1, наместо Xt да ја земе предвид секоја состојба што е пред Xt. Значи во компактна форма, на крај ова претставува производ на сите Xt+1 биграми за секвенцата, почнувајќи од i=1 до i=t-1. Во контекст на ова, подолу е прикажан еден практичен пример:

$$P(d, a, b) = P(X_1=d)\, P(X_2=a|X_1=d)\, P(X_3=b|X_2=a)$$

$$= 1.0 \times 0.7 \times 0.8$$

$$= 0.56$$

Од примерот погоре, се поставува прашањето за тоа која е веројатноста која се случува во секвенцата d-a-b? Во овој случај, се добива дека

веројатноста на првата состојба d, помножена со веројатноста на втората состојба a, земајќи ја предвид првата состојба d, помножено со веројатноста на третата состојба b, добиена со земање предвид на претходната состојба *a.* Кога сето ова ќе се замени со бројни вредности од конкретен пример, за првото добиваме вредност 1, за второто 0.7, додека за третото вредност 0.8, и кога ќе се помножат можеме да заклучиме дека секвенцата d-a-b има веројатност 0.56. Истата постапка се прави и за која било друга низа на веројатности, додека во овој пример, бидејќи се претпоставува дека почетната состојба е секогаш состојбата d, тоа значи дека друга низа која ќе започне со друга состојба освен d ќе има веројатност нула.

4.5 Скриен Марков модел

Скриен Марков модел е посоодветен за моделирање на јазик за разлика од видливиот Марков модел. Така, во делот на part-of-speech, се врши набљудување на секвенца од знаци, без притоа да е познат редоследот на состојбите кои придонесле за генерирање на тие симболи. Затоа, симболите кои ние ги гледаме всушност се вистински зборови, и редоследот на состојбите се дел од говорот кои одговараат на овие зборови.

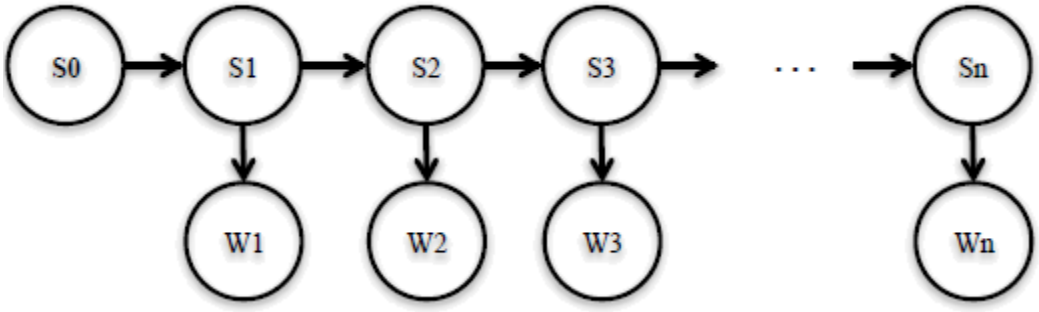
Дефиницијата на Скриен Марков модел е следната:

- Q – секвенца од состојби;
- O – секвенца на набљудувања;
- q0,qf – специјални (почетна и крајна) состојби;
- A – состојба на веројатност на транзиција;
- B – веројатност за емисија на симболи;
- П – иницијална состојба на веројатности;
- μ = (A,B,П) – целосен модел на веројатност

Структурата на Скриен Марков модел ги опфаќа следните параметри: Q е низа од опслужени состојби, или низа од набљудувања која е составена од голем вокабулар, q0 и qf претставуваат специфични почетна и крајна состојба. A е множество од состојби на транзициски веројатности, исто како и во видливиот модел, но овде има нов сет на параметри кој се означува со B, кој е познат како веројатност за емисија на симболи. Всушност, доколку се наоѓаме

во дадена состојба, се поставува прашањето за тоа колкава би била веројатноста дека ќе биде емитуван одреден симбол од таа состојба, така што може да го има и П, кој исто како и кај видливиот модел се одликува со почетната состојба на веројатности. μ, кој претставува унија од А, В и П, е целосно веројатен модел со кој се утврдува скриениот Марков модел.

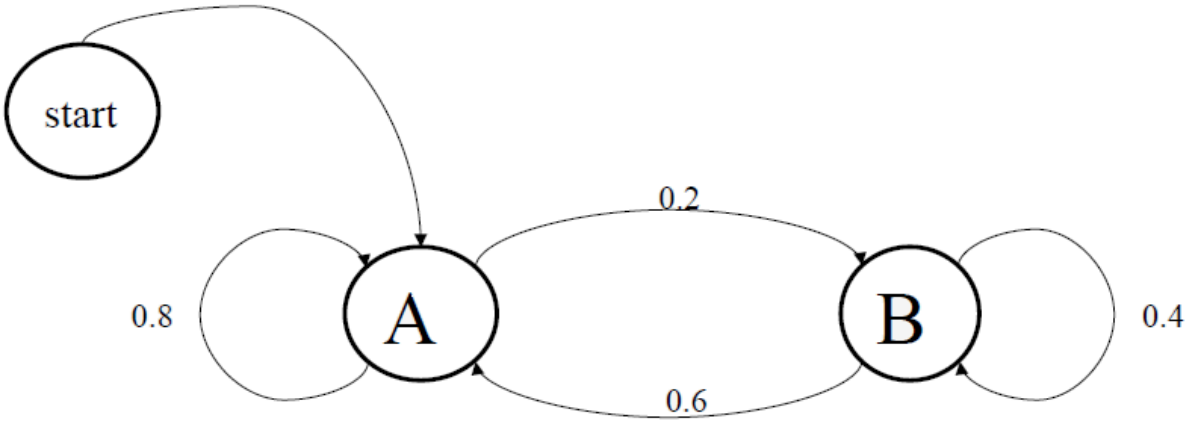
Скриен Марков модел се користи кај тагирањето на дел од говорот, препознавање на говорот и секвенционирање на ген, и може да се користи за да се моделираат секвенци на состојби и секвенци за набљудување. На пример, сакаме да ја дознаеме веројатноста дека одредена група на состојби е во корелација со одредена група на зборови. Па така, во биграм моделот ние всушност можеме да го напишеме ова како производ на сите биграм веројатности да излегуваат од една состојба до пронаоѓање на следната веројатност за да се емитира одреден симбол Wi кога сме во состојба Si. На сликата подолу е претставен едноставен пример.



***Слика бр. 5 Пример 1 на* Скриен Марков модел**

Редоследот на состојбите во примерот на сликата погоре е од S_0 до Sn, и кога се наоѓаме во состојба еден, се произведува W_1 како излезен симбол, кога сме во состојба два, се произведува W_2, и така натаму се до состојба Sn кога се произведува зборот Wn како излезен резултат.

Алгоритамот HMM е генеративен алгоритам. Подолу е претставен друг пример кој се состои од две состојби А и В.



***Слика бр. 6 Пример 2 за* Скриен Марков модел**

Дадени ни се две состојби A и B. Овие состојби се поврзани едни со други со одредени веројатности. Од примерот погоре можеме да видиме дека веројатноста која оди од A до B изнесува 0,2, веројатноста која останува во A изнесува 0,8, веројатноста која оди од B до A е 0,6, а веројатноста која останува во B е 0,4. Она што е важно да се спомне, е дека веројатностите кои произлегуваат од еден јазол или од една состојба треба да изнесуваат 1, што може да е случај и во овој пример. Исто така се дефинира и почетен симбол start или почетна состојба или почетна веројатност. Во овој случај се започнува од состојбата A, па веројатноста за почнување од состојбата B има вредност нула.

Сега, ајде да погледнеме поконкретен пример. Сега, ние сме заинтересирани за веројатноста на секоја низа на набљудување во дадено време t. Значи, која е веројатноста за набљудување на времето да биде еднакво на k, со оглед на тоа дека во моментов сме во одредена состојба и дека во претходната состојба е прикажана веројатноста? Во продолжение е претставена емисијата на матрицата на веројатностите.

	x	y	z
A	0.7	0.2	0.1
B	0.3	0.5	0.2

Доколку се наоѓаме во состојба A, најмногу веројатно е да се произведе симбол X со веројатност од 0,7. Помала е веројатноста да се произведат Y и Z симболите, бидејќи нивните веројатности кои произлегуваат од состојбата A се 0,2 и 0,1 соодветно, но сепак постои шанса за генерирање на истите. Од состојбата B, најмногу веројатно е да се произведе симболот Y со веројатност од 0,5. X и Z може да ги произведе со веројатност 0,3 и 0,2 соодветно. На прашањето кои се параметрите на моделот, одговорот е дека првичните параметри се A со даден почеток 1,0 и состојба B со даден почеток 0,0.

$$P(A|start) = 1.0, P(B|start) = 0.0$$

Во однос на примерот, во продолжение е прикажана транзицијата на веројатностите.

$$P(A|A) = 0.8, P(A|B) = 0.6, P(B|A) = 0.2, P(B|B) = 0.4$$

Покрај транзицијата на веројатностите, овој модел ја прикажува и емисијата на веројатностите.

$$P(x|A) = 0.7, P(y|A) = 0.2, P(z|A) = 0.1\ , P(B|B) = P(x|B) = 0.3, P(y|B) = 0.5, P(z|B) = 0.2$$

За да утврдиме колкава е веројатноста при набљудување на секвенцата yz, примерот кој ќе го разработиме, низ неколку чекори ќе ни ја објасни целата постапка. Ќе се започне со состојбата A, а потоа ќе се разгледаат и останатите секвенци на другите состојби. Најпрво, може да се започне од состојбата A и да се врати во истата, да започне од A и да се префрли на B, и може да оди до B и потоа да оди до A. И, конечно, можеме да одиме од B и да останеме во таа состојба. Секоја од овие секвенци на состојбите може да доведе до одредена веројатност за yz набљудувањето.

Така, доколку сакаме да ја разгледаме веројатноста на yz за дадена секвенца од два чекори, ќе треба да се разгледа веројатноста на yz која е генерирана од секвенцата AA, веројатноста за yz која е генерирана од секвенцата AB, веројатноста на yz која е генерирана од BA и на крај веројатноста на yz која е генерирана од BB. Збирот на сите овие четири

состојби ќе ни ја даде целосната веројатност за секвенците на набљудувањето на yz (Слика 7).

$$
\begin{aligned}
P(yz) &= P(yz|AA) + P(yz|AB) + P(yz|BA) + P(yz|BB) = \\
&= .8 \times .2 \times .8 \times .1 \\
&+ .8 \times .2 \times .2 \times .2 \\
&+ .2 \times .5 \times .4 \times .2 \\
&+ .2 \times .5 \times .6 \times .1 \\
&= .0128 + .0064 + .0080 + .0060 = .0332
\end{aligned}
$$

***Слика бр. 7 Пример за* Скриен Марков модел**

Така, првиот од четирите услови има вредност 0,8. Тоа е веројатноста која останува во состојбата A, со оглед на тоа дека започнавме со A, помножена со веројатноста која го прикажува *y* од состојбата A, која изнесува 0,2, помножена со веројатноста за престојување во A по втор пат. Тоа изнесува 0,8, која е помножена со веројатноста да се емитува z од состојбата A, кое е еднакво на 0,1. Во вториот дел, имаме 0,8 помножено со 0,2 кое се совпаѓа со претходниот дел. Сега имаме веројатност која оди од A до B, која изнесува 0,2 помножена со веројатноста за емитување на z од состојбата B, која исто така има вредност 0,2. Веројатноста за да се продуцира yz од дадена состојба BA, е еднаква на 0,2, која е всушност веројатноста на движење од состојба A до состојбата B, помножена со 0,5, која претставува вредност на веројатноста за производство на y во состојбата B, помножено со 0,4, која ја претставува веројатноста на движење од B до A, помножена со 0,2, која ја прикажува веројатноста за генерирање од состојбата A. Последната линија е многу слична. Тоа е 0,2 помножено со 0,5, помножено со веројатноста на движење од B до B, која изнесува 0,6, помножена со веројатноста на емитување на z од состојбата B, која изнесува 0,1. Доколку се соберат сите овие зборови, ќе видиме дека вкупната веројатност на yz секвенцата е еднаква на 0,0332, или околу 3%. Условно кажано, ова претставува многу малку веројатна низа.

Со оглед на тоа што постојат девет можни секвенци на x, y и z, секој од нив во просек ќе се очекува да се појави околу 11% од времето. Особено за првото, ова е многу малку веројатно. Таа има само веројатност од 0,3. Ова е повеќе од очигледно ако се погледне во оригиналниот пример, бидејќи за да се

произведат y и z, ниту една од состојбите A и B не им дава голема веројатност. Тоа би било многу поверојатно со оглед на тоа дека почетокот е од состојбата A, дека ќе се произведе низа која ќе содржи најмалку еден x.

Слично на ова, може да се пресметаат веројатностите на секвенците zz или yx или xz итн. Веројатноста за сите овие секвенци ќе треба да се додаде до 1, бидејќи тоа се сите можни алтернативи кои може да се добијат како забелешки доколку сè започне од состојбата A. Во основа, состојбите се користат за шифрирање на најновата историја. Тие не мора секогаш да о вклучуваат најновиот дел од говорот. Тие можат да вклучат неколку најнови делови од говорот. Во рамките на заднинскиот модел, тие само ќе вршат кодирање на најновиот дел од говорот. Транзициите ги кодираат секвенците на состојбите. Така на пример, еден натпис не може да се следи од страна на глаголот. Затоа веројатноста за преминување од натпис во глагол ќе биде многу ниска. Она што е поверојатно е да се премине од придавка во именка, па веројатноста за секвенцата придавка именка ќе биде релативно повисока.

На прашањето за тоа како се проценуваат веројатностите на транзициите, одговорот е во примена на максималната проценка на веројатноста. Еден можен начин за да се примени ова е доколку имаме корпус. На пример, во делот за означување на говорот, може да се погледне во секвенцата на соседните делови на говорот и тоа да се искористи за да се процени веројатноста на транзицијата.

Проценувањето на веројатноста на емисијата, всушност е малку потешка за разлика од веројатноста на транзицијата, бидејќи може да има нови начини на употреба на специфични комбинации на зборови и делови од говорот. Така, специфичен збор може да се појави уште во податоците за обука како дел од говорот, но потоа во тесната околина да се појави со сосема различен дел од говорот. Постојат неколку предлози кои можат да се користат, како на пример со употреба на стандардна рамномерност. Исто така може да се користи и хевристики начин, врз основа на правописот на зборовите.

Редослед на набљудувања

Многу често, при примена на Скриен Марков модел, која во основа е и причината зошто овој модел е измислен, е случај кога набљудувачот може само да ги види симболите кои се емитираат, но не и состојбите кои довеле до нивно генерирање. Па така, нешто што би сакале да се пресмета во ваков случај е веројатноста на набљудување. Доколку е познат редоследот на набљудување и моделот μ, кој пак се состои од матрицата на транзиција, од емисијата на матрица и од почетната состојба на веројатноста, ние сакаме да ја пресметаме веројатноста на секвенцата, со оглед на дадениот модел μ. Така, која е веројатноста дека оваа низа е генерирана од страна на тој модел. Па, излегува дека е можно пресметаните секвенци на веројатноста на набљудувањата да го претвораат *HMM* во модел на јазик. Затоа се поставува прашањето, што претставува моделот на јазик? Тоа е начин да се пресмета веројатноста на секвенца или реченица или секвенца на набљудувања.

Скриен Марков модел - *задачи*

Постојат неколку важни задачи кои се поврзани со Скриен Марков модел.

- Кај првата, со оглед на моделот на веројатност $\mu = (\mathrm{A}, \mathrm{B}, \Pi)$ сакаме да се најде веројатноста за набљудување $P\ (O\ I\ \mu)$.
- Втората задача е со даден сет од опсервации и вредноста на μ, да дознаеме каков е редоследот на состојбите кои придонеле да се случи оваа опсервација.
- На крај, за позната вредност на опсервацијата О и сите можни познати вредности на μ, сакаме да се пронајде онаа вредност на μ, која најдобро ќе ја опише опсервацијата.

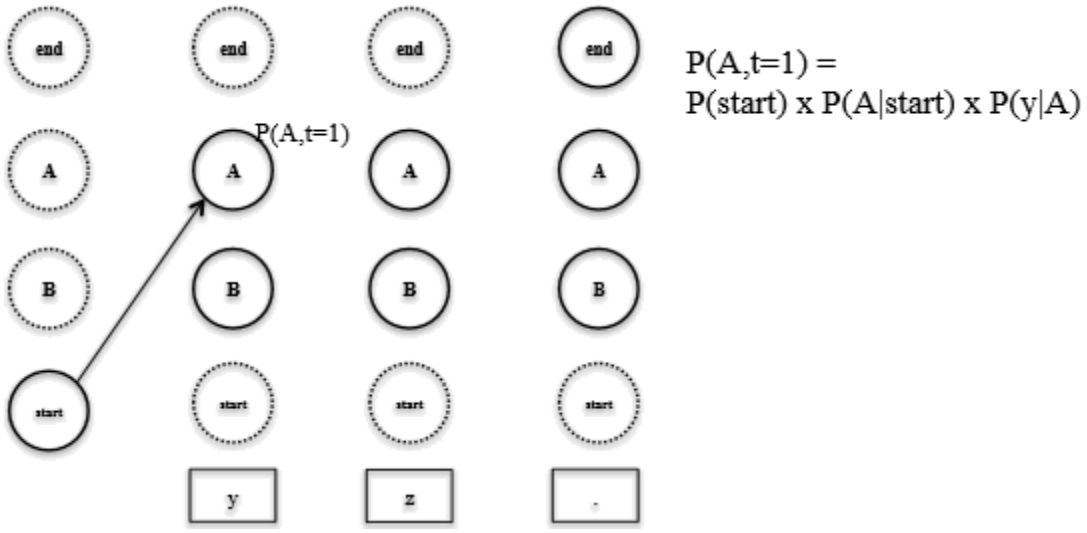
Една од најважните задачи во обработката на Скриен Марков модел се нарекува декодирање, чија што задача е да ја пронајде најверојатната низа. Значи, целта е да се тагира секој знак со етикета. Исто така може да пронајде и веројатноста за набљудување на секвенци, а може да се научи и преку модели за обука за да ги соберат емпириските податоци.

4.6 Viterbi алгоритам

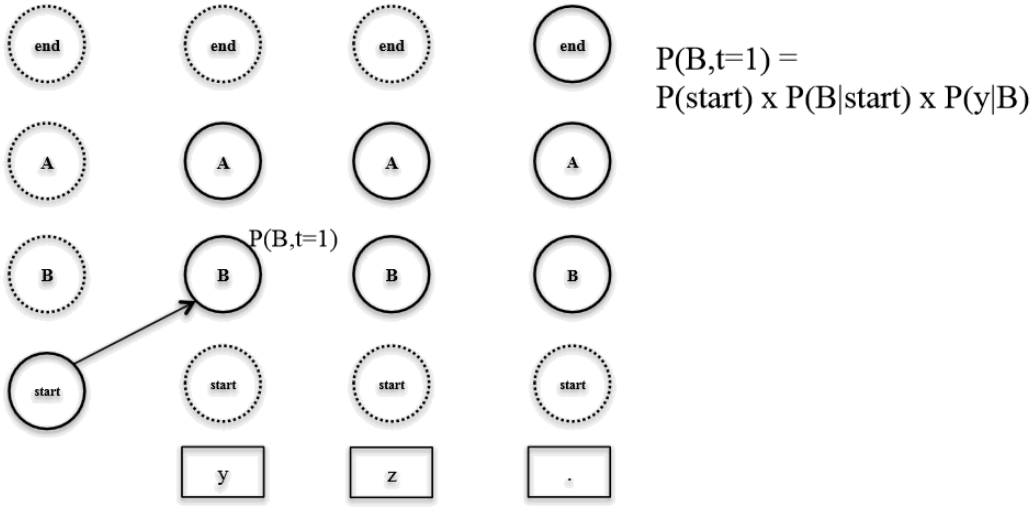
Алгоритамот Viterbi е еден од најпознатите основни алгоритми во обработката на природните јазици. Се базира на динамично програмирање и се користи за да се најде најдобриот пат S до конкретните опсервации I. Доколку се примени на целата низа, овој алгоритам ќе даде одговор за набљудувањето, односно за веројатноста на целата реченица. Но доколку се примени само до првиот или до вториот збор, тој само ќе ни го прикаже патот до таа точка. Значи, користи динамичко програмирање и меморизација, која во суштина е начин на чување на веројатностите на која било потсеквенца која веќе е пресметана, така што во нема да мора да ја пресметува во иднина повторно. Друга важна карактеристика на овој алгоритам се backpointers, која се користи доколку сакаме да ја зачуваме патеката, не само на најдобриот пат кој нè води до одредена состојба, туку ни го прикажува целиот процес како е стигнато до тој момент.

Во продолжение ќе разгледаме еден Пример за овој алгоритам. Овој пример се состои од четири редови и четири колони. Секоја од колоните одговара на една единица време. Се започнува од состојба T0, потоа T1 се до состојба T3. Редовите пак од првиот ред соодветствуваат на првиот дел од говорот. Тоа е почетниот симбол. Вториот и третиот ред, одговараат на два од другите делови на говорот, A и B. И на крај, имаме еден кој одговара на крај на состојбата. Она што го бараме е секвенца која нè води од почетниот јазол во долниот лев агол до крајот на состојбата во горниот десен агол, и оди преку низа од As до Bs во секоја колона.

Ќе го започнеме овој алгоритам подетално, започнувајќи од почетната состојба. Примерот е прикажан на сликите подолу.



Слика бр. 8. Пример за Viterbi алгоритам со почетна состојба А

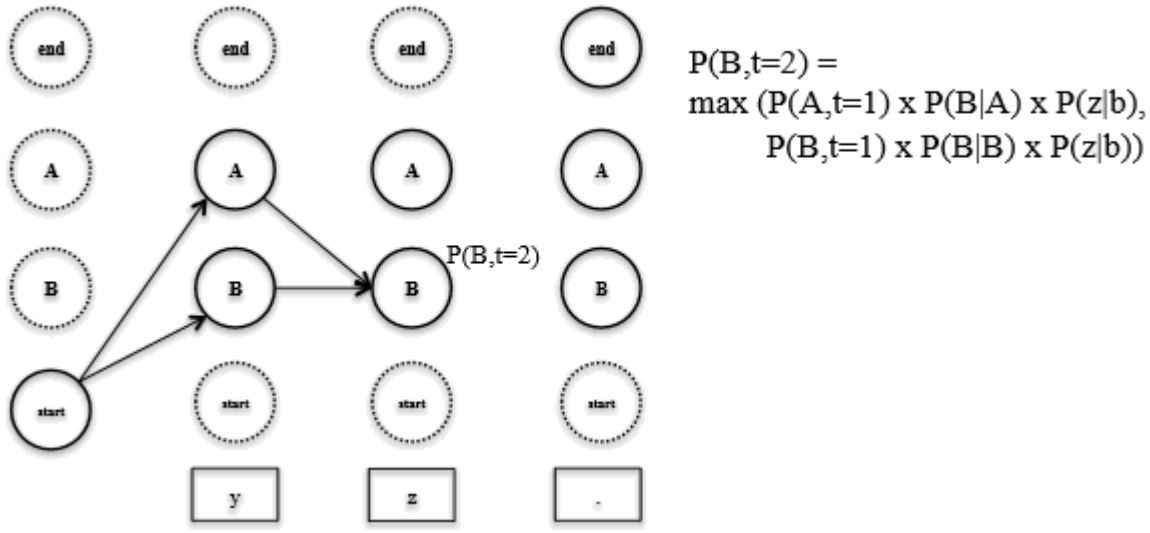


Слика бр. 9 Пример за Viterbi алгоритам со почетна состојба В

Од сликите погоре, можеме да видиме дека од почетната состојба може да се оди или до состојбата А или до В со одредена веројатност. Се поставува прашањето која би била веројатноста доколку се оди од состојба А во време t=1? Па така, произлегува дека веројатноста од оваа состојба за време еден е еднаква на веројатноста на почетната состојба помножена со веројатноста за А, со зададена почетна состојба, помножена со веројатноста за прием на симбол у, со оглед на тоа дека се наоѓаме во состојба А. Исто така, може да одбереме да одиме од состојба В. Во тој случај, се пресметува веројатноста на В, како производ од веројатноста на почетната состојба помножена со

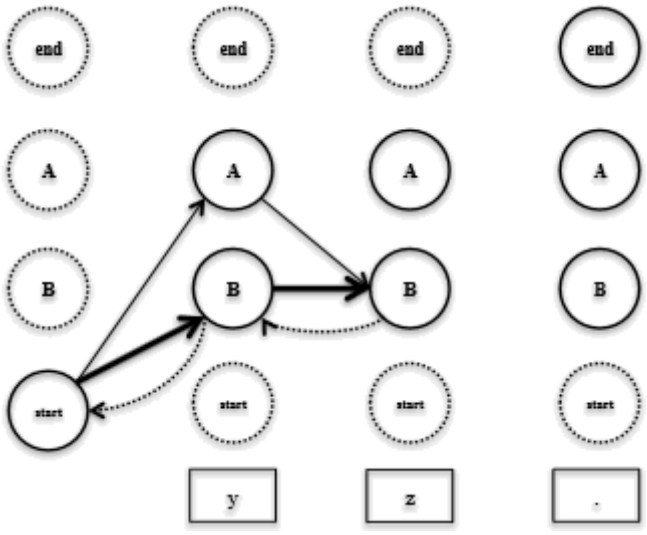
веројатноста за добивање на B од почетната состојба, со веројатноста за прием на симболот *y* од состојбата B.

Сега, како следна итерација, сакаме да се пресмета веројатноста на B во време t=2, слика бр. 10



Слика бр. 10 Веројатност на B во време t=2

Како што можеме да видиме од сликата, постојат два начина. Може да се оди од почетната состојба до A, потоа од A до B. Или, може да се оди од почетната состојба до B и од B до B. Веројатноста на состојбата B во време два, ќе биде еднаква на збирот на веројатностите на секоја од овие две патеки. Првата патека, има веројатност од A во време еден помножена со веројатноста од одење од A до B, помножена со веројатноста која емитува симбол z кога сме во состојба B. Втората патека пак, е производ почнувајќи од состојба B во време еден, потоа пренесување на состојбата B во време два, и на крај емитува симбол *z* кога сме во состојба Б. На крај произлегува дека најдобра варијанта во овој пример е, да се дојде до состојба B со дадена опсервација yz ако се отиде прво до состојба B и потоа да се остане во состојба B, отколку да се отиде прво во состојба A и да се префрли во состојба B. На крај, се доделува backpointer кој оди од состојба B во време два, во состојба B во време еден, за потоа повторно да започне од стартната состојба во состојба нула. Сликата подолу ни претставува пример при применување на backpointer-и.



Слика бр. 11 Користење на backpointer-и за враќање во почетната состојба

4.7 Триаграм јазичен модел

Постојат различни модели на дефинирање на јазичните модели, но еден од особено важни јазични модели е триграм јазичниот модел. Ова претставува директна примена на Марковиот модел за проблемот на моделирање на јазикот.

Исто како и кај Марковиот модел, и овде секоја реченица се моделира како низа од n случајни променливи, $X_1, X_2 \ldots X_n$. Должината на n, која и самата претставува случајна променлива, може да варира во различни реченици. Секогаш имаме дека X_n = STOP. Во рамките на Марковиот модел од втор ред, веројатноста за која било реченица $X_1 \ldots X_n$ е

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) = \prod_{i=1}^{n} P(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})$$

каде што претпоставуваме дека како и досега X_0 = X – 1 = *.

Ќе претпоставиме дека за било кое *i,* за секое $Xi-2, Xi-1, Xi$,

$$P(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1}) = q(x_i | x_{i-2}, x_{i-1})$$

каде што $q\ (w\ I\ u, v)$, за секое (u, v, w) е параметар на моделот. Нашиот модел, тогаш е во облик на:

$$p(x_1 \dots x_n) = \prod_{i=1}^{n} q(x_i | x_{i-2}, x_{i-1})$$

за која било низа $X_1 \dots X_n$.

Ова доведува до следната дефиниција:

Дефиниција 2 (Триграм јазичен модел): Триграм јазичниот модел се состои од ограничен сет V и параметар

$$q\ (w\ I\ u, v)$$

за секој триграм u, v, w, така што $w \in$ V U {STOP}, и $u, v \in$ V U {*}. Веројатноста за гледање на $q\ (w\ I\ u, v)$, може да се толкува како веројатноста за гледање на зборот w веднаш по биграмот (u, v). За секоја реченица $X_1 \dots X_n$, каде $Xi \in V$ за $i = 1 \dots (n-1)$, и $Xn = STOP$, веројатноста на реченицата според триграм јазичниот модел е:

$$p(x_1 \dots x_n) = \prod_{i=1}^{n} q(x_i | x_{i-2}, x_{i-1})$$

каде што дефинираме $X_0 = X - 1 = *$.

На пример, за реченицата

Како се викаш STOP

ќе го добиеме следново:

p (Како се викаш STOP)= q (Како|,*) x q (се|*, Како) x q (викаш| Како, се) x q(STOP| се, викаш)*

Од примерот може да забележиме дека имаме еден термин за секој збор во реченицата (Како, се, викаш и STOP). Секој збор зависи од претходните два збора: ова се нарекува триграм претпоставка.

Параметрите ги задоволуваат ограничувањата за секоја триграм *u, v, w,*

$$q(w|u,v) \geq 0$$

и за кој било биграм *u, v,*

$$\sum_{w \in \mathcal{V} \cup \{\text{STOP}\}} q(w|u,v) = 1$$

Така q *(w|u,v)* ја дефинира дистрибуцијата низ можните зборови *w,* условена од контекстот на биграмот *u,v.*

Клучниот проблем што ни останува е да се проценат параметрите на моделот, односно:

$$q(w|u,v)$$

каде што *w* може да биде кој било член од $\mathcal{V} \cup \{\text{STOP}\}$, и $u, v \in \mathcal{V} \cup \{*\}$. Постојат околу $|\mathcal{V}|^3$ параметри во моделот. Ова најверојатно ќе биде многу голем проблем.

Триаграм Estimation problem

Во однос на проблемите кои се јавуваат кај овој модел, најпрво ќе ја разгледаме максималната проценка на веројатност, којшто се јавува кај триграм моделот и претставува многу интуитивна проценка.

Значи, да претпоставиме дека имаме примери на реченици во нашиот јазик. Тоа би можело да бидат неколку милиони или неколку милијарди зборови или реченици. Од овие примери на реченици може да се изведат различни пресметки и варијации.

Значи се дефинира *c (u, v, w)* кој ќе ни кажува колку пати триграмот *(u, v, w)* се појавува во корпусот. Исто така, се дефинира *c (u, v)* кој ќе ни кажува колку пати биграмот *(u, v)* се појавува во корпусот. За било кое *u, v, w* се дефинира

$$q(w|u,v) = \frac{c(u,v,w)}{c(u,v)}$$

На пример:

$$q(\text{ викаш} \mid \text{Како, се }) = \frac{c(\text{ Како, се , викаш })}{c(\text{ Како, се })}$$

Од дадената равенка, можеме да заклучиме дека максималната проценка на веројатност се добива како сооднос од триграм и биграм.

Оценување на јазичен модел: Perplexity

За да може да се измери нивото на квалитет на еден јазичен модел, потребно е да се изгради еден заеднички метод кој ќе може лесно да ги процени нејасностите на тој модел кај податоци кои се чуваат надворешно.

Да претпоставиме дека имаме некои реченици за тест $x^{(1)}$, $x^{(2)}$,..., $x^{(m)}$. Секоја реченица за тест $x^{(i)}$ за $i \in \{1 \ldots m\}$ претставува секвенца од зборови $x_1^{(i)}$,…, $x_{ni}^{(i)}$, каде што *ni* ја претставува должината на *i-mata* реченица. Како и досега, се претпоставува дека секоја реченица на крај завршува со симболот STOP.

Она што е од суштинско значење е дека реченициите за тест се „надворешни“, во смисла на тоа дека тие не се дел од корпусот кој се користи за проценка на јазичниот модел. Во овој контекст, тие се примери за нови, невидени реченици.

За секоја реченица за тест *x(i),* може да се измери нејзината веројатност p (*x(i)*) во рамките на јазичниот модел. Природна мерка за квалитетот на

јазичниот модел ќе биде веројатноста која го назначува целиот сет на реченици за тест, а тоа е:

$$\prod_{i=1}^{m} p(x^{(i)})$$

Крајниот резултат би бил следниов: Колку оваа вредност е повисока, толку е подобар јазичниот модел во моделирање на невидени реченици.

Вредноста на Perplexity на тестен корпус е изведена како директна трансформација на оваа вредност. Се дефинира M да биде вкупниот број на зборови во корпусот за тестирање. Поточно, според дефиницијата дека *ni* ја претставува должината на *i-mama* тест реченица,

$$M = \sum_{i=1}^{m} n_i$$

Тогаш просечната веројатност на log-от под моделот се дефинира како:

$$\frac{1}{M} \log_2 \prod_{i=1}^{m} p(x^{(i)}) = \frac{1}{M} \sum_{i=1}^{m} \log_2 p(x^{(i)})$$

Ова е само log веројатноста на целиот тест корпус, поделена со вкупниот број на зборови во корпусот за тестирање. Овде се користи $\log_2 (z)$ за секое $z > 0$ да се повикува на log-от во однос на основното 2 од z. Повторно, колку е повисока вредноста толку е подобар јазичниот модел.

Така, Perplexity се дефинира како:

$$2^{-l}$$

каде што:

$$l = \frac{1}{M} \sum_{i=1}^{m} \log_2 p(x^{(i)})$$

Така го земаме негативното од просечната веројатност и го зголемуваме за две на таа вредност. Perplexity е позитивен број. Колку е помала нејзината вредност, толку подобро јазичниот модел е во моделирање на невидени податоци. Некои претпоставки зад овој метод на Perplexity е како што следува. Да претпоставиме дека имаме даден речник V, каде што I V U {STOP}I = N, и моделот го предвидува следново

$$q(w|u, v) = \frac{1}{N}$$

за сите u, v, w. Според тоа, ова претставува тивок модел кој едноставно предвидува униформна дистрибуција заедно со симболот STOP. Во тој случај може да се покаже дека вредноста на perplexity може да биде еднаква на N. Така, под единствен модел на веројатност, големината на perplexity ќе биде еднаква на големината на вокабуларот.

5. Алатки што се користат при анализа на корпус

Со преминот од традиционалното кон електронското и модерно учење и работа, се наметна и потребата за приспособување на говорните јазици кон ваквата технологија. Постојат голем број на алатки и програми кои се користат за анализа на природните јазици. Употребата на ваквите алатки е од огромно значење бидејќи овозможуваат детална и прецизна обработка на одреден јазик. Ваквите алатки сами од себе се приспособени со добро структурирани интерфејси и точно дефинирани функции кои овозможуваат креирање на анализи, статистики и различни извештаи кои му се потребни на корисникот. Во продолжение се прикажани само дел од алатките од ваков тип.

5.1 WordSmith алатка

WordSmith алатката претставува интегриран пакет од програми за анализирање како зборовите се однесуваат во текстовите. Може да се врши анализа на голем број на текстови и да се дознае како зборовите се користат во сопствениот или во некој друг текст. WordList алатката дава можност да се види листата на сите зборови и кластери и тоа по азбучен ред или според фреквенцијата. Concord алатката се користи за вршење на конкорданција и овозможува преглед на секој збор или фраза во корпусот. Со KeyWord пак алатка која исто така е во склоп на WordSmith, може да се пребаруваат сите клучни зборови во даден текст.

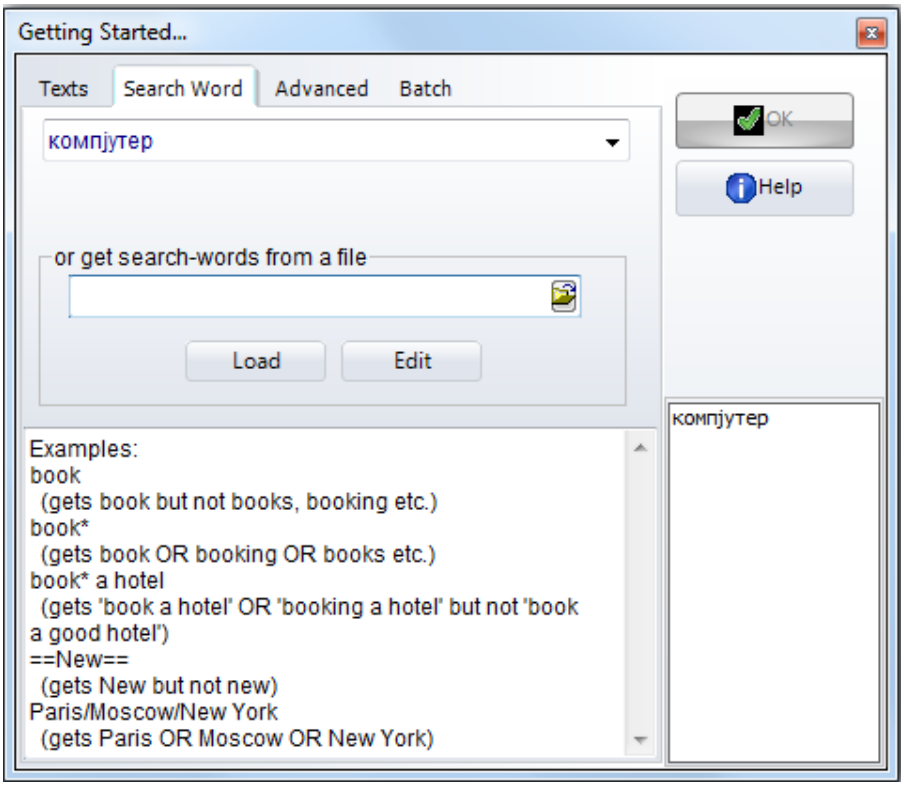
Значи, алатката WordList креира листи врз основа на еден или повеќе обични текстуални датотеки. Ваквите Word листи се прикажуваат и во азбучен ред и според фреквенцијата. Тие можат да се зачуваат за повторна употреба, за менување, печатени или зачувани како текстуални датотеки. Concord е програма која прави конкорданција со користење на DOS, ASCII или ANSI текстуални датотеки. При употреба, се пребарува одреден збор при што оваа алатка ќе изврши пребарување за ваков збор во сите датотеки кои се одбрани. Откако ќе заврши со пребарувањето, на крај прикажува екран на кој се дадени информации за колокацијата на зборот за кој пребарувавме. Листите можат да

се зачуваат за повторна употреба, за менување, печатени или да бидат зачувани како текстуални датотеки. Целта на KeyWord програмата пак е да се лоцираат и идентификуваат клучните зборови во даден текст. За да го стори тоа, програмата врши споредување на зборови во текстот со референтна група на зборови која обично се зема од голем корпус. Секој збор кој што ќе се пронајде како извонреден во неговата фреквенција се смета за „клуч“. Распределбата на клучните зборови може да се црта. Листите можат да се зачуваат за повторна употреба, за менување, печатени или да бидат зачувани како текстуални датотеки. Оваа програма треба да има пристап до 2 или повеќе збор листи, кои најпрво мора да бидат креирани со користење на WordList.

5.1.1 Примена на Concord

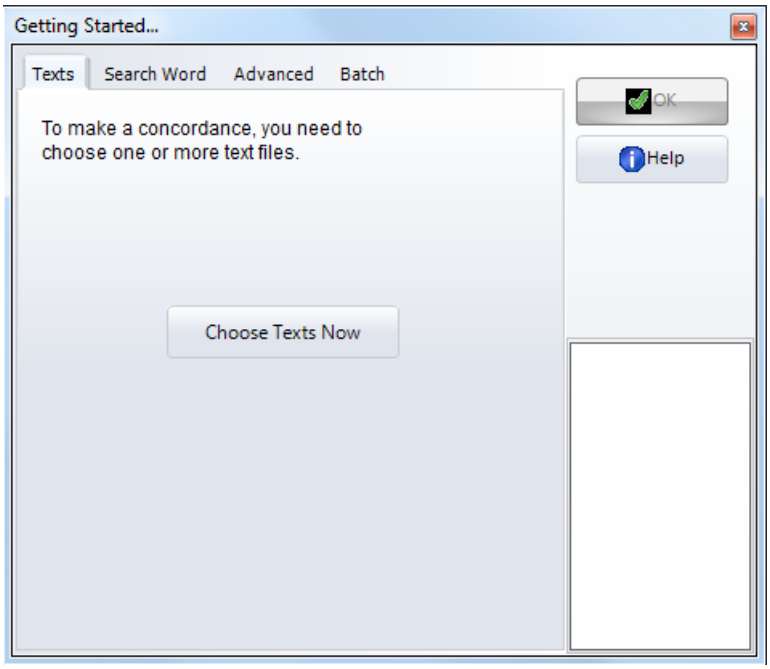
Алатката Concord, како што кажавме, се користи за вршење на конкорданција на одбраните фајлови за даден збор. Во продолжение во неколку чекори ќе го прикажеме начинот на поставување на алатката од импортирање сопствени датотеки до пребарување и приказ на излезни резултати за даден збор.

На почетниот екран кој се појавува со стартување на програмот WordSmith се одбира опцијата Concord со едноставен клик врз копчето именувано Concord. Следно, од главното мени се одбира Choose File | New и ќе се прикаже текст како на сликата подолу за да се одбере текст или да се направи конкорданција и сл.



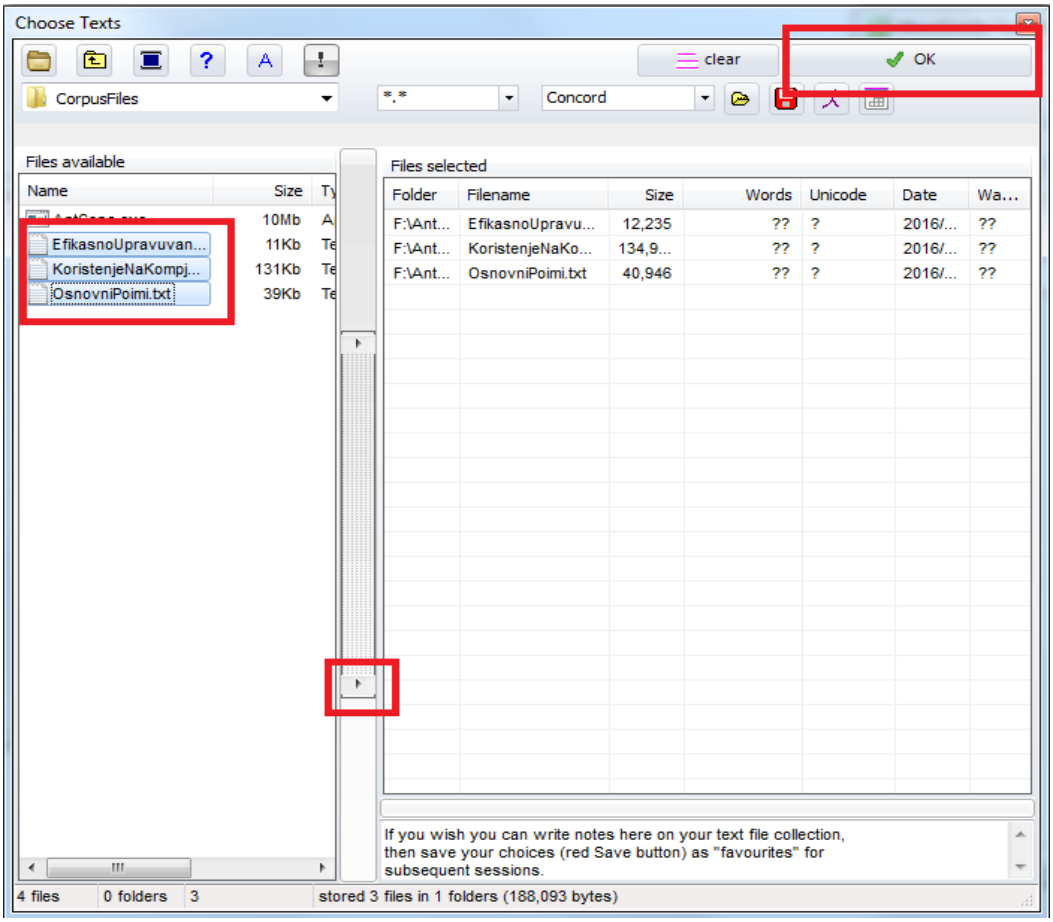
Слика бр. 12 Почетен екран на алатката WordList

За вршење на конкорданција кај оваа алатка, од менито се одбира текст, се внесуваат потребните датотеки и во горниот дел за пребарување во Search Word делот се внесува зборот за кој сакаме да ни се прикажат резултати.



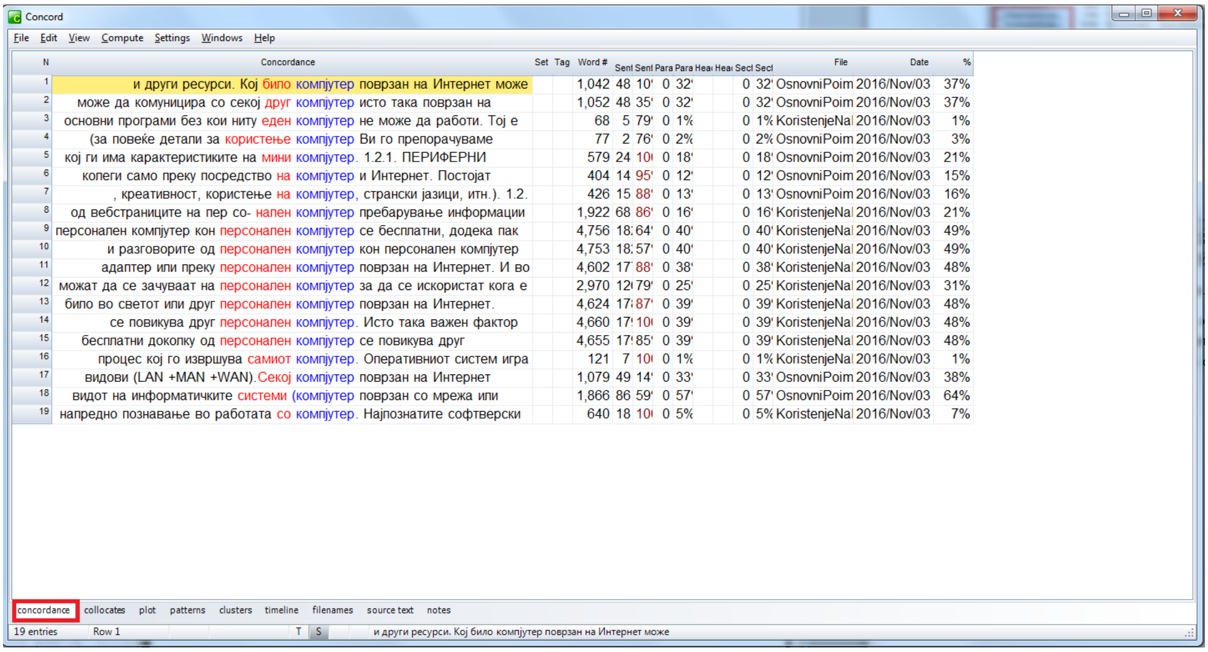
Слика бр. 13 Почетен екран на алатката WordList

Откако ќе кликнеме на копчето Choose Texts Now ќе ни се прикаже екран како на следната слика во кој е потребно да се внесат потребните датотеки кои ќе се излистаат од левата страна, а потоа со клик врз секоја поединечно и со притискање на стрелката ги внесуваме и притискаме ОК.



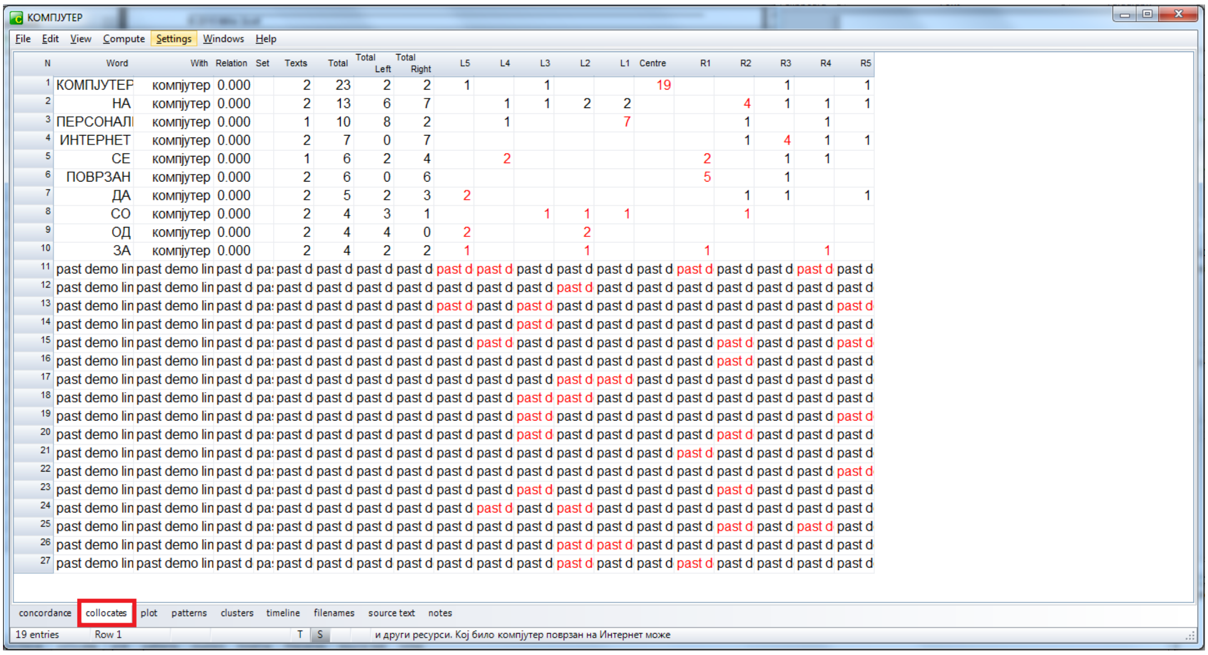
Слика бр. 14 Внесување на датотеки

При пребарување на даден збор, во овој случај терминот „компјутер“, како што напоменавме, во Search Word делот и со притискање ОК ќе ни се прикажат резултатите од пребарувањето на овој збор кое е прикажано на следната слика.

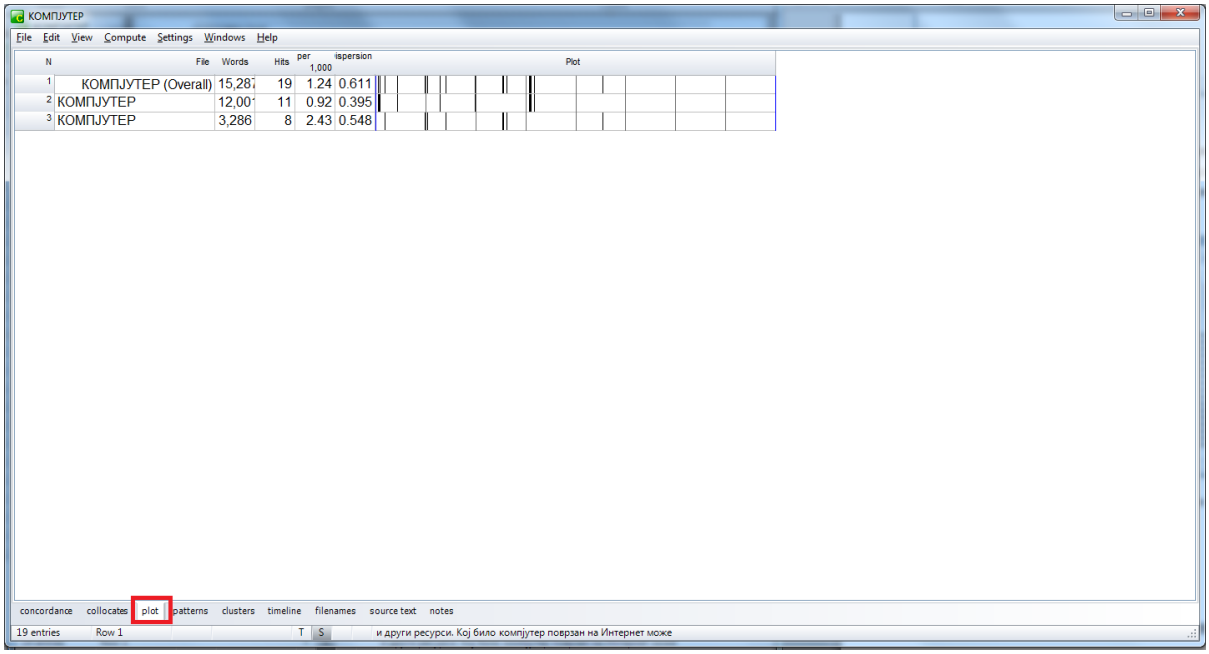


Слика бр. 15 Излезни резултати со примена на concordance

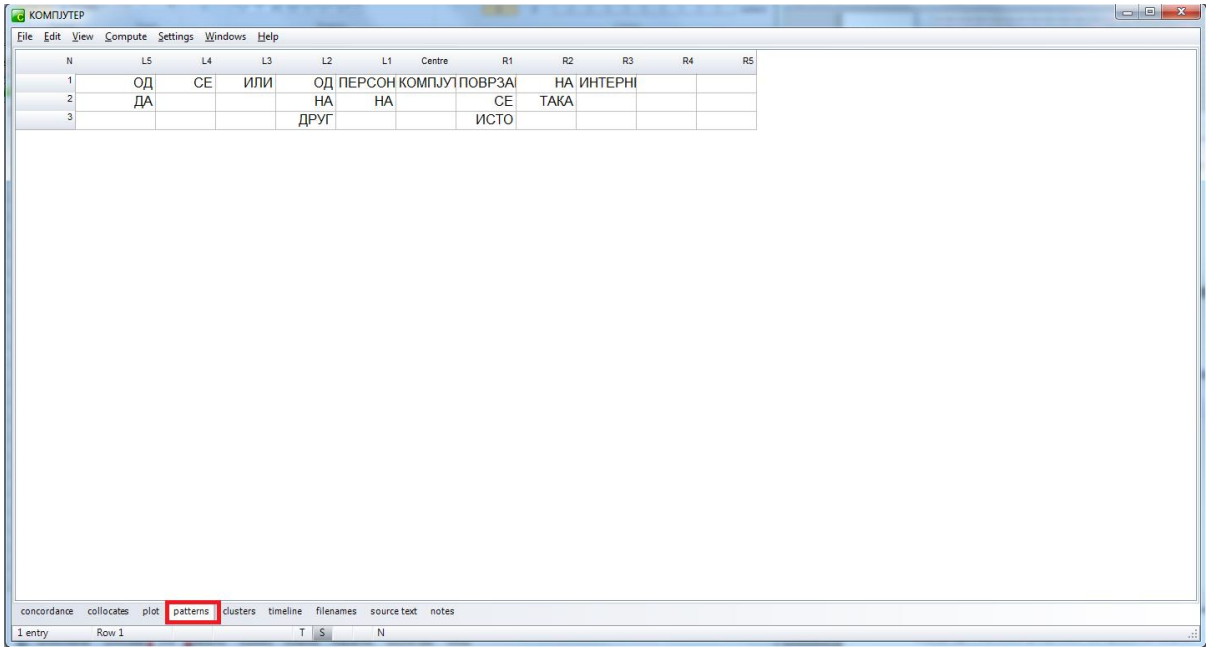
Освен конкорданција, оваа алатка ни дава извештаи и приказ за колокација, плот анализа, кластери, имиња на датотеки кои се вклучени во анализата итн. Во продолжение се прикажани излезни резултати за секое од гореспоменативе термини.



Слика бр. 16 Излезни резултати со примена на colocates



Слика бр. 17 Излезни резултати со примена на plot

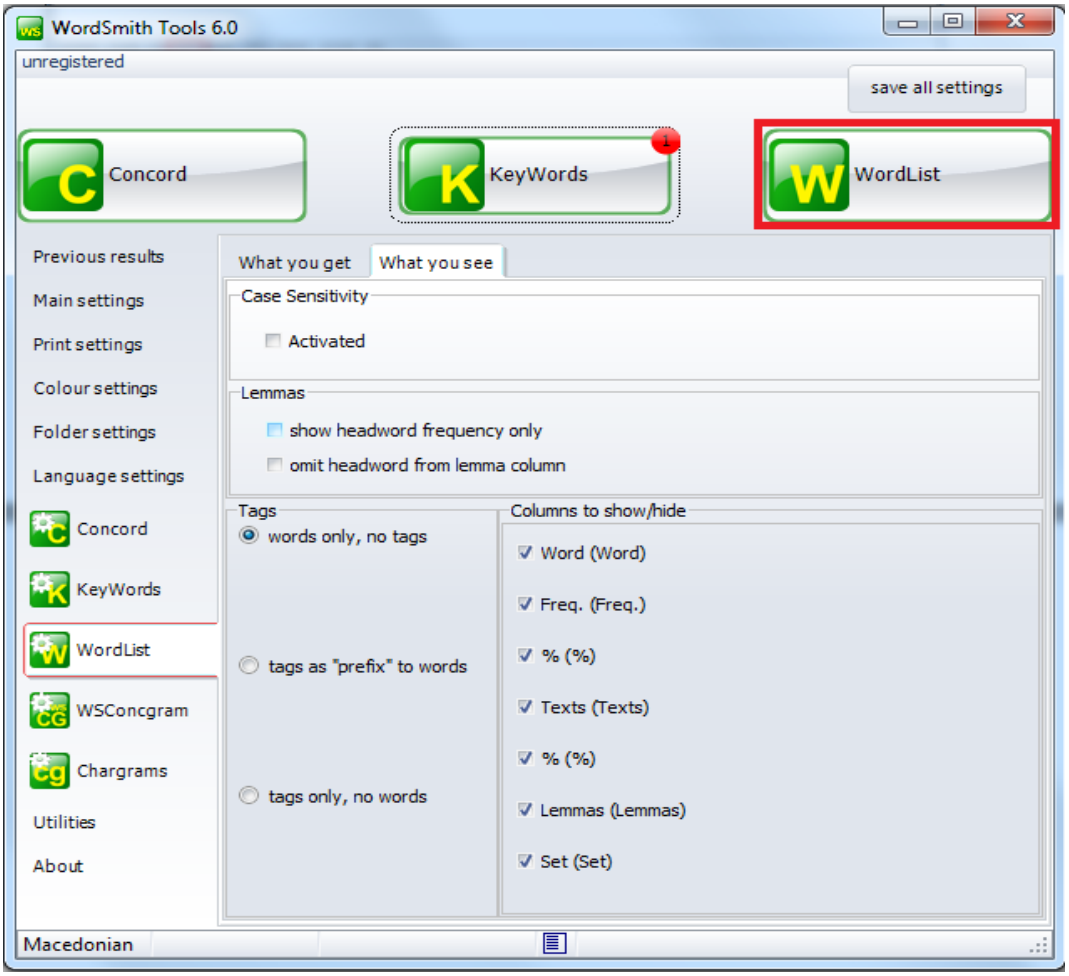


Слика бр. 18 Излезни резултати со примена на patterns

5.1.2 Примена на WordList

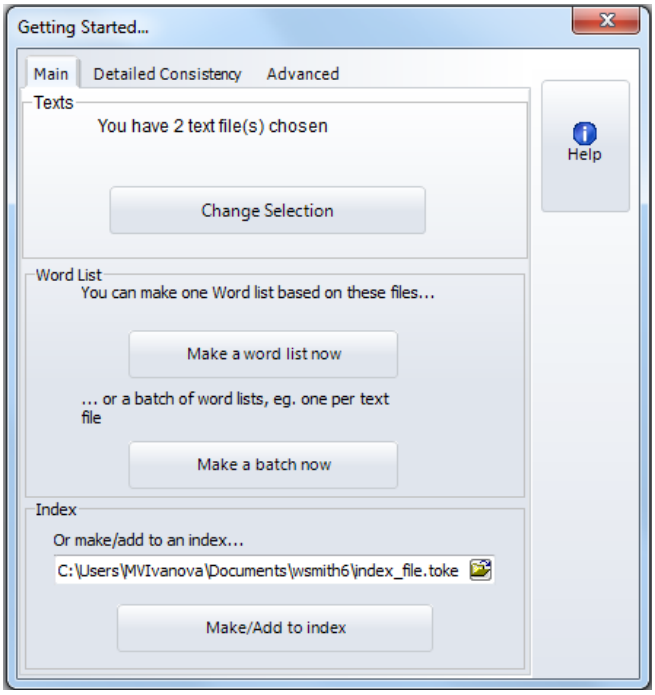
Алатката WordList се користи за креирање на листи наречени WordLists, кои подоцна се користат, на пример при генерирање на клучни зборови и за некои дополнителни анализи или споредби помеѓу неколку листи.

Стартувањето на алатката WordList се врши со кликање на WordList копчето од почетниот екран.



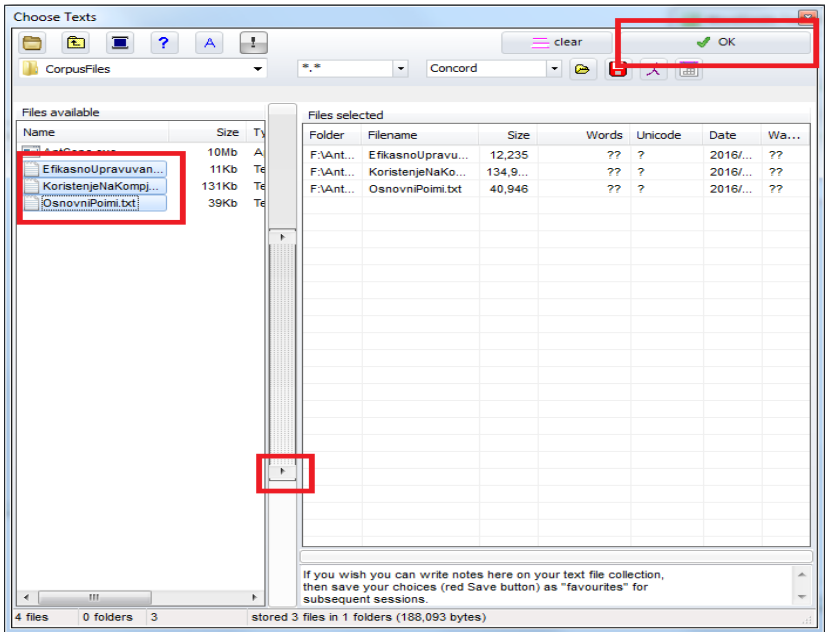
Слика бр. 19 Стартување на алатката WordList

Следно, од главното мени се одбира Choose File | New и ќе се прикаже екран како на сликата подолу за креирање на WordList.



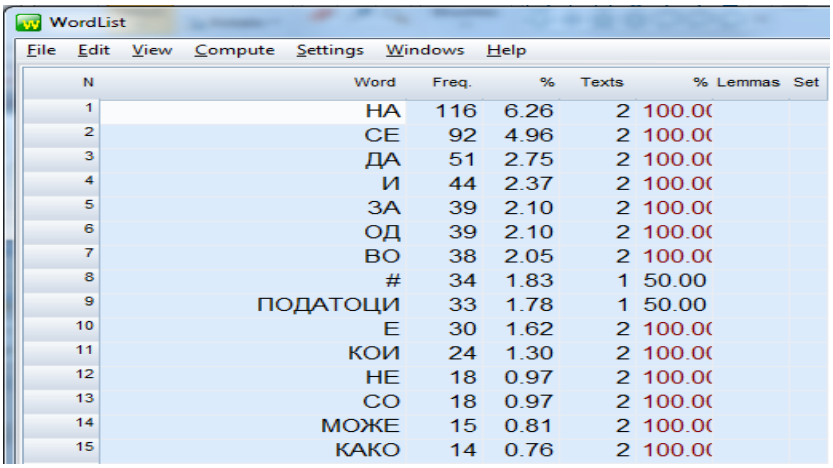
Слика бр. 20 Почетен екран на алатката WordList

Со кликање на Change Selection се појавува дијалог прозорец како на сликата подолу и се вметнува потребаната датотека од левата кон десната страна и се клика на копчето OK.



Слика бр. 21 Одбирање на текст

Потоа од почетниот екран на алатката WordList се одбира копчето Make a word list now и се прикажува резултат како на сликата подолу кој е потребно да биде зачуван за да може да се користи во иднина.



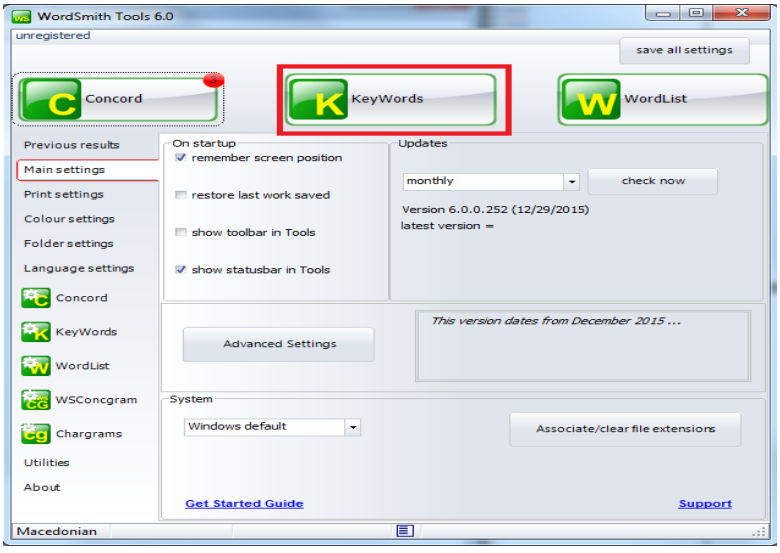
N	Word	Freq.	%	Texts	% Lemmas	Set
1	НА	116	6.26	2	100.0(	
2	СЕ	92	4.96	2	100.0(	
3	ДА	51	2.75	2	100.0(	
4	И	44	2.37	2	100.0(	
5	ЗА	39	2.10	2	100.0(	
6	ОД	39	2.10	2	100.0(	
7	ВО	38	2.05	2	100.0(	
8	#	34	1.83	1	50.00	
9	ПОДАТОЦИ	33	1.78	1	50.00	
10	Е	30	1.62	2	100.0(	
11	КОИ	24	1.30	2	100.0(	
12	НЕ	18	0.97	2	100.0(	
13	СО	18	0.97	2	100.0(	
14	МОЖЕ	15	0.81	2	100.0(	
15	КАКО	14	0.76	2	100.0(	

Слика бр. 22 Пример од генериран WordList

5.1.3 Примена на KeyWords

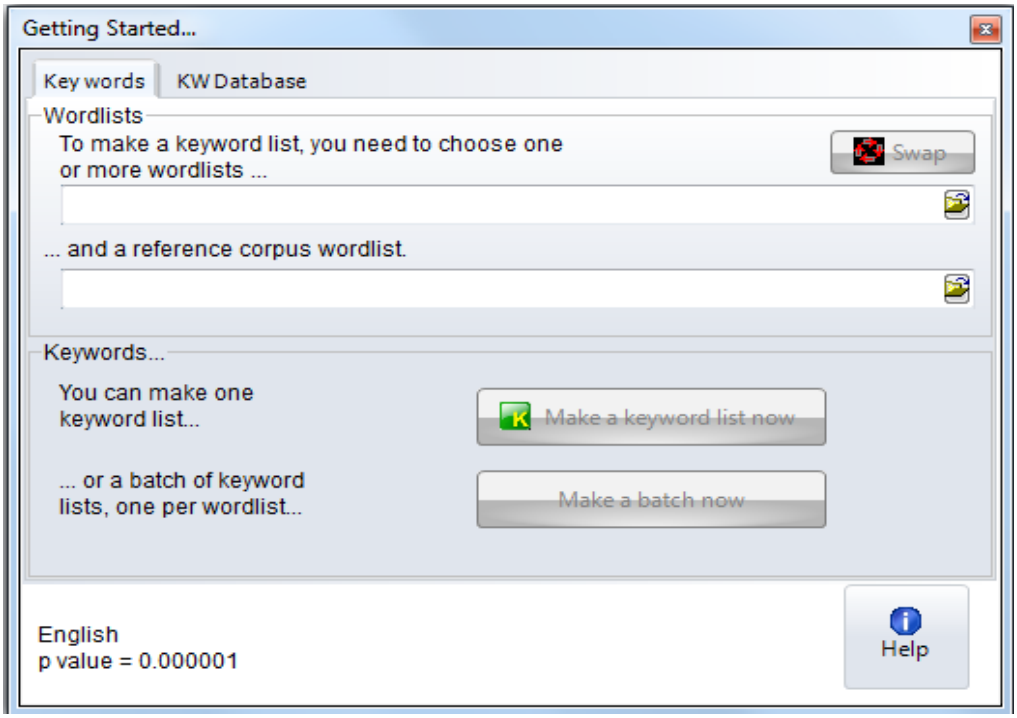
Алатката KeyWord се користи за да се лоцираат и идентификуваат клучните зборови во даден текст. За да го стори тоа, програмата врши споредување на зборови во текстот со референтна група на зборови која обично се зема од голем корпус.

Стартувањето на алатката KeyWord се врши со кликање на KeyWord копчето од почетниот екран.



Слика бр. 23 Стартување на алатката KeyWord

Следно, од главното мени се одбира Choose File | New и ќе се прикаже екран како на сликата подолу за креирање на KeyWord листи.



Слика бр. 24 Почетен екран на алатката KeyWord

Овде, ќе треба да се изберат две листи и да се направи листа на клучни зборови: една врз основа на еден текст или еден корпус, и уште една врз основа на друг текст или корпус, што е доволно за да се направи анализа и споредба на две датотеки. Значи, во првото поле од сликата горе се лоцира првата датотека, во вториот се лоцира втората датотека и на крај се одбира копчето за креирање на листата KeyWord.

Ова е само дел од прикажаното што го нуди програмот WordSmith. Станува збор за еден комплексен програм кој нуди голем број на опции со кои можат да се добијат различни анализи, статистики и комбинации од корпусите. Во зависност од потребата на корисникот или на организацијата воопшто може да се користат голем број од опциите за да се добие посакуваниот резултат.

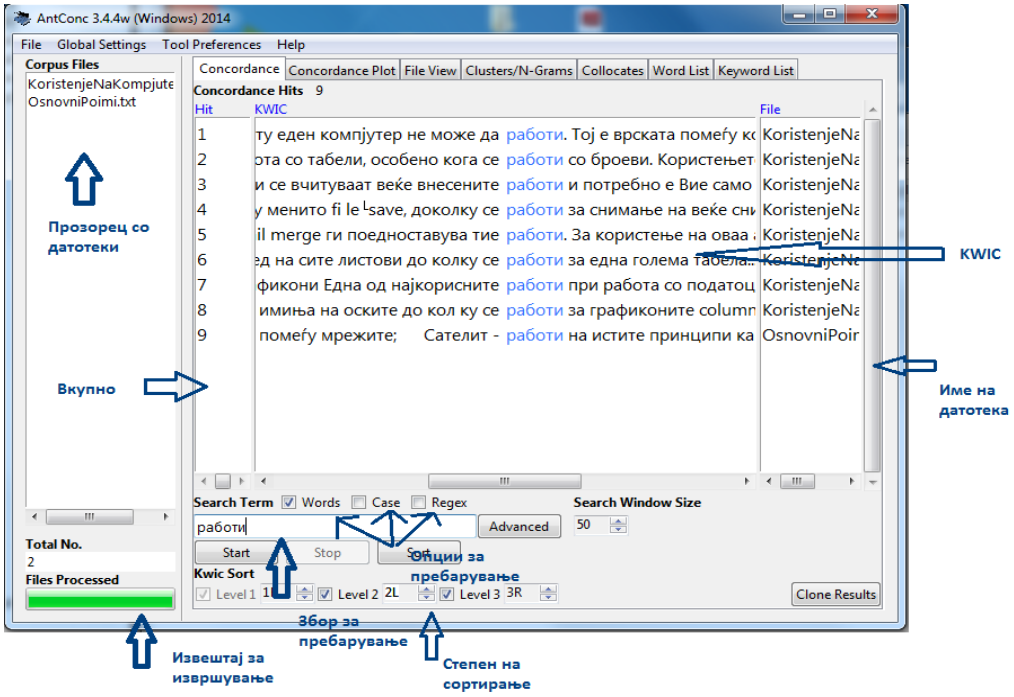
5.2 Алатка AntConc

AntConc претставува мулти платформа за корпус анализа која е дизајнирана за употреба во училница. Оваа алатка овозможува сеопфатен сет од алатки, вклучувајќи конкорданција, зборови, генератори за фреквенција на клучни зборови, алатки за кластери и лексички пакети за анализа. Корпусот е практично бескорисен без употреба на некаков вид на компјутерски софтвер или алатка за да обработува и прикажува резултати на разбирлив начин. Во денешно време, развиени се голем број на програми за анализа на корпус.

AntConc е бесплатна апликација, што ја прави идеална за поединци, училишта и колеџи со ограничен буџет. Се состои од многу лесен за употреба интуитивен графички кориснички интерфејс и нуди моќна функција за конкорданција, зборови, генератори за фреквенција на клучни зборови, алатки за кластери и лексички пакети.

5.2.1 Concordancer Tool

Централната алатка која се користи во повеќето софтвери за корпус анализа, вклучувајќи ја AntConc, е конкорданцијата. Оваа алатка се покажа како доста ефикасна во стекнувањето на втор странски јазик или странски јазик воопшто, олеснување при учење на вокабулар, колокации, граматика и пишување на стилови. Слика 25 ни прикажува екран при употреба на AntConc додека се применува конкорданција. Како и сите останати функции во програмата, и конкорданцијата е дизајнирана така што најчестите операции директно се достапни од главниот екран.



Слика бр. 25 KWIC (Key Word In Context), алатка за конкорданција

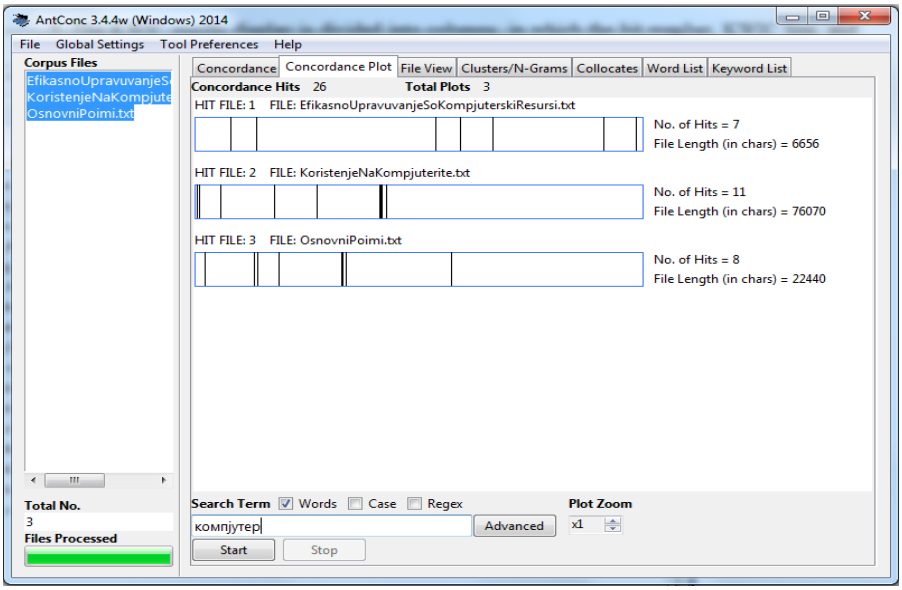
Алатката за конкорданција има широк спектар на функции кои ја прават крајно ефективна алатка, не само за учениците, но исто така и за истражувачите. Во продолжение ќе наведеме некои од важните функции на оваа алатка:

1. Условите за пребарување може да бидат поднизи, зборови или фрази, и да бидат со големи или со мали букви. Тие може да бидат вградени со широк спектар на специјални знаци, кон кои корисникот може да додели специјален знак или низа од знаци преку опција од менито.
2. Условите за пребарување може да се дефинираат како целосно регуларни изрази (називи), кои на корисникот му нудат пристап до исклучително моќни и комплексни пребарувања.
3. Три нивоа на сортирање на KWIC (Key Word In Context) се можни, со кои корисникот може да одреди бои во секое ниво.
4. Доколку корисникот кликне на некој израз во KWIC дисплејот за прикажување на резултатите, програмата автоматски ќе ја активира алатката за преглед на датотеки (View Files tool) и ќе ја прикаже датотеката на податоци каде што се наоѓа изразот.

5. Екранот за резултати во KWIC е поделен на колони, во една се прикажува бројот на резултати, KWIC делот и името на датотеките кои се прикажани одделно. Како и во сите други алатки, секоја колона може да биде прикажана или сокриена, и стандардните методи на избор може да се користат за зачувување на податоци во редови или колони на табелата со исечоци или текстуална датотека.

5.2.2 Concordance Search Term Plot Tool

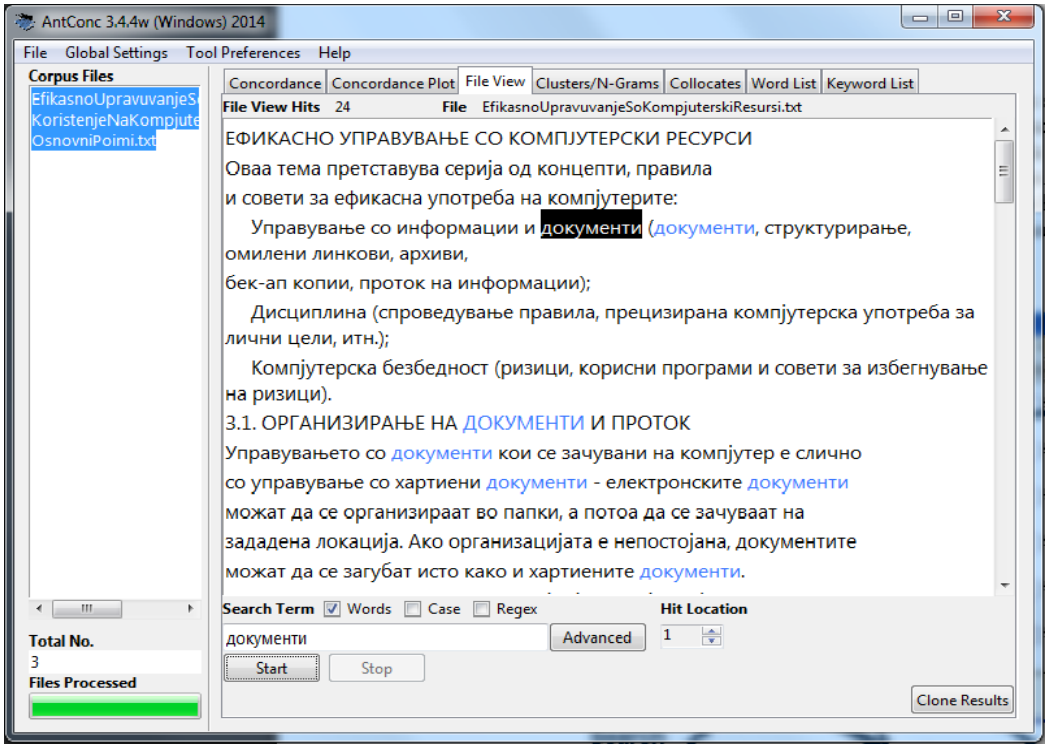
Главната цел на алатката за конкорданција е да ги покаже терминте за пребарување кои се користат во цел корпус. За корисниците кои сакаат да видат каде точно се наоѓа терминот кој го пребаруваат, AntConc им ја нуди оваа дополнителна алатка која прикажува точно за секој фајл поединечно каде се наоѓа терминот. Како што е прикажано на сликата подолу, секој фајл е претставен како еден вид на правоаголник кој е исполнет со сини линии кои го означуваат местото каде се наоѓа пребаруваниот термин. Значи секој правоаголник ги прикажува релативните положби на даден термин кои се изразени во хитови за секој фајл поединечно.



Слика бр. 26 Излезни резултати со користење на Concordance Plot

5.2.3 View Files Tool

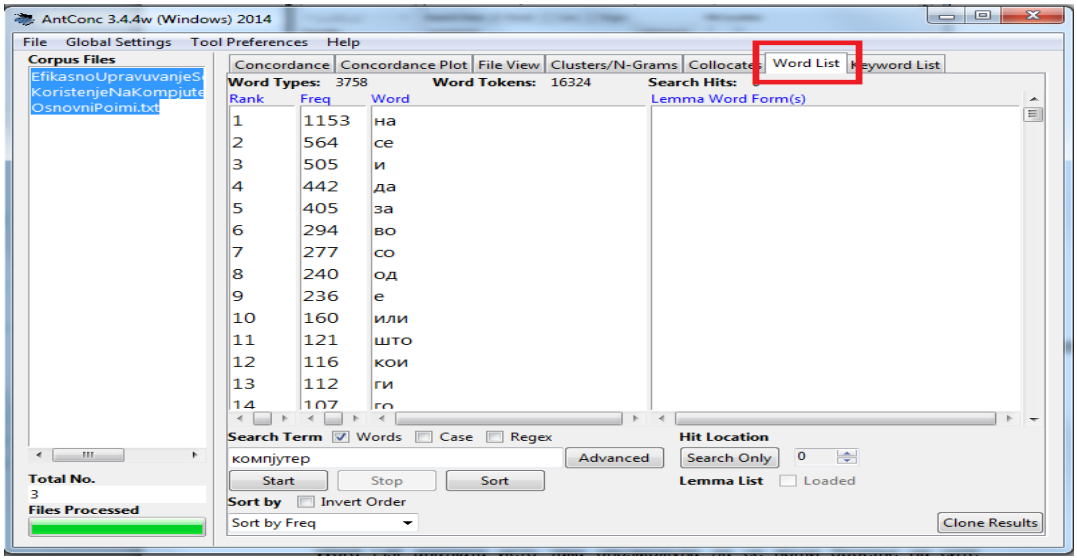
Алатката за прикажување на датотеките всушност ни овозможува, за секој импортиран фајл поединечно да имаме преглед каде и колку пати се појавува пребаруваниот термин само во тој фајл. Оваа алатката за конкорданција ни покажува колку пати се појавува терминот вклучувајќи ги сите фајлови во програмот. Оваа алатка прави еден вид на филтер, при што самиот корисник одбира за кој фајл да му бидат прикажани резултатите за тој термин. Значи може да се пребарува за секоја подниза, збор, фраза или регуларен израз во целата датотека.



Слика бр. 27 Излезни резултати со користење на View Files

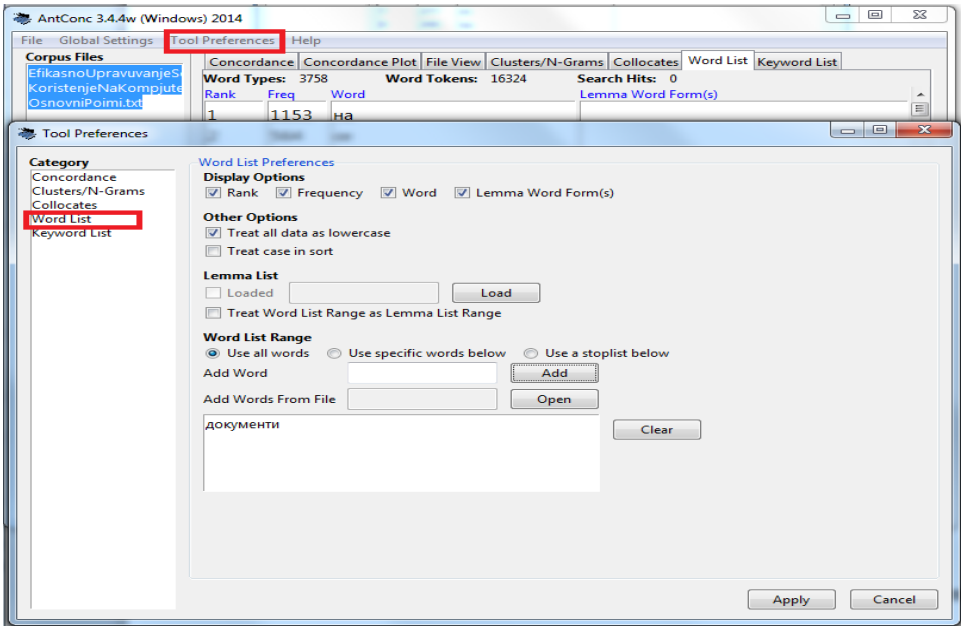
5.2.4 Word List / Keyword List Tools

Една од првите работи, којашто корисникот ќе ја направи при анализа на нов корпус, е да генерира листа на сите зборови кои се наоѓаат во корпусот. Алатката Word List е доста корисна за истакнување на интересни области во корпус и за укажување на проблематични области. Исто така, може да се користи за пронаоѓање на речнички единици или сличности кај поврзани форми на зборови. Пример од алатката Word List е прикажана на слика 28.



Слика бр. 28 Излезни резултати со користење на Word List

Алатката Word List исто така овозможува да се врши броење на сите зборови врз основа на нивните основни форми. Со цел да се избегне броење на голем број на функционални зборови при создавање на листа на зборови, стоп листа може да се специфира при предефинирање на одредени опции во делот за дополнителни привилегии за оваа алатка, или со директно внесување од тастатура или од посебна датотека.



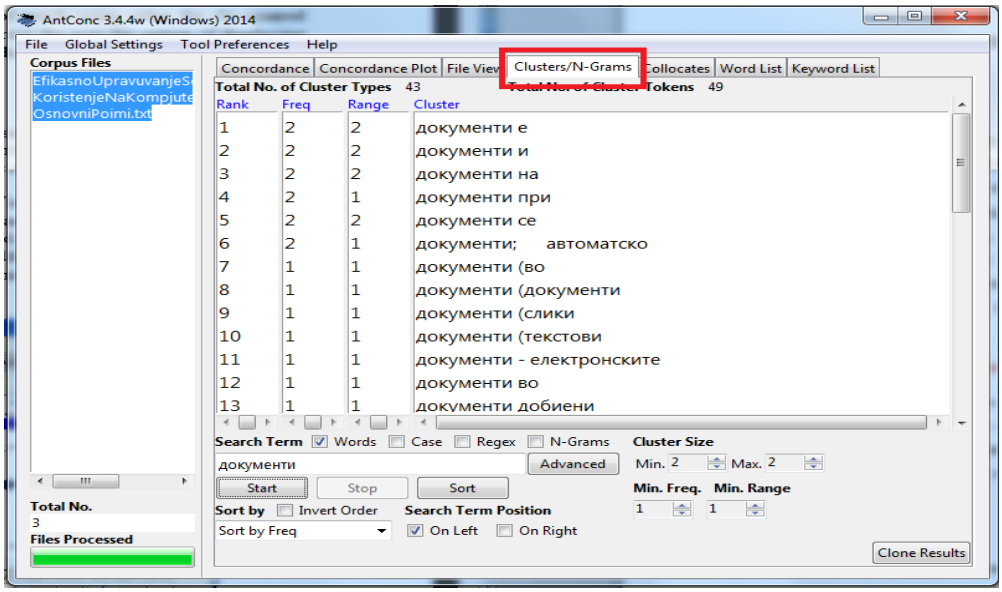
Слика бр. 29 Приказ на Tool Preferences додатоци

Како искусни корисници на алатките за корпус анализа, ќе забележиме дека алатката Word List обично ни ги прикажува сите зборови, и не ни кажува за тоа колку еден збор е важен во корпусот. Затоа, AntConc ја нуди алатката Key Word List , која ги пронаоѓа зборовите кои не се појавуваат многу често во корпусот во споредба со останатите зборови кои се наоѓаат во корпусот.

5.2.5 Word Clusters / Bundles Tool

Истражувањата покажале дека колокациите и останатите мулти-збор единици како фразалните глаголи и идиомите се особено тешки за учење од страна на учениците. Значењето на ваквите зборови станува уште позначајно доколку ученикот со текстови кои се од мошне технички или научно поле, како што е на пример лексичката единица која е многу подолга од еден збор.

Кај AntConc, мулти-збор единиците може да се истражуваат со помош на алатката Word Clusters прикажана на сликата подолу. Оваа алатка ги прикажува групите на зборови кои го опкружуваат терминот за кој пребаруваме и ги подредува по азбучен ред или по фреквенција. Термин за пребарување може да биде подниза, збор, фраза или регуларен израз, и исто како кај конкорданцијата и останатите алатки, и може да се регулира бројот на дополнителни зборови од десна и од лева страна од пребаруваниот израз. Исто така е можно да се постави минимален праг на фреквенција за генерирани кластери.



Слика бр. 30 Излезни резултати со користење на Clusters/N-Grams

Друг алтернативен начин на пребарување на мулти-збор единици е да се најдат лексички пакети кои се еднакви на n-грамите, каде што n може да варира помеѓу два или пет зборови. Неколку програми за корпус анализа ја нудат оваа функција но AntConc вклучува пакети за лексички пребарувања како опција во алатката Word Clusters Tool.

5.2.6 Ограничувања на AntConc

AntConc ги врши сите операции директно на дадениот текст од корпусот. Ова е корисно затоа што корисникот може да го модифицира корпусот за одредени потреби, така што програмата не мора да врши претходна обработка на податоците, како на пример, создавање на индекс. Од друга страна пак, бидејќи AntConc не користи индекси, тој може да работи ефикасно само со корпуси од помал размер. И покрај тоа, еден од главните трендови во корпус лингвистиката во текот на изминативе неколку години е зголемениот интерес за многу мали, високо специјализирани корпуси. Малите корпуси може да се користат за голем број на различни цели.

Повеќето програми за корпус анализа, на корисниците им нудат можност да ги видат врските на терминот за кој пребаруваат во табела, каде што е прикажана фреквенцијата на најчестите зборови на лево или десно од пребарувањето. Некои програми исто така нудат детални статистички податоци во врска со резултатите од корпусот и пребарувањето. Имено, можеби програмите не би требало да вклучуваат статистики, туку да овозможат лесен начин за копирање на резултатите од програмата за подоцна да можат да се вршат табеларни анализи во зависност од моменталната потреба. Во тој поглед, AntConc овозможува лесно копирање на податоците во програма за табеларни пресметки со користење на едноставни кратенки од тастатура.

AntConc е програм за анализа на корпус, едноставен и лесен за употреба кој се покажа како исклучително ефикасна алатка. Вклучува алатки кои се од суштинско значење и кои се неопходни за анализа на корпуси, со интуитивен и едноставен интерфејс како и бесплатна лиценца, кој го прави актуелен за употреба за голем број на корисници.

6. Модел на македонски дигитален јазичен корпус

Покрај претходно наведените и обработени алатки кои се користат во процесот на обработка на природните јазици, во овој труд акцентот ќе го ставиме на едно поинакво решение, односно креирање и дефинирање на сопствен модел на македонски дигитален јазичен корпус.

Она што е клучно кај овој модел, е дека во целост опфаќа процес на дефинирање на еден поинаков дигитален приказ на македонскиот јазик, во кој преку дефинирање на граматичките и синтаксичките правила, прикажуваме даден текст кој што го пребаруваме, е детално разработен од аспект на дефинирање и опис на секој збор поединечно во реченицата.

Овој начин на дефинирање и обработка на корпусот, вклучува еден мал сегмент од целата проблематика на проучување на обработката на природните јазици, а тоа е POS Tagging проблематиката, односно креирање, дефинирање и приспособување на наш, македонски концепт на граматички правила кои се одликуваат само за нашиот јазик. Дефинирањето на ваков модел нè води на исто рамниште со голем број на светски јазици, како англискиот, германскиот, шпанскиот, а српскиот, бугарскиот и хрватскиот од нашите словенски јазици.

6.1 Постапка на креирање и дефинирање на дигиталниот корпус

Начинот на кој е дефиниран и замислен процесот на дефинирање на македонскиот дигитален јазичен корпус е утврден во неколку чекори кои се објаснети во продолжение.

1. Најпрво е започнато со дефинирање и утврдување на идејата за развој на ваков тип на корпус, при што детално се разгледувани и проучувани различни тематики кои сметаме дека би биле од голема важност во целата постапка од дефинирање, планирање и имплементација на идејата.

2. Откако е дефиниран прототипот, следува постапка на анализа на наши и надворешни веќе дефинирани слични форми и примери, за што сметаме дека нивните идеи и процеси би ни помогнале во процесот на креирање на алатката.

3. При деталната анализа, следува постапка на конкретно прецизирање на генералните задачи, преку кои се дефинира текот на функционирање на алатката, на сите важни делови, структура, влезни податоци и крајни резултати.

4. По успешно прецизирање на целата проблематика, следува најсуштинскиот сегмент, а тоа е тестирање на прототипот, при што ќе можат да се утврдат квалитетот на ликвидноста или пак сите оние непредвидени неправилности. Предвидено е тестирање чекор по чекор за да биде успешно тестирањето, и доколку се појави проблем, да може да се интервенира во даден момент.

6.2 Дефинирање на проблемот

Процесот на креирање на македонскиот дигитален јазичен корпус подразбира креирање на алатка која е замислена за секоја реченица како краен резултат да прикажува детален опис на истата, на начин што за секој збор што го содржи истата да се дава опис што всушност претставува тој збор во реченицата, дали именка, глагол, придавка и сл. Постапката е утврдена во неколку чекори кои се објаснети во продолжение.

1. Најпрво се подразбира дефинирање на граматичките правила со кои се одликува македонскиот јазик. Дефинирањето на граматичките правила се дефинира во посебен фајл по строго дефинирана структура, во кој прикажуваме дали еден збор во текстот кој е дефиниран претставува именка, глагол, придавка и сл. Дефинирањето на граматичките правила е многу сензитивен дел кој е многу важен, затоа што крајниот резултат што ќе се прикажува се

заснова и повикува всушност од овие предефинирани правила. Многу е важно да се внимава на родот, дали е множина или еднина и да се препознае во кое време е дефинирана реченицата. Овој детаљ сам по себе укажува на важноста на овој фајл во понатамошниот процес на извршување на целата постапка до прикажување на крајниот резултат.

2. Понатаму, употребата на база на податоци, која е треба да ја наполниме со речник на зборови на македонски јазик. Овој речник се состои од маса на зборови за кои е прикажано какво значење би можеле да имаат во реченициte, дали дадениот збор е глагол, именка, придавка итн.

3. За да може да се користат правилата и базата, потребно е да има текст кој што ќе се споредува. Во нашиот пример е замислено да креираме корпус, кој ќе се состои од текст фајлови. Ваквите фајлови се креирани според потребните стандарди за унифицирање, со што би можеле, без проблем, да се вчитуваат и да се пуштат во понатамошна детална анализа.

4. Откако сите претходно наведени процедури ќе се креираат и прецизно дефинираат, крајниот резултат би бил во дадено поле да постои опција за внесување на даден текст, при што првиот филтер би бил проверка - дали дадената реченица или зборовите се содржат во дефинираниот корпус. Од таму се проверува дали зборовите од зададената реченица се содржат во нашиот дефиниран речник и доколку некој збор недостасува, на излез се прикажува резултат кој ќе ни даде детална анализа на целата реченица врз основа на дефинираните граматички правила, и во случај некој збор да не постои во речникот истиот ќе биде означен како нов збор, и ќе врати кое му е значењето на тој збор во реченицата.

Оваа алатка претставува интересен и едноставен концепт која наоѓа широка примена. Доколку во иднина се продолжи на усовршување, прецизирање и додавање на дополнителни функционалности, би била од големо значење за сите, бидејќи ќе овозможи брз и едноставен пристап, кој во секое време ќе може да одговори на потребите на корисниците.

6.3 Функционални барања

Дефинирањето на овие барања е од суштинско значење во процесот на дефинирање на структурата на оваа алатка, бидејќи овие барања се однесуваат на примената на македонскиот дигитален јазичен корпус и на целокупната структура и сите видови на врски и интеракции кои треба да се извршуваат. Овие функционални барања се основа за развивање на каква било алатка или апликација. Во нашиот случај, функционалните барања можеме да ги поделиме на:

- Дефинирање на граматичките правила кои се неопходни при генерирање на POS Tagger анализа;
- Креирање на речник на зборови за македонскиот јазик;
- Креирање на јазичен корпус;
- Функционално барање за можност за пребарување и прикажување на крајниот резултат од целокупната анализа

6.4 Приказ на практичното решение

Практичното решение на овој магистерски труд претставува софтвер кој се состои од речник на македонскиот јазик, интегрирани библиотеки, развиени функции и останати пакети. Ова е веќе дефиниран модел, тоа е моделот Stanford POS Tagger. Овој модел нуди низа од датотеки кои се неопходни за анализа на јазиците. Со ваквите, веќе дефинирани библиотеки овозможува креирање на неопходните фајлови во коишто се дефинирани поставките со кои се одликува секој јазик.

Овој модел е достапен во повеќе програмски јазици како Python, Php, Java и други. Дефинирањето на логиката, дизајнот и нивното поврзување во овој случај е развиено на платформата Java.

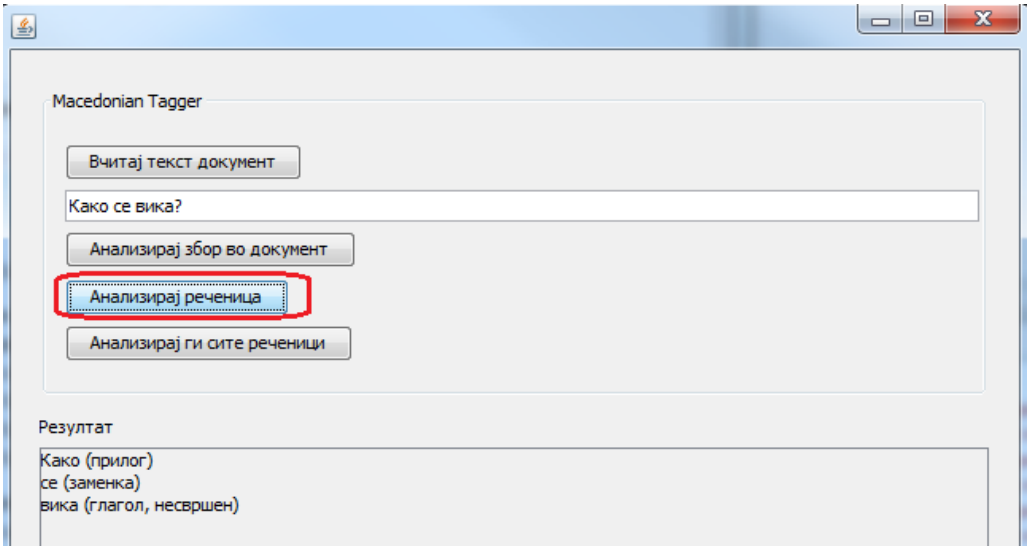
Софтверот на дигиталниот македонски корпус, всушност се состои од едноставен дизајн кој вклучува неколку копчиња за управување со програмот, поле за внесување и простор за приказ на резултатите.

Опфатени се неколку начини на обработка на податоците и тоа:

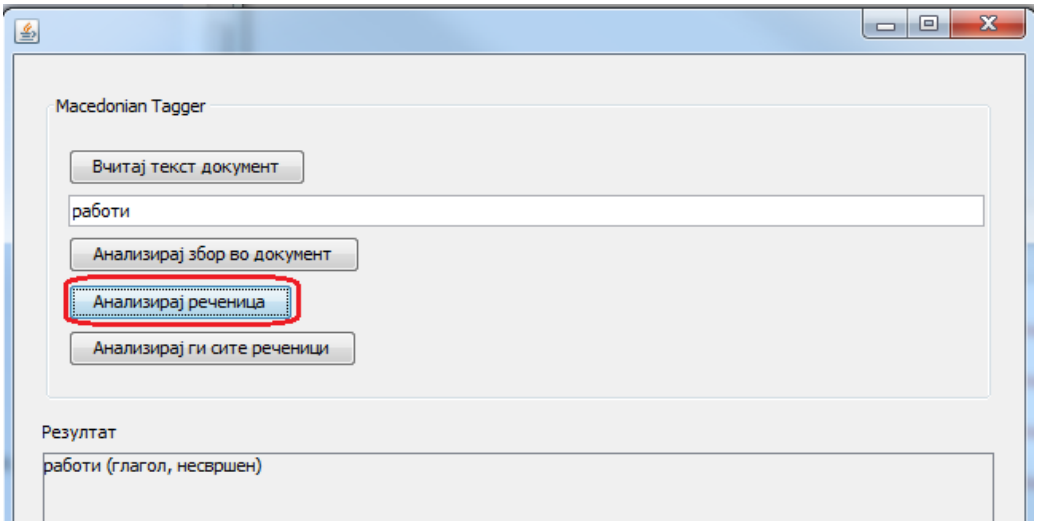
- Анализирање на реченица или збор;
- Проверка и анализа во даден документ;
- Проверка и анализа на целиот документ.

Анализирање на реченица или збор

Овој дел опфаќа анализа на цела реченица или на само еден збор. Тоа значи, внесување на саканиот збор или речени во полето за внесување и со клик на копчето „Анализирај ја реченица“ се прикажуваат резултатите обработени за секој збор поединечно. Сликите, подолу, претставуваат приказ на анализа на реченица или збор.



Слика бр. 31. Анализа на реченица

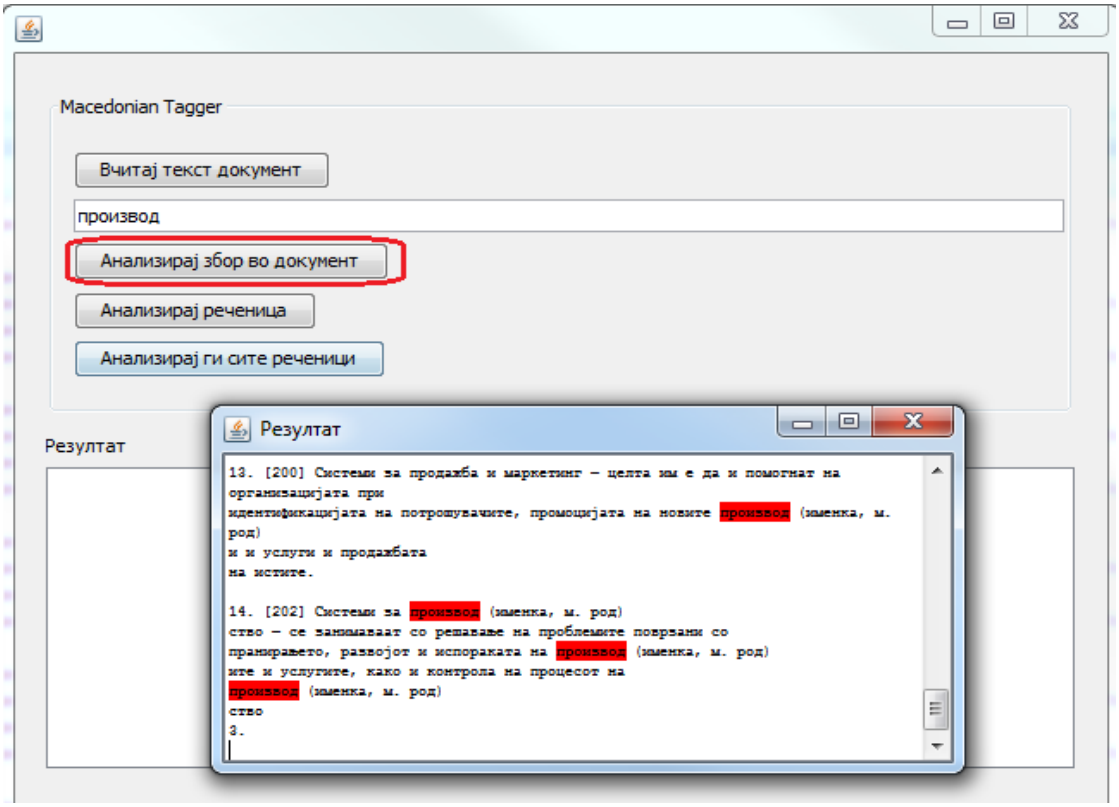


Слика бр. 32. Анализа на збор

Анализирањето на реченица или само на еден збор се одвива на тој начин што програмот го проверува секој збор дали постои во речникот, кој претставува база на зборови на македонскиот јазик, и потоа врз база на дефинирани правила, како резултат, прикажува опис и значење за секој збор.

Проверка и анализа во документ

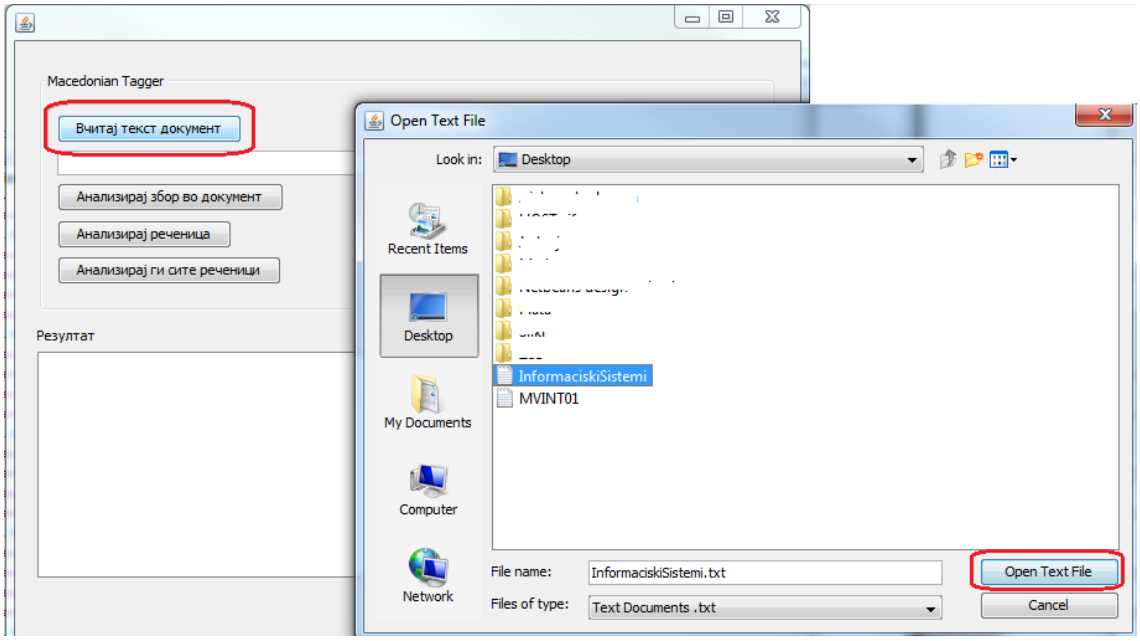
Овој дел опфаќа пребарување и анализа по даден збор. Тоа значи, внесување на саканиот збор во полето за внесување и со клик на копчето „Анализирај збор во документ“ се прикажуваат обработените резултати за пребаруваниот збор. За да се изврши ваквата обработка по збор, неопходно е претходно внесување на текстуален документ во кој сакаме да ни се направи потребната анализа. Импортирањето на текстуалниот документ се врши со клик на копчето „Вчитај текст документ“ при што се отвора нов прозорец за одбирање на локацијата каде што се наоѓа документот. Овој документ треба да биде во формат на .txt фајл. Сликата подолу претставува приказ на анализа на збор од одбран документ.



Слика бр. 33 Проверка и анализа на збор од избран документ

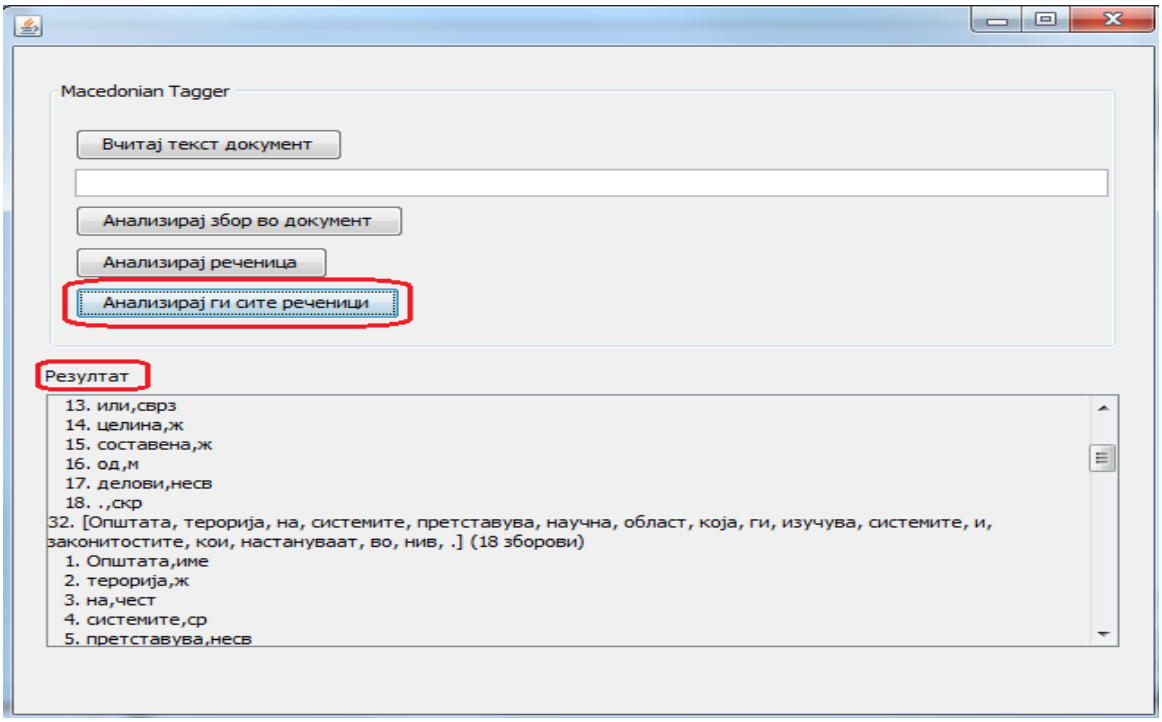
Проверка и анализа на целиот документ

Покрај претходно наведените опции, апликацијата нуди можност и за анализа на целиот документ кој што ќе биде одбран за проверка. Анализата се одвива на тој начин што се врши поделба по реченици и потоа се дава опис за значењето на секој збор во нив. Сликата подолу претставува пример за таков опис.



Слика бр. 34 Анализа на даден документ

Откако ќе се одбере документот, со клик на „Анализирај ги сите реченици“, како резултат ќе врати обработени и анализирани реченици кои се во документот.



Слика бр. 35 Анализа на даден документ

6.5 Технологии кои се користат при градење на корпусот

Целиот модел на дефинирање на македонски POS Tagger всушност се одликува преку употреба на веќе достапен дефиниран модел, Stanford POS Tagger – от кој е базиран на платформата Java.

Stanford POS Tagger – от во себе вклучува предефинирани библиотеки, модели, процеси и сл. кои се унифицирани и важат за еден од најдобрите вакви модели, за што потврдува и фактот дека на ист начин се креирани и голем број на странски јазици. Овој Tagger сам по себе се одликува со стандарди и спецификации по кои треба да се водиме и кои се неопходни за креирање на еден корпус. Во себе вклучува меѓусебно поврзани голем број на модели и библиотеки кои се зависни и потребни за постигнување на саканиот резултат. За дефинирање на сопствен корпус POS Tagger – от има поставено правила кои се унифицирани за кој било јазик.

Според горенаведеното, со практична примена на функционалните барања во прецизирана архитектура, и сето ова приспособено во согласност на стандардите кои се неопходни за реализација на еден ваков проект, на крај успеавме да креираме прототип на ваков дигитален корпус, приспособен според правилата на македонскиот јазик. Креирана е една целина, концепт на дигитален македонски јазик кој овозможува внесување на текст, анализа на тој текст и прикажување на краен резултат. Понатамошната надградба и усовршувањето во иднина би овозможило и имплементација на ваквиот дигитален корпус кој би бил од големо значење за голем број корисници.

7. Заклучок

Секојдневниот напредок на информатичките технологии и претставувањето на сѐ посовремени алатки кои се појавуваат како многу практични во сите сфери, ги поттикна сетилата на сите луѓе за развивање на современи системи кои ќе го олеснат секојдневниот живот. Ваквите технологии кои овозможуваат голем број на функционалности, го нудат токму тоа што пред неколку деменции се сметало за невозможно или за многу тешко за спроведување.

Гледајќи од аспект на лингвистиката и говорните јазици воопшто, веќе долго време се развија апетитите на голем број стручни лица кои сакаат да ја заживеат оваа област. Па така, како и во сите останати области и во лингвистиката, примената на ваквите современи информатички технологии се покажа како многу ефективна. Се овозможи развивање на голем број апликации кои вклучуваат зборови и поими од секојдневниот живот во јазичната употреба.

Дигиталната ера на лингвистиката се состои од електронски корпус чија големина може да биде различна и претставува збир од текстови од одреден јазик кои се складираат, чуваат и обработуваат електронски. Типовите на корпуси можат да бидат различни, како еднојазични така и повеќејазични, со што се дава можност за вршење на различни статистики, анализи и проверки кои се потребни.

Во поглед на ова, современото информатичко општество разви голем број на алатки и системи за обработка на природните јазици. Се овозможи креирање на дигитални лексикони, речници и останати збирки на текстови кои претставуваат основа за понатамошна лексичка анализа.

Напредокот на компјутерската технологија додаде нова димензија во лингвистиката. Се разви корпус лингвистиката како ново поле во поглед на истражувањата на јазикот и кулминираше кон развивање на компјутерска и применета лингвистика со вклучување на компјутерска технологија. Сето ова придонесе , корпус лингвистиката да еволуира како една од најперспективните

области во ова поле. Затоа развивањето на обработката на природните јазици е многу важен сегмент, бидејќи овозможи заживување на голем број на јазици, дијалекти и сл.

8. Користена литература

Bowker, L. and Pearson, J. 2002. Working with Specialized Language: A Practical Guide to Using Corpora. London/New York: Routledge.

Bird, Steven, and Mark Liberman. 2001. A formal framework for linguistic annotation. Speech Communication 33.23–60.

Collins, M. Language Modeling (Natural Language Processing Course). Columbia University

Collins, M. Tagging Problems, and Hidden Markov Models (Natural Language Processing Course). Columbia University

Collins, M. Parsing, and Context-Free Grammars (Natural Language Processing Course). University of Columbia

Collins, M. Markov Processes (Natural Language Processing Course). University of Columbia

Collins, M. Trigram Language Models (Natural Language Processing Course). University of Columbia

Collins, M. The Tagging problem (Natural Language Processing Course). University of Columbia

Conrad, Susan. 2000. Will corpus linguistics revolutionize grammar teaching in the 21st century? TESOL Quarterly 34.548–60.

Coniam, David. 2004. Concordancing oneself: Constructing individual textual profiles. International Journal of Corpus Linguistics, 9(2), 271–298.

Conrad, Susan. 2000. Will corpus linguistics revolutionize grammar teaching in the 21st century? TESOL Quarterly 34.548–60.

Gera, R. Bennett. 2010. Using Corpora in the Language Learning Classrom: Corpus Linguistics for Teachers. Michigan

Gries, Th. Stefan, 2009. What is Corpus Linguistics? University of California

Hundt, Marianne, Nadja Nesselhauf, and Carolin Biewer (eds) 2007. Corpus linguistics and the web. Amsterdam: Rodopi.

Hunston, S. 2006. Corpus Linguistics. University of Birmingham, Birmingham, UK

Kibble, R. 2013. Introduction to Natural Language Processing. University of London

Laurence, A. 2004. AntConc: A Learner and Classroom Friendly, Multi-Platform Corpus Analysis Toolkit, Japan

Lapata,M and Keller, F. 2005. Web-based models for natural language processing. ACM Transactions on Speech and Language Processing.

McEnery, Anthony, Richard Xiao, and Yukio Tono. 2006. Corpus-based language studies: an advanced resource book. London, New York: Routledge.

McEnery, T. and Wilson, A. 2001. Corpus Linguistics. An Introduction. Second edition. Edinburgh: Edinburgh University Press.

Nesselhauf, N. and Tschichold, C. 2002 Collocations in CALL. An investigation of vocabulary-building software for EFL. Computer Assisted Language Learning, 15(3), 251-279.

Nesselhauf, N. 2011. Corpus linguistics: A practical Introduction.

Noguchi, J. 2004. A genre analysis and mini-corpora approach to support professional writing by nonnative English speakers. English Corpus Studies, 11, 101-110.

Ponorac, T: The Basics of Corpus Linguistics, Banja Luka

Semino E and Short M (2004). Corpus stylistics: speech, writing and thought presentation in a corpus of English writing. London: Routledge.

Sun, Y. C. & Wang, L. Y. 2003. Concordancers in the EFL Classroom: Cognitive Approaches and Collocation Difficulty. Computer Assisted Language Learning, 16 (1), p. 83-94.

Vinci, L and Curran, R. James. 2006. Web Text Corpus for Natural Language Processing. University of Sydney, Australia

Winfred, P. 2006. Introduction to Natural Language Processing

http://borel.slu.edu/crubadan/

http://www.auburn.edu/~mitrege/cv-big.html

http://www.makedonski.info/

http://www.ahds.ac.uk/creating/guides/linguistic-corpora/appendix.htm