تاثیر زبان مغولی بر زبان فارسی

استاد: سرکار خانم لیلا کرمی

دانشجویان: مهدی وجهی - سید علیرضا میرشفیعی - امیرحسین میرمحمدی

مقدمه

یکی از ارکان فرهنگ هر کشوری ، زبان آن کشور است که در طول تاریخ دستخوش تغییرات کوچک و بزرگ شده است ولی در هر صورت میراثی است که از گذشتگان برای آیندگان باقی مانده است و یکی از ابعاد هویتی هر ملیتی هست.

حال میزان تاثیر اتفاقات مختلف تاریخی در زبان یک کشور ، متفاوت خواهد بود و در این مسیر جنگها یکی از راههای اصلی تعامل فرهنگی بین دو کشور می باشد ، که شامل سنت ها ، اعتقادات ،روش زندگی، تغذیه و البته زبان آن دو می باشد و نسل آینده هر کشوری زبانی دارد که نمی داند کدام قسمت آن زبان اصلی اوست و کدام قسمت زبان وارداتی ملیتی دیگر است.

در این پژوهش برآنیم تا ببینیم واژگان مغولی بعد از حمله مغول به ایران تا چه حد در آثار نوشتاری بزرگان ادب وارد شده است.

سوالات و اهداف پژوهش

هدف این پژوهش پاسخ به سوالات ذیل است:

۱. چه تعداد واژگان مغولی قبل ، بعد و در حین حمله مغول در آثار برخی نویسندگان و شعرا وجود داشته است؟
۲. حمله مغول چه تاثیری بر تعداد این واژگان گذاشته است؟

چکیده:

تا بحال مقالات و نوشته های فراوانی درباره حمله مغول و آسیب های آن بر کشور نوشته شده است.

اما ما در این پژوهش برآنیم تا به طور مختصر تاثیر این اتفاق را در ورود واژگان مغولی به اشعار و نوشته های اشخاص بزرگ ادبی بررسی کنیم. از این رو نویسندگانی را مورد بررسی قرار دادیم و نتیجه آن شد که آمار استفاده واژگان مغولی در آثار ادبی قبل از حمله مغول ناجیز و سپس با سیری صعودی در حین حمله بالا رفته و دوباره پس از این دوره با شیبی ملایم تر نزول پیدا کرده است.

کلیدواژه:

زبان فارسی – زبان مغولی – حمله مغول ها- هجویری – کیکاووس – مولوی – جامی – زاکانی

مستندات فنی:

برای شروع مواردی که باید انجام شوند را فهرست می کنیم:

۱. پیدا کردن فایل متنی کتب و واژه نامه مغولی
۲. تبدیل فایل های متنی به فرمت خوانا برای برنامه (مانند txt)
۳. نوشتن برنامه با قابلیت های زیر:
   - باز کردن و دریافت متن از فایل و واژه نامه
   - جدا کردن کلمات از یکدیگر (Word Tokenizing)
   - بررسی ریشه کلمات فایل و مقایسه با واژه نامه
     - ❖ تبدیل کلمات به ریشه ها (کتابهایش ← کتاب، می خوردیم ← خور)
     - ❖ مقایسه لغات و شمارش آنها
     - ❖ خروجی نتیجه خام
۴. دادن ورودی ها (فایل های کتب، واژه نامه مغولی و کلمات ایست) به برنامه و دریافت خروجی
۵. تحلیل خروجی ها

پیدا کردن فایل کتب و نوشته های شاعران

سایت منبع باز (Open source) گنجور[1] که به عنوان مرجع انتخاب شده وارد می شویم. در بخش کتابخانه گنجور می توانیم کتاب ها را با فرمت epub دانلود کنیم (تصویر ۱).

برای مطالعهٔ محتوای سایت گنجور (و بیشتر از آن) بر روی کامپیوتر و گوشیهای هوشمند بدون نیاز به اینترنت -با استفاده از نرم‌افزارهای کتابخوان با قابلیت نمایش کتابهای با پسوند epub- فایل آثار شاعر مورد نظر را از فهرست زیر دریافت کنید.

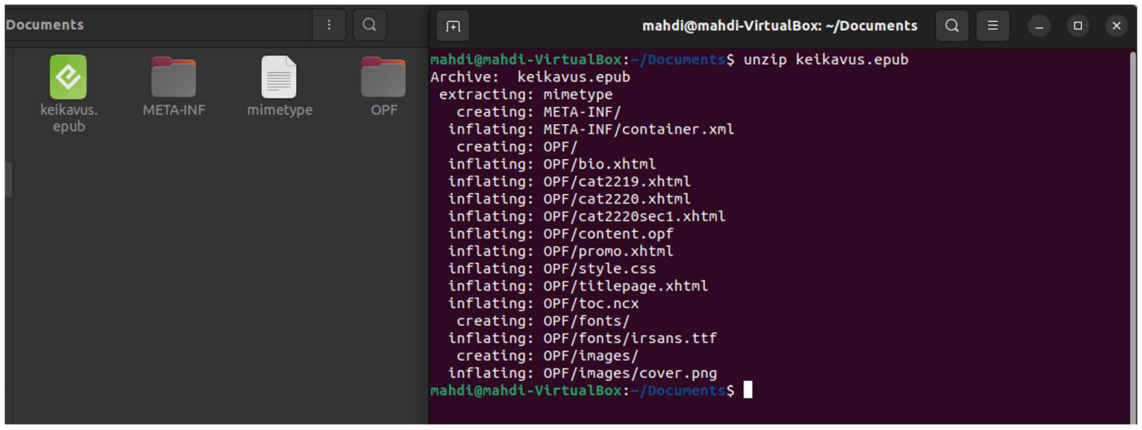
جهت مشاهدهٔ فهرستی از نرم‌افزارهای کامپیوتری یا فایل اجرا بر روی انواع گوشیهای تلفن همراه برای باز کردن فایلهای epub این نوشته را مطالعه بفرمایید.

ردیف	شاعر	اندازهٔ فایل	دریافت
۱	ابن حسام خوسفی	۱۵۱ کیلوبایت	دریافت
۲	ابن یمین	۷۳۴ کیلوبایت	دریافت
۳	ابوالحسن فراهانی	۱۷۴ کیلوبایت	دریافت
۴	ابوالفرج رونی	۱۶۱ کیلوبایت	دریافت
۵	ابوسعید ابوالخیر	۱۷۳ کیلوبایت	دریافت
۶	ابوعلی عثمانی	۲۹۰ کیلوبایت	دریافت
۷	اثیر اخسیکتی	۲۵۰ کیلوبایت	دریافت
۸	احمد پروین	۱۰۷ کیلوبایت	دریافت
۹	احمد شاملو	۳۰۲ کیلوبایت	دریافت
۱۰	ادیب الممالک	۷۲۸ کیلوبایت	دریافت
۱۱	ادیب صابر	۳۳۱ کیلوبایت	دریافت
۱۲	ازرقی هروی	۱۸۱ کیلوبایت	دریافت
۱۳	اسدی توسی	۳۵۰ کیلوبایت	دریافت
۱۴	اسیر شهرستانی	۳۶۹ کیلوبایت	دریافت
۱۵	اسیری لاهیجی	۳۸۸ کیلوبایت	دریافت
۱۶	افسر کرمانی	۲۵۶ کیلوبایت	دریافت
۱۷	افسرالملوک عاملی	۱۴۶ کیلوبایت	دریافت
۱۸	اقبال لاهوری	۴۱۶ کیلوبایت	دریافت
۱۹	الهامی کرمانشاهی	۶۳۲ کیلوبایت	دریافت
۲۰	الیار	۱۹۹ کیلوبایت	دریافت
۲۱	امام خمینی	۱۵۳ کیلوبایت	دریافت
۲۲	امامی هروی	۱۵۳ کیلوبایت	دریافت
۲۳	امیر شاهی	۱۲۵ کیلوبایت	دریافت
۲۴	امیر معزی	۸۳۰ کیلوبایت	دریافت

تصویر ۱-نمایی از سایت کتابخانه ی گنجور

۱.«گنجور» مجموعه‌ای تحت وب از آثار سخنسرایان پارسی‌گو است که با استفاده از آن می‌توان ضمن مرور به تفکیک نام شاعر و نام آثار او، در بین آثار جستجو کرد. این مجموعه در ابتدا به کمک نرم‌افزار مدیریت محتوای وردپرس راه‌اندازی شده بود و از سال ۱۴۰۰ به بعد به پشتوانه نرم‌افزار اختصاصی بازمتن خود کار می‌کند.

سپس فایل را از حالت کتاب الکترونیک (epub) خارج می کنیم.



تصویر۲ – نمای متحوای باز شده کتاب نمونه (کیکاووس) با فرمت epub

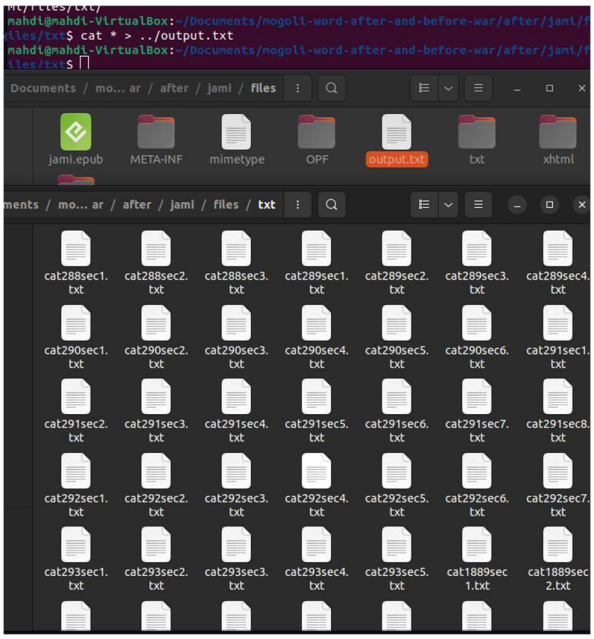
با بررسی فایل های باز شده ی epub متوجه می شویم، فایل های در پوشه OPF به شکل *sec*.xhtml حاوی محتوای کتاب ما هستند.

حال باید فایل های xhtml را به فرمت خوانا تری برای بررسی (همچون txt) در برنامه تبدیل کنیم.

سایت ها و برنامه های مختلفی برای تبدیل xhtml به txt وجود دارد. ما از aspose استفاده کردیم. هر فایل xhtml به یک فایل txt تبدیل می شود.

سپس با ادغام فایل های txt، همه را به یک فایل تبدیل کردیم. با دستور زیر می توانیم این کار را انجام دهیم. (فرض شده که تمامی فایل ها در یک پوشه هستند و هیچ فایل دیگری در آن پوشه نیست.) (تصویر ۳)

```
cat * > your_address/file_name.txt
```



تصویر۳ – نمای تبدیل فایل های txt به یک فایل txt

درنهایت یک فایل را برای دادن به برنامه آماده می شود.

نوشتن برنامه

برای نوشتن برنامه از زبان پایتون و کتابخانه ی wordcloud_fa که متشکل از کتابخانه های wordcloud، hazm و nltk است استفاده کردیم. ابتدا توضیحات مختصری درباره ی کتابخانه ها می دهیم.

کتابخانه های hazm و nltk به ترتیب برای پردازش زبان های فارسی و انگلیسی کاربرد دارند.

پردازش زبان طبیعی (Natural Language Processing) که به اختصار NLP نیز نامیده می‌شود، روشی است برای درک زبان انسانی برای رایانه؛ این علم یکی شاخه‌های دانش هوش مصنوعی محسوب می‌شود و به رایانه‌ها کمک می‌کند تا با آگاهی از چگونگی استفاده بشر از زبان، زبان انسانی را درک کند. پردازش زبان طبیعی یک دانش پیچیده و دشوار است. این شاخه از هوش مصنوعی خود به دو شاخه دیگر یعنی NLG و NLU تقسیم می شوند.

تصویر۴ – ارتباط بین NLP، NLG و NLU با یکدیگر

Natural Language Understanding یا به اختصار NLU به فارسی درک زبان طبیعی بر درک خواندن ماشینی از طریق قواعد دستور زبانی و موضوع تمرکز دارد و ماشین‌ها را قادر می‌سازد تا معنای مورد نظر یک جمله را تعیین کنند. Natural Language Generation یا به اختصار NLG به فارسی تولید زبان طبیعی بر تولید متن یا ساخت متن به زبان انگلیسی یا زبان‌های دیگر توسط یک ماشین و بر اساس یک مجموعه داده معین تمرکز می‌کند (تصویر ۴).

در جدول زیر کتابخانه های معروف NLP آورده شده اند (جدول ۱).

License	Project link	Other Valuable Tools/Properties	Persian Language Supported	Word Embeddings	Text Categorizer/ Sentiment Analysis Semantic Role Labeler (SRL)	Coreference	Dependency Parsing	Shallow Parser/ Chunker	Named Entity Recognition (NER)	POS Tagging	Stemming/ Lemmatization	Tokenizer/ Sentence Splitter	Github Stars	Community Support (Stackoverflow)	Programming languages	Developer, Initial release	Start Year- Last Update	Name
Apache License, V2.0;	http://www.nltk.org/	Get various text corpus (data)	Hazm		✓			✓	✓	✓	✓	✓	7.5k	15.2k	Python	The University of Pennsylvania	2001-2018	Natural Language Toolkit (NLTK)
http://textblob.readthedocs.io/en/dev/license.html	http://textblob.readthedocs.io/en/dev/	Noun phrase extraction WordNet integration Spelling correction			✓		✓			✓	✓	✓	6k	838	Python	Steven Loria	2013-2018	TextBlob
MIT License	https://spacy.io/	Support for 7+ languages Convenient string-to-hash mapping Pre-trained word vectors Visualizers for syntax and NER		✓			✓		✓	✓	✓	✓	12.4k	2.3k	Python	Explosion AI	2016-2018	SpaCy
GNU LGPLv2.1 license	https://radimrehurek.com/gensim/	Language & Topic Modelling (TFIDF, LDA, LSI, ...) Word2 Representation (Word2Vec, FastText, ...)		✓	✓							✓	8.7k	3.5k	Python	RaRe Technologies	2009-2019	Gensim
Apache License, Version 2.0	https://opennlp.apache.org/	Language detector tool UIMA integration			✓		✓	✓	✓	✓	✓	✓	1k	1.9k	Java	Apache Software Foundation	2004-2018	Apache OpenNLP
GNU General Public License	https://stanfordnlp.github.io/CoreNLP/	Support for 7+ languages open information extraction bootstrapped pattern learning			✓		✓	✓	✓	✓	✓	✓	6k	1.5k	Java (Python, C# & ... API)	The Stanford NLP Group	2010-2019	Stanford CoreNLP
Apache License, V2.0;	https://stanfordnlp.github.io/stanfordnlp/	53 (human) languages supported The modules are built on top of PyTorch	✓				✓			✓	✓	✓	3.5k	256	Python	The Stanford NLP Group	2018-2019	StanfordNLP
University Agreement	https://cogcomp.org/page/software_view/Curator	Illinois Wikifier Text Similarity relation-extraction				✓	✓	✓	✓	✓	✓	✓	425	16	Java (Python API)	Cognitive Computation Group	2001-2019	Illinois NLP Curator
3SD	https://www.clips.uantwerpen.be/pattern	Get various text corpus (data) API Web crawler, HTML DOM parser Interface to WordNet Topics model(LDA) & Clustering Graph centrality and visualization			✓		✓				✓	✓	6.8k	235	Python	T. De Smedt & W. Daelemans	2012-2018	Pattern
GPLv3 license	http://polyglot-nlp.com/	Multilingual Tools Transliteration Language detection	✓	✓	✓				✓	✓	✓	✓	1.2k	2k	Python	Rami Al-Rfou	2014-2016	Polyglot
Apache License, V2.0;	https://opensource.google.com/projects/syntaxnet	Trained models for 40 languages A TensorFlow based framework	✓				✓			✓	✓	✓	49k	903	C++, Python	Google	2016 - 2018	SyntaxNet
Affero GNU General Public License	http://nlp.lsi.upc.edu/freeling/	More than 15 (human) languages supported Word Sense Disambiguation Semantic graph extraction				✓	✓	✓	✓	✓	✓	✓	1k	155	C++ (Java, Python, and Perl API)	TALP Research Center, Universitat Politècnica de Catalunya	2009-2018	FreeLing
the GNU licenses and other	https://gate.ac.uk/	12 (human) languages supported Gazetteer Plugins for Weka, LibSVM, ... Managing ontologies like WordNet Annotate on text (JAPE)				✓			✓	✓		✓	47	50	Java – provides a GUI	GATE research team, University of Sheffield	1995-2018	General Architecture for Text Engineering (GATE) – ANNIE Package
GPL	https://www.clips.uantwerpen.be/pages/MBSP	Relation Finding and Prepositional Phrase Attachment						✓		✓	✓	✓	62	14	Python	Vincent Van Asch, Tom De Smedt	2005-2011	Memory-Based Shallow Parser (MBSP)
Apache License, Version 2.0	https://nlp.johnsonwlabs.com/	English, Italian Languages support Spell Checker Built on top of Apache Spark		✓	✓		✓	✓	✓	✓	✓	✓	554	92	Scala, Python & Java	Apache Software Foundation	2017-2019	Spark-NLP
AGPL	https://rapidminer.com/	Sentiment Analysis & Classification by Taxonomy Keyword, Entity and Concept Extraction Article Extraction Language Detection			✓							✓	281	1.2k	RapidMiner provides a GUI to design and execute analytical workflows	RapidMiner	2006	RapidMiner (AYLIEN)
Common Public License	http://mallet.cs.umass.edu/	Topic Modeling (LDA) clustering Information Extraction Sequence Prediction Models			✓							✓	685	814	Java	Andrew Kachites McCallum, University of Massachusetts Amherst	2002-2019	MAchine Learning for LanguagE Toolkit (MALLET)

جدول ۱ – جدول مقایسه کتابخانه های مشهور NLP

از کتابخانه های فارسی معروف در این حوزه می توان به Hazm، Stanford، farsiNLPTools و parsivar اشاره کرد.

کتابخانه NLTK (Natural Language Toolkit) یکی از جامع‌ترین و قدیمی‌ترین کتابخانه‌های پردازش زبان طبیعی در پایتون است. این کتابخانه پایه و استانداردی برای کتابخانه‌های پردازش متن محسوب شده و برای کاربردهای پژوهشی فوق‌العاده است. یکی از ویژگی‌های خوب این کتابخانه امکان اتصال به پیکره‌های مختلف متنی است.

کتابخانه هضم با استفاده از کتابخانه NLTK در سال ۱۳۹۲ توسط دانشجویان دانشگاه علم و صنعت برای پردازش زبان فارسی توسعه داده‌شده است. در ابتدا هضم تنها برای زبان پایتون و سیستم‌عامل لینوکس طراحی‌شده بود، اما اکنون برای زبان‌های جاوا و C# نیز قابل‌استفاده است. نسخه جاوایی این کتابخانه با عنوان JHazm منتشرشده است. ازجمله ویژگی‌های این کتابخانه مرتب کردن متون، بخش‌بندی، ریشه‌یابی کلمات، تحلیل صرفی و تجزیه نحوی جملات، سازگاری با NLTK و پشتیبانی از نسخه‌های ۲ و ۳ پایتون هست.

همچنین میزان دقت این کتابخانه در تحلیل متون در جدول زیر آمده است (جدول ۲).

Module name	Precision
Lemmatizer (ریشه یابی)	89.9%
Chunker (تشخیص گروه های اسمی و فعلی)	89.9%
POSTagger (تحلیل نقش کلمات)	97.1%
DependencyParser (درخت وابستگی)	97.1%

جدول ۲ – میزان دقت کتابخانه هضم

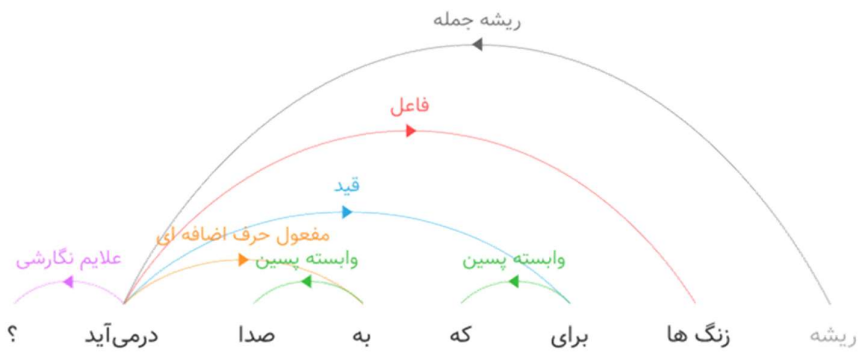
در ادامه تعدادی دستورات کاربردی هضم آورده شده:

```
>>> normalizer = Normalizer()
>>> normalizer.normalize ( 'اصلاح نویسه ها و استفاده از نیم فاصله پردازش را آسان می کند' )
'اصلاح نویسه ها و استفاده از نیم فاصله پردازش را آسان می کند'
>>> sent_tokenize ( 'ما هم برای وصل کردن آمدیم! ولی برای پردازش، جدا بهتر نیست؟' )
[ 'ما هم برای وصل کردن آمدیم!' , 'ولی برای پردازش، جدا بهتر نیست؟' ]
>>> word_tokenize ( 'ولی برای پردازش، جدا بهتر نیست؟' )
[ 'ولی', 'برای', 'پردازش', '،', 'جدا', 'بهتر', 'نیست', '؟' ]
>>> stemmer = Stemmer()
>>> stemmer.stem ( 'کتاب ها' )
'کتاب'
>>> lemmatizer = Lemmatizer()
>>> lemmatizer.lemmatize ( 'می روم' )
```

```
'رفت#رو'
>>> tagger = POSTagger ( model = 'resources/postagger.model' )
>>> tagger.tag ( word_tokenize( 'ما بسیار کتاب می‌خوانیم' ))
[ ( 'ما' , 'PRO' ), ( 'بسیار' , 'ADV' ), ( 'کتاب' , 'N' ), ( 'می خوانیم' , 'V' ) ]
>>> chunker = Chunker( model= 'resources/chunker.model' )
>>> tagged = tagger.tag ( word_tokenize( 'کتاب خواندن را دوست داریم' ))
>>> tree2brackets( chunker.parse (tagged))
'[کتاب خواندن NP] [را POSTP] [دوست داریم VP]'
>>> parser = DependencyParser (tagger = tagger, lemmatizer = lemmatizer )
```

همچنین این کتابخانه حتی توانایی رسم درخت تحلیل نحوی جملات را دارد. به عنوان مثال دستور زیر درخت تحلیل نحوی جمله " زنگ‌ها برای که به صدا درمی‌آید؟" را رسم می کند (تصویر ۵)

```
>>> parser.parse (  word_tokenize( 'زنگ‌ها برای که به صدا درمی‌آید؟' ))
<DependencyGraph with 8 nodes>
```



تصویر ۵ – نمودار تحلیل نحوی

کتابخانه wordcloud_fa برای تولید ابر کلمات در زبان فارسی کاربرد دارد. ابر کلمات به تصویری می گویند که در آن لغات متنی به نمایش در می آید به صورتی که در آن کلمه ای که بزرگ تر است بیشتر تکرار شده اند. در تصویر زیر می توانید ابر کلمات مجموعه آثار مولانا مشاهده کنید (تصویر ۶).

تصویر ۶ – ابر کلمات مجموعه آثار مولانا

از آنجایی که در این کتابخانه واژه ها از هم جدا و شمرده می شوند ما می توانیم با تغییراتی در کد های آن، خروجی مد نظر خود را ستخراج کنیم و با ایجاد شرطی مغولی بودن آن را نیز بررسی کنیم.

موضوع دیگری که در این برنامه اهمیت دارد داشتن فهرستی از کلمات ایست (stop word) است. کلمه ایست به کلماتی از جمله از، به، است، گفت، یا، ای و... می گویند که مفهوم خاصی ندارند ویا صرفا برای تکمیل جمله به کار می روند و اهمیت معنایی زیادی ندارند. به همین علت و برای بهتر شدن تحلیل، لازم است این لغات از متن حدف شوند. با جست و جویی در اینترنت به فهرستی از این کلمات رسیدیم. این فهرست شامل ۱۹۰۰ کلمه بود. درهنگام بررسی و تحلیل تعدادی لغت نیز به آن اضافه کردیم. لازم به ذکر است که این فهرست کاربرد عمومی دارد. بنابراین برای تحلیل متون مختلف نیاز به تغییر و اختصاصی کردن این فهرست می باشد. به عنوان مثال فهرست کلمات ایست برای متون عادی، مقاله، نظرات مشتریان، موجود در شبکه های اجتماعی و... متفاوت می باشد. بنابراین برای تحلیل و بررسی دقیق تر اشعار و نوشته های ادیبان لازم است که این لغات تک به تک بررسی شوند و در صورت نیاز کم ویا اضافه شوند اما به دلیل کمبود وقت این بررسی انجام نشده است.

درنهایت بعد از جدا کردن و شمارش لغات برای خروجی خام برنامه، فرمت CSV انتخاب شد. «CSV» مخفف عبارت « Comma Separated Values» و به معنای «مقادیر جدا شده با ویرگول» است. اینگونه فایل‌ها در واقع نوعی فایل خام متنی هستند که شامل لیستی از داده‌ها می‌شوند. به صورت رایج فایل‌های یاد شده برای جابه‌جایی داده‌ها بین نرم‌افزارهای متفاوت مورد استفاده قرار می‌گیرند. برای مثال، پایگاه‌های داده و نرم‌افزارهایی از جمله اکسل معمولا از فایل‌های CSV پشتیبانی می‌کنند. همچین این فایل ها سازگاری بسیار خوبی با زبان پایتون دارند و به آسانی نیز قابل تولید هستند.

با توجه به توضیحات بالا، در ارتباط با پروژه "بررسی نفوذ زبان مغولی قبل و بعد از حمله مغول به ایران" به صورت زیر برنامه نویسی و پیاده سازی شد.

ابتدا برای نصب wordcloud_fa (که شامل تمامی کتابخانه های مورد نیازمان است) دستور زیر را در خط فرمان وارد می کنیم:

```
pip install wordcloud-fa
```

سپس طبق توضیحات موجود در اسناد کتابخانه کد های اولیه را می نویسیم. حاصل کد main.py، تصویر ابر کلمات است. که مشابه آن را در تصویر ۶ مشاهده شد.

main.py

```
from wordcloud_fa import WordCloudFa

wodcloud = WordCloudFa(no_reshape=True, persian_normalize=True,
include_numbers=False, collocations=False, width=1600, height=800,
background_color="white")
text = ""
with open('book.txt', 'r') as file:
    text = file.read()
wc = wodcloud.generate(text)
image = wc.to_image()
image.show()
image.save('output.png')
```

حال لازم است فایل های کتابخانه ی wordcloud_fa را طبق نیاز خود تغییر بدهیم. کتابخانه ی wordcloud_fa برای codespase github که سیستم عامل اوبونتو[1] است در آدرس زیرقابل دستیابی است:

```
/usr/local/python/3.10.4/lib/python3.10/site-packages/wordcloud_fa/WordCloudFa.py
```

بعد از باز کردن فایل مذکور برای اینکه با استفاده از تمامی ابزار های هضم، Header File را به صورت زیر تغییر می دهیم:

```
# from hazm import Normalizer, word_tokenize
from hazm import *
```

با بررسی ساختار فایل متوجه می شویم لغات در شئی (Class) و تابع (def) زیر جدا می شوند (tokenize).

```
class WordCloudFa(WordCloud):
    def process_text(self, text: str) -> Dict[str, int]
```

۱. اوبونتو (به انگلیسی: Ubuntu) یک توزیع گنو/لینوکس بر مبنای دبیان است.

با ایجاد دو ویژگی برنامه را به صورتی که می خواهیم تغییر میدهیم.

الف) تبدیل کلمات به ریشه ی آنها

ب) فیلتر کردن واژگان و جدا سازی واژگان مغولی

برای ایجاد ویژگی های ذکر شده تکه کد زیر را اضافه می کنیم.

```
        if self.regexp:
            words = re.findall(self.regexp, text, flags)
        else:
            words = word_tokenize(text)

        ##############added

        #####Stemmer
        root = False
        if root:
            buffer = list()
            for i in words:
                buffer.append(Stemmer().stem(i))
            words = buffer

        #####tokeniz and add mogol dic
        mogoli = True
        if mogoli:
            with open("../../mogholi_dic.txt") as file_dic:
                text = file_dic.read()
                mogoli_words = word_tokenize(text)
            find_mogoli = True
            if find_mogoli:
                buffer = list()
                for i in words:
                    for j in mogoli_words:
                        if i == j :
                            buffer.append(i)
                words = buffer

        ###############
```

با تغییر درستی و نادرستی متغیر های root و mogoli در اول دستورات می توان آنها را فعال یا غیر فعال کرد. سپس برای تنظیم قالب خروجی، در شئی و تابع

```
class WordCloudFa(WordCloud):
    def generate_from_frequencies(self, frequencies: Dict[str, float], max_font_size=None)
```

قطعه کد زیر را اضافه می کنیم:

```
        if self.persian_normalize:
            words = WordCloudFa.normalize_words(words)

        new_frequencies = dict(zip(words, values))

        ############ added
        sort_dic = sorted(new_frequencies.items(), key=lambda x:x[1],
reverse=True)
        for i in sort_dic:
            coma = True
            for j in i:
                print(j, end="")
                if coma:
                    print(",", end="")
                    coma = False
            print()
        #############
```

حاصل کد بالا خروجی ای به شکل فایل CSV است که هم با پایتون و هم با اکسل قابل بررسی است.

سپس فهرست لغات ایست اختصاصی مورد نظر خود را اضافه کنیم برای این منظور به فایل زیر رفته و آن را با فهرست خود جایگزینش می کنیم.

```
/usr/local/python/3.10.4/lib/python3.10/site-packages/wordcloud_fa/stopwords
```

با اجرای برنامه به صورت زیر، خروجی در فایلی با فرمت CSV (مثلا log.csv) ذخیره می شود.

```
Python main.py > log.csv
```

خروجی تفکیک و شمارش کلمات به ۴ حالت (همه کلمات، همه کلمات مغولی، ریشه ی همه کلمات، همه ی کلمات با ریشه مغولی) گرفته شد، اما در نهایت فقط خروجی "همه ی کلمات مغولی" و "همه ی کلمات" مورد ارزیابی قرار گرفت. لازم است بررسی شود که چرا ریشه های کلمات به درستی تحلیل نمی شوند.

دو خروجی CSV هر ادیب که شامل "همه کلمات مغولی" و "همه ی کلمات" می شود را باهم ادغام کرده و به صورت یک فایل اکسل در می آوریم که برای تحلیل آماده شوند. (جدول ۳)

جان	6521		بیم	183					
دل	5247		سعادت	115					
عشق	3352		دوا	100					sum 550073
غزل	3309		افندی	67					
اندر	2892		خان	66					
گر	2646		ایاز	66					sum 1255
آب	2376		یاغی	52					
جهان	2018		قوی	48					
دست	2006		قلاوز	30					0/00228152
رباعی	1992		خاتون	28					sum/sum
چشم	1969		تتق	27					
شب	1494		خاقان	25					
همچو	1481		سغراق	23					
کز	1476		سنجر	23					
زان	1467		قفق	23					
نی	1447		فرما	21					
پر	1432		ترجمان	21					
عقل	1414		چارق	21					
نور	1405		بالش	19					
بهر	1304		تغار	18					
شمس	1274		موران	17					
جمله	1269		یغما	16					
تن	1265		یوش	16					
عالم	1256		اینی	16					
گل	1254		سنجق	15					
آتش	1252		اورا	13					

جدول ۳ – نمایی از ادغام دو فایل خروجی حاصل از کتب مولانا که به صورت یک جدول اکسل درآمده است

تمامی کد ها، ورودی ها و خروجی ها در مخزن گیت هاب مربوطه بارگزاری شده است.

همواره در طی زمان موضوعاتی از این قبیل توسط افرادی متخصص و به صورت جامع مورد پژوهش و بررسی قرار گرفته اند ، اما ما در این پژوهش قصد داریم فقط به بررسی آثار برخی از بزرگان ادبیات بپردازیم که عبارتند از:

هجویری (کشف المحجوب)

عنصرالمعالی کیکاووس (قابوسنامه)

مولوی (دیوان شمس – مثنوی معنوی – فیه ما فیه – مجالس سبعه)

جامی (دیوان اشعار – هفت اورنگ – بهارستان – رساله اربعین)

عبید زاکانی (دیوان اشعار – عشاق نامه – اخلاق الشراف – موش وگربه)

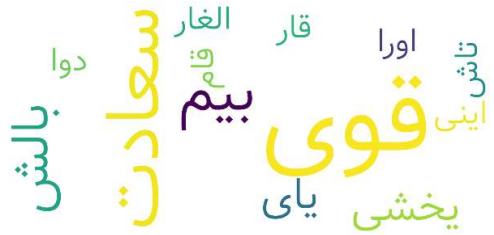
برای بررسی بهتر تاثیر این اتفاق آن را به سه دوره زمانی قبل ، هنگام و بعد از آن تقسیم میکنیم ، هرکدام از این بزرگان مربوط به یکی از سه دوره زمانی مربوط به حمله مغول (۶۱۶-۶۵۴ هـ . ق) می باشند که در ادامه به بررسی هرکدام خواهیم پرداخت.

1- قبل از حمله مغول:

1-1- علی بن عثمان بن علی جلابی هُجویری غَزنَوی (قرن پنجم هـ . ق)

وی نوشتن کتاب کشف المحجوب را حدود ۱۵۰ سال قبل از حمله مغول به ایران کمی پیش از سال ۴۶۵ قمری آغاز کرد.

و تعجبی ندارد که در اثر او تعداد انگشت شماری از کلمات مغولی یافت میشود ، با این حال نگاه کوچکی به این کلمات در غالب یک تصویر می اندازیم.



(کشف المحجوب)

احتمالا در نگاه اول این سوال مانند ما برایتان ایجاد شده است که این کلمات شباهتی به کلمات مغولی ندارند! و باید در پاسخ به شما بگوییم که حق با شماست ، زیرا برخی از واژگان مغولی با شباهت به واژگان فارسی اما در معنایی متفاوت وجود دارند که نمیتوانیم از آن ها صرف نظر کنیم پس اگر در ادامه باز هم از این قشر کلمات دیدید تعجب نکنید.

1-2- عنصرالمعالی کیکاووس بن اسکندر بن قابوس زیاری (۴۱۲ – ۴۸۰ هـ . ق)

وی نصیحت نامه مشهور به قابوسنامه را در سال ۴۷۵ هجری به اتمام رساند و در این اثر تعداد واژگان بسیار کم است.

(قابوسنامه)

با توجه به دو نمونه دیده شده و در نظر گرفتن این موضوع که تنها این کلمات اندک از مجموعه ۵۰۰ واژه مغولی جمع آوری شده توسط ما در این آثار یافت شده اند میتوان نتیجه گرفت که قبل از حمله مغول فارسی زبانان با واژگان مغولی آشنایی نداشته و از آنها استفاده نمی کردند.

حال وقت آن است که کمی در زمان جلوتر برویم و به دوره حمله مغول به ایران برسیم.

2- در حین حمله مغول: (۶۱۶-۶۵۴ هـ . ق)

2-1- جلال‌الدین محمد بلخی معروف به مولوی(۶۰۴ – ۶۷۲ هـ . ق)

با وجود این موضوع که عموم آثار مولوی پس از حمله مغول تالیف شده اند ، ما به این دلیل این آثار را در این بخش آوردیم که شخص مولوی در این دوره زمانی حضور داشته و آثار وی بازتاب مناسبی از تاثیر واژگان مغول بر زبان فارسی خواهد بود.

مثنوی معنوی و دیوان شمس آثاری از مولانا اند که به دست وی تالیف شده اند ، در حالی که دو اثر فیه ما فیه و مجالس سبعه برگرفته از سخنان مولوی میباشند و پس از وفات وی از دست نوشته ها و مکتوبات دیگران جمع آوری شده اند. به همین علت این دو اثر را از نظر زمان پس از مثنوی معنوی و دیوان شمس مورد بررسی قرار خواهیم داد.

در هر کدام از این دو آثار حدود ۴۰ کلمه مغلولی با تکرار تقریبا ۴۰۰ بار بکار رفته است ، که در مقایسه با آثار هجویری و کیکاووس این ارقام معنادار میشوند.

(مثنوی)

(دیوان شمس)

کتب فیه ما فیه و مجالس سبعه آثاری هستند که به قلم شخص مولوی نوشته نشده اند و زمان تالیف این دو نیز به پس از وفات وی یعنی سال ۶۷۲ هجری باز میگردد . این دو برخلاف آثار قبل دارای کلمات مغولی بسیار اندکی هستند که تعداد تکرارشان مشابه آثار هجویری و کیکاووس میباشد.

(فیه ما فیه)

(مجالس سبعه)

(کل آثار)

حال با دانستن مطالب فوق میتوانیم حداقل درباره شخص مولوی این را بگوییم که قرار گرفتن او در این دوره زمانی بر واژگانی که وارد نوشته هایش میشوند تاثیر بسزایی داشته است. هرچند که این راهم نباید نادیده گرفت که حجم کتبی که که در این دو قسمت تا به اینجا مورد بررسی قرار گرفته است بسیار متفاوت بوده و دارای شرایط یکسانی نیستند.

اما این را میدانیم که تنوع کلمات ، ناشی از حجم کتاب نیست.

3- بعد از حمله مغول:

دوران پس از حمله مغول دورانیست که واژگان جای خودشان را پیدا میکنند ، یا در میان باقی واژگان جاخوش میکنند و یا در کتب قدیمی می مانند ، اگر زمانی در آن ها بوده اند.

آثار نویسندگان این دوران به ما نشان خواهد داد که آیا واژگان مغولی توانسته اند وارد زبان رسمی یک ملیت شوند یا خیر.

3-1- خواجه نظام‌الدین عبیدالله زاکانی (۷۰۱-۷۷۲ هـ . ق)

عبید زاکانی ۳۰ سال پس از حمله مغول ها بدنیا آمد.

اما در آثار او اثری از کثرت تنوع و تکرار کلمات نیست .

(آثار زاکانی)

بدون توضیحی بیشتر سراغ آثار بعدی میرویم.

3-2- نورالدّین عبد الرّحمن بن احمد بن محمد جامی (۸۱۷ – ۸۷۱ هـ . ق)

وی نیز حدود ۱۲۰ سال پس از حمله مغول ها بدنیا آمده است و مانند زاکانی در آثارش از کلمات بیگانه زیادی استفاده نکرده است.

(آثار جامی)

در این دوران تعداد واژگان مغولی نسبت به دوران حمله مغول ها کمتر شده است ولی نسبت به دوران قبل از حمله مغول تا حدودی بیشتر است.

نتیجه گیری:

در دوره قبل از حمله مغول ، واژگان بسیار کمی در ادبیات فارسی بوده است با توجه به اینکه قوم مغول اهل ادبیات و نوشتاره نبوده اند.

و در هنگام حمله مغول با سیر بسیار صعودی آن مواجه شدیم و پس از حمله مغول بزرگان ادبیات دوباره شروع به کاهش استفاده از این قبیل واژگان کردند.

به عبارتی این سه دوره زمانی به صورت یک اوج در ابتدا و یک کاهش در انتها دیده میشود.

منابع:

۱. پردازش زبان طبیعی (NLP) چیست؟ - مهرداد توکلی – همیار آی تی (لینک)
۲. معرفی بهترین کتابخانه‌های پردازش متن - بخش اول – textmining – ویرگول (لینک)
۳. پردازش زبان طبیعی چیست؟ - نویسنده ناشناس – هوشینو (لینک)
۴. هضم (لینک)
۵. فایل CSV چیست و چه کاربردی دارد؟ - نویسنده ناشناس – فرادرس (لینک)
۶. مقاله کلمات ترکی – مغولی در کلیات شمس (محمود عابدی - محمد شادرویمنش- بدریه قوامی)
۷. واژگان ترکی و مغولی تاریخ جهانگشای جوینی(مصطفی موسوی)
۸. بررسی لغات ترکی و مغولی تاریخ گیتی گشا (رضا الوندی- مینا سهرابی)
۹. ارتباطِ زبانهای ترکی و مغولی و خطا در تشخیص واژههای دخیل ترکی و مغولی در زبان فارسی (مهدی رضائی)
۱۰. کلمات (ترکی مغولی و عثمانی) در غزلیات مولوی (محمود عابدی – بدریه قوامی)
۱۱. توضیح برخی از لغات و اصطلاحات مغولی در زبان و ادبیات فارسی (دکتر مهری باقری)
۱۲. ویکی پدیا
۱۳. مهر میهن
۱۴. ویستا