مدرسه بیگ دیتا

به نام خداوند بخشاینده مهربان

دومین دوره مجموعه سخنرانی های علم داده و هوش مصنوعی

دانشگاه صنعتی امیرکبیر(پلی تکنیک تهران)

محمد حیدری

BigDataWorld.ir


@BigDataSchool


@BigData_School


@BigData_School

آشنایی با من

محمد حیدری (بنیانگذار مدرسه بیگ دیتا، پژوهشگر علوم داده و یادگیری ماشین)

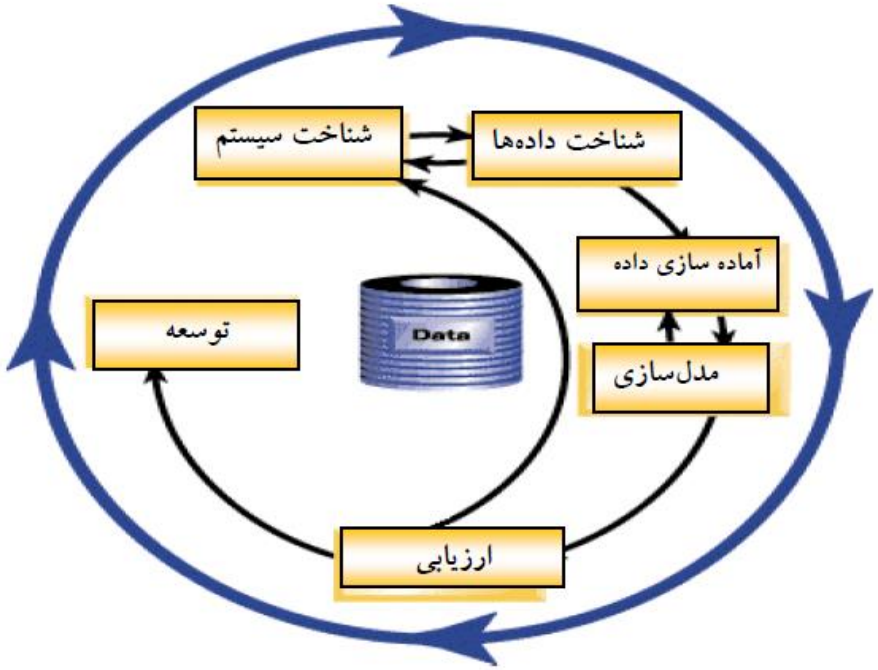
- پژوهشگر یادگیری ماشین در HiTS
- مدرس دوره های نرم افزاری در مرکز آموزش های تخصصی جهاد دانشگاهی
- کارشناسی مهندسی نرم افزار و ارشد مهندسی فناوری اطلاعات، دانشگاه تربیت مدرس تهران
- سخنران کارگاه آموزشی Graph Analytics در پنجمین سمینار زمستانه علوم کامپیوتر، صنعتی شریف
- ارائه دهنده مقاله برتر در 6th IEEE International Conference on Web Research, ICWR

منظور از داده چیست ؟

- داده انواع و اقسام متنوعی دارد.
  - صدا
  - تصاویر
  - کلمات
  - اعداد
- حمل و نقل
- علمی
- مالی
- آماری
- هواشناسی
- جغرافیایی
- و ...

گام های انجام یک پروژه علم داده

- گام اول، شناخت سیستم (**فهم** کسب و کار)
- گام دوم، شناخت داده ها (**درک** داده)
- گام سوم، آماده سازی داده (**پاکسازی**)
- گام چهارم، **مدل سازی** داده
- گام پنجم، **ارزیابی** داده
- گام ششم، **توسعه**

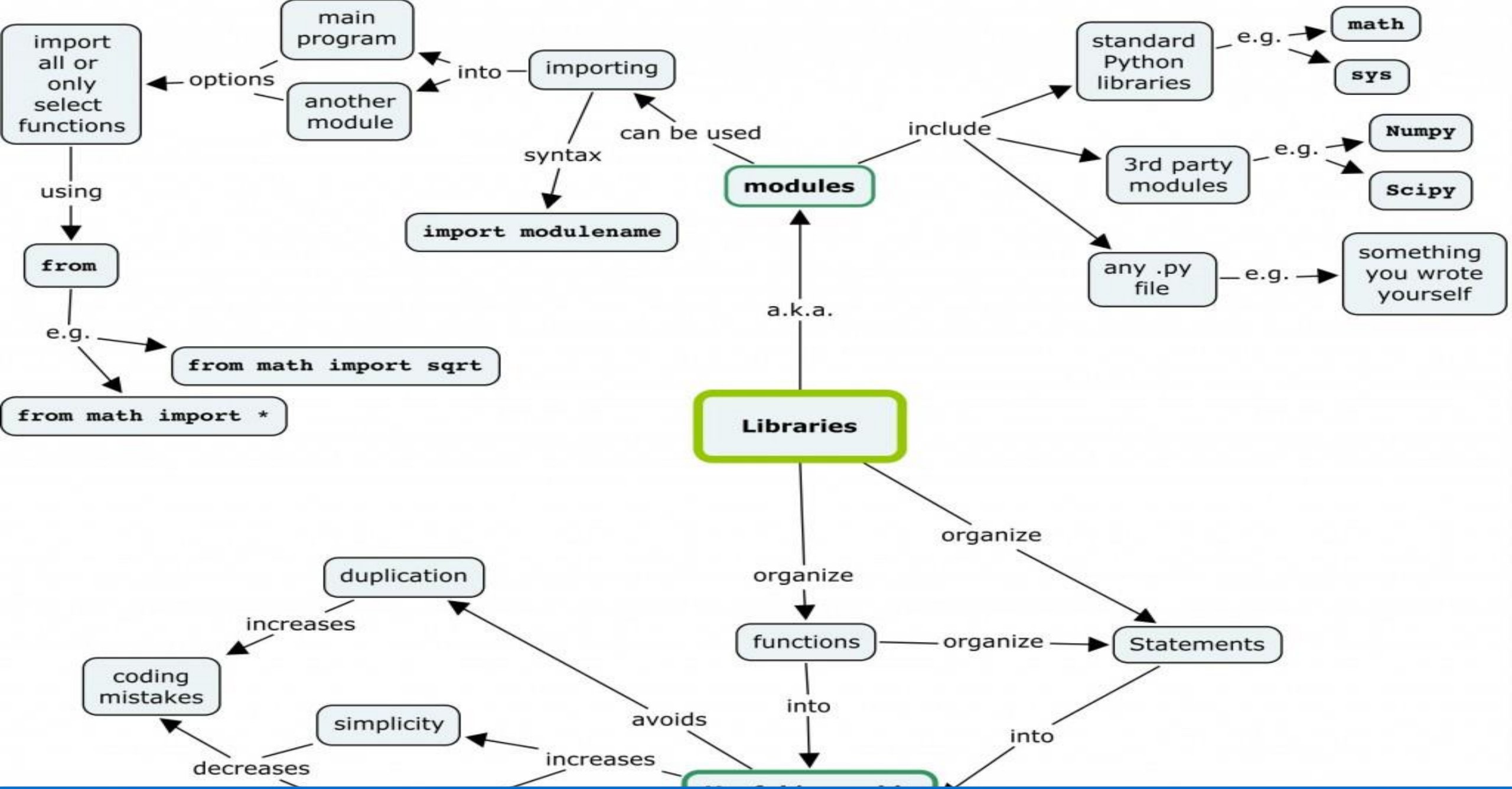


مفهوم کتابخانه (Library)

- کتابخانه مجموعه ای از **ماژول** هاست! (ماژول چی هست اصلا؟)
- ماژول مجموعه ای از کدها است که برای یک هدف خاص استفاده می شود و می توان در برنامه های گوناگونی از آن استفاده کرد.
- مزیت کلیدی
  - **پرهیز از تکرار کد**
- **اختراع دوباره چرخ، ممنوع**
  - کسی که در حال ساخت ماشین است
    زمان خود را صرف دوباره چرخ نمی کند،
    بنابراین، چرخ یک ماژول است، چون می تواند در سایر تولیدات استفاده شود.

طبقه بندی کتابخانه ها در پایتون

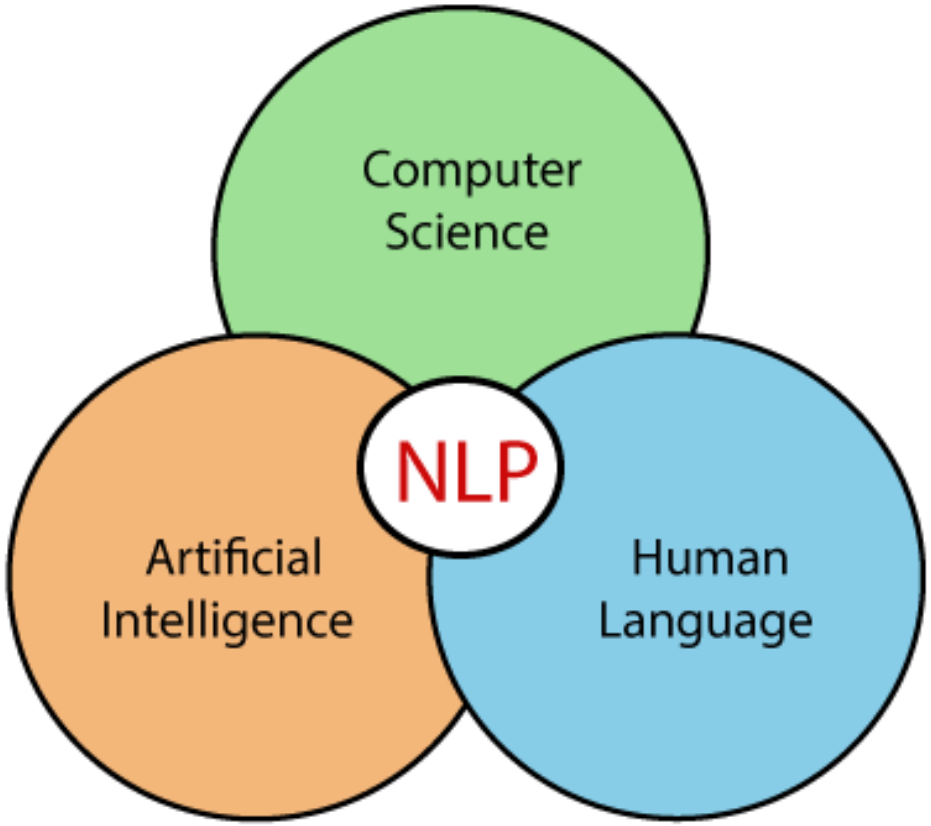
- ماژول ها یا کتابخانه هایی که در پایتون نوشته می شوند سه دسته اند:
  - توسط شخص ما نوشته شده اند.
  - توسط دیگران یا منابع خارجی نوشته شده اند. (PyPI)
  - به صورت پیش فرض به همراه پایتون نصب شده اند.


main program
import all or only select functions
options
another module
into
importing
syntax
import modulename
can be used
modules
include
standard Python libraries
e.g.
math
sys
3rd party modules
Numpy
Scipy
any .py file
something you wrote yourself
using
from
e.g.
from math import sqrt
from math import *
a.k.a.
Libraries
organize
duplication
increases
coding mistakes
functions
organize
Statements
into
avoids
simplicity
decreases
increases

پیش نیازهای لازم برای شرکت در کارگاه آموزشی

دانش مقدماتی **پایتون**

اطلاعات مقدماتی در رابطه با **داده کاوی**

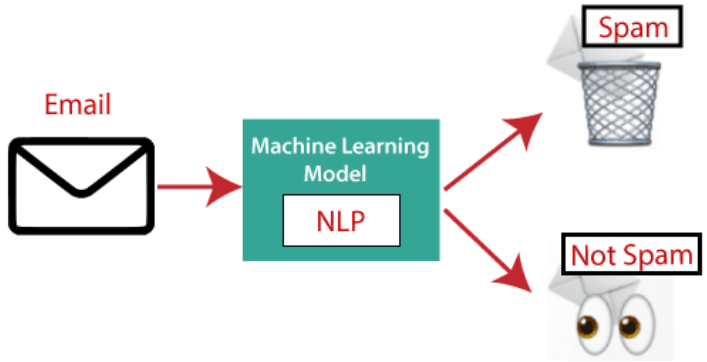


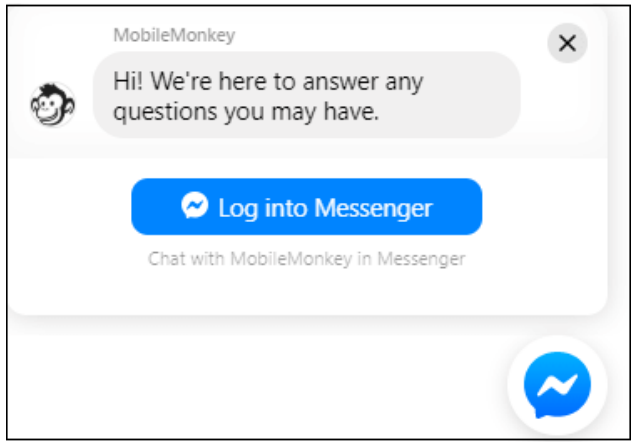
نصب ابزارهای لازم

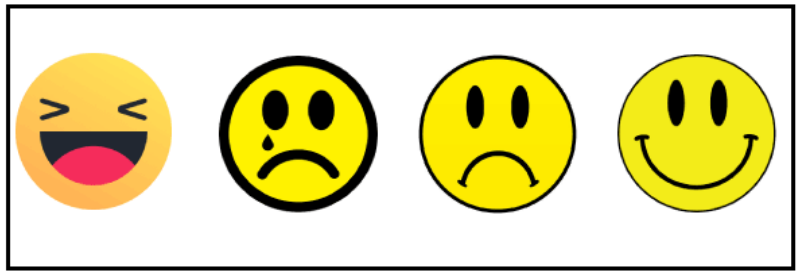
Anaconda Jupyter Notebook

PyCharm

کاربردهای پردازش متن







ابزارهای پردازش زبان طبیعی در زبان فارسی

هضم

- تمیز و مرتب کردن متن
- تقطیع جمله‌ها و واژه‌ها
  - Sentence Segmentation
  - Tokenization
- ریشه‌یابی واژه‌ها
- تحلیل صرفی جمله
- تجزیه نحوی جمله
- واسط استفاده از داده‌های زبان فارسی
- سازگاری با بسته NLTK
- پشتیبانی از پایتون نسخه ۲ و ۳

ابزارهای پردازش زبان طبیعی در زبان فارسی

پارسی ور

- تمیز و مرتب کردن متن
- تقطیع جمله‌ها و واژه‌ها
- Sentence Segmentation
- Tokenization
- ریشه‌یابی واژه‌ها
- تحلیل صرفی جمله
- تجزیه نحوی جمله
- **قابلیت استخراج مقادیر تاریخی عددی از متن**
- **تبدیل کلمات فـــــینگلیش یا پـــــینگلیش به فارسی**

چالش‌های پردازش زبان طبیعی در زبان فارسی (مقاله ۲۰۱۹)

Sentimental Analysis Challenges in Persian Language

Mohammad Heydari[1]

[1]Tarbiat Modares University, Tehran, Iran

m_heydari@modares.ac.ir

Abstract. The rapid growth in data on the internet requires a data mining process to reach a decision support insight. Persian language has strong potential for deep research in any aspect of natural language processing especially sentimental analysis approach. Thousands of websites and blogs updates and modifies by Persian users around the world that contains millions of Persian context. This range of application requires a comprehensive structured framework to extract beneficial information for helping enterprises to enhance their business and initiate customer-centric management process by producing effective recommender systems .Sentimental analysis is an intelligent approach for extracting useful information from huge amounts of data to help an enterprise for smart management process. In this road, machine learning and deep learning techniques will become very helpful but there are number of challenges which are face to them. This paper tried to present and assort most important challenges of sentimental analysis in Persian language. This language is an Indo-European language which spoken by over 110 million person around the world and is official language in Iran, Tajikistan and Afghanistan. It's also widely used in Uzbekistan, Pakistan and Turkish by order.

چالش‌های پردازش زبان طبیعی در زبان فارسی، (مقاله ۲۰۲۰)

Sentiment Analysis in Persian Language, Obstacles and Recent Developments

Mohammad Heydari
School of Industrial and System Engineering
Tarbiat Modares University
Tehran, Iran
m_heydari@modares.ac.ir

Alireza Rezvanian
School of Computer Science
Institute for Research in Fundamental Sciences, IPM
Tehran, Iran
rezvanian@usc.ac.ir

***Abstract*— Currently, sentiment analysis is a field of great interest and development since it has various practical applications in different fields. The Persian language has a strong potential for deep research in any aspect of natural language processing, especially the sentimental analysis approach. Thousands of websites, blogs, social networks like Telegram, Instagram and Twitter update, and modify by Persian users around the world that contains millions of contexts. to extract knowledge of these huge amounts of raw data, deep learning techniques became increasingly popular but there is a number of challenges that the novel models encounter with them. In this study, we review the latest studies in Persian Natural Language Processing and introduce state-of-the-art Deep Learning models in the field of Natural Language Processing. In the following we express most important Sentimental Analysis Challenges in Persian ancient language and demonstrate related Persian Information Retrieval Tools, Models, Libraries and Techniques. Finally, we create the network of Research Centers and their products in Persian Natural Language Processing for the first time in our knowledge and analyze it by utilizing Social Network Analysis techniques. Utilization of SNA approach in the field gives us a comprehensive big picture of latest advancement and development in the Persian Sentiment Analysis topic and identifying the Blind spots.**

Keywords— Persian Natural Language Processing, Persian Text Mining, Persian Language Processing Difficulties, Persian Sentiment Analysis Challenges

eliminate the high-level problems. Famous companies like Google, YouTube, Apple and Amazon employed Natural Language Processing techniques to extract useful insight of user's reviews. Currently social network platforms are under scrutiny of many academic researchers and developers to create automatic users comments extractors' tools, the topic is widely used in various political, social, commercial and industrial aspects[2]. "The application of text mining increased by politicians who aims data sources acquisition to gain the superior seat or stabilize their position. Studying on text mining effects NLP[1] and various topic of science"[3][4].

Sentimental analysis is an interdisciplinary field tries to familiar machine learning and artificial intelligence with human emotion. It is the extraction process of insight from the writer's beliefs and viewpoints in a specific domain of texts. The field is gained a lots of attention in NLP in recent years. [5] It also roles an important task in web mining, data mining specially when it is crucial to initiate a recommendation platform in various business case such as YouTube, Instagram or Twitter. It is also guiding the business and brands to represents best service collection for their customers. Almost studies in sentimental analysis initiated on English language. regrettably, Persian ancient language didn't get enough attention by computation linguistic scholars. Since 2000 classification algorithms in data mining utilized for opinion mining[6][7][8].

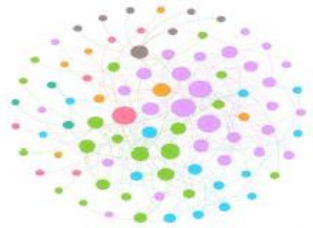


Figure 8 - Community Detection of the Network based on Modularity Method

Figure 9 - Research Centers and Products Graph

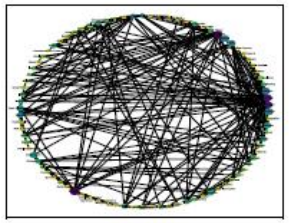


Figure 10 - Circular Visualization of The Network

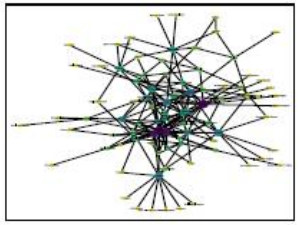
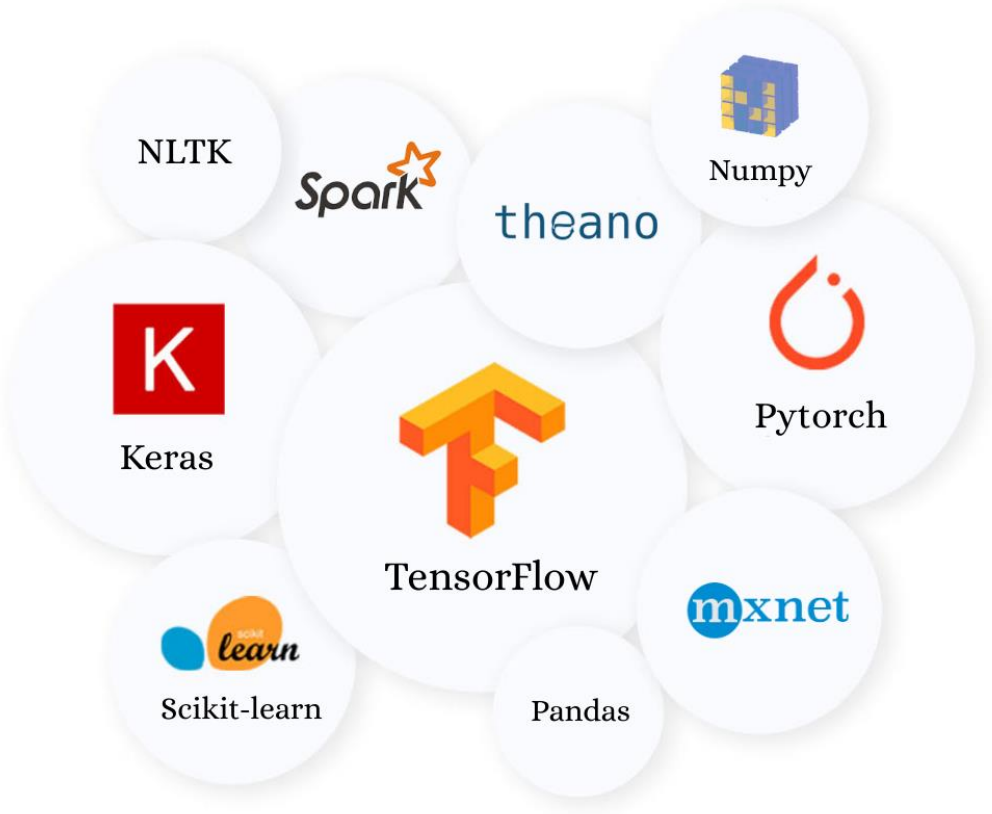


Figure 11 - Betweenness Centrality-based Visualization of The Network

Table 6 - Top ten Degree Centrality of the Network

Rank	Nodes	Type	Degree Centrality
1	**Normalizer**	**Technique**	0.2093
1	**FarsiYar**	**Startup**	0.2093
2	Tokenizer	Technique	0.1976
2	Stemmer	Technique	0.1976
4	Sentiment Check	Technique	0.1395
4	POS Tagger	Technique	0.1395
1	**Amirkabir**	**Academic laboratory**	0.1279
1	Tehran	Academic laboratory	0.1279
6	Topic Classification	Technique	0.1279
1	**Arman Rayan Sharif**	**Private company**	0.1162
3	Beheshti	Academic laboratory	0.1162
2	Amerandish	Private company	0.1046
2	Dataak	Private company	0.1046
2	Cafebazar	Private company	0.1046

کتابخانه های برتر به منظور بهره برداری در پروژه های یادگیری ماشین



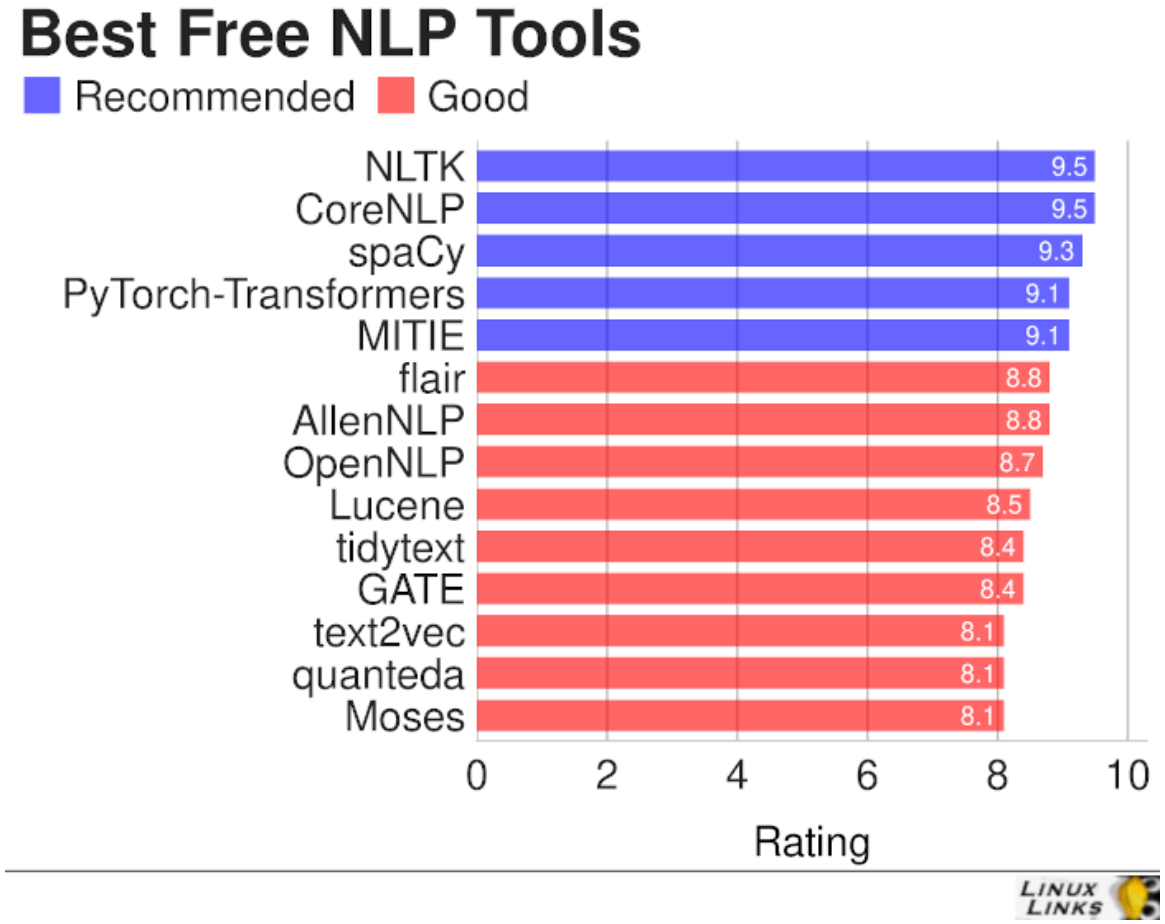
گام‌ها و مراحل متن کاوی

- انتخاب متن
- پردازش متن
- تبدیل متن به صفات خاصه
- انتخاب صفات خاصه از متن
- داده‌کاوی بر روی متن
- تفسیر و ارزیابی خروجی متن کاوی

کتابخانه های برتر NLP

Persian

- Hazm (Persian Supported)
- Parsivar(Persian Supported)
- FastText (Persian Supported)



مهمترین روش های پیش پردازش متن

به ترتیب

- تبدیل متن به حروف کوچک
- پاک کردن اعداد
- پاک کردن علائم نقطه گذاری
- پاک کردن فضاهای خالی
- توکن بندی کلمات
- حذف کردن کلمات فاقد اهمیت کافی
- ریشه یابی کلمات
- برچسب گذاری نقش کلمات در متن، Part of Speech (POS) Tagging
- تجزیه و تحلیل سطحی جملات
- شناسایی موجودیت های اسمی در جملات، NER
- شناسایی هم رُخدادی کلمات در جملات

پیش پردازش متن، انواع تحلیل ها

تحلیل صرفی

- POS
- Tokenizer
- Lemmatizer

تحلیل نحوی

- Parser
- Chunker

تحلیل معنایی

- NER
- Event Detection

ابزارهای پیش پردازش متن فارسی

نرمال ساز

تقطیع واژه و جمله

ریشه یابی

برچسب ادات سخن

حذف کلمات توقف

پارسِر

نرمال ساز، Normalizer

هدف: تمیز، مرتب کردن متن و یکسان‌سازی کاراکترها با جایگزین کردن کاراکترهای استاندارد در متن ورودی

همه‌ی حروف متن با جایگزینی معادل استاندارد آنها، یکسان‌سازی گردند.

- مثال: چالش شباهت رسم الخَط فارسی با عربی و استفاده از کاراکترهای عربی به جای فارسی
  - حروف "ک" و "ی"
- اصلاح و یکسان‌سازی نویسه‌ی نیم‌فاصله و فاصله
  - حذف نویسه‌های اعراب، تشدید، تنوین و «ـ» برای کشش نویسه‌های چسبان
- مطابق با یک سری قاعده دقیق و مشخص، فاصله‌ها و نیم‌فاصله‌های موجود در متن برای وندهایی نظیر
  - "ها"، "تر" و "ی" غیرچسبان (در انتهای لغات)
  - و همچنین پیشوندها و پسوندهای فعل‌ساز نظیر "نمی"، "می"، "ام"، "ایم"، "اید".

تقطیع واژه و جمله، Sentence Splitter and Tokenizer

هدف: تشخیص جملات در متن ورودی

- با مشخص سازی مرز جملات و کلمات

تشخیص‌دهنده‌ی لغات

هدف: با استفاده از علامت‌های فضای خالی، "،"، ",‌"، "–" واحدهای با معنی مانند واژه‌ها را شناسایی می‌نماید.

حذف کلمات توقف، Stop Word Removal

هدف: حذف علائم، اعداد، کلمات عمومی و بدون ارزش معنایی (مثل: از، در، با، به، است، پس، ...) در جمله
- می‌تواند بدون از بین بردن معنا باعث بهبود دقت و سرعت الگوریتم‌های متن‌کاوی شوند.

لیست کلمات توقف وابسته به کاربرد مورد نظر باید تهیه شود.
- مثال: کلمات "هست" و "نیست" برای دسته‌بندی موضوعی متن حائز اهمیت نیستند
- ولی در تحلیل حس، می‌توانند حس جمله را معکوس کنند!

ریشه‌یابی کلمات، Stemmer and Lemmatizer

هدف: ریشه‌یابی، حذف پیشوند و پسوندهای کلمات و تعیین ریشه اصلی کلمه

معروفترین الگوریتم ریشه یابی در انگلیسی

- Porter

در روش‌های ریشه‌یابی رایج، بعد از حذف انواع وندها ممکن است معنای کلمه **تغییر یابد.**

ولی در Lemmatizer سعی در ریشه‌یابی بُن کلمه **بدون تغییر مفهوم اصلی کلمه** در جمله می باشد.

Word	*Lemmatization*	*Stemming*
was	be	wa
studies	study	studi
studying	study	study

بن واژه یاب، Lemmatizer

سطوح مختلف

- مثال: کلمه دانشجویان
  - سطح ۱: دانشجو
  - سطح ۲: دانش
  - سطح ۳: دان

تصحیح کننده خطاهای املایی، Spell Correction

به ازای کلمات بدون مفهوم (که کلمه یا ریشه‌ی آن در لیست کلمات رسمی یا محاوره‌ای زبان فارسی وجود ندارند)، شبیه‌ترین کلمه براساس تحلیل‌های آماری و از نظر املایی جایگزین آن خواهد شد.

معیار استفاده شده برای آستانه شباهت املایی تبدیل کلمات

- جداسازی کلمات بهم چسبیده
- اصلاح/تغییر تنها یک حرف با یکی از کاراکترهای مجاور آن در صفحه کلید استاندارد زبان فارسی
- اصلاح حروف هم صدا از نظر تلفظ
  - مثال: مانند حروف: س، ص، ث که همگی به صدای S اشاره دارند.

برچسب‌زنی ادات سخن یا نقش کلمات در جمله، POS Tagging

انتساب برچسب‌های نحوی (از قبیل اسم، انواع صفت، انواع قید، نوع فعل، انواع حروف و ...) به واژه‌ها
- Part of Speech Tagging

نقش کلمات در جملات از مهم ترین پیش پردازش‌هایی است که در Chunker هم استفاده می‌شود.

قطعه‌بند Chunker **یا پارسر کم‌عمق** Shallow Parser

ابزاری برای تشخیص گروه‌های اسمی، فعلی، صفات و .... در یک جمله است.

برای مثال **اولین دارنده مدال طلای المپیک** یک عبارت اسمی است.

جهت تقویت ابزارهای سطح بالاتر از قبیل

- پارسر،
- برچسب‌زن نقش معنایی
- و تشخیص موجودیت‌های نامدار

لازم است نه تنها نقش‌های کلمات مشخص گردند، بلکه **باید وابستگی‌های کلمات مجاور هم به لحاظ نقشی در جمله** مشخص شوند. از اینرو به این ابزار پارسر سبک light یا کم عمق Shallow می گویند.

تشخیص موجودیت‌های نامدار، Named Entity Recognition

به این معناست که اسامی خاص در یک متن را بتوان تشخیص داد و آنها را به رده‌های مشخصی دسته‌بندی کرد.

از قبیل اعم از

- اسامی افراد،
- اماکن،
- سازمان‌ها،
- مقادیر عددی و ...

شبکه واژگان، WordNet

پایگاه داده لغوی شامل مجموعه‌ای از لغات در قالب گروه‌های هم‌معنی و ارتباطات مختلف معنایی بین آنهاست.

ارتباطات معنایی در داخل این مجموعه شامل ۱۶ نوع رابطه می‌باشد.

این مجموعه به عنوان یک مرجع در بسیاری از پردازش‌های زبانی برای توجه به معنای لغات استفاده می شود.

نمونه های انگلیسی

- Princeton WordNet
- EuroWordNet

نمونه های فارسی

- فارس‌نت (شهید بهشتی و مرکز تحقیقات مخابرات)
- فردوس نت (دانشگاه فردوسی)
- شبکه واژگان (دانشگاه تهران)

کتابخانه NLTK: Natural Language Toolkit

یکی از جامع‌ترین و قدیمی‌ترین کتابخانه‌های پردازش زبان طبیعی در پایتون است

پایه و استانداردی برای کتابخانه‌های پردازش متن محسوب شده و برای کاربردهای پژوهشی فوق‌العاده است.

دارای بیش از ۵۰ مجموعه متن و ۹ تکنیک مختلف برای ریشه یابی کلمات است.

یکی از ویژگی‌های خوب این کتابخانه امکان اتصال به پیکره‌های مختلف متنی است.

برای کار با این کتابخانه لازم است تا در ابتدا مجموعه‌ای از متون را دانلود کنیم.

- این مجموعه متون که با نام Corpus نیز شناخته می شوند از طریق NLTK قابل دانلود هستند.
- یک Corpus که صورت جمع آن Corpora است مجموعه ای از داده های متنی است که در توصیف و تحلیل زبان کاربرد دارند
- حجم تمامی Corpus های NLTK نزدیک به ۱۱ گیگابایت است.

منبع

کتابخانه Stanford’s CoreNLP

ابزار بسیار مناسب برای تحلیل‌های دستوری برای زبان است.

این ابزار با زبان جاوا نوشته شده اما API آن برای زبان‌های مختلف از قبیل پایتون و سی‌شارپ موجود است.

چندزبانه بودن و **تمرکز بر استخراج اطلاعات باز از متن** از ویژگی‌های اصلی این ابزار است.

از **زبان فارسی** پشتیبانی **می کند**.

https://github.com/stanfordnlp/python-stanford-corenlp

https://nlp.stanford.edu/pubs/StanfordCoreNlp2014.pdf

کتابخانه SpaCy

با زبان **پایتون** برای متن کاوی تهیه شده و با اجرا روی **Cython** به سرعت مشهور است.

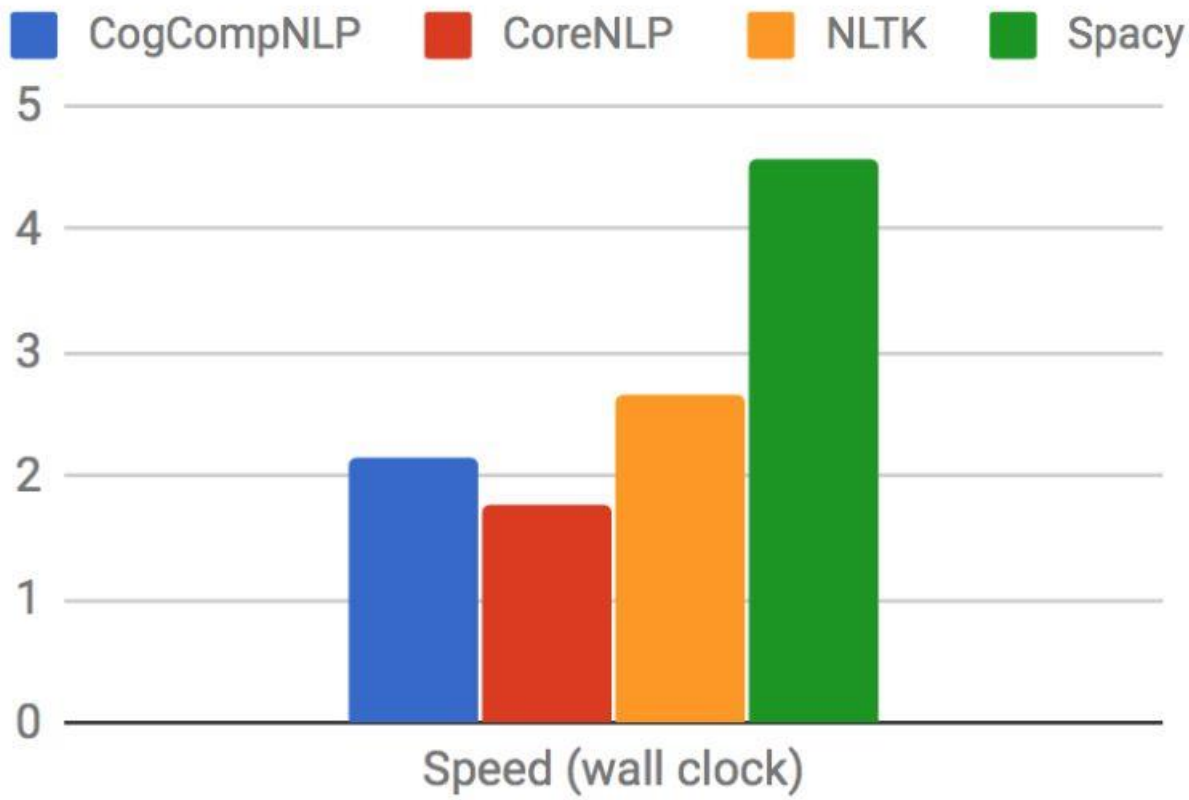
ارتباط خوبی با ابزارهای یادگیری ماشین و یادگیری عمیق از قبیل ذیل دارد.

- Gensim
- Keras
- TensorFlow
- scikit-learn

از زبان فارسی پشتیبانی **نمی کند.**

http://spacy.io/

مقایسه جالبی بین سرعت اجرای چهار کتابخانه مطرح



https://pdfs.semanticscholar.org/5930/efbf01efa8944258b1c0f7349111702f779e.pdf

مقایسه دقت در شناسایی کلمات و برچسب‌زنی نقش ادات سخن چهار کتابخانه مطرح

	Comparison	tokens for manual annotation	tokens for library	identical tokens	identical tokens %	identical token/POS_g	identical token/POS_g %	identical token/POS_s	identical token/POS_s %
Stack Overflow	Manual vs. Stanford	375	385	371	97.63	339	89.21	317	83.42
	Manual vs. SyntaxNet	375	366	357	96.36	332	89.61	317	85.56
	Manual vs. spaCy	375	377	369	98.14	347	**92.29**	338	**89.89**
	Manual vs. NLTK	375	374	373	**99.60**	331	88.38	306	81.71
GitHub ReadMe	Manual vs. Stanford	361	371	357	97.54	310	84.70	286	78.14
	Manual vs. SyntaxNet	361	358	350	97.36	308	85.67	297	**82.61**
	Manual vs. spaCy	361	370	344	94.12	309	84.54	291	79.62
	Manual vs. NLTK	361	360	354	**98.20**	312	**86.55**	278	77.12
Java API Doc.	Manual vs. Stanford	380	398	374	96.14	328	84.32	303	77.89
	Manual vs. SyntaxNet	380	379	373	98.29	298	78.52	286	75.36
	Manual vs. spaCy	380	380	369	97.11	345	**90.79**	298	**78.42**
	Manual vs. NLTK	380	382	379	**99.48**	345	90.55	294	77.17

https://www.computer.org/csdl/proceedings/msr/2017/1544/00/07962368.pdf

کتابخانه TextBlob

یکی از ابزارهای بسیار کامل و راحت برای پردازش داده‌های متنی در زبان **پایتون** است.

از قابلیت های NLTK و کتابخانه Pattern استفاده می کند و فرآیند تحلیل احساسات در آن بسیارساده است.

این کتابخانه شامل ابزارهای مختلف پردازش زبان طبیعی از قبیل

- استخراج عبارات اسمی،
- بن‌واژه‌یابی،
- برچسب‌زنی نقش ادت سخن،
- پارسر (تجزیه‌گر) جملات،
- تحلیل احساسات،
- دسته‌بندی (نایوبیز و درخت تصمیم)،
- ترجمه (بوسیله مترجم گوگل)،
- تصحیح اشتباهات املایی،
- و اتصال به WordNet است.

StanfordNLP **کتابخانه**

یکی از **جدیدترین** کتابخانه‌های پردازش متن چندزبانه است که در **۲۰۱۸** ایجاد شد.

پشتیبانی بیش از ۵۳ زبان با استفاده از **آموزش مدل‌های یادگیری عمیق بوسیله PyTorch** است.

مدل‌های یادگیری عمیق ابزارهای مختلف این کتابخانه، براساس **پیکره خانم دکتر سراجی**، برای زبان فارسی نیز آموزش داده شده و در زبان فارسی نیز قابل استفاده هستند.

تصمیم جالب دانشگاه استنفورد این بود که برخلاف کتابخانه Stanford CoreNLP که با زبان جاوا نوشته شده بود، این کتابخانه کلاً با زبان محبوب **پایتون** نوشته شده است.

از زبان فارسی پشتیبانی **می کند**.

https://stanfordnlp.github.io

کتابخانه Gensim

یکی از محبوب‌ترین و بهترین ابزارهای **مدلسازی موضوع و بازنمایی متن** است و برای تشخیص شباهت متون مختلف بسیار مناسب است.

- Topic Modeling
- Vector representation

در این کتابخانه اغلب روش‌های مشهور در زبان پایتون پیاده‌سازی شده و به خوبی بروزرسانی و پشتیبانی می‌شوند

- Word embedding
- Word Representation

برای **پیش‌پردازش** متن بهتر است از کتابخانه‌های NLTK, SpaCy استفاده کنید.

برای تحلیل‌های بَعدی **مانند استخراج کلمات کلیدی یا موضوعات درون متن** از Gensim استفاده نمایید.

https://radimrehurek.com/gensim/index.html

کتابخانه SparkNLP

برای استفاده از ماژول‌های کاربردی **پردازش متن روی بستر Spark و کتابخانه Spark ML** توسعه یافته است

برای استفاده در زبان‌های متنوع و با هدف **محاسبات مقیاس‌پذیر برای بیگ دیتا بصورت توزیع شده**، ایجاد شده است.

- Python
- Java
- Scala

اغلب امکانات این کتابخانه بصورت **رایگان** در اختیار پژوهشگران قرار داده شده است

و تنها امکانات بخش یادگیری عمیق و ارتباط با Tensorflow بصورت **تجاری** میسر است.

nlp.johnsnowlabs.com/quickstart.html

github.com/JohnSnowLabs/spark-nlp/blob/2.0.0/python/example/vivekn-sentiment/sentiment.ipynb

John Snow LABS

FULL

- ✓ Spelling Correction
- ✓ Sentiment Analyzer
- ✓ Object Character Recognition
- ✓ PDF Parsing
- ✓ Assertion Status Detection
- ✓ Pretrained Models
- ✓ Pretrained Pipelines
- ✓ Light NLP Pipelines
- ✓ Recursive NLP Pipelines
- ✓ Tokenizer
- ✓ Document Assembler
- ✓ Text Matcher
- ✓ Normalizer
- ✓ Sentence Detector
- ✓ Regex Matcher
- ✓ Stemmer
- ✓ Part of Speech Tagger
- ✓ Date Matcher
- ✓ Lemmatizer
- ✓ Dependency Parser
- ✓ Chunker
- ✓ Named Entity Recognition

+

Spark MLlib

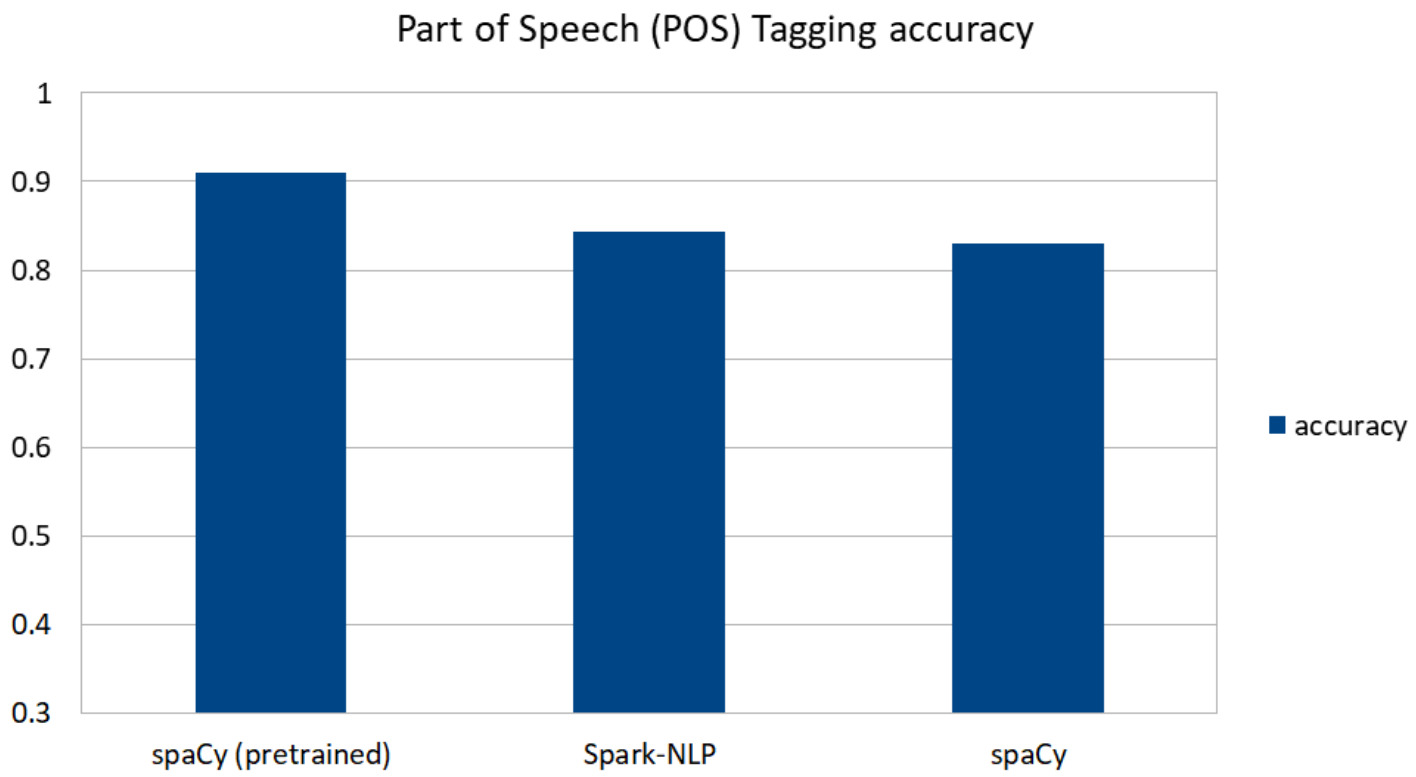
- ✓ Word2Vec
- ✓ Topic Modeling
- ✓ TF / IDF
- ✓ Stop word removal
- ✓ n-grams
- ✓ String distance
- ✓ Custom ML Pipelines

+

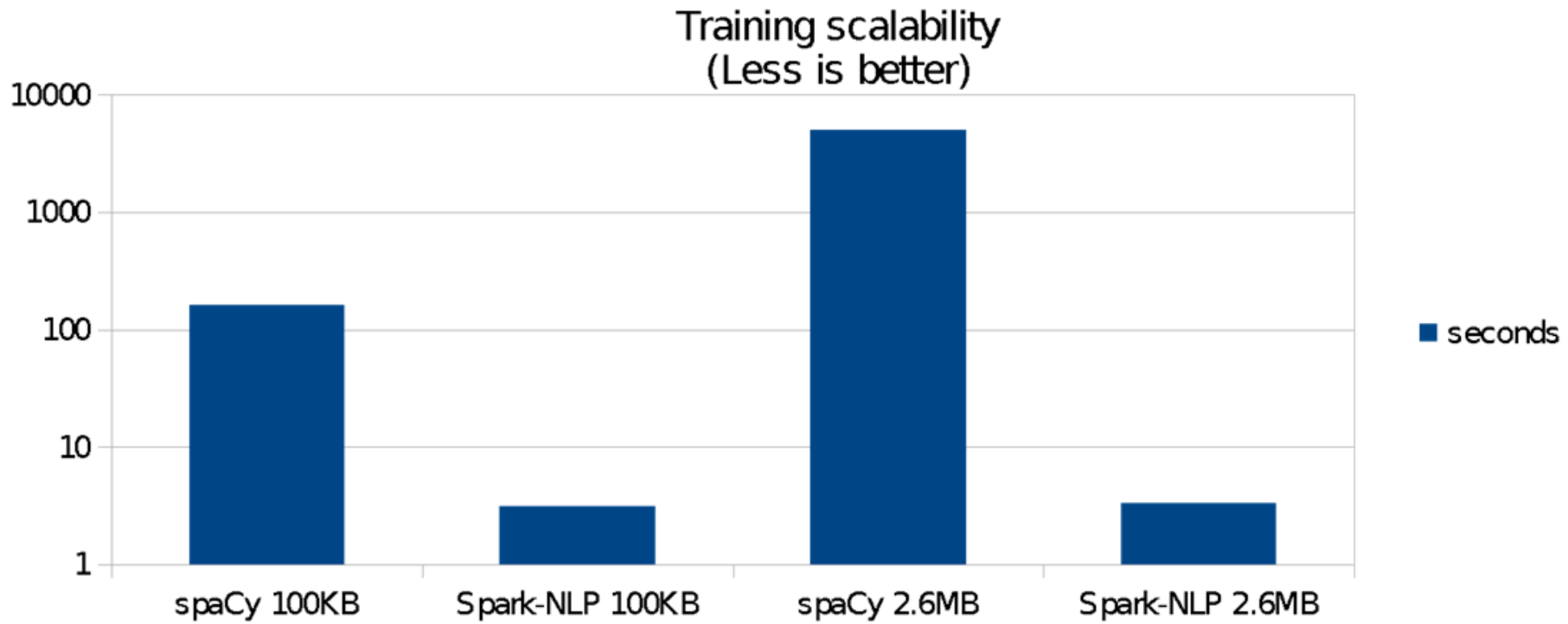
TensorFlow

- ✓ Prebuilt LSTM+CNN Graphs
- ✓ Prebuilt RNN Graphs
- ✓ Prebuilt Embeddings
- ✓ Custom DL Pipelines
- ✓ GPU Optimized Training

مقایسه دقت SpaCy و SparkNLP برای برچسب‌زنی نقش کلمات در جمله برای زبان انگلیسی



مقایسه کارایی SpaCy و SparkNLP در آموزش مدل یادگیر برای برچسب‌زنی نقش کلمات



کتابخانه FastText

کتابخانه ای به منظور یادگیری Word Embedding و Text Classification
- توسط آزمایشگاه Facebook's AI Research (FAIR) ساخته شده است.

امکان ساخت الگوریتم های Supervised و UnSupervised به منظور دریافت بازنمایی بُرداری کلمات
- در حال حاضر مدل های Pretrain شده FastText برای ۲۹۴ زبان در دسترس است.

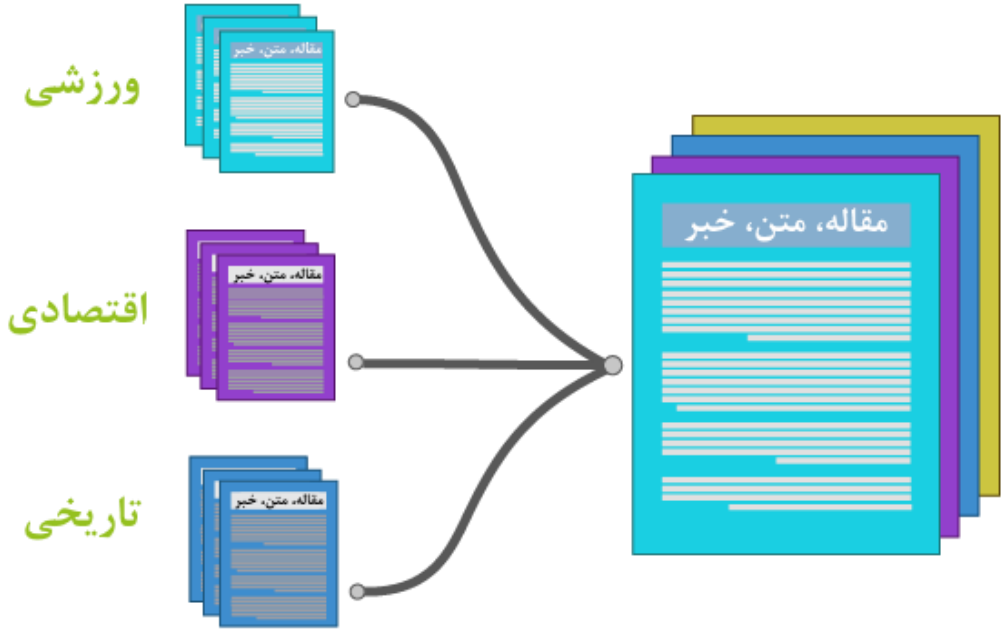
این کتابخانه از یک شبکه عصبی به منظور Word Embedding استفاده می کند.

الگوریتم FastText بر اساس این ۲ مقاله پژوهشی است.

- Enriching Word Vectors with Subword Information , Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov, 2016
- Bag of Tricks for Efficient Text Classification, Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov, 2016

دسته بند متن

دسته بند متن، امکان **دسته‌بندی خودکار و هوشمند متن** را در دسته های دلخواه فراهم می کند.



خلاصه ساز

خلاصه‌ساز، بخش‌های مهم متن‌های طولانی را تشخیص می دهد و **متن خلاصه شده** را ارائه می کند.

تعبیه سازی کلمات یا Words Embeddings

تعبیه سازی کلمات یا Words Embeddings

برای بسیاری از روشهای پردازش متن، نیاز به نمایش عددی کلمات و متون داریم تا بتوانیم از انواع روشهای عددی حوزه یادگیری ماشین مانند اکثر الگوریتم های دسته بندی روی لغات و اسناد استفاده کنیم.

تعبیه سازی کلمات یا Words Embeddings

روش Bag of Words یا صندوقچه کلمات

- برای هر لغت در صندوقچه یا بردار ما، مکانی درنظر گرفته شده است

مثال

- Sentence 1: The cat sat on the hat
- Sentence 2: The dog ate the cat and the hat
- برای این دو جمله فرهنگ لغت ما عبارت خواهد بود از :
- { the, cat, sat, on, hat, dog, ate, and }
- نمایش صندوقچه کلمات این دو جمله:
- Sentence 1: { 2, 1, 1, 1, 1, 0, 0, 0 }
- Sentence 2: { 3, 1, 0, 0, 1, 1, 1, 1}

- با این روش ما **دو بردار عددی** داریم که حال می توانیم از این دو در الگوریتم های عددی خود استفاده کنیم

تعبیه سازی کلمات یا Words Embeddings

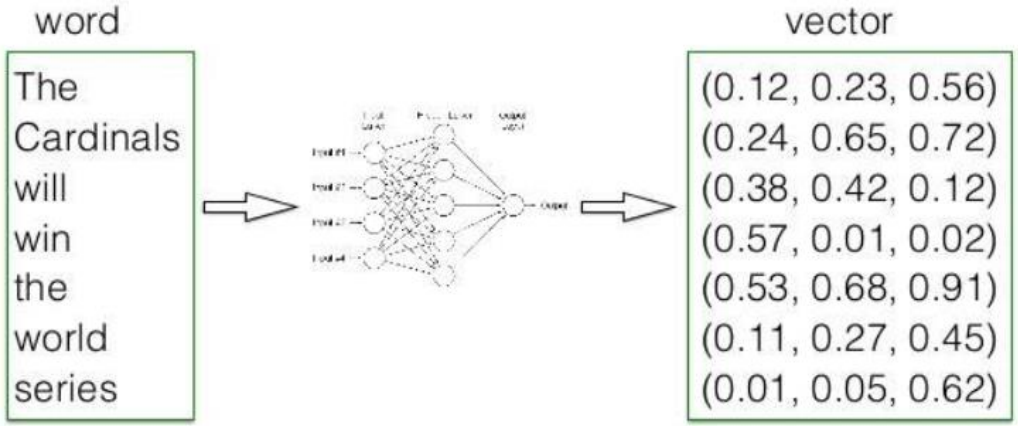
- با وجود سادگی این روش ، اما معایب بزرگی دارد.
- مثلاً اگر فرهنگ لغت ما صد هزار لغت داشته باشد ، به ازای هر متن ما باید برداری صد هزارتایی ذخیره کنیم
  - که هم نیاز به **فضای ذخیره سازی** زیادی خواهیم داشت
  - و هم **پیچیدگی الگوریتم ها و زمان اجرای** آنها را بسیار بالا می برد .
- در این نحوه مدلسازی، فقط کلمات و تکرار آنها برای ما مهم بوده است و ترتیب کلمات یا زمینه متن (اقتصادی ، علمی ، سیاسی و ...) تاثیری در مدل ما نخواهد داشت (و این ضعف است)

الگوریتم Word2Vec

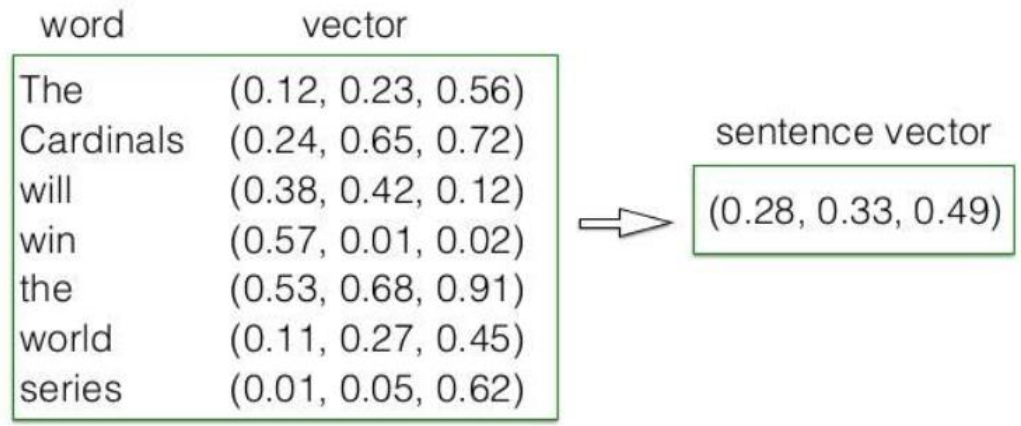
- **روش Word2Vec**
  - روشی بسیار کارآمد و مناسب برای نمایش لغات و متون و پردازش آنها
  - نمایش برداری کلمات که می تواند در بسیاری از کاربردهای نوین پردازش متن مانند سنجش احساسات، جستجوی متون مشابه یا پیشنهاد اخبار یا کالای مشابه استفاده شود.
  - در این روش به کمک شبکه عصبی یک بردار با اندازه کوچک و ثابت برای نمایش تمام لغات و متون در نظرگرفته شده و با اعداد مناسب در فاز آموزش مدل، برای هر لغت این بردار محاسبه می شود.
  - اگر این بردار را ۴۰۰ تایی فرض کنید، یک فضای ۴۰۰ بعدی خواهیم داشت که هر لغت در این فضا یک نمایش منحصر بفرد خواهد داشت.
  - برای افزایش دقت این روش، مجموعه داده اولیه که برای آموزش مدل مورد نیاز است، باید حدود چند میلیارد لغت را که درون چندین میلیون سند یا متن به کار رفته اند، در برگیرد.

الگوریتم Word2Vec

محاسبه وکتور هر کلمه پس از اعمال مدل روی کلمات



محاسبه وکتور جمله با احتساب میانگین کلمات جمله



مقایسه این الگوریتم با دو رهیافت مختلف، از لحاظ حجم داده ها ، زمان لازم برای ایجاد بردارها، دقت و ابعاد بردارها

در این جدول، روش Word2Vec با دو الگوریتم ایجاد بردار CBOW و Skip-gram با روشهای مدلسازی زبانی با شبکه عصبی یاNNLM مقایسه و ارزیابی شده است.

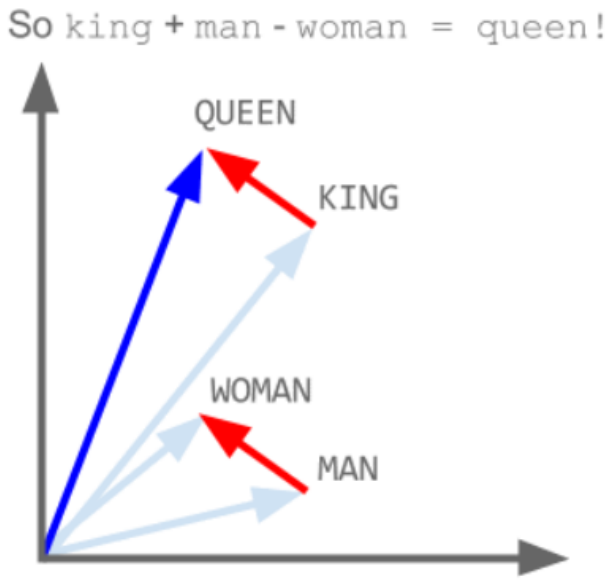
Model	*Vector Dimensionality*	*Training Words*	*Training Time*	*Accuracy [%]*
Collobert NNLM	50	660M	2 months	11
Turian NNLM	200	37M	few weeks	2
Mnih NNLM	100	37M	7 days	9
Mikolov RNNLM	640	320M	weeks	25
Huang NNLM	50	990M	weeks	13
Our NNLM	100	6B	2.5 days	51
Skip-gram (hier.s.)	1000	6B	hours	66
CBOW (negative)	300	1.5B	**minutes**	**72**

الگوریتم Word2Vec

این روش که الگوریتم آن به صورت متن باز نیز منتشر شده است و کتابخانه های مختلفی برای زبانهای مختلف برای کار با آن تولید شده است، زمانی که توسط گوگل بر روی حجم بالای متون و اطلاعات به کار رفته است ، نتایج بسیار شگرفی را به همراه داده است .

امکان جمع و تفریق جبری روی کلمات

- مثلا اگر بردار لغت **پادشاه** را **منهای** بردار لغت **مرد** کنیم ، نتیجه به بردار کلمه **ملکه** بسیار نزدیک است.



مثالهایی از روابط تولید شده توسط این الگوریتم

فرانسه به پاریس مثل ؟ است به ایتالیا

شباهت ویندوز به مایکروسافت مثل شباهت ؟ است به گوگل

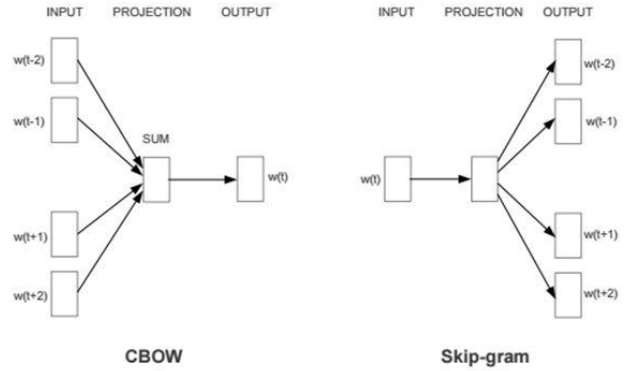
Expression	*Nearest token*
Paris - France + Italy	Rome
bigger - big + cold	colder
sushi - Japan + Germany	bratwurst
Cu - copper + gold	Au
Windows - Microsoft + Google	Android
Montreal Canadiens - Montreal + Toronto	Toronto Maple Leafs

Skip-gram

این روش برعکس این روش کار می کند به این صورت که بر اساس یک لغت داده شده ، می خواهد چند لغت قبل و بعد آنرا تشخیص دهد و با تغییر مداوم اعداد بردارهای لغات، نهایتا به یک وضعیت باثبات می رسد که همان بردارهای مورد بحث ماست.

CBOW and Skip-gram

- CBOW stands for “continuous bag-of-words”
- Both are networks without hidden layers.



Reference: Efficient Estimation of Word Representations in Vector Space by Tomas Mikolov, et al.

روش های CBOW, Skip-Gram

این دو روش که هر دو یک شبکه عصبی ساده هستند که بدون وجود لایه پنهانی که در اغلب روشهای شبکه عصبی وجود دارد، به کمک چند قانون ساده، بردارهای مورد نیاز را تولید می کنند.

در روش **کیف لغات پیوسته**، ابتدا به ازای هر لغت یک بردار با طول مشخص و با اعداد تصادفی (بین صفر و یک) تولید می شود.

سپس به ازای هر کلمه از یک سند یا متن، تعدادی مشخص از کلمات بعد و قبل آنرا به شبکه عصبی می دهیم (به غیر از خود لغت فعلی) و با عملیات ساده ریاضی، بردار لغت فعلی را تولید می کنیم (یا به عبارتی از روی کلمات قبل و بعد یک لغت، آنرا حدس می زنیم) که این اعداد با مقادیر قبلی بردار لغت جایگزین می شوند.

زمانی که این کار بر روی تمام لغات در تمام متون انجام گیرد، بردارهای نهایی لغات همان بردارهای مطلوب ما هستند.

پیاده سازی در Gensim

```
>>> model.most_similar("man")
[(u'woman', 0.6056041121482849), (u'guy', 0.4935004413127899), (u'boy',
0.48933547735214233), (u'men', 0.4632953703403473), (u'person',
0.457422494888830566), (u'lady', 0.4487500488758087), (u'himself',
0.4288588762283325), (u'girl', 0.4166809320449829), (u'his',
0.3853422999382019), (u'he', 0.38293731212615967)]

>>> model.most_similar("queen")
[(u'princess', 0.519856333732605), (u'latifah', 0.476443171501159967),
(u'prince', 0.459142267704001), (u'king', 0.4466976821422577), (u'elizabeth',
0.4134873151779175), (u'antoinette', 0.41033703088760376), (u'marie',
0.4061327874660492), (u'stepmother', 0.4040161967277527), (u'belle',
0.388272881507873354), (u'lovely', 0.3866859376430511 5)]
```

آشنایی با TensorFlow

یک کتابخانه رایگان و متن باز به منظور پیاده سازی الگوریتم های یادگیری ماشین

- برای
  - دسک تاپ ، موبایل ، وب و ابر
  - را برای مبتدیان و متخصصان آسان می کند

TensorFlow

پایه و اساس TensorFlow را با آموزش هایی برای مبتدیان و متخصصان یاد بگیرید تا به شما در ایجاد پروژه یادگیری بعدی ماشین خود کمک کنند.

برای JavaScript

از TensorFlow.js برای ایجاد مدلهای جدید یادگیری ماشین و استقرار مدل های موجود با JavaScript استفاده کنید.

برای موبایل و IoT

استنتاج را با TensorFlow Lite در دستگاه های تلفن همراه و جاسازی شده مانند Edge ، iOS ، Android و TPU و Raspberry Pi انجام دهید.

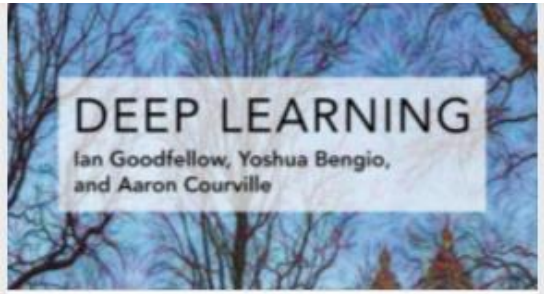
برای تولید

با استفاده از TensorFlow Extended (TFX) یک خط لوله ML آماده تولید برای آموزش و استنباط را مستقر کنید.

سوئیفت برای TensorFlow

به طور مستقیم با Swift برای TensorFlow ، بستر نسل بعدی برای یادگیری عمیق و برنامه نویسی متفاوت ، ادغام شوید.

کتب مفید



کتابها

Deep Learning: یک کتاب مطبوعاتی MIT ، توسط Ian Goodfellow ، Yoshua Bengio و Aaron Courville

این کتاب درسی Deep Learning منبعی است که برای کمک به دانشجویان و دست اندرکاران در زمینه یادگیری ماشین بطور کلی و یادگیری عمیق بطور خاص فراهم می شود.

کتابها

Hands-on Learning Machine با Scikit-Learn ، Keras و TensorFlow ، نسخه 2 ، توسط Aurélien Géron

این کتاب با استفاده از نمونه های بتن ، نظریه حداقل ، و دو چارچوب آماده تولید پایتون — Scikit-Learn و TensorFlow helps به شما کمک می کند تا درک بصری از مفاهیم و ابزارهای ساخت سیستم های هوشمند را بدست آورید.

کتابها

Deep Learning with Python ، توسط فرانسوا چولت

این کتاب یک مقدمه عملی و مفید برای Deep Learning with Keras است.

تفسیر و ارزیابی خروجی متن کاوی

تشکیل Confusion Matrix

معیار Accuracy: دقت به این معناست که مدل تا چه اندازه خروجی را درست پیش‌بینی می‌کند

معیار Precision: وقتی که مدل نتیجه را مثبت پیش‌بینی می‌کند، این نتیجه تا چه اندازه درست است

معیار Recall: زمانی که ارزش False Negatives بالا باشد، معیار Recall، معیار مناسبی خواهد بود.

معیار F1: یک معیار مناسب برای ارزیابی دقت یک آزمایش است.

- این معیارPrecision و Recall را با هم در نظر می‌گیرد.
- معیارF1 در بهترین حالت، یک و در بدترین حالت صفر است.