Institute of Data

2021

© 2021 Institute of Data

Data Science and AI

Module 9

Natural Language Processing (NLP)

© 2021 Institute of Data

Agenda: Module 9 - NLP

- Overview
- Regular Expressions
- Web Scraping
- Working with Text
- Sentiment Analysis
- Text Classification

© 2021 Institute of Data

NLP - Overview

- Natural Language Processing (NLP) focuses on enhancing **interactions** between human (natural) languages and computers. Specifically how to use computers to process natural language data.

- NLP uses techniques many fields such information engineering, computer science and artificial intelligence.

- Challenges in natural language processing frequently involve:

  - **Natural language understanding** (NLU), and

  - **Natural language generation** (NLG)

© 2021 Institute of Data

Rule-based vs Statistical NLP

- Initially, most language processing systems were designed by **hand-coding a set of rules**, e.g. by writing grammars or devising **heuristic rules** for parsing.
- Since the so-called “statistical revolution”, much natural language processing research has relied heavily on **statistical machine learning.**
- The earliest algorithms, such as decision trees, produced systems of hard if-then rules similar to the methods of hand-written rules.
- The machine-learning paradigm uses statistical inference to automatically learn the rules through the analysis of large **corpus** of typical real-world examples.
    - A corpus (plural, “corpora”) is a set of documents, possibly with human or computer annotations.

© 2021 Institute of Data

NLP – Key tasks

- **Syntax**
  - Breaking text into **sentences**, **words**, **tokens**.
  - **Stemming**: Reduce the inflected words to their word stem, base or root form.
  - **Lemmatisation**: Removal of inflectional endings return the base dictionary form of a word (lemma).
  - **Part-of-speech** tagging: Determine the part of speech for each word in a sentence.
- **Semantics**
  - **Named Entity Recognition** (NER): Determine which items in a text map to proper names and what such name is (e.g. person, location, organisation)
  - **Sentiment analysis**: Gather subjective information from a set of documents, to establish “polarity” about known objects
  - **Question answering**: Answer a human-language question.

© 2021 Institute of Data

NLP - Applications

- Chatbot
- Sentiment analysis
- Information extraction (IE)
- Summarisation
- Spam detection
- Document classification
- Spelling correction
- Machine Translation

© 2021 Institute of Data

NLP – Why is it too hard

- **Ambiguity** is pervasive
- **Synonymy**: One concept maps to many words
- **Polysemy**: One word maps to many concepts, e.g. bat
- **Idioms**, Metaphors, sarcasm, irony, slang, jargon, etc.
- World knowledge (**common sense**)
- Tricky entity names

- Examples of **ambiguous** newspaper headlines:
  - Drunk gets nine years in violin case
  - Stolen painting found by tree
  - Red tape holds up new bridge
  - Deer kill 30,000
  - Include children when baking cookies
  - Miners refuse to work after death
  - Prostitutes appeal to Pope

© 2021 Institute of Data

NLP – What is ahead

- Regular expression
- Web scraping
- Working with Text
- Sentiment Analysis
- Text Classification

© 2021 Institute of Data

Regular expressions

- **Overview**
- **Syntax**
- **Lab**

© 2021 Institute of Data

Regular expressions

- A **formal language** for specifying patterns text strings. It is used to identify whether a pattern exists in a given sequence of characters or not.
- Regular expressions play a surprisingly **large role in NLP** applications.
- Sophisticated sequences of regular expressions are often the **first model** for any text processing text
- For harder tasks, we use **Machine Learning classifiers**.
- Regular expressions could be used as **features** in the classifiers.
- Many programming languages provide regex capabilities, built-in or via libraries
  - Python has the **'re’** standard library and a **'regex'** package with extra functionality.

© 2021 Institute of Data

Text data - Encoding

- **Unicode**: Computing standard for consistent encoding, representation, and handling of text expressed in writing systems
  - Unicode 11.0 contains 137,439 characters over 146 modern and historic scripts, various symbol sets and emoji

- **UTF-8**: Variable width character encoding
  - The name derives from Unicode Transformation Format - 8-bit.

- **Older encoding**
  - There are many older encoding protocols such as ASCII, EBCDIC, etc

© 2021 Institute of Data

Regular expressions – Functions

- **Split()** splits text
- **Search()** scans through a string, looking for any location where this RE matches.
- **Findall()** **f**inds all substrings where the RE matches, and returns them as a list.
- **finditer()** finds all substrings where the RE matches, and returns them as an iterator.
- **Match()** returns a match object on success, None on failure. We use group(num) or groups() function of match object to get matched expression
- **sub()** to replace texts with another
- Regular expressions may be compiled into pattern objects using **compile()**.

© 2021 Institute of Data

Regular expressions – key rules

- **Ordinary characters** are the simplest regular expressions. They match themselves exactly.
- Letters inside square **brackets []** matches
  - [wW]oodstock matched woodstock and Woodstock
  - [0123456789] matches any
- **Range**
  - [A-Z] matches any uppercase letter
- **Negation**
  - [^A-Z] not an uppercase letter
- **Or** (pipe symbol)
  - [a|b] matches a or b

© 2021 Institute of Data

Regular expressions – key rules

- **Predefined character classes**
  - . Any character.
  - \d A digit: [0-9]
  - \D A non-digit: [^0-9]
  - \s A whitespace character: [\t\n\x0B\f\r]
  - \S A non-whitespace character: [^\s]
  - \w A word character: [a-zA-Z_0-9]
  - \W A non-word character: [^\w]

© 2021 Institute of Data

Regular expressions – key rules

- **Boundary characters**
  - ^ The beginning of a line.
  - $ The end of a line.
  - \b A word boundary.
  - \B A non-word boundary.
  - \A The beginning of the input.
  - \G The end of the previous match.
  - \Z The end of the input but for the final terminator, if any. \z The end of the input

© 2021 Institute of Data

Demo 9.1: Encoding

- Purpose
  - Work encoding and decoding UTF-8 content

- Materials
  - Jupyter Notebook (Demo-9_1)

© 2021 Institute of Data

Demo 9.2: Regular Expression

- Purpose
  - Overall idea of dealing with regular expression

- Materials
  - Jupyter Notebook (Demo-9_2)

© 2021 Institute of Data

Lab 9.1: Regular Expression

- Purpose
  - Practice with Regular Expressions

- Materials
  - Jupyter Notebook (Lab-9_1)

© 2021 Institute of Data

Web scraping

- **Uses**
- **Techniques**
- **Tools**
- **Legal Issues**
- **Prevention**

© 2021 Institute of Data

Web Scraping

- Data retrieval used for extracting data from **websites**.
- Access the World Wide Web (www) directly using the Hypertext Transfer Protocol (**HTTP**)
- Can be done by automated processes using a **web crawler** or manually.
- Data is collected and copied from the web, into a local database or flat files, for later retrieval or analysis.
- Involves **fetching** a page and **extracting** information from it.
    - Fetching: download of a page.
    - Extraction: parse, search, reformatted or save.

© 2021 Institute of Data

Web Scraping - uses

- Price change monitoring and comparison
- Product review
- Gathering real estate listings
- Weather data monitoring
- Website change detection
- Research
- Tracking online presence
- Reputation profiling
- Web data integration
- Contact retrieval

© 2021 Institute of Data

Web Scraping - techniques

- **HTTP programming:** Retrieved by posting HTTP requests to the remote web server using socket programming
- **HTML Document Object Model (DOM) parsing**: Parsing of the Document Object Model page's tree structure

© 2021 Institute of Data

Web Scraping - Tools

- Typical command line utilities common in Unix and Unix like systems
  - **cURL, wget**
- Libraries, either generic or common to a programming language
  - **Beautiful Soup**
- Programs designed for specifically for scraping
- Web browser add-ons
- Bots

© 2021 Institute of Data

Web Scraping – legal issues

- The legality of web scraping varies across the world
- Web scraping may be against the terms of use of some websites, but enforceability is unclear
- **United States**: Websites can use three major legal claims to prevent undesired web scraping: 1) copyright infringement, 2) violation of the Computer Fraud and Abuse Act (CFAA), and 3) trespass to chattel
- **The EU**: Varies but a court ruled that the terms and conditions were plainly visible and that placing the onus on the user to agree to terms and conditions to gain access to online services is sufficient to comprise a contractual relationship
- **Australia**: The Spam Act 2003 outlaws some forms of web harvesting, but restricted to email addresses

© 2021 Institute of Data

Demo 9.3: Web Scraping

- Purpose
  - Overall idea of dealing with web pages

- Resources
  - BeautifulSoup

- Materials
  - Jupyter Notebook (Demo-9_3)

© 2021 Institute of Data

Lab 9.3: Web Scraping

- Purpose
  - Practice with Web Scraping
- Resources
  - BeautifulSoup
- Materials
  - Jupyter Notebook (Lab-9_3)

© 2021 Institute of Data

Working with Text

- Tokenisation
- Stop Words
- Stemming and Lemmatisation
- Parsing and Tagging
- NLP Packages (NLTK, SpaCy)

© 2021 Institute of Data

Tokenisation

- Text components
  - **Corpus:** a set of documents within a context
  - **Document**: a container of text
  - **Span**: a slice of text (sentence, phrase, multi-token term)
  - **Token**: An individual token — i.e. a word, punctuation symbol, whitespace, etc.

© 2021 Institute of Data

Tokenisation

- Break up a string into logical chunks known as **tokens**
- Based on rules and recently also on statistical models
- Rules vary with the use and intent
- Examples
  - Words of a sentence
  - Remove punctuation
  - Identify or remove tags from markup languages

© 2021 Institute of Data

Tokenisation

- The main question of the tokenisation phase is “what are the correct tokens to use?”
  - Initially, it is relatively trivial: chop on **whitespace** and throw away punctuation characters
  - White space splits compound words
    - Proper nouns: New South Wales; compound expression: Bon appétit
    - Phone numbers: 1300 23-2333; dates 11 Mar 2013
    - York University opposed to New York University
  - Different spelling
    - Lowercase, lower-case and lower case

© 2021 Institute of Data

Tokenisation - Language issues

- German writes compound nouns without spaces
  - Computerlinguistik (computational linguistics)
  - Lebensversicherungsgesellschaftsangestellter (life insurance company employee)

- French has an use of the apostrophe for a reduced definite article before a word beginning with a vowel
  - “l’ensemble” and “un ensemble”

- Major East Asian Languages (e.g., Korean, Chinese, Thai, and Japanese), where texts written without any spaces between words
  - A approach is to forgo word-based indexing and to do the indexing via short strips of characters (character -grams)
  - Another approach is to perform word segmentation as prior linguistic processing

© 2021 Institute of Data

Tokenisation

- In English, there are many extreme cases
  - For example: How to handle distinct uses of the apostrophe for possession and contractions?

- **“Mr O’Connor thinks that the girls’ stories about Peru’s capital aren’t amusing.”**
  - What is the desired tokenisation?

O’Connor	connor	oconnor	o’connor	o’ connor	o connor
aren’t	aren’t	arent	are n’t	aren’ t	aren t

© 2021 Institute of Data

Stop Words

- Vastly common words which carry little value in selecting documents matching a user need are usually removed from the vocabulary entirely
- The strategy for determining a stop list is to sort the terms by their frequency and take the most common terms.
- Using a stop word’s list significantly reduces the number of terms that a system has to store
- Frequently, not indexing stop words does little harm: keyword searches with terms like “the” and “by” do not seem very useful.
- This is not true for phrase searches
  - The meaning of “flights to Melbourne” is likely to be lost if the word “to” is removed.
- Most of the modern NLP libraries include definition of the stop words.

© 2021 Institute of Data

Stemming and Lemmatisation

- For grammatical reasons, documents use **different forms** of a word, such as “organise”, “organises”, and “organising”

- In many situations, it would be relevant for a search of one of these words to retrieve documents that contain another word in the set

- Also, there are groups of derivationally related words with equivalent meanings, such as “autocracy”, “autocratic”, and “autocrat”

Source: Stanford-NLP

© 2021 Institute of Data

Stemming and Lemmatisation

- The goal of both **Stemming** and **Lemmatisation** is to reduce forms of inflection and sometimes forms derivationally related of a word to a common radical

am	are	is		»	**be**
car	cars	car’s	cars’	»	**car**

- The result of this mapping of text is something like

Original	The	boy’s	car	are	of	different	colours
Modified	The	boy	car	be	of	differ	colour

© 2021 Institute of Data

Stemming and Lemmatisation

- Stemming refers to a **simple heuristic** process that cuts off the last characters of words, which most of the time includes the removal of derivational affixes
  - The token “saw” might become just “s”

- Lemmatisation usually refers to removing the suffixes with the use of **vocabulary** and **morphology** of words, to remove inflectional endings only and to keep the root or the lexicon form of a word, known as the lemma
  - The token “saw” might become either “see” or “saw”, depending on the token being a verb or a noun

© 2021 Institute of Data

Stemming and Lemmatisation

- The **Porter's algorithm** is the most common for stemming English and is also very effective
  - The entire algorithm is too long
  - It consists of five sequential phases of word reductions
  - There are various conventions to select rules at each phase, like selecting the rule that applies to the longest suffix from each rule group

© 2021 Institute of Data

Stemming and Lemmatisation

- **Sample text:** Such an analysis can reveal features that are not easily visible from the variation in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

- **Lovins stemmer:** Such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpres

- **Porter stemmer:** Such an analysi can reveal featur that ar not easili visibl from the variat in the individu genes and can lead to a pictur of express that is more biolog transpar and access to interpret

- **Paice stemmer:** Such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret

Source: Stanford-NLP

© 2021 Institute of Data

Parsing and Tagging

- For most language, syntax and structure go hand in hand

- A set of precise **rules, principles and conventions** govern the way how
  - Words are organised into phrases
  - Phrases get organised into clauses, and
  - Clauses get organised into sentences

- In English, words usually combine to form other participant units
  - These participants include **words, clauses, phrases, and sentences**

© 2021 Institute of Data

Parsing and Tagging

- Parsing
  - Do the grammatical analysis by creating the **parse tree** of a given sentence
  - The grammar of sentences in natural languages have multiple possible analyses making them **ambiguous**
  - There are two primary types of parsing
    - Constituency Parsing builds the Parse Tree using a **Probabilistic Context-Free Grammar** (PCFG)
    - **Dependency Parsing** looks for the relationships between words in a sentence (tagging objects like Primary Objects and predicates)

© 2021 Institute of Data

Parsing and Tagging

- Considering the expression
  - **“The brown fox is quick, and he is jumping over the lazy dog.”**
- Words alone by themselves do not tell much

Text	The	brown	fox	is	quick	and	he	is	jumping	over	the	lazy	dog
PoS Tag	DET	ADJ	N	V	ADJ	CONJ	PRON	V	V	ADV	DET	ADJ	N

© 2021 Institute of Data

Parsing and Tagging

- **Parts of speech** (POS) are defined lexical categories assigned to words based on their role and syntactic context

- Major categories are nouns (N), verbs (V), adjectives (ADJ) and adverbs (ADV)

- Other categories that frequently occur in the English language are pronouns, prepositions, interjections, conjunctions, determiners, and others

© 2021 Institute of Data

Parsing and Tagging - Main PoS Categories

- **N(oun, tag: N)**: Words that depict some object or entity, which may be living or nonliving (fox, dog, book). Can be subdivided as singular nouns (NN), singular proper nouns (NNP), and plural nouns (NNS)

- **V(erb, tag: V):** Words that are used to describe certain actions, states, or occurrences (running, jumping, read, and write)

- **Adj(ective, tag: ADJ):** Words used to describe or qualify other words, typically nouns (in “beautiful flower" beautiful (ADJ) describes or qualifies flower (N))

- **Adv(erb, tag: ADV):** Usually act as modifiers for other words including nouns, adjectives, verbs, or other adverbs (in “very beautiful flower” very (ADV) modifies beautiful (ADJ) indicating its degree)

© 2021 Institute of Data

Named-Entity Recognition

- Named-entity recognition (NER) is also known as entity identification, entity chunking and entity extraction

- Subtask that seeks to **locate** and **classify** named entity mentions in unstructured text into pre-defined categories such as the **person** names, organisations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc.

- State-of-the-art NER systems for English produce near-human performance.

© 2021 Institute of Data

Named-Entity Recognition

- **"He killed the two officers, Marla Finn and William Hodges, then disposed of their bodies in the plasma stream."**
  - two (CARDINAL)
  - Marla Finn (PERSON)
  - William Hodges (PERSON)

- Some entities have more than one token

© 2021 Institute of Data

Named-Entity Recognition

- **“Captain Jean-Luc Picard took command of the ship on stardate 41148 at the order of Rear Admiral Norah Satie.”**
  - Jean-Luc Picard (PERSON)
  - 41148 (DATE)
  - Norah Satie (PERSON)

- Some entities have more than one token
- A “Stardate” was recognised

© 2021 Institute of Data

NLP Packages

- Most NLP techniques require the pre-processing of large collections of annotated text to learn specific language rules

- There are many tools available for English and other common languages

- Each tool typically requires a large amount of data and large databases of particular use-cases, including language inconsistencies and slang

- In Python, two popular NLP packages are NLTK and spaCy
  - NLTK is more popular but not as advanced and well maintained
  - spaCy is more modern and build for industrial-strength performance

© 2021 Institute of Data

SpaCy

- spaCy is an open-source software library for advanced NLP, written in the programming languages Python and Cython.
- It offers statistical neural network models for English and a number of other languages.
- Unlike NLTK, which is widely used for teaching and research, spaCy focuses on providing software for production usage. This means that spaCy is integrated and opinionated. spaCy tries to avoid asking the user to choose between multiple algorithms that deliver equivalent functionality. Keeping the menu small lets spaCy deliver generally better performance and developer experience.
- Main features include tokenisation, NER, pretrained word vectors, POS tagging, text classification and syntax visualisation.

© 2021 Institute of Data

Word embedding and word to Vector representation

- When we are looking at words (tokens) alone, it is difficult for a machine to understand connections that a human would understand immediately.
- Word to vector (word2vec) is a token representation captures joint-occurrence, similarity and connections between tokens based on a **corpus** of text.
- Word embedding and dimensionality reduction are used to produce word vectors from a given corpus.
- Word embedding is the technique of hot-one encoding of all word in a given text.
- A word vector is a numeric representation of a word that indicates fits relationship to other words. Each word is interpreted as a unique and lengthy array of numbers.
- Word2Vec technique was first described in a paper “Efficient Estimation of Word Representations in Vector Space” by Thomas Mikolov et al from Google.

© 2021 Institute of Data

Lab 9.5: Working with Text

- Purpose
  - Practice with NLP libraries

- Resources
  - NLTK
  - spaCy

- Materials
  - Jupyter Notebook (Lab-9_5)

© 2021 Institute of Data

Sentiment Analysis

- Scope
- Types
- Algorithms
- Feature Extraction
- Challenges
- Applications

© 2021 Institute of Data

Sentiment Analysis

- It is a part of NLP that builds systems to **identify** and **extract** opinions from a text
- It has become a vital tool for making sense of that data and has allowed companies to get critical insights and **automate** all kind of processes
- These systems extract attributes of the expression aside of assessing opinion
  - Opinion holder: the individual or entity that expresses the opinion
  - Subject: the target the text refers to
  - Polarity: if the speaker expresses a **positive** or **negative** opinion

Source/Reference: Monkey Learn

© 2021 Institute of Data

Sentiment Analysis - Opinion

- Text information can be broadly categorised into two main types
  - Facts are objective expressions about something
  - Opinions are subjective expressions that describe how people feels, assesses or values a subject or topic

- Sentiment analysis can be modelled as a **classification** problem where two sub-problems must be solved
  - Subjectivity classification: Labelling a sentence as **objective** or **subjective**
  - Polarity classification: Labelling a sentence as expressing a **positive**, **negative** or **neutral** opinion

© 2021 Institute of Data

Sentiment Analysis - Direct/Comparative Opinion

- Direct opinions give an opinion about an entity **directly**
  - “The screen resolution of notebook A is **bad**.”
  - States a **negative** opinion about notebook A

- **Comparative** opinions are expressed by comparing an entity with another
  - “The screen resolution of notebook A is **better** than that of notebook B.”
  - Express differences or similarities between two or more entities using a superlative or comparative form of an adjective or adverb
  - In the example, there is a **favourable** opinion about notebook A and, conversely, **negative** opinion about notebook B

© 2021 Institute of Data

Sentiment Analysis - Explicit/Implicit Opinion

- An explicit opinion on a subject is expressed in a subjective sentence
- The following sentence expresses an explicit favourable opinion
  - “The sound quality of this speaker is **amazing**.”
- An implicit opinion on a subject is in an objective sentence
- The next sentence expresses an implied negative opinion
  - “The glass frame **broke** in two days.”
- Implicit opinions can include metaphors that may be the most **challenging** type of opinions to analyse as they include much semantic information

© 2021 Institute of Data

Sentiment Analysis - Scope

- Sentiment analysis applies to different **scopes**
  - **Document level**: The sentiment of a complete document or paragraph
  - **Sentence level**: The sentiment of a single sentence
  - **Sub-sentence level**: The sentiment of sub-expressions within a sentence

© 2021 Institute of Data

Sentiment Analysis - Types

- Polarity ([very] positive, neutral, [very] negative)
  - Some systems provide different flavours of polarity associated with positive feelings (i.e. love, happiness or enthusiasm) or negative feelings (i.e. sadness, anger or worries)

- Emotion detection such as happiness, frustration, anger, sadness, and the like
  - Many systems resort to **lexicons** (i.e. lists of words and the emotions they convey) or complex machine learning algorithms
  - A downside of lexicons is that people expression varies a lot (i.e. “kill”)
    - “Customer support is **killing** me” (negative)
    - “You are **killing** it” (positive)

© 2021 Institute of Data

Sentiment Analysis - Types

- Aspect-based, i.e. about particular aspects or features of a product
  - “The battery life of this phone is too short.”
  - An adverse opinion about the phone, but more precisely, about the battery life

- Intent analysis detects what people want rather than what they say with that text
    - “The customer support is a disaster. I’ve been on hold for 20 minutes” (complaint)
    - “I would like to know how to swap the batteries” (question)
    - “Can you help me fill out this application?” (request)
  - A human has no problems detecting the differences while machines do
  - Sometimes the intent can be inferred from the text in others it requires context

© 2021 Institute of Data

Sentiment Analysis - Algorithms

- Sentiment analysis’ methods and algorithms can be categorised as
  - Automatic systems that learn from data using machine learning techniques
  - Rule-based systems that execute sentiment analysis using a set of manually written rules
  - Hybrid systems that combine both automatic and rule-based approaches

© 2021 Institute of Data

Sentiment Analysis - Algorithms - Rule-based

- Rule-based defines a set of rules to identify polarity, subjectivity, or the subject of an opinion

- The rules may use a mix of inputs, such as the following
  - Classic NLP techniques like **tokenisation**, stemming, parsing and part of speech tagging
  - Other resources like **dictionaries** (i.e. lists of words and expressions)

- These systems are very hard to maintain as new rules are needed to add support for new expressions and vocabulary

© 2021 Institute of Data

Sentiment Analysis - Algorithms - Rule-based

- A basic example of a rule-based implementation
  1. Define lists of **polarised** words (e.g. negative words such as evil, worst, ugly, and positive words such as good, best, sweet)
  2. Given a text **Count** the numbers of both positive and negative words that appear in the text
  3. If the number of positive words is greater than the number of negative word return a positive sentiment, if lesser, return a negative sentiment, if equal, return neutral

- This system is **very simplistic** as it does not take into account how words are combined in a sequence

© 2021 Institute of Data

Sentiment Analysis - Algorithms - Automatic

- Automatic methods rely on **machine learning** techniques

- The sentiment analysis is usually modelled as a classification problem where a **classifier** is fed with text and returns the corresponding category, e.g. positive, negative, or neutral

  a. **Train** a model to associate a particular input (i.e. a text) to the corresponding output (label) based on test samples

  b. **Predict** unseen text inputs as positive, negative, or neutral

© 2021 Institute of Data

Sentiment Analysis - Feature Extraction

- Transform the text into a numerical representation
  - Usually, each component represents the frequency of a word or expression in a predefined dictionary (e.g. a lexicon of polarised words)

- This task is known as text **vectorisation** or feature extraction
  - The classical approaches are **bag-of-words** or **bag-of-n-grams** with their frequency

- New feature extraction approaches have been used based on **word embeddings**, also known as word vectors
  - This representation makes allow words with similar meaning to have a close measures, which can improve the performance of classifiers

© 2021 Institute of Data

Sentiment Analysis - Classification Algorithms

- Naïve Bayes: a group of probabilistic algorithms that uses Bayes's Theorem to predict the class of a text
- Linear Regression: an algorithm used to predict a value *y* given a set of variables *X*
- Support Vector Machines: a non-probabilistic model that represents text examples as points in a multidimensional space
  - The examples are mapped so that different categories (sentiments) belong to distinct regions of that space, so new texts are mapped onto that same space and classified to a category based on the respective region
- Deep Learning: a set of algorithms that attempts to imitate how the human brain works by employing artificial neural networks to process data

© 2021 Institute of Data

Sentiment Analysis - Challenges

- Subjectivity and **Tone**
  - "The package is nice."
  - "The package is red."
  - All predicates should not be treated the same; above, nice is more subjective than red

- Context and Polarity
  - Consider the questions
    - "What did you like about the event?"
    - "What did you **DIS**like about the event?"
  - And answers to the questions above, like
    - "Everything of it."
    - "Absolutely nothing!"

© 2021 Institute of Data

Sentiment Analysis - Challenges

- **Irony** and **Sarcasm**, consider
  - “Yeah. Sure.”
  - “Not one, but many!”
  - As answers to the question “Have you had a nice customer experience?”

- Comparisons
  - “This product is second to none.” (does not need context)
  - “This is better than old tools.” (are the old tools just old or are they inferior?)

© 2021 Institute of Data

Sentiment Analysis - Challenges

- **Emojis**
  - There are emojis encoded in only one character or a couple of them (e.g. :D) or longer combination of characters of a vertical nature (e.g. ¯ \ _ (ツ) _ / ¯)
  - Emojis represent **sentiment** using characters, particularly in Tweets
  - Sentiment analysis over Tweets requires attention to characters as well as words
  - Much pre-processing might be needed
    - **Preprocess** social media content and transform emojis into **tokens** to help improve sentiment analysis performance

© 2021 Institute of Data

Sentiment Analysis - Applications

- Social media monitoring
- Brand monitoring
- Voice of the customer (VoC)
- Customer service
- Workforce analytics and voice of the employee
- Product analytics
- Market research and analysis

© 2021 Institute of Data

Lab 9.6: Sentiment analysis

- Purpose
  - Practice with NLP libraries to identify sentiment

- Resources
  - NLTK
  - spaCy

- Materials
  - Jupyter Notebook (Lab-9_6)

© 2021 Institute of Data

Text Classification

- Supervised
- Feature Engineering
- Model Training
- Improve Models
- Unsupervised

© 2021 Institute of Data

Overview

- Vastly used NLP task in different business problems
- The goal is to **classify** text content into some categories automatically (either defined or unknown
- Examples of text classification
  - Understanding audience **sentiment** from social media
  - **Detection** of spam and non-spam emails
  - **Auto-tagging** of customer queries
  - The categorisation of news articles into defined **topics**

Source/Reference: Monkey Learn

© 2021 Institute of Data

Text Classification - Supervised

- Defined classification **categories**
- Input data has **labels** with email spam filtering being an example
- Works on training and **testing** principle
- **Language detection**, emotion, **intent** and sentiment analysis are all based on supervised learning systems
- Supervised classification is comparable to asking computers to imitate humans
  - The algorithms generate **AI models** from a given set of categorised text
  - These models then automatically classify new untagged text

© 2021 Institute of Data

Text Classification - Supervised Workflow

1. Dataset Preparation: Loading a dataset and performing necessary pre-processing, then splitting into train and test sets

2. Feature Engineering: The raw dataset is transformed into flat features that can be used in a machine learning model and also creating new features from the existing data

3. Model Training: A machine learning model is trained on a labelled dataset

4. Improve Performance: Try to improve the performance of the models

© 2021 Institute of Data

Text Classification - Feature Engineering

- Transform raw text data into feature vectors

- Common approaches
  - Count Vectors as features
  - **TF-IDF** (Term Frequency–Inverse Document Frequency) vectors as features
    - Word level
    - N-gram level
    - Character level
  - Word Embeddings as features
  - Text/NLP based features
  - **Topic** Models as features

© 2021 Institute of Data

Text Classification - Count Vectors as Features

- **Count Vector** represents the dataset in a **matrix** form
  - **Rows** represent a **document** from the corpus
  - **Columns** represent a **term (token)** from the corpus
  - Every cell has the frequency count of a particular term in a particular document

- Also known as **Bag of Words** (BoW)

© 2021 Institute of Data

Text Classification - TF-IDF Vectors as Features

- TF-IDF represents the relative **importance** of a term in the document and the entire corpus.

- TF-IDF is a numerical **statistic** that is intended to reflect the **importance** of a word is to a document in a collection or corpus. Its value increases proportionally to the **number of times a word appears in the document** and is **offset** by the **number of documents** in the corpus that contain the word.

- Distinct levels of input tokens generate TF-IDF Vectors from
  - Word: Matrix of TF-IDF scores of every term in different documents
  - N-gram (N-grams are N terms together): Matrix of TF-IDF scores of N-grams
  - Character: Matrix of TF-IDF scores of character level n-grams in the corpus

© 2021 Institute of Data

Text Classification - Word Embeddings as Features

- TF-IDF is composed by
  - Term Frequency (TF): occurrences of term ***t*** in document ***d*** (***$n_{t,d}$***) over the number of all terms in that document (***T_d***)

$$TF_{t,d} = \frac{n_{t,d}}{T_d}$$

  - Inverse Document Frequency (IDF): Number of documents with term ***t*** (***DF_t***) related to all documents in the corpus (***D***)

$$IDF_t = \log\left(\frac{D}{DF_t}\right)$$

- The weight ***$W_{t,d}$*** of term ***t*** in document ***d*** is given by $W_{t,d} = TF_{t,d} \cdot IDF_t$

© 2021 Institute of Data

Text Classification - Word Embeddings as Features

- Word embedding represents words (**tokens**) and documents using a dense vector
- The position of a word in the vector space is learned from the text and is based on the **surrounding words**
- It can be trained using the input corpus itself or can be generated using pre-trained word embeddings such as Word2Vec, Glove or similar.
- Any pre-trained can be downloaded and used as **transfer learning**
- There are four essential steps to use pre-trained word embeddings in a model
  - Loading the pre-trained word embeddings
  - Creating a tokeniser object
  - Transforming text documents into a sequence of tokens
  - Create a mapping of tokens and their respective embeddings

© 2021 Institute of Data

Text Classification - Text / NLP Based Features

- Other text-based features could sometimes be helpful to improve text classification
  - Word Count: number of words in the documents
  - Character Count: number of characters in the documents
  - Average Word Density: average length of the words used in the documents
  - Punctuation Count: number of punctuation marks in the documents
  - Upper Case Count: number of upper count words in the documents
  - Title Word Count: number of proper case (title) words in the documents
  - Frequency distribution of Part of Speech Tags
    - Noun, Verb, Adjective, Adverb and Pronoun counts
- These features should be tested and used according to the problem

© 2021 Institute of Data

Text Classification - Topic Models as Features

- Topic Modelling is an approach to identify **groups of words** (i.e. topic) from a collection of documents that contains the best information in the collection

- **Latent Dirichlet Allocation** (LDA) is an iterative model
  - It starts with a **known** number of topics
  - **Topics** are represented by a distribution on **words**, and **documents** are represented by a distribution on topics
  - Although the tokens are meaningless by themselves, the **probability distributions** on words provided by the topics offers a notion of the **distinct ideas** contained in the documents

© 2021 Institute of Data

Text Classification - Model Training

- The final step is to train a classifier using the features created in the previous steps
- There are various choices of machine learning modelling techniques
  - Naïve Bayes Classifier
  - Linear Classifier
  - Support Vector Machine
  - Bagging and Boosting Models
  - Shallow and Deep Neural Networks
    - Convolutional Neural Network (CNN)
    - Long Short-Term Models (LSTM)
    - Gated Recurrent Unit (GRU)
    - Bidirectional RNN
    - Recurrent Convolutional Neural Network (RCNN)

© 2021 Institute of Data

Text Classification - Naïve Bayes

- Naïve Bayes is a classification technique based on Bayes' Theorem
- A Naïve Bayes classifier assumes that the presence of a particular feature in a class is **independent** to the presence of any other feature (independent predictors)
    - A thing the is round, red, about 10 cm in diameter and is a fruit can be identified as an apple
    - All of these features contribute independently to the probability that this fruit is an apple, regardless if these properties depend on each other or others
        - This is the reason is known as "Naïve"

© 2021 Institute of Data

Text Classification - Linear Classifier

- Linear Classifier (Logistic Regression)
- Measures the **association** between the factor dependent variable and one or more independent features
- Estimates **probabilities** using a logistic/sigmoid function

© 2021 Institute of Data

Text Classification - Support Vector Machine

- SVM is a supervised machine learning approach used for both regression and classification problems

- The model extracts the best possible **hyper-plane** / line that segregates the two classes

© 2021 Institute of Data

Text Classification - Bagging

- Bagging with **Random Forest** Model

- A popular type of **ensemble models** (bagging models) is Random Forest

- Part of the tree based model group

© 2021 Institute of Data

Text Classification - Boosting

- **Boosting** models are also ensemble models from tree-based models
- Boosting is a machine learning ensemble **meta-algorithm** for primarily reducing **bias**, and also **variance** in supervised learning
- Also a group of machine learning methods that convert weak into strong learners
    - A classifier is called a weak learner if is only slightly correlated with the actual classification (it can classify examples better than random guessing)
    - Boosting combines learnings from a number of week learners. When they are added, they are typically **weighted** in some way that is usually related to the weak learners' accuracy. Misclassified input data **gain a higher weight** and examples that are classified correctly lose weight.

© 2021 Institute of Data

Text Classification - Improve Models

- Text Cleaning: Helps to reduce the noise present in text data in the form of stop words, punctuations marks, suffix variations, and others
- Stacking Text/NLP Features: Generated some different feature vectors, combining them to improve accuracy
- Hyper-parameter Tuning: An important step it to optimise parameters like tree length, leaves, network parameters, and others
- Ensemble Models: Stacking different models and blending their outputs to improve the results further

© 2021 Institute of Data

Text Classification - Unsupervised

- Works without external information
- The algorithms try to **discover** natural structures in data
  - A natural structure might not be the same as humans think of as logical division
  - Looks for similar patterns in the values and combine them into **clusters**
  - The clusters formed are the basis for the classification of the data
- Every data point is embedded into a **hyperspace**
- The data exploration is done to find similar data points based on textual similarity
- It is highly customisable as no tagging is required
- Can operate on any textual data without training and tagging, therefore is language agnostic

© 2021 Institute of Data

Lab 9.7: Text Classification

- Purpose
  - Work with feature engineering and text classification

- Materials
  - Jupyter Notebook (Lab-9_7)

© 2021 Institute of Data

Questions?

© 2021 Institute of Data

Appendices

© 2021 Institute of Data

End of Presentation!

© 2021 Institute of Data