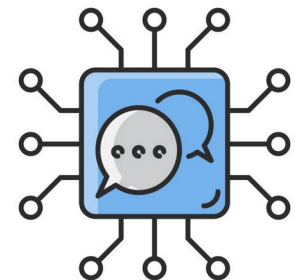


Deep Parsing and Tools for NLP

Tushar B. Kute,
<http://tusharkute.com>



Syntactic Parsing

- The language syntax is fundamental for generative text and sets the foundation for parts of speech and parse trees.
- The word syntax originates from the Greek word **syntaxis**, meaning “**arrangement**”, and refers to how the words are arranged together.
- Henceforth, language syntax means how the language is **structured or arranged**.

Syntactic Parsing

- There are many different ways to categorize these structures or arrangements.
- One way to classify how the words are arranged is by grouping them as the words behave as a single unit or phrase, which is also known as a **constituent**.
- A sentence can have different language rules applied to it and have different types of structure.
- As different parts of the sentence are based on different parts of the syntax that follow the same grammar rules that are of a noun phrase, verb phrase, and prepositional phrase.

Syntactic Parsing

- A sentence is structured as follows:

Sentence = S = Noun Phrase + Verb Phrase + Preposition Phrase

$$S = NP + VP + PP$$

- The different word groups that exist according to English grammar rules are:
 - Noun Phrase(NP): Determiner + Nominal Nouns = DET + Nominal
 - Verb Phrase (VP): Verb + range of combinations
 - Prepositional Phrase (PP): Preposition + Noun Phrase = P + NP

Syntactic Parsing

- We can make different forms and structures versions of the noun phrase, verb phrase, and prepositional phrase and join in a sentence.
- For instance, let us see a sentence:
- The boy ate the pancakes. This sentence has the following structure:
 - The boy: Noun Phrase
 - ate: Verb
 - the pancakes: Noun Phrase (Determiner + Noun)
- This sentence is correct both structurally and contextually.

Syntactic Parsing

- However, now taking another sentence: The boy ate the pancakes under the door.
 - The boy: Noun Phrase
 - ate: Verb
 - the pancakes: Noun Phrase (Determiner + Noun)
 - under: preposition
 - the door: Noun Phrase (Determiner + Noun)
- Here, the preposition under is followed by the noun phrase the door, which is syntactically correct but not correct contextually.

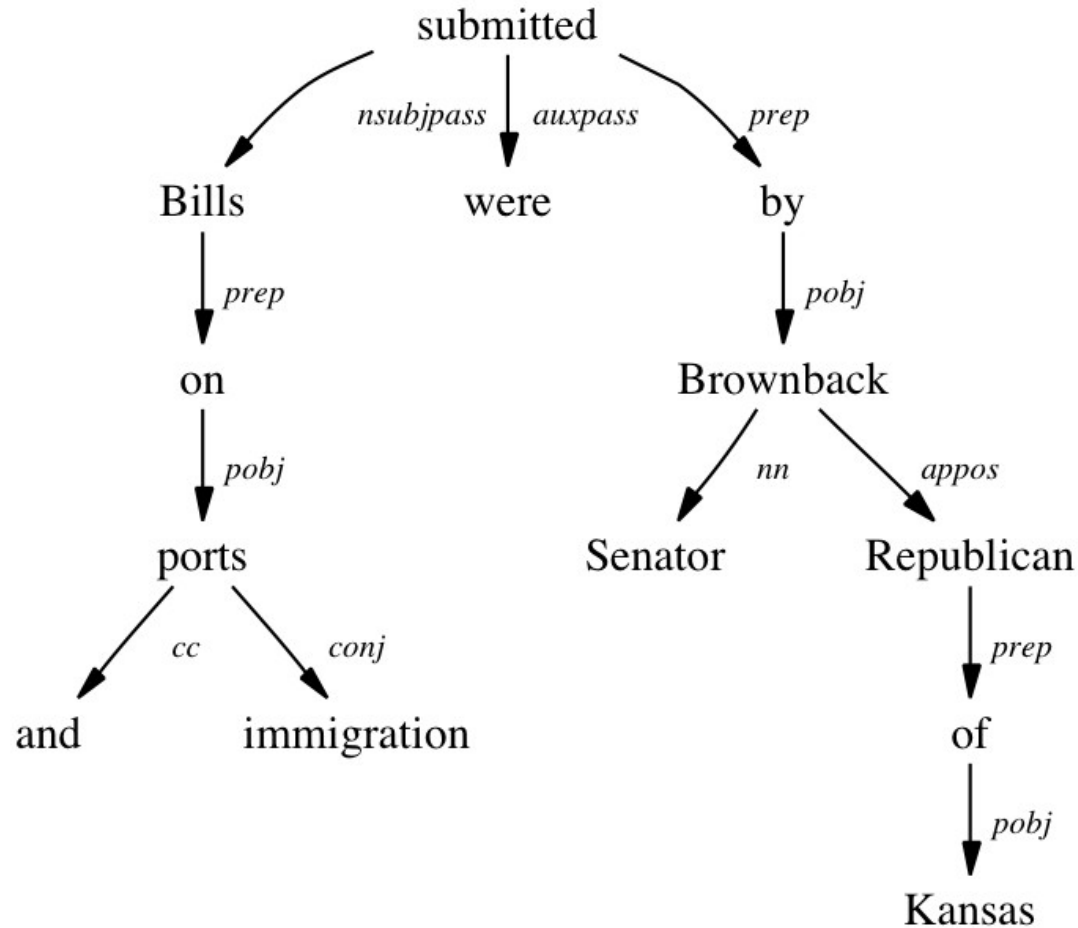
Syntactic Parsing

- Taking the same sentence in another way: The boy ate the pancakes from the jumping table.
 - The boy: Noun Phrase
 - ate: Verb
 - the pancakes: Noun Phrase (Determiner + Noun)
 - from: preposition
 - jumping table: Verb Phrase
- This sentence is syntactically incorrect as the preposition form is followed by a verb phrase jumping table.

Text Syntax Components

- There are two imperative attributes of text syntactic: Part of Speech tags and Dependency Grammar.
- **Part of Speech** tagging or POS tagging specifies the property or attribute of the word or token.
- Each word in a sentence is associated with a part of speech tag such as nouns, verbs, adjectives, adverbs.
- The POS tags define the usage and function of a word in the sentence.

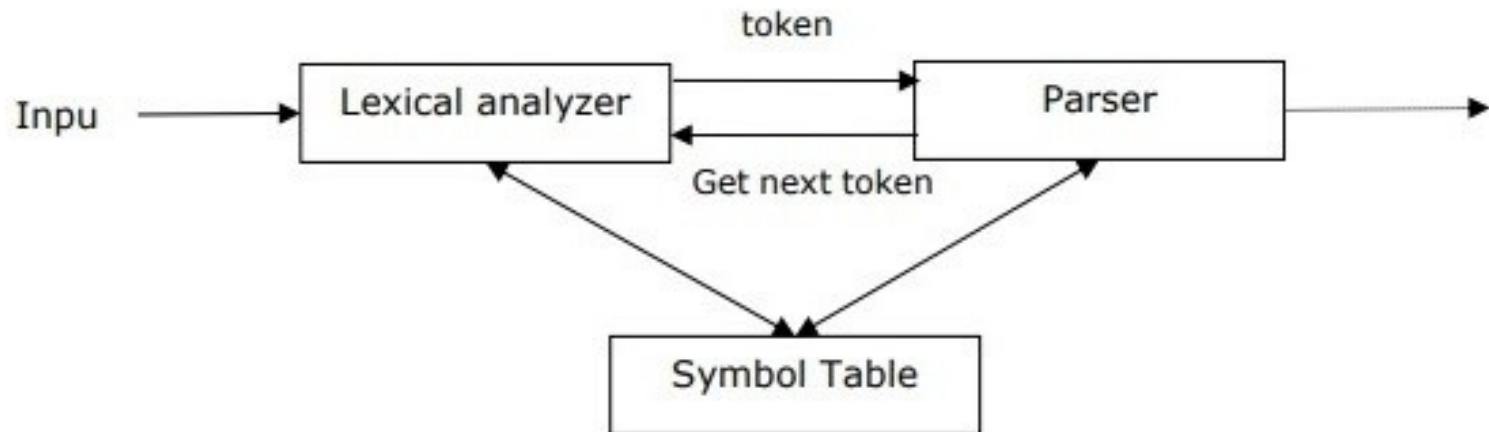
Dependency Tree



Role of a Parser

- It is used to implement the task of **parsing**.
- It may be defined as the software component designed for taking input data (text) and giving **structural representation** of the input after checking for correct syntax as per formal grammar.
- It also builds a data structure generally in the form of **parse tree** or **abstract syntax tree** or other hierarchical structure.

Role of a Parser



How does a parser work?

- The first step is to **identify the subject** of the sentence.
- As the parser splits the sequence of text into a bunch of words that are related in a sort of **phrase**. So, these bunch of words that we get that are related to each other is what is referred to as the subject.
- Syntactic parsing and parts of speech, these language structures are context-free grammar is based on the **structure or arrangement** of words. It is not based on the context.
- The important thing to note is that the grammar is always **syntactically correct** i.e. syntax wise and may not make contextual sense.

Syntactic Parsing

- Practical

Semantic Parsing

- Semantic parsing is the task of **converting a natural language utterance to a logical form**: a machine-understandable representation of its meaning.
- Semantic parsing can thus be understood as extracting the **precise meaning** of an utterance.
- Applications of semantic parsing include machine translation, question answering, ontology induction, automated reasoning, and code generation.
- The phrase was first used in the 1970s by **Yorick Wilks** as the basis for machine translation programs working with only semantic representations

Semantic Parsing

- Semantic frame parsing may be used for applications that needed to understand deeper about the **meaning** of words, like question answering.
- It tries to, determine **what** is the text talking about (oversimplified paraphrasing of frame) and **who** did **what** to **whom** (oversimplified paraphrasing of frame elements or semantic roles) around it.

Semantic Parsing

- Consider an example
 - [The price of bananas] increased [5%]
 - [The price of bananas] rose [5%]
 - There has been a [5%] rise in [the price of bananas]
- The phrases in the bracket are the arguments, while “increased”, “rose”, “rise” are the **predicates**.

Semantic Parsing

- All of these sentences mean the same thing, but how can a computer understand them? We wanted to be able to ask a computer, for example,
 - “How much has the price of bananas increased?”
- Given a mixed structure, it may be confused and couldn't find a correct answer.

Information Extraction

- The process of **sifting through unstructured data and extracting vital information** into more editable and structured data forms is known as information extraction.
- Working with a large volume of text data is usually stressful and time-consuming.
- As a result, many businesses and organizations rely on Information Extraction techniques to use clever NLP algorithms to automate manual tasks.
- Information extraction can save time and money by reducing human effort and making the process less error-prone and efficient.

Information Extraction

- Deep Learning and NLP techniques like Named Entity Recognition may be used to extract information from text input.
- If we're starting from scratch, though, we should evaluate the sort of data we'll be dealing with, such as bills or medical records.
- Information Extraction System is used in a variety of NLP-based applications.
- For example, extracting summaries from vast collections of text like Wikipedia, conversational AI systems like chatbots, extracting stock market announcements from financial news, and so on.

Web Scrapping

- Web scraping is the process of **collecting and parsing raw data from the Web**, and the Python community has come up with some pretty powerful web scraping tools.
- The Internet hosts perhaps the greatest source of information—and misinformation—on the planet.
- Many disciplines, such as data science, business intelligence, and investigative reporting, can benefit enormously from collecting and analyzing data from websites.

Scrape and Parse Text From Websites

- Collecting data from websites using an **automated process** is known as web scraping. Some websites explicitly forbid users from scraping their data with automated tools Websites do this for two possible reasons:
 - The site has a good reason to protect its data. For instance, Google Maps doesn't let you request too many results too quickly.
 - Making many repeated requests to a website's server may use up bandwidth, slowing down the website for other users and potentially overloading the server such that the website stops responding entirely.

Scrapping the Web

- Important: Before using your Python skills for web scraping, you should always check your target website's acceptable use policy to see if accessing the website with automated tools is a violation of its terms of use.
- Legally, web scraping against the wishes of a website is very much a gray area.

Python Package

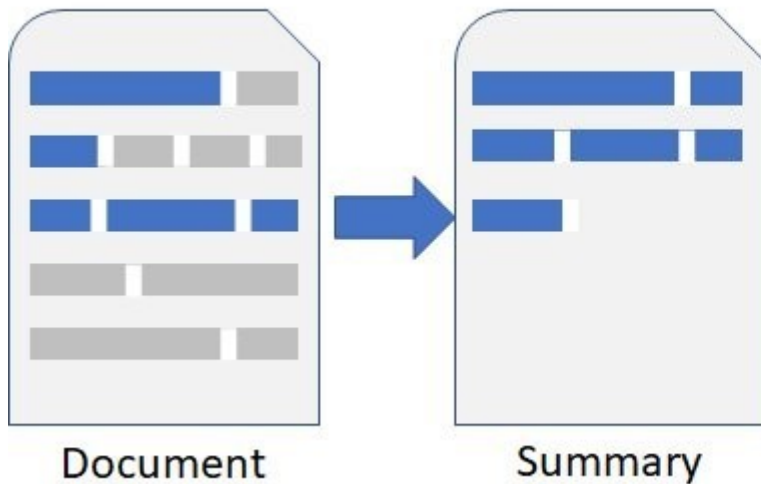
- One useful package for web scraping that you can find in Python's standard library is urllib, which contains tools for working with URLs.
- In particular, the urllib.request module contains a function called urlopen() that can be used to open a URL within a program.

Text Summarization

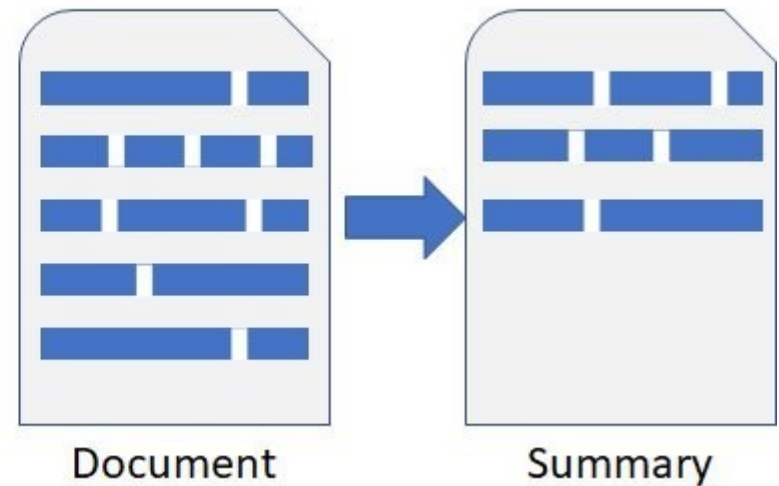
- Text summarization is the process of **generating short, fluent, and most importantly accurate summary of a respectively longer text document.**
- The main idea behind automatic text summarization is to be able to find a short subset of the most essential information from the entire set and present it in a **human-readable format.**
- As online textual data grows, automatic text summarization methods have the potential to be very helpful because more useful information can be **read in a short time.**

Text Summarization

Extractive Summarization



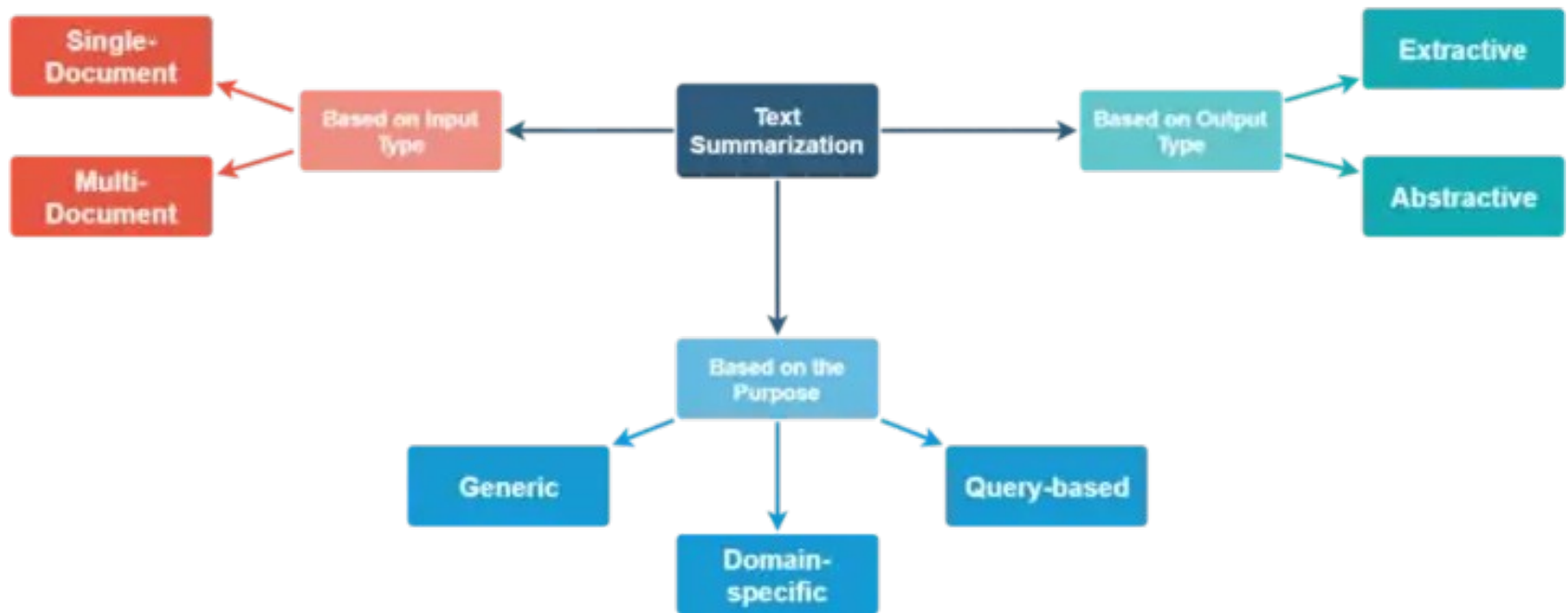
Abstractive Summarization



Why Auto Text Summarization?

- Summaries **reduce** reading time.
- When researching documents, summaries make the **selection process easier**.
- Automatic summarization improves the **effectiveness of indexing**.
- Automatic summarization algorithms are **less biased** than human summarization.
- Personalized summaries are useful in **question-answering** systems as they provide personalized information.
- Using automatic or semi-automatic summarization systems enables commercial abstract services to **increase the number of text documents** they are able to process.

Text Summarization Types



Text Summarization

- Based on input type:
 - **Single Document**, where the input length is short. Many of the early summarization systems dealt with single-document summarization.
 - **Multi-Document**, where the input can be arbitrarily long.

Text Summarization

- Based on the purpose:
 - **Generic**, where the model makes no assumptions about the domain or content of the text to be summarized and treats all inputs as homogeneous. The majority of the work that has been done revolves around generic summarization.
 - **Domain-specific**, where the model uses domain-specific knowledge to form a more accurate summary. For example, summarizing research papers of a specific domain, biomedical documents, etc.
 - **Query-based**, where the summary only contains information that answers natural language questions about the input text.

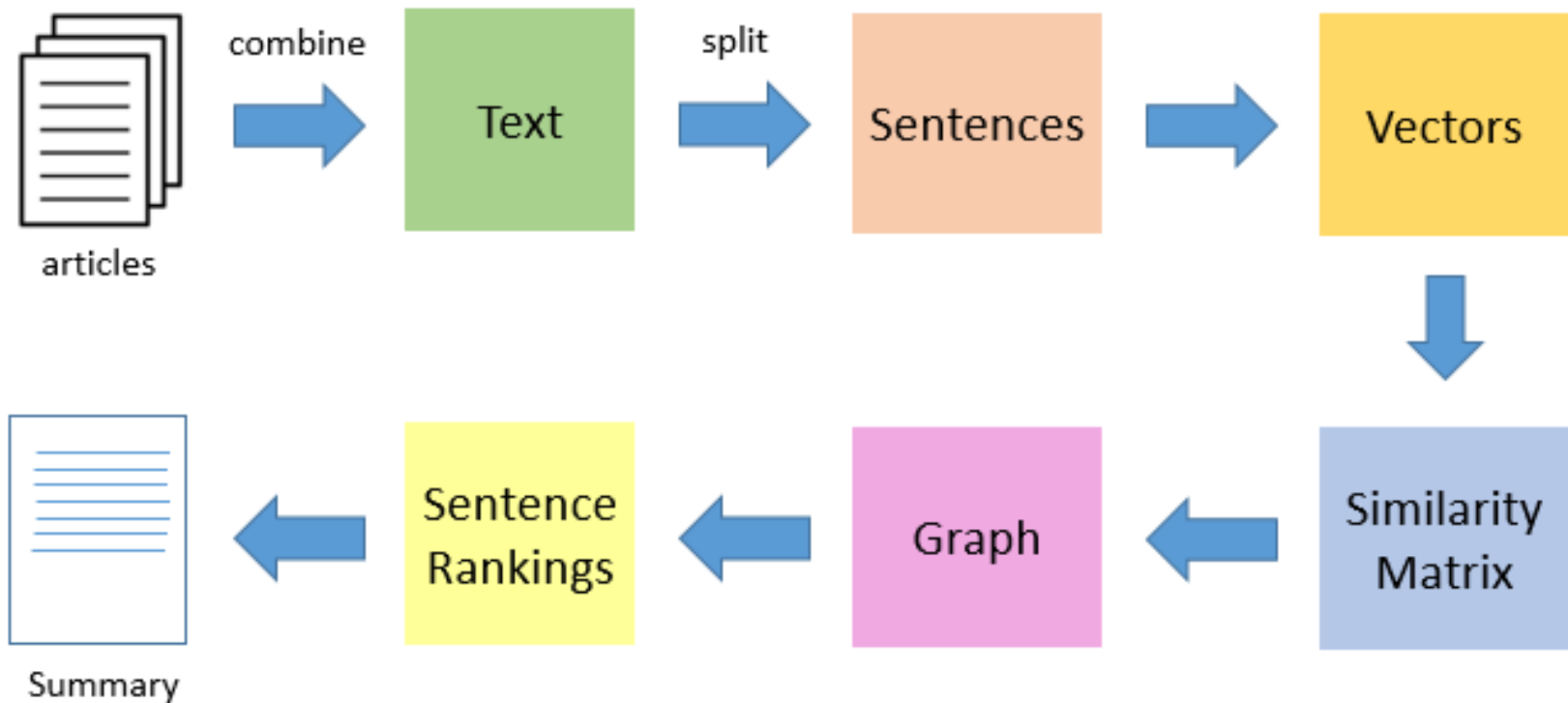
Text Summarization

- Based on output type:
 - **Extractive**, where important sentences are selected from the input text to form a summary. Most summarization approaches today are extractive in nature.
 - **Abstractive**, where the model forms its own phrases and sentences to offer a more coherent summary, like what a human would generate. This approach is definitely more appealing, but much more difficult than extractive summarization.

TextRank Algorithm

- TextRank is an **extractive** summarization technique.
- It is based on the concept that **words which occur more frequently are significant**. Hence, the sentences containing highly frequent words are important .
- Based on this , the algorithm assigns scores to each sentence in the text . The **top-ranked sentences** make it to the summary.

TextRank Algorithm



TextRank Algorithm

- Practical

LexRank Algorithm

- A sentence which is similar to many other sentences of the text has a **high probability** of being important.
- The approach of LexRank is that a particular sentence is recommended by other similar sentences and hence is ranked higher.
- Higher the rank, higher is the priority of being included in the summarized text.

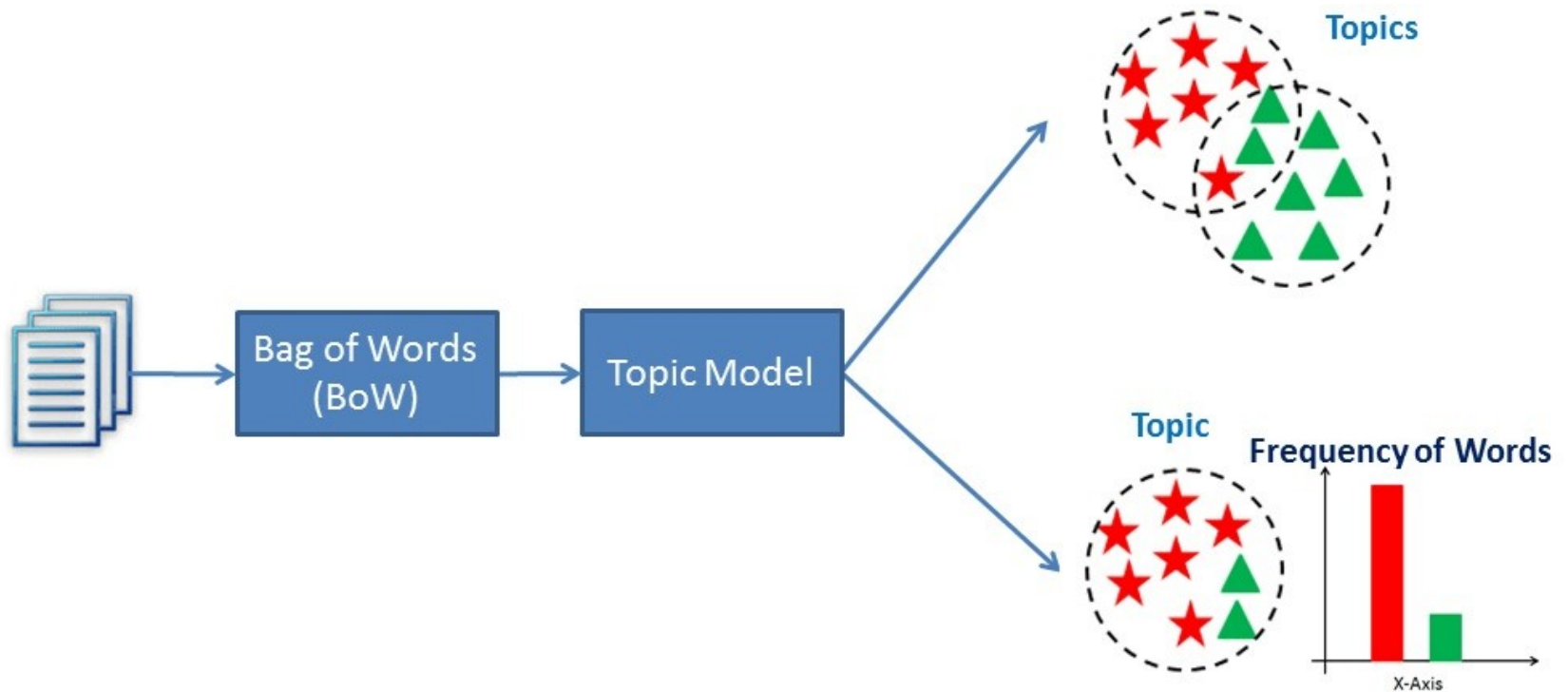
LexRank Algorithm

- Practical

Latent Semantic Analysis LSA

- Latent Semantic Analysis is a unsupervised learning algorithm that can be used for extractive text summarization.
- It extracts semantically significant sentences by applying **singular value decomposition** (SVD) to the matrix of term-document frequency.

Latent Semantic Analysis LSA



GPT Transformers

- Abstractive summarization is the new state of art method, which generates new sentences that could best represent the whole text.
- GPT-2 transformer is another major player in text summarization, introduced by OpenAI.
- First, you have to import the tokenizer and model. Make sure that you import a LM Head type model, as it is necessary to generate sequences.
- Next, load the pretrained gpt-2 model and tokenizer .

GPT Transformers

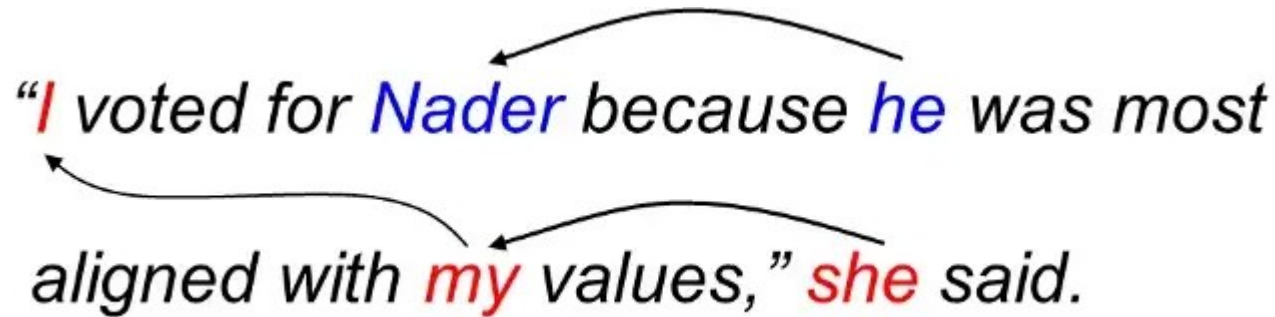
- Practical

Anaphora Resolution

- Anaphora Resolution == the problem of resolving what a pronoun, or a noun phrase refers to.
- In the following example, 1) and 2) are utterances; and together, they form a discourse.
 - 1) John helped Mary.
 - 2) He was kind.
- As human, readers and listeners can quickly and unconsciously work out that the pronoun "he" in utterance 2) refers to "John" in 1). The underlying process of how this is done is yet unclear... especially when we encounter more complex sentences:

Anaphora / Coreference Resolution

*"I voted for Nader because he was most
aligned with my values," she said.*



The diagram illustrates coreference resolution in the sentence "I voted for Nader because he was most aligned with my values," she said. Three curved arrows indicate the relationships: one from "I" to "she", one from "he" to "Nader", and one from "my" to "she".

Anaphora Resolution

- An example involving Noun phrases (Webber 93)
 - 1a) John traveled around France twice.
 - 1b) They were both wonderful. ??

 - 2a) John took two trips around France.
 - 2b) They were both wonderful.
- Consequently, anaphora resolution presents a challenge, and is an active area of research.

Discourse Integration

- In this phase, the impact of the sentences before a particular sentence and the effect of the current sentence on the upcoming sentences is determined.
- For example, the word “**that**” in the sentence “He wanted that” depends upon the prior discourse context.

Pragmatic Analysis

- The last phase of natural language processing is Pragmatic analysis. Sometimes the discourse integration phase and pragmatic analysis phase are combined.
- The actual effect of the text is discovered by applying the set of rules that characterize cooperative dialogues.
- E.g., “close the window?” should be interpreted as a request instead of an order.

Pragmatics and Discourse

- Pragmatics as the study of how the meaning of spoken and written discourse is related to the context in which that speech and writing occurs.
- Pragmatics is specifically concerned with **how speakers' shared interests** and purposes shapes discourse.
- The role of Pragmatics and Discourse is central to the research of various faculty in the department, from a variety of perspectives, including **syntax, semantics, typology and sociolinguistics.**

Ontology

- In Natural Language Processing (NLP), ontology refers to a formal representation of a domain's knowledge, including the concepts, relationships, and axioms that define that domain.
- It acts as a structured vocabulary that allows computers to understand the meaning of words and sentences more accurately.

Ontology: Representing Knowledge

- Ontologies capture the essential concepts and relationships within a specific domain.
- They use formal languages, such as OWL or RDF, to represent these concepts in a machine-readable format.
- This allows computers to reason about the relationships between different concepts and infer new knowledge.

Ontology: Disambiguating Ambiguity

- Many words in natural language have multiple meanings depending on the context.
- Ontologies help disambiguate such ambiguity by providing a clear definition of each concept and its relationship to other concepts.
- This allows NLP systems to interpret the meaning of words and sentences more accurately.

Ontology: Supporting Tasks

- Ontologies play a crucial role in various NLP tasks, such as:
 - Information extraction: Identifying relevant entities and relationships within text.
 - Question answering: Understanding the intent of the question and providing accurate answers.
 - Text summarization: Extracting the main points from a text while maintaining coherence.
 - Machine translation: Ensuring the translated text accurately reflects the meaning of the original text.
 - Sentiment analysis: Identifying the sentiment or opinion expressed in a text.

Ontology: Types

- Domain-specific ontologies:
 - Focus on a specific domain, such as healthcare, finance, or law.
- General-purpose ontologies:
 - Represent common concepts and relationships that are applicable across different domains.
- Upper ontologies:
 - Provide a top-level framework for organizing domain-specific ontologies.

Ontology: Benefits

- Improved accuracy:
 - Ontologies help disambiguate ambiguity and provide a richer representation of meaning, leading to more accurate NLP results.
- Interoperability:
 - Ontologies allow different NLP systems to share and understand information, facilitating collaboration and knowledge sharing.
- Scalability:
 - Ontologies can be easily extended and adapted to new domains or tasks.

Web Ontology Language

- The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies.
- Ontologies are formal ways to describe taxonomies and classification networks, essentially defining the structure of knowledge for various domains.
- They represent the nouns representing classes of objects and the verbs representing relations between the objects.

WOL: Example

- Describing products in an e-commerce website:
 - We can create OWL classes for different product types, such as "Book", "Movie", and "MusicAlbum".
 - We can then define relationships between these classes, such as "Book isPublishedBy Publisher".
 - This allows the website to understand the relationships between products and make recommendations to users.

WOL: Example

- Representing medical knowledge in a healthcare system:
 - We can create OWL classes for different diseases, such as "Diabetes" and "Cancer".
 - We can then define relationships between these classes, such as "Diabetes isTreatedBy Insulin".
 - This allows healthcare professionals to use the system to diagnose and treat patients.

WOL: Example

- Organizing scientific data in a research project:
 - We can create OWL classes for different types of scientific objects, such as "Gene" and "Protein".
 - We can then define relationships between these classes, such as "Gene encodes Protein".
 - This allows scientists to share and collaborate on research data more effectively.

Thank you

This presentation is created using LibreOffice Impress 7.4.1.2, can be used freely as per GNU General Public License



@mitu_skillologies



@mITuSkillologies



@mitu_group



@mitu-skillologies



@MITUSkillologies

kaggle

@mituskillologies

Web Resources

<https://mitu.co.in>

<http://tusharkute.com>



@mituskillologies

contact@mitu.co.in
tushar@tusharkute.com