

Tokenization is like breaking down a sentence into smaller pieces, like words or even individual letters, to make it easier to understand by a computer.

Morphology helps us understand how words change their form to show different meanings, like adding -s to make a word plural or -ed to show past tense. It's basically all about the shapes and patterns of words!

Lemmatization is the process of reducing words to their base or dictionary form, known as the lemma, by removing inflections and variations. The lemma represents the canonical form of a word, which is typically a valid word found in a dictionary.

Stemming is the process of reducing words to their base or root forms by removing suffixes or prefixes, often through heuristic rules or algorithms. Stemming is a cruder and rule-based approach compared to lemmatization, as it does not consider the context of words or part-of-speech information.

section splitting might refer to breaking a document or a text into smaller sections or segments.

Part of speech refers to the category that a word belongs to based on its function and usage in a sentence. In English, there are eight main parts of speech

Dependency parsing It involves identifying the relationships between words in a sentence and representing these relationships as a hierarchical structure, typically in the form of a directed graph called a dependency tree.

Semantic analysis focuses on understanding the meaning of text beyond its surface structure.

Ambiguity refers to a situation where something, such as a word, phrase, statement, or situation, has more than one possible interpretation or meaning.

Lexical Ambiguity: Words with multiple meanings or senses, such as "bank" (which can refer to a financial institution or the edge of a river).

Syntactic Ambiguity: Ambiguity resulting from the structure or arrangement of words in a sentence, where different interpretations arise from different possible parses or grammatical structures.

Semantic Ambiguity: Ambiguity stemming from the meaning of words or phrases, where context or additional information is needed to determine the intended interpretation.

Chunking is a natural language processing technique that involves grouping adjacent words or tokens in a sentence into meaningful units, or "chunks," based on their syntactic structure.

Coreference refers to a linguistic phenomenon where two or more words or expressions in a text refer to the same entity or concept. In other words, coreference occurs when one word or phrase (called the "anaphor") refers back to another word or phrase (called the "antecedent") that represents the same entity.

Named Entity Recognition (NER) is a natural language processing (NLP) technique that focuses on identifying and categorizing named entities within a text into predefined

categories such as names of persons, organizations, locations, dates, numerical expressions, and more.

EXAMINATION

"He saw the ****likes**** and dislikes of everyone at the party."

- In this sentence, "likes" is used as a noun, referring to preferences or tastes, whereas in the original sentence "He likes swimming," "likes" is used as a verb.

"After a long day of work, he felt a sense of ****swimming**** in his head."

"swimming" is used as a noun, indicating a sensation or feeling of dizziness or disorientation, whereas in the original sentence "He likes swimming," "swimming" is used as a gerund (a verb form used as a noun).

The NLP technique used to determine the correct part-of-speech of a word in context is part-of-speech tagging.

"she went to bank there was a long queues," the meaning of the word "bank" depends on the context. **Named Entity Recognition (NER):** NER can be used to identify named entities in text, including organizations, locations, and other entities. **Dependency Parsing:** Dependency parsing can be used to analyse the syntactic structure of the sentence and identify relationships between words. **Word Embeddings:** Word embeddings are dense, low-dimensional vector representations of words that capture semantic similarities between words based on their context in a corpus of text. **Coreference resolution** is another relevant NLP technique that could aid in determining the meaning of "bank" in the given sentence

Converting text from uppercase to lowercase or vice versa can be a good idea

Preservation of Intended Meaning: converting text to lowercase may not always be appropriate, especially in cases where the original case carries semantic or syntactic

To find the context-free (CF) rules and hence the context-free grammar (CFG) for the given sentences, let's analyze the structure of each sentence:

1. ****S1: The cat likes tuna fish.****

- S -> NP VP
- NP -> Det Noun
- VP -> Verb NP
- Det -> "The"
- Noun -> "cat" | "tuna" | "fish"
- Verb -> "likes"

2. ****S2: The dog chased the cat.****

- S -> NP VP
- NP -> Det Noun
- VP -> Verb NP
- Det -> "The"
- Noun -> "dog" | "cat"
- Verb -> "chased"

3. **S3: James loves the dog.**

- S -> NP VP
- NP -> ProperNoun
- VP -> Verb NP
- ProperNoun -> "James"
- Noun -> "dog"
- Verb -> "loves"

Based on the analysis of these sentences, we can derive the following context-free grammar:

- S -> NP VP
- NP -> Det Noun | ProperNoun
- VP -> Verb NP
- Det -> "The"
- ProperNoun -> "James"
- Noun -> "cat" | "tuna" | "fish" | "dog"
- Verb -> "likes" | "chased" | "loves"

With this CFG, we can generate other sentences by substituting non-terminals with their corresponding terminals and applying the production rules. Some examples of sentences that can be generated by this CFG include:

- "The cat likes fish."
- "James loves the tuna."
- "The dog chased James."
- "The cat loves James."

N-grams are contiguous sequences of N items (words, characters, or other units) within a text.

The correct bi-gram (2-gram) representation of the given sentence "Udacity is a great source to learn Natural Language Processing Online" would be:

- **Bi-grams:** ["Udacity is", "is a", "a great", "great source", "source to", "to learn", "learn Natural", "Natural Language", "Language Processing", "Processing Online"]

The correct tri-gram (3-gram) representation of the same sentence would be:

- **Tri-grams:** ["Udacity is a", "is a great", "a great source", "great source to", "source to learn", "to learn Natural", "learn Natural Language", "Natural Language Processing", "Language Processing Online"]

Bag-of-words (BoW) limitations:

- Loss of Word order
- Sparse Representation
- No Semantics
- Vocabulary Size
-

After text normalization (stop word removal and lemmatization),

"natural", "language", "processing", "become", "important", "since", "soon", "begin", "talk", "computer", "understand", "much", "simpler", "use", "speech", "recognition", "first", "step", "build", "like"})

Bigrams in D2:

- "computers understand"
- "understand natural"
- "natural language"
- "language become"
- "become much"
- "much simpler"
- "simpler use"

Trigrams in D2:

- "computers understand natural"
- "understand natural language"
- "natural language become"
- "language become much"
- "become much simpler"
- "much simpler use"

BoW Document Vector for D3 (TF-based):

D3 = [1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1]

publisher's Problem

1. Dataset Preparation:

For the supervised classification problem of identifying "good comments," you would need to prepare a labeled dataset consisting of comments labeled as either "good" or "bad." The dataset should include features that represent characteristics of the comments, such as text content, sentiment, length, etc. Each comment should be associated with a label indicating whether it is considered "good" or "bad" based on some predefined criteria.

2. Precision Requirement:

The publisher's requirement of achieving 90% precision means that out of all the comments classified as "good" by the model, at least 90% of them should actually be good comments. Precision is calculated as the ratio of true positive (correctly identified good comments) to the sum of true positive and false positive (incorrectly identified good comments).

3. Importance of High Recall:

High precision alone may not be sufficient to ensure a successful solution. High recall is also crucial, as it indicates the ability of the model to correctly identify all the relevant instances of good comments, without missing any. A high recall ensures that the model captures as many good comments as possible, even if it means some bad comments are misclassified as good. Achieving a balance between precision and recall is essential to ensure that the model effectively identifies good comments while minimizing the risk of incorrectly deleting good ones.

4. Additional Requirements:

In addition to high precision and recall, a good solution would also require:

- **Robustness:** The model should be able to generalize well to unseen data and handle variations in comment content, language, and style.
- **Efficiency:** The solution should be scalable and computationally efficient, especially if deployed in a real-time or high-volume setting.
- **Regular Updates:** The model should be regularly updated with new data to adapt to changing trends and patterns in comment quality.

5. Treatment as Unsupervised Problem:

If treating the problem as an unsupervised problem, you would focus on clustering or anomaly detection techniques to identify patterns or anomalies in the comment data without relying on labeled examples. In this case, you would need to define criteria or thresholds to determine what constitutes a "good" comment based on characteristics such as sentiment, language quality, relevance, etc. The approach would require more

Difference:::

Term Frequency (TF):

- Term Frequency measures the frequency of a term (word) within a document.

Inverse Document Frequency (IDF):

- Inverse Document Frequency measures the rarity or uniqueness of a term across the entire document collection (corpus).
- It is calculated as the logarithm of the total number of documents in the corpus divided by the number of documents containing the term.

Pipeline for Text Classification: steps

Text Preprocessing:

Tokenization: Splitting the raw text into individual words or tokens.

Stopword Removal: Removing common stopwords that do not contribute much to the meaning of the text.

Lemmatization or Stemming: Reducing words to their base or root forms to normalize the text.

Feature Extraction:

Bag-of-Words Representation: Constructing a feature vector representing the frequency of each term (word) in the document.

TF-IDF Transformation: Optionally transforming the bag-of-words representation using TF-IDF weighting to give more importance to informative terms.

Model Training:

Training a classification model, such as Logistic Regression, Naive Bayes, Support Vector Machine (SVM), or Neural Network, using the feature vectors and corresponding labels (class labels) of the documents.

Model Evaluation:

Evaluating the trained model's performance using metrics such as accuracy, precision, recall, F1-score, etc., on a separate validation or test dataset.

a. False Positive and False Negative:

False Positive (FP): The number of negative instances (actual negatives) incorrectly classified as positive by the model.

False Negative (FN): The number of positive instances (actual positives) incorrectly classified as negative by the model.

b. Precision and Recall:

Precision: The ratio of true positive (TP) to the sum of true positive and false positive. It measures the proportion of correctly predicted positive instances among all instances predicted as positive.

Recall: The ratio of true positive (TP) to the sum of true positive and false negative. It measures the proportion of correctly predicted positive instances among all actual positive instances.

Preprocessing data for Natural Language Processing (NLP):

- Tokenization:
- Lowercasing:
- Stop word Removal:
- Normalization:
- Entity Recognition:
- Part-of-Speech (POS) Tagging:
- Dependency Parsing:
- Feature Engineering:

Syntactic Analysis, also known as syntax analysis or parsing, is a crucial component of natural language processing (NLP) that focuses on analyzing the grammatical structure of sentences to understand their syntactic relationships.

- Morphological Analysis:
- Constituent Structure Analysis:
- Parsing:

Popular libraries and frameworks available for Natural Language Processing (NLP),

- NLTK (Natural Language Toolkit):
- spaCy:
- Stanford NLP:

- Gensim:
- Transformers (Hugging Face):
- TensorFlow and PyTorch:

Difference

Syntax: Syntax refers to the structure or arrangement of words and phrases in a sentence and the rules governing their combination to form grammatically correct sentences.

Semantics: Semantics, on the other hand, refers to the meaning conveyed by words, phrases, and sentences in a language.

parsing refers to the process of analyzing the syntactic structure of sentences according to the rules of a formal grammar.

Stop words are common words in a language that occur frequently but typically do not carry much meaning or semantic information.

common techniques used for keyword normalization in natural language processing (NLP) are:

- Lowercasing:
- Stemming:

Text classification, involves assigning predefined categories or labels to text documents based on their content.

Common metrics are used to evaluate the performance of NLP models, such as text classifiers.

- Accuracy:
- Precision:
- Recall (Sensitivity):
- F1-Score:

Word Sense Disambiguation (WSD) is a natural language processing (NLP) task that aims to determine the correct meaning or sense of a word within a given context.

Relation extraction is a natural language processing (NLP) task that involves identifying and extracting relationships between entities mentioned in text. The goal is to identify

semantic relationships or associations between entities and represent them in a structured format

Several NLP techniques are typically used as part of the pipeline for information extraction.

- Tokenization:
- Part-of-Speech (POS) Tagging:
- Named Entity Recognition (NER):
- Relation Extraction:
- Chunking and Parsing:

Semantic similarity refers to the degree of closeness or relatedness in meaning between two linguistic elements, such as words, phrases, or sentences.

Lexical similarity refers to the degree of similarity or overlap in the surface forms of linguistic elements, such as words, phrases, or sentences.

The screenshot shows a web browser window with multiple tabs. The active tab is ChatGPT 3.5, displaying a chat conversation. The chat history on the left includes topics like 'CCS 412 Summary', 'Ranking with Cosine Similarity', and 'Logistic Regression Training'. The main chat area shows a user query about Jaccard similarity and the AI's response with calculations and results.

Set of Words in Doc3:

- Words in Doc3: {sun, sky, bright}

Jaccard Similarity Calculations:

1. $J(\text{Doc1}, \text{Doc3}): J(\{sky, blue\}, \{sun, sky, bright\}) = \frac{|\{sky\}|}{|\{sky, blue, sun, bright\}|} = \frac{1}{4} = 0.25$
2. $J(\text{Doc2}, \text{Doc3}): J(\{sun, bright\}, \{sun, sky, bright\}) = \frac{|\{sun, bright\}|}{|\{sun, sky, bright\}|} = \frac{2}{3} \approx 0.67$
3. $J(\text{Doc4}, \text{Doc3}): J(\{shining, sun, bright\}, \{sun, sky, bright\}) = \frac{|\{sun, bright\}|}{|\{shining, sun, sky, bright\}|} = \frac{2}{4} = 0.5$

Results:

- Jaccard similarity between Doc3 and Doc1: 0.25
- Jaccard similarity between Doc3 and Doc2: 0.67
- Jaccard similarity between Doc3 and Doc4: 0.5

The document with the highest Jaccard similarity coefficient to Doc3 is Doc2, with a similarity coefficient of approximately 0.67. Therefore, Doc2 is the document most similar to Doc3 based on Jaccard similarity.

Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.

Apr 14 20:19

CCS 412 Summary x Tokenization is like brea x (16) WhatsApp x New Tab x +

https://chat.openai.com/c/6d6dc0e4-d4cf-4d3d-b119-99ab50b4fde3

Apps Folder All Bookmarks

Finish update

New chat

Today

CCS 412 Summary

Ranking with Cosine Similarity

CCS 414 Summarized

Previous 7 Days

Logistic Regression Training Proc

Overcoming NLP Challenges

Setup Bluetooth ESP32: Hello Wor

Update Min SDK Version

PDF Generacija i Preuzimanje

Flutter Reports Fetching & Display

Adaptive Coding Style Learning

Upgrade plan
Get GPT-4, DALL-E, and more

Anilda Kimtal

ChatGPT 3.5

Set of Words in Doc3:

Words in Doc3: {sun, sky, bright}

Jaccard Similarity Calculations:

1. $J(\text{Doc1}, \text{Doc3}): J(\{sky, blue\}, \{sun, sky, bright\}) = \frac{|\{sky\}|}{|\{sky, blue, sun, bright\}|} = \frac{1}{4} = 0.25$

2. $J(\text{Doc2}, \text{Doc3}): J(\{sun, bright\}, \{sun, sky, bright\}) = \frac{|\{sun, bright\}|}{|\{sun, sky, bright\}|} = \frac{2}{3} \approx 0.67$

3. $J(\text{Doc4}, \text{Doc3}): J(\{shining, sun, bright\}, \{sun, sky, bright\}) = \frac{|\{sun, bright\}|}{|\{shining, sun, sky, bright\}|} = \frac{2}{4} = 0.5$

Results:

Jaccard similarity between Doc3 and Doc1: 0.25

Jaccard similarity between Doc3 and Doc2: 0.67

Jaccard similarity between Doc3 and Doc4: 0.5

The document with the highest Jaccard similarity coefficient to Doc3 is Doc2, with a similarity coefficient of approximately 0.67. Therefore, Doc2 is the document most similar to Doc3 based on Jaccard similarity.

Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.