A screenshot of a computer

Description automatically generatedA screenshot of a computer screen

Description automatically generated

If you're using a search engine or an information retrieval system, you might manually or programmatically expand your query using a thesaurus or a similar resource. In academic databases or advanced search tools, thesaurus-based query expansion is often automated to improve search results and retrieval efficiency.

The **Permuterm Index** is an indexing technique used in information retrieval systems, particularly for supporting wildcard queries. Wildcard queries are those where the user may not know the exact term they want to search for and may use a wildcard symbol (e.g., \* or ?) to replace unknown characters.

**How Permuterm Index Works:**

**Permutations of Terms:** For each term in the vocabulary, all possible rotations (or permutations) of the term are generated and stored in the index.A special symbol, often $, is appended to the end of each term to handle the rotation properly. **Example:**

Consider the term "apple".

The term with the special symbol would be "apple$".

All possible rotations of "apple$" would be:

"apple$"

"pple$a"

"ple$ap"

"le$app"

"e$appl"

"$apple"

**Indexing:** Each of these rotations is stored in the index along with a pointer to the original term "apple".

The index might look something like this:

"apple$" → apple

"pple$a" → apple

"ple$ap" → apple

"le$app" → apple

"e$appl" → apple

"$apple" → apple

**Querying with Wildcards:** When a user submits a query with a wildcard, the system rotates the query and matches it against the permuterm index.

**Example Query:** \*ple

Rotate the query by appending the wildcard to the end: "ple\*$"

The system searches for all terms in the index that start with "ple$".

In our index, "ple$ap" would match, leading to the original term "apple" being retrieved.

**Handling Different Wildcard Queries:**

**Prefix Wildcard:** \*ple → search for ple\*$

**Suffix Wildcard:** app\* → search for $app\*

**Infix Wildcard:** a\*e → rotate and search for e$a\*

**Advantages of Permuterm Index:**

**Efficient Wildcard Querying:** Allows for fast and efficient searching for terms with leading, trailing, or embedded wildcards.

**Comprehensive Matching:** All permutations ensure that every possible wildcard pattern can be matched.

**Disadvantages:**

**Index Size:** The permuterm index can be quite large, as it stores multiple rotations for each term, leading to increased storage requirements.

**Complexity:** Generating and maintaining the permuterm index can be computationally intensive, especially for large vocabularies.

**Example Use Case:**

Imagine a user is searching for documents containing any word that starts with "bio" (e.g., "biology", "biochemistry"). With a permuterm index:

The user queries with "bio\*"

This is rotated to "\*bio$"

The system searches for all terms in the permuterm index that start with "bio$", retrieving terms like "biology", "biochemistry", etc.

The Permuterm Index is a powerful technique in information retrieval systems, particularly when supporting complex query patterns that involve wildcards.

**Introduction to Sentiment Analysis**

**Sentiment Analysis**, also known as opinion mining, is a natural language processing (NLP) technique used to determine the sentiment expressed in a piece of text. It involves analyzing text data to classify it into categories such as positive, negative, or neutral. Sentiment analysis is widely applied in various fields, including social media monitoring, customer feedback analysis, market research, and more.

**Key Concepts in Sentiment Analysis**

**Sentiment Classification:**

The primary goal of sentiment analysis is to classify the sentiment expressed in a text. This is typically done at different levels:

**Document-Level Sentiment Analysis:** Determines the overall sentiment of an entire document.

**Sentence-Level Sentiment Analysis:** Determines the sentiment of individual sentences within a document.

**Aspect-Level Sentiment Analysis:** Focuses on specific aspects or entities mentioned in the text and determines the sentiment related to them.

**Polarity:**

Sentiment is often expressed in terms of polarity:

**Positive Sentiment:** Indicates a favorable or optimistic opinion.

**Negative Sentiment:** Indicates an unfavorable or critical opinion.

**Neutral Sentiment:** Indicates an objective or impartial opinion.

**Emotion Detection:**

Beyond simple polarity, some sentiment analysis systems also detect specific emotions, such as joy, anger, sadness, or surprise.

**Opinion Holder and Target Identification:**

**Opinion Holder:** The person or entity expressing the sentiment.

**Target:** The object or entity about which the sentiment is expressed.

**Applications of Sentiment Analysis**

**Social Media Monitoring:**

**Example:** Brands use sentiment analysis to monitor mentions of their products or services on social media platforms like Twitter, Facebook, or Instagram. This helps them gauge public opinion, track brand reputation, and respond to customer feedback in real-time.

**Problem Solved:** Provides insights into how customers feel about a brand, allowing for timely interventions and reputation management.

**Customer Feedback Analysis:**

**Example:** Companies analyze customer reviews, survey responses, and support tickets to understand customer satisfaction and identify areas for improvement.

**Problem Solved:** Helps businesses improve their products and services by addressing customer concerns and capitalizing on positive feedback.

**Market Research:**

**Example:** Businesses use sentiment analysis to understand public opinion on market trends, competitor products, or new product launches.

**Problem Solved:** Provides valuable insights into market dynamics, helping companies make informed strategic decisions.

**Political Sentiment Analysis:**

**Example:** Sentiment analysis is used to analyze public opinion during elections by analyzing social media posts, news articles, and comments on political events.

**Problem Solved:** Helps political analysts, parties, and candidates understand voter sentiment and adjust their campaigns accordingly.

**Financial Market Analysis:**

**Example:** Investors use sentiment analysis to predict stock market movements by analyzing news articles, tweets, and other financial data sources.

**Problem Solved:** Aids in making investment decisions by identifying trends in market sentiment that could affect stock prices.

**Product Development:**

**Example:** Sentiment analysis is used during product development to understand customer needs and preferences by analyzing feedback from beta testers, forums, and user reviews.

**Problem Solved:** Ensures that products are developed in line with customer expectations, leading to higher customer satisfaction and successful product launches.

**Challenges in Sentiment Analysis**

**Sarcasm and Irony:**

Sentiment analysis systems often struggle with detecting sarcasm and irony, which can lead to incorrect sentiment classification.

**Context Understanding:**

The meaning of a sentence can depend heavily on the context, making it challenging for sentiment analysis systems to accurately interpret the sentiment.

**Ambiguity:**

Words or phrases can have different meanings in different contexts, leading to ambiguity in sentiment detection.

**Multilingual Sentiment Analysis:**

Analyzing sentiment in different languages or across multilingual datasets adds complexity due to language-specific nuances and cultural differences.

**Conclusion**

Sentiment analysis is a powerful tool for extracting insights from text data. It enables businesses, researchers, and analysts to understand public opinion, customer satisfaction, and market trends, driving better decision-making. Despite the challenges, advances in NLP and machine learning continue to enhance the accuracy and applicability of sentiment analysis across various domains.

A screenshot of a document

Description automatically generatedA white paper with black text

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a math problem

Description automatically generated

A screenshot of a math test

Description automatically generatedA screenshot of a computer

Description automatically generated

1. Precision is 0.6, and recall is 0.75.

A screenshot of a math test

Description automatically generatedA screenshot of a black and white page

Description automatically generated

A screenshot of a paper

Description automatically generatedA screenshot of a paper

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a math problem

Description automatically generated

A screenshot of a paper

Description automatically generatedA screenshot of a paper

Description automatically generated

A screenshot of a paper

Description automatically generatedA screenshot of a computer program

Description automatically generated

**1. Introduction to Topic Modelling**

**Problem:** Explain the concept of topic modeling. Given a collection of documents, how would topic modeling help in understanding the underlying themes?

**Solution:**

**Concept:** Topic modeling is an unsupervised machine learning technique used to identify abstract topics within a collection of documents. It assumes that each document is a mixture of topics and that each topic is a mixture of words.

**Application:** Given a collection of documents, topic modeling helps in:

Identifying key themes (topics) that are prevalent across the documents.

Clustering words with similar meanings together under a common topic.

Organizing large sets of textual data into interpretable structures for better understanding and analysis.

For instance, if we have a collection of news articles, topic modeling might identify topics such as "Politics," "Sports," "Technology," etc., and associate each article with these topics based on the words it contains.

**2. Topic Modelling using LDA**

**Problem:** Describe how Latent Dirichlet Allocation (LDA) works for topic modeling. What are the key components involved in LDA?

**Solution:**

**Components of LDA:**

**Document-Topic Distribution:** Each document is represented as a distribution over topics.

**Topic-Word Distribution:** Each topic is represented as a distribution over words.

**Dirichlet Priors:** LDA uses Dirichlet distributions as priors for the document-topic and topic-word distributions.

**How LDA Works:**

**Initialize Random Topic Assignments:** Start with random assignment of topics to words in the document.

**Iterative Process:**

For each word in each document, reassign the topic based on the probability that the word belongs to a certain topic given the other words in the document and the words assigned to the same topic across all documents.

**Convergence:** The process continues until the topic assignments stabilize, meaning the distributions over topics for each document and words for each topic no longer change significantly.

LDA is generative, meaning it assumes documents are generated through a mixture of topics and works backward to uncover the topics based on the observed words.

**3. Mathematical Foundations for LDA: Multinomial and Dirichlet Distributions (Part 1)**

**Problem:** Explain the role of the multinomial distribution in LDA. How is it used in the context of topic modeling?

**Solution:**

**Multinomial Distribution:** The multinomial distribution is a generalization of the binomial distribution. It represents the probability of obtaining a certain combination of outcomes from multiple categories in a single trial.

**Role in LDA:**

**Topic-Word Distribution:** LDA uses a multinomial distribution to model the probability of words given a topic. Each topic is represented as a multinomial distribution over the vocabulary.

**Document-Topic Distribution:** Similarly, LDA models the probability of topics given a document as a multinomial distribution.

In LDA, the multinomial distribution helps in modeling the process of selecting words based on the topic distribution and topics based on the document distribution.

**4. Mathematical Foundations for LDA: Multinomial and Dirichlet Distributions (Part 2)**

**Problem:** What is a Dirichlet distribution, and how is it used as a prior in LDA? Explain with an example.

**Solution:**

**Dirichlet Distribution:** The Dirichlet distribution is a family of continuous multivariate probability distributions parameterized by a vector of positive reals. It is often used as a prior for multinomial distributions in Bayesian statistics.

**Role in LDA:**

**Document-Topic Distribution:** A Dirichlet prior is placed on the document-topic distribution, which means that the distribution of topics in each document is governed by a Dirichlet distribution.

**Topic-Word Distribution:** Similarly, a Dirichlet prior is placed on the topic-word distribution, governing the distribution of words in each topic.

**Example:** Suppose we have three topics and a document. The Dirichlet prior might indicate that it’s likely for a document to be heavily focused on one topic but still have some contribution from the others. If the Dirichlet parameter is [2, 2, 2], it suggests that each topic is equally likely in the document. If the parameter is [5, 1, 1], it indicates a preference for the first topic.

**5. Gibbs Sampling for LDA (Part 1)**

**Problem:** Describe the concept of Gibbs sampling in the context of LDA. Why is it used?

**Solution:**

**Gibbs Sampling:** Gibbs sampling is a Markov Chain Monte Carlo (MCMC) algorithm used to sample from a high-dimensional probability distribution when direct sampling is difficult.

**In the Context of LDA:**

**Purpose:** LDA involves complex probability distributions over documents, topics, and words. Direct computation of these distributions is intractable due to their high dimensionality.

**How It’s Used:** Gibbs sampling is used to approximate the distributions by iteratively sampling from the conditional distribution of each variable given the others. In LDA, this means iterating over each word in each document, sampling a new topic assignment based on the current assignments of all other words.

**Why It’s Used:** Gibbs sampling helps LDA converge to a stable distribution of topics without having to compute the full joint distribution directly, making the algorithm computationally feasible.

**6. Gibbs Sampling for LDA (Part 2)**

**Problem:** Given an initial set of topic assignments, demonstrate how Gibbs sampling updates the topic assignment for a specific word in a document.

**Solution:**

**Initial Setup:** Suppose we have a document with words [w1, w2, w3], and initial topic assignments [T1, T2, T1].

**Gibbs Sampling Update:**

1. **Remove the Word:** Temporarily remove w2 and its current topic assignment T2.
2. **Calculate the Conditional Probability:**

Compute the probability of assigning each possible topic to w2, given the current topic assignments of the other words.

This probability is proportional to the product of:

The probability of w2 given the topic (how likely the word is to occur under each topic).

The probability of the topic given the document (how likely the topic is given the remaining topic assignments in the document).

**Sample a New Topic:** Based on these probabilities, sample a new topic for w2.

**Update the Assignment:** Assign the new topic to w2 and proceed to the next word.

**Iteration:** Repeat this process for all words in all documents until convergence.

**Steps Explanation:**

1. **Preprocessing:** Tokenize and clean the documents by removing stopwords.
2. **Create Dictionary:** Map each word to a unique ID.
3. **Create Corpus:** Convert the documents into a bag-of-words format.
4. **Train LDA Model:** Use the gensim library to train an LDA model on the corpus.
5. **Display Topics:** Print the top words associated with each topic.

**1. Topic Modelling using LDA**

**Problem:** You have a corpus of 10,000 documents, and you wish to identify 5 topics using LDA. Explain how LDA assigns topics to the documents and what output you can expect after running the algorithm.

**Solution:**

**How LDA Assigns Topics:**

* **Initialization:** LDA starts by randomly assigning each word in each document to one of the 5 topics.
* **Iterative Process:** For each word in each document, LDA reassigns a topic based on two factors:
  + The proportion of words in the document that are currently assigned to each topic.
  + The proportion of times the word is assigned to each topic across the entire corpus.
* **Convergence:** After several iterations, the assignments stabilize, meaning the algorithm converges.
* **Expected Output:**
* **Document-Topic Distribution:** For each document, LDA provides a probability distribution over the 5 topics, indicating the strength of each topic within that document.
* **Topic-Word Distribution:** For each topic, LDA provides a probability distribution over words, indicating which words are most strongly associated with each topic.
* **Top Words per Topic:** LDA outputs the most significant words for each topic, allowing interpretation of the topics.
* **Example Output:** For a document discussing technology trends:
* Topic 1: { "AI": 0.3, "Machine Learning": 0.25, "Deep Learning": 0.2, "Data": 0.15, "Neural Networks": 0.1 }

**2. Introduction to Topic Modelling**

**Problem:** Given a collection of customer reviews, describe how topic modeling could help a business understand customer sentiments and identify areas for improvement.

**Solution:**

**Application of Topic Modelling:**

**Identify Common Themes:** Topic modeling can identify the most common themes across the reviews, such as "customer service," "product quality," or "pricing."

**Sentiment Analysis:** By associating topics with sentiment words, businesses can determine the overall sentiment (positive, negative, neutral) related to each theme.

**Actionable Insights:** For example, if a significant topic is "delivery time" and it’s associated with negative sentiments, the business might focus on improving its delivery process.

**Process:**

1. **Preprocessing:** Tokenize and clean the reviews.
2. **LDA Application:** Apply LDA to extract topics.
3. **Interpretation:** Analyze the top words for each topic and associate them with specific themes.
4. **Sentiment Analysis:** Combine with sentiment analysis to understand the sentiment associated with each topic.

**Conclusion:** Topic modeling can reveal hidden patterns in customer reviews, helping businesses make data-driven decisions.

**3. Mathematical Foundations for LDA: Multinomial and Dirichlet Distributions (Part 1)**

**Problem:** Consider a simplified scenario where you have 3 documents and a vocabulary of 5 words. Assume there are 2 topics. Write the multinomial distribution for a single document, assuming equal probabilities for each word within a topic.

**Solution:**

**Vocabulary:** Vocab={"word1","word2","word3","word4","word5"}\text{Vocab} = \{ \text{"word1"}, \text{"word2"}, \text{"word3"}, \text{"word4"}, \text{"word5"} \}Vocab={"word1","word2","word3","word4","word5"}

**Topics:** Topic 1=Uniform distribution over vocab\text{Topic 1} = \text{Uniform distribution over vocab}Topic 1=Uniform distribution over vocab Topic 2=Uniform distribution over vocab\text{Topic 2} = \text{Uniform distribution over vocab}Topic 2=Uniform distribution over vocab

**Multinomial Distribution for a Single Document:**

Assume a document is composed of 10 words.

Topic 1 and Topic 2 each have an equal probability (0.5) of generating a word.

A white background with black text

Description automatically generated

**4. Mathematical Foundations for LDA: Multinomial and Dirichlet Distributions (Part 2)**

**Problem:** Suppose the Dirichlet parameter for the topic distribution in a document is α = [2, 2]. Calculate the expected proportion of topics in the document.

**Solution:**

**Dirichlet Distribution:** The Dirichlet distribution is parameterized by α, which in this case is [2, 2].

**Expected Proportions:** For a Dirichlet distribution with parameters α = [α1, α2], the expected value for each topic is given by:

A math equations on a white background

Description automatically generated

**5. Gibbs Sampling for LDA (Part 1)**

**Problem:** Explain how Gibbs sampling helps in inferring the topic distribution for each document in LDA. Use an example with a small corpus to illustrate the steps involved.

**Solution:**

**Concept:** Gibbs sampling is a way to approximate the posterior distribution of the model parameters in LDA, especially when direct computation is infeasible.

**Steps in Gibbs Sampling for LDA:**

1. **Initialize Randomly:** Start with a random assignment of topics to words in the corpus.
2. **Iterate Over Words:**

For each word wiw\_iwi​ in document ddd, remove its current topic assignment.

Calculate the probability of assigning each possible topic kkk to the word based on:

The proportion of words in document ddd currently assigned to topic kkk.

The proportion of times word wiw\_iwi​ is assigned to topic kkk across all documents.

Sample a new topic for word wiw\_iwi​ based on this probability.

**Update Assignments:** Reassign the new topic to word wiw\_iwi​ and proceed to the next word.

**Repeat:** Continue the process until convergence, i.e., when topic assignments stabilize.

**Example:** Given a small corpus of 3 documents:

Doc1: "apple orange banana"

Doc2: "banana apple"

Doc3: "orange banana"

After several iterations of Gibbs sampling:

Doc1 might have a high probability of Topic 1 (related to fruits).

Doc2 might be a mix of Topic 1 and another topic.

Doc3 might also have a high probability of Topic 1.

**Conclusion:** Gibbs sampling enables LDA to infer the topic distribution for each document by iteratively refining the topic assignments for words.

**Gibbs Sampling for LDA (Part 2)**

**Problem:** Suppose after running Gibbs sampling for several iterations, the following topic-word assignments for a word in a document are observed:

Topic 1: 3 times

Topic 2: 7 times

If the topic distribution is Dirichlet(α = [1, 1]), what is the probability of assigning the word to Topic 1 or Topic 2 in the next iteration?

**Solution:**

**Gibbs Sampling Probability:** The probability of assigning a word to a topic kkk during Gibbs sampling is proportional to:

A math equations on a white background

Description automatically generated

Given:

Topic 1: 3 times

Topic 2: 7 times

α=[1,1]

A math problem with numbers and equations

Description automatically generated

1. Sentiment Scoring with Lexicons

Problem: You have the following sentences and a sentiment lexicon:

Sentence: "The movie was surprisingly good but a bit too long."

Lexicon:

"good" : +2

"surprisingly" : +1

"too" : -1

"long" : -2

Calculate the overall sentiment score of the sentence using the lexicon and explain what this score represents.

Solution:

Identify Words in the Sentence:

"good" → +2

"surprisingly" → +1

"too" → -1

"long" → -2

Calculate the Total Sentiment Score:

Total score = (+2) + (+1) + (-1) + (-2) = 0

Interpretation:

A sentiment score of 0 indicates that the sentence is neutral overall. The positive and negative sentiments cancel each other out.

Conclusion: Despite having both positive and negative elements, the sentence has a neutral sentiment when using this lexicon.

2. Probability Calculation in a Naive Bayes Sentiment Classifier

Problem: Consider a Naive Bayes classifier trained on the following sentiment-labeled dataset:

Positive Reviews: 200

Negative Reviews: 100

You encounter a new sentence with the following word probabilities:

P(word="great" | Positive) = 0.02

P(word="great" | Negative) = 0.01

Calculate the posterior probability that the sentence "The product is great" is positive using Naive Bayes, assuming the word "great" is the only feature used in the model. Assume uniform priors.

Solution: Prior Probabilities:

P(Positive)=200300=23,P(Negative)=100300=13P(Positive) = \frac{200}{300} = \frac{2}{3}, \quad P(Negative) = \frac{100}{300} = \frac{1}{3}P(Positive)=300200​=32​,P(Negative)=300100​=31​

Likelihood of "great":

P(word="great"∣Positive)=0.02,P(word="great"∣Negative)=0.01P(word="great" | Positive) = 0.02, \quad P(word="great" | Negative) = 0.01P(word="great"∣Positive)=0.02,P(word="great"∣Negative)=0.01

Posterior Probability: Using Bayes’ Theorem:

P(Positive∣"great")=P("great"∣Positive)×P(Positive)P("great")P(Positive | "great") = \frac{P("great" | Positive) \times P(Positive)}{P("great")}P(Positive∣"great")=P("great")P("great"∣Positive)×P(Positive)​

Where P("great")P("great")P("great") is:

P("great")=P("great"∣Positive)×P(Positive)+P("great"∣Negative)×P(Negative)P("great") = P("great" | Positive) \times P(Positive) + P("great" | Negative) \times P(Negative)P("great")=P("great"∣Positive)×P(Positive)+P("great"∣Negative)×P(Negative) P("great")=(0.02×23)+(0.01×13)=0.0133+0.0033=0.0166P("great") = (0.02 \times \frac{2}{3}) + (0.01 \times \frac{1}{3}) = 0.0133 + 0.0033 = 0.0166P("great")=(0.02×32​)+(0.01×31​)=0.0133+0.0033=0.0166

Now, calculate the posterior probability:

P(Positive∣"great")=0.02×230.0166=0.01330.0166≈0.80P(Positive | "great") = \frac{0.02 \times \frac{2}{3}}{0.0166} = \frac{0.0133}{0.0166} \approx 0.80P(Positive∣"great")=0.01660.02×32​​=0.01660.0133​≈0.80

Conclusion: The posterior probability that the sentence "The product is great" is positive is 0.80, indicating that it is highly likely to be positive.

3. Term Frequency-Inverse Document Frequency (TF-IDF) Calculation

Problem: Given the following two documents:

Document 1: "I love programming in Python." Document 2: "Python programming is fun."

Calculate the TF-IDF score for the word "Python" in both documents. Assume a corpus of these two documents.

Solution: Term Frequency (TF): TF("Python", Document 1) = 1/5 (Python appears once in a document of 5 words) = 0.2

TF("Python", Document 2) = 1/4 (Python appears once in a document of 4 words) = 0.25

Document Frequency (DF): DF("Python") = 2 (Python appears in both documents)

A close-up of a math problem

Description automatically generated

TF-IDF Calculation: TF-IDF("Python", Document 1) = 0.2 × 0 = 0 TF-IDF("Python", Document 2) = 0.25 × 0 = 0