# Datametica - Topic Modeling

Friday, September 8, 2023 3:51 PM

| Step | Description |
|---|---|
| Problem Definition | Topic modeling on customer feedback can be immensely valuable, allowing you to discover underlying patterns and themes in your customer complaints, which can help drive process improvements. I'll guide you through the steps on how to approach this: |
| 1. Data Collection | Gathering data from various sources - Customer support channels<br>• (e.g., ticketing system) -<br>• Web scraping tools for social media:<br>• Beautiful Soup,<br>• Scrapy -<br>• Tweepy for Twitter -<br>• Speech-to-text tools:<br>• Google's Speech-to-Text API,<br>• IBM's Watson |
| 2. Text Preprocessing | Cleaning and preparing the text data for analysis -<br>• Text normalization<br>  • Stop words removal<br>  • Tokenization<br>  • Stemming<br>  • Lemmatization<br>  • Case - Folding<br>• Text standardization<br>  • Jargon removal - defaulter, bail-out<br>  • Removing grammatical mistakes<br>• Part-of-Speech tagging<br>• Named Entity Recognition<br>• Dependency parsing |
| 3. Text Analysis similar to EDA | Analyzing Word Distributions and Relationships:<br>• Word Distributions:<br>  • Refers to the frequency or probability distribution of words appearing in a text corpus.<br>  • Often visualized using histograms or bar plots to identify the most common words or phrases in a given dataset.<br>• N-gram Analysis:<br>  • N-grams are continuous sequences of n items (words, characters, etc.) from a text.<br>  • Helps in understanding context and capturing phrases. E.g., "New York" would be a 2-gram (bigram).<br>• Sparsity Visualization:<br>  • Often related to the term-document matrix, where many entries are zero because a large vocabulary results in many words not appearing in individual documents.<br>  • Visualization can be done using a matrix plot, highlighting which terms appear in |

| | |
|---|---|
| | which documents. |
| Count Vectorization | • Count Vectorizer:<br>• Converts a collection of text documents to a matrix of word/token counts.<br>• The result is a sparse matrix where rows are documents and columns represent word counts.<br>• TF-IDF (Term Frequency-Inverse Document Frequency):<br>• Similar to the Count Vectorizer, but instead of raw counts, the matrix represents the weighted count where weights are determined by the importance of a word to a document relative to its frequency across all documents. |
| Word Embeddings | • A form of representing words as dense vectors such that words with similar meanings are close to each other in the vector space.<br>• Result in dense vectors, typically with hundreds of dimensions.<br>• Word2Vec<br>○ Developed by Google, it captures semantic relationships between words. Has two architectures: Skip-gram and Continuous Bag of Words (CBOW).<br>• GloVe (Global Vectors for Word Representation)<br>○ Developed by Stanford, it focuses on word co-occurrence statistics.<br>• FastText<br>○ Developed by Facebook, it represents words as bags of character n-grams, allowing it to handle out-of-vocabulary words. |
| Contextualized Word Embeddings | More advanced than traditional word embeddings.<br>They consider the context around a word, allowing the same word to have different vectors based on its surrounding words.<br>• a. ELMo (Embeddings from Language Models):<br>○ Developed by AllenNLP, it uses character-based word representations and bidirectional LSTMs.<br>• b. BERT (Bidirectional Encoder Representations from Transformers)<br>○ Developed by Google, it's a transformer-based model pre-trained on a large corpus, known for its bidirectionality.<br>• c. GPT (Generative Pre-trained Transformer)<br>○ Developed by OpenAI, it's a transformer model trained to predict the next word in a sequence. |
| 4. Topic Modeling | Identifying underlying topics in the data<br>• LSA: Latent Semantic Analysis (LSA)<br>• SVD: Decompose term-document matrix using Singular Value Decomposition<br>• LDA: Latent Dirichlet Allocation (LDA): Assign topics to documents and words<br>• NNMF: Non-Negative Matrix Factorization (NMF) - Sklearn for NMF, TruncatedSVD Gensim for LDA |
| 5. Visualization & Interpretation | Visual representation of topics -<br>• pyLDAvis for LDA visualization -<br>• Word Clouds for each topic |
| 6. Evaluation | Assessing the quality of the topics -<br>• Examine the distinctness of topics -<br>• Topic coherence scores (using Gensim) |

| | |
|---|---|
| 7. Application and Business Impact | • Utilizing the topics in practical scenarios -<br>• Classify new complaints -<br>• Route complaints to relevant departments -<br>• Drive business improvements based on topics |
| | |
| | |

## How does LDA work Latent Dirichlet Allocation

- **In traditional LDA, the model operates on a document-term matrix, which can be either BoW or TF-IDF representations. However, LDA doesn't directly work with dense embeddings like word2vec.**

- **If you wanted to leverage word embeddings, you'd likely look into other topic modeling or document clustering methods that can work in the dense vector space.**

| # | Concept | Description |
|---|---|---|
| 1 | Document | A piece of text. In the context of LDA, a document can be an article, a paragraph, or any text segment. |
| 2 | Corpus | A collection of documents. |
| 3 | Topic | A collection of words that frequently appear together and represent a particular theme or subject. |
| 4 | Dirichlet Distribution | A family of multivariate probability distributions parameterized by a vector of positive reals. LDA uses the Dirichlet distribution for the creation of topics and documents. |
| 5 | Bag of Words (BoW) | A representation of text where each document is represented as a vector of word frequencies, disregarding the order of words. |
| 6 | Latent Variables | Variables that are not directly observed but inferred from mathematical models. In LDA, topics are latent variables. |
| 7 | Gibbs Sampling | A Markov chain Monte Carlo (MCMC) technique for obtaining a sequence of samples from a multivariate distribution, especially when direct sampling is challenging. |
| 8 | Hyperparameters (Alpha and Beta) | Parameters of the Dirichlet distribution. Alpha affects document-topic density and Beta affects topic-word density. |
| 9 | Posterior Distribution | The probability distribution of an unknown quantity, treated as a random variable, conditional on the evidence obtained from an experiment or survey. |
| 10 | Topic Coherence | A measure used to judge the quality of the topics produced by a topic model. It scores a single topic by measuring the degree of semantic similarity between high scoring words. |

## Metrics for LDA model

| # | Evaluation Metric | Description |
|---|---|---|
| 1 | Perplexity | A measure of how well a probability distribution predicts a sample. Lower perplexity value indicates better generalization to unseen documents. However, lower perplexity might not always correlate with human interpretability. |
| 2 | Topic Coherence | Measures the degree of semantic similarity between high scoring words within a topic. Models with higher coherence scores generally produce more interpretable topics. |
| 3 | Human Judgement | Involves having humans, especially domain experts, interpret and validate topics. It could also involve crowdsourcing feedback. |
| 4 | Visual Inspection with pyLDAvis | Visualization tools that help visually interpret and validate the quality and separability of topics. |
| 5 | Model Stability | Tests the robustness of the topics by checking if the same topics are found across multiple samples (subsamples or bootstraps) of data. |
| 6 | Topic Diversity | Measures the lexical diversity of topics. A model that produces topics with overlapping terms isn't producing distinct enough topics. |
| 7 | Document Classification | If labeled data is available, the topic distributions can be used as features for document classification. Good classification performance indicates the topics capture meaningful patterns. |
| 8 | Document Retrieval or Recommendation | Topics can be used as features to build retrieval or recommendation systems. Improved recommendation or retrieval quality indicates the topics have meaningful structure. |

=========================================================================

## How does LSA work: Latent Semantic Analysis

- For Latent Semantic Analysis (LSA), we typically use Term-Document matrices. These matrices can be generated using either the Count Vectorization method or the TF-IDF Vectorization method.

- Word embeddings like Word2Vec, GloVe, etc., are not directly used in LSA. Instead, LSA works on the principle of singular value decomposition (SVD) of the Term-Document matrix.

| # | Concept/Component | Explanation/Details |
|---|---|---|
| 1 | Term-Document Matrix (TDM) | A matrix representation of a corpus where rows represent terms and columns represent documents. Entries are typically raw term frequencies or TF-IDF scores. |
| 2 | Singular Value Decomposition (SVD) | A matrix factorization method. When applied to the TDM, it breaks it down into three matrices: �U (term-topic matrix), ΣΣ (diagonal matrix of singular values), and ��VT (document-topic matrix). |
| 3 | Dimensionality Reduction | In LSA, the goal is to capture the underlying structure by reducing the number of topics (latent factors). This is done by keeping only the top �k largest singular values in ΣΣ and discarding the rest. |
| 4 | Latent Topics | The reduced matrices �U and ��VT capture 'latent' topics in the data. Each row in �U and each column in ��VT can be viewed as a topic represented as a distribution over terms or documents, respectively. |
| 5 | Semantic Space | The space in which documents and terms are mapped after reduction. In this space, the cosine similarity between vectors can be used to measure the semantic similarity between terms or documents. |
| 6 | Noise Reduction | By keeping only the top �k singular values, LSA effectively filters out noise, helping in capturing the more significant patterns or structures in the data. |
| 7 | Term & Document Similarity | In the reduced semantic space, cosine similarity between term vectors or document vectors can be used to find similar terms or documents. |
| 8 | Challenges | 1. The method is linear, so it might not capture complex relationships. 2. LSA does not have a probabilistic foundation, unlike LDA. 3. SVD can be computationally expensive for very large matrices. |
| 9 | Applications | Document retrieval, information retrieval, query disambiguation, and other text mining tasks. |
| 10 | Comparisons | LSA focuses on reducing dimensionality to detect latent patterns. LDA, in contrast, uses a probabilistic approach to model topic distributions. |

===============================================================

## Advanced methods for TOPIC MODELING

LDA and LSA are generally based on Count Vectorizer kind of DISCRETE word embeddings
For continuous word embeddings like Word2Vec, which are DENSE vector representations of words there are other methods like:

Below, Doc2Vec is directly related to Word2Vec

| # | Method | Description |
|---|---|---|
| 1 | Document Embedding | Extends word2vec to represent entire documents. Documents can then be clustered using algorithms like KMeans. |

| | | |
|---|---|---|
| | (Doc2Vec) | |
| 2 | Clustering on Embeddings | Convert each document into a vector by averaging its word vectors. Then apply clustering algorithms such as KMeans or DBSCAN. Clusters represent topics. |
| 3 | BERTopic | Uses sentence transformers to create embeddings, followed by UMAP for dimensionality reduction and HDBSCAN for clustering. |
| 4 | Top2Vec | Combines Doc2Vec, word2vec, or universal sentence encoder embeddings with UMAP for dimensionality reduction. Automatically identifies topics. |
| 5 | LDA2Vec | Hybrid of word embeddings and LDA, designed to learn topic vectors in the same embedding space as word vectors. |
| 6 | Neural Variational Document Model (NVDM) | Neural approach combining variational autoencoders (VAE) and neural networks to model documents as topic mixtures. |
| 7 | NMF on Embeddings | Uses Non-negative Matrix Factorization on dense document embeddings derived from word embeddings. |

==========================================================================================

How does LDA work:  Linear Discriminant Analysis

- This is NOT used for Topic Modeling
- This is intended for Dimensionality Reduction and Classification
- This is a Supervised model

| # | Concept/Component | Explanation/Details |
|---|---|---|
| 1 | Objective | To find the linear combination of features that best separate two or more classes in a dataset. |
| 2 | Assumption | Assumes that the independent variables are normally distributed and that the classes have identical covariance matrices. |
| 3 | Between-Class Variance | Variance between the mean of different classes. LDA aims to maximize this to ensure that classes are well separated. |
| 4 | Within-Class Variance | Variance within individual classes. LDA aims to minimize this to ensure that members of the same class are close to each other. |
| 5 | Fisher's Linear Discriminant | A criterion that seeks to maximize the ratio of between-class variance to the within-class variance, thereby ensuring maximum separability. |
| 6 | Eigenvalues & Eigenvectors | Upon computing the between-class scatter matrix and the within-class scatter matrix, the goal is to determine the eigenvectors and eigenvalues which dictate the new feature space. |
| 7 | Dimensionality Reduction | LDA can be used for dimensionality reduction by projecting data onto a lower-dimensional space. The number of dimensions is at most �−1C−1 where �C is the |

| | | |
|---|---|---|
| | | number of classes. |
| 8 | Multiclass LDA | While originally developed for two classes, LDA can be generalized to more than two classes. |
| 9 | Applications | Pattern recognition, feature extraction, and dimensionality reduction in fields like face recognition and predictive modeling. |
| 10 | Challenges | Assumes linear separability, requires normally distributed classes, and equal class covariance. Sensitive to outliers. |

========================================================================================
========================================================================================

++