**TOPIC MODELLING DESIGN DOCUMENT**

**INDEX**

1. Introduction
2. Use Cases
3. Parameters
4. Expected Input and Output
   * LDA Model
     + Parameters
     + Service Design
     + Model Design
   * NMF Model
     + Parameters
     + Service Design
     + Model Design
5. Data Source
6. Limitations
7. Improvements
8. References

**1.** **Introduction – Topic Modelling**

Topic modeling is technique to extract abstract topics from a collection of text documents. It is a process to automatically identify topics present in a text object and to derive hidden patterns exhibited by a text corpus. Thus, assisting better decision making.

**2. Use Cases**

There are several scenarios when topic modeling can prove useful. Here are some of them:

* **Text classification** – Topic modeling can improve classification by grouping similar words together in topics rather than using each word as a feature
* **Recommender Systems** – Using a similarity measure, topic modelling can recommend articles with a topic structure similar to the articles the user has already read.
* **Uncovering Themes in Texts** – Useful for detecting trends in online publications
* Finding latent topics in a large corpus of document.
* Finding recurring actions in video-surveillance streams
  + <http://www.idiap.ch/~odobez/publications/VaradarajanOdobez-ICCV-VS_2009.pdf>
* Behavior mining of internet users
* Feature Engineering. Reduce redundant text documents used for modelling based on topics

**3. Parameters:**

The parameters required for the Topic Modelling Service are

|  |  |  |
| --- | --- | --- |
| Parameter | Requirement | Description |
| languageModel | default | Th This defines the model that needs to be loaded based on the text being analyzed.  The current default language model is English. |
| Mode | required | Th This defines the mode of modelling required i.e LDA |
| num\_topics | required | This defines the number of topics that need to be extracted from the text. Default is set to 4 |
| words | required | This defines the number of words to be displayed in each topic. Default is set to 6 |
| content | required | The content holds the text that needs to be analyzed to generate the topics in it. |

**4. Expected Input and Output**

**LDA Model:**

Expected input is a text paragraph

Expected output is a set of topics that describe the text

**Example:**

**Input:** documents = ["Human machine interface for lab abc computer applications",

"A survey of user opinion of computer system response time",

"The EPS user interface management system",

"System and human system engineering testing of EPS",

"Relation of user perceived response time to error measurement",

"The generation of random binary unordered trees",

"The intersection graph of paths in trees",

"Graph minors IV Widths of trees and well quasi ordering",

"Graph minors A survey"]

**Parameters:** num\_topics = 4

num\_words = 10

**Output:**

| **Topic # 01** | **Topic # 02** | **Topic # 03** | **Topic # 04** |
| --- | --- | --- | --- |
| **0** | user | system | graph | trees |
| **1** | time | eps | minors | interface |
| **2** | response | human | trees | human |
| **3** | survey | interface | survey | computer |
| **4** | computer | user | computer | graph |
| **5** | system | trees | interface | survey |
| **6** | trees | computer | human | user |
| **7** | interface | survey | user | system |
| **8** | human | graph | system | minors |
| **9** | graph | minors | response | time |

**Service Design**

The service has been designed to accept the parameters in the body of the request and returns the summary of the text entered in a JSON format. A sample input and the output corresponding to the same is mentioned below.

**Input:**

{

“mode”: “LDA”

“content”: “ ”

}

**Output:**

{

}

**LDA Model Design**

LDA is a matrix factorization technique. LDA assumes documents are produced from a mixture of topics. Those topics then generate words based on their probability distribution. Given a dataset of documents, LDA backtracks and tries to figure out what topics would create those documents in the first place.

In vector space, any corpus (collection of documents) can be represented as a LDA’s core use case is detecting underlying core topics across a corpus of text documents

**1st Fundamental assumption:**

LDA suggests that words carry strong semantic information and documents discussing similar topics will use similar set of words. Latent topics are therefore discovered by identifying groups of words in the corpus that frequently occur together within documents.

**2nd Fundamental assumption:**

Is concerning the structure of the documents themselves. LDA suggests that documents are probability distributions of latent topics and topics are probability distribution of words. So, according to LDA every document contains a number of topics and each topic consists of a distribution of words.

**NMF Model:**

**Service Design**

The service has been designed to accept the parameters in the body of the request and returns the summary of the text entered in a JSON format. A sample input and the output corresponding to the same is mentioned below.

**Input:**

To achieve a robust model, a text file with 4551 large size documents have been used

Based on coherence measure TC-W2V, number of topics extracted = 8

Number of words in output = 10

**Output:**

Topic 01: eu, brexit, uk, britain, referendum, leave, vote, european, cameron, labour

Topic 02: trump, clinton, republican, donald, campaign, president, hillary, cruz, sanders, election

Topic 03: film, films, movie, star, hollywood, director, actor, story, drama, women

Topic 04: league, season, leicester, goal, premier, united, city, liverpool, game, ball

Topic 05: bank, banks, banking, financial, rbs, customers, shares, deutsche, barclays, lloyds

Topic 06: health, nhs, care, patients, mental, doctors, hospital, people, services, junior

Topic 07: album, music, band, song, pop, songs, rock, love, sound, bowie

Topic 08: internet, facebook, online, people, twitter, media, users, google, company, amazon

**NMF Model Design**

NMF stands for Non-Negative Matrix Factorization

Find two non-negative matrices (W, H) whose product approximates the non- negative matrix X. This factorization can be used for example for dimensionality reduction, source separation or topic extraction.

The W factor contains the document membership weights relative to each of the k topics. Each row corresponds to a single document, and each column correspond to a topic.

The objective function is:

0.5 \* ||X - WH||\_Fro^2

+ alpha \* l1\_ratio \* ||vec(W)||\_1

+ alpha \* l1\_ratio \* ||vec(H)||\_1

+ 0.5 \* alpha \* (1 - l1\_ratio) \* ||W||\_Fro^2

+ 0.5 \* alpha \* (1 - l1\_ratio) \* ||H||\_Fro^2

Where:

||A||\_Fro^2 = \sum\_{i,j} A\_{ij}^2 (Frobenius norm)

||vec(A)||\_1 = \sum\_{i,j} abs(A\_{ij}) (Elementwise L1 norm)

The objective function is minimized with an alternating minimization of W and H.

To select the number of topics, here we will use a topic coherence measure called TC-W2V. This measure relies on the use of a word embedding model constructed from our corpus. So in this step we will use the Gensim implementation of Word2Vec to build a Word2Vec model based on our collection of news articles.

**5. Reference dataset used for the Topic Modelling**

* <https://dumps.wikimedia.org/simplewiki/latest/> (Wikipedia Dataset)
* <https://www.kaggle.com/therohk/million-headlines/version/2/data> (Headlines Dataset)
* <https://github.com/derekgreene/blob/master/data/articles.txt>

**6. Limitations of the current Topic Model**

1. Not tested against multiple paragraphs
2. Not tested against multi – documents
3. Can’t be measured in terms of accuracy as it’s an unsupervised approach
4. Not comparable to other approaches as well due to unsupervised in nature

**7. Improvements**

1. Adding a frequency filter
2. Batch wise LDA
3. Use of Lemmatization/Stemming might further improve model

**Pros and Cons of Different Implementations:**

* LDA is good in identifying coherent topics where as NMF usually gives incoherent topics. However, in the average case NMF and LDA are similar but LDA is more consistent.
  + <http://aclweb.org/anthology/D/D12/D12-1087.pdf> (Reference comparison of the methods)
* The only difference is that LDA adds a Dirichlet prior on top of the data generating process, meaning NMF qualitatively leads to worse mixtures. It fixes values for the probability vectors of the multinomials, whereas LDA allows the topics and words themselves to vary.  
    
  Thus, in cases where we want the topic probabilities should remain fixed per document (oftentimes unlikely)—or in small data settings in which the additional variability coming from the hyperpriors is too much—NMF performs better.

**8. References**

1. Latent Dirichlet Allocation (LDA) and Topic modeling: models, applications, a survey - <https://arxiv.org/ftp/arxiv/papers/1711/1711.04305.pdf>
2. Latent Dirichlet Allocation - <http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>
3. Latent Dirichlet Allocation - <https://ai.stanford.edu/~ang/papers/nips01-lda.pdf>
4. Nonnegative matrix factorization for interactive topic modeling and document clustering - <https://www.cc.gatech.edu/~hpark/papers/nmf_book_chapter.pdf>
5. Initializations for the Nonnegative Matrix Factorization - <http://meyer.math.ncsu.edu/Meyer/PS_Files/NMFInit.pdf>
6. A document exploring system system on LDA topic model foe Wikipedia articles - <http://aircconline.com/ijma/V8N4/8416ijma01.pdf>