**2.1. Practical Task: Text processing, feature extraction and representation by using both TF and TF-IDF schemes**

**1. Data Preprocessing**

The data is about film reviews, maybe it can be used to make a recommendation engine for viewers. The dataframe includes 4803 rows and 21 columns. There are 3091 missing values in column homepage, 844 missing values in tagline.

Table

Description automatically generated

New column description was created by concatenating the strings from two columns: tagline and overview as followed:

**2. Text processing**

The first step of text analysis is tokenization. Tokenization is a fundamental of natural language processing (NLP) and it is described separating texts into smaller pieces called tokens. These tokens can be either words, numbers, characters, or other types of word forms but in our study, we prefer to eliminate everything except words and numbers. Basic preprocessing steps were applied to the texts along with token creation by ‘preprocessing(corpus)’ function, which involves token lemmatization to reduce token to their base form, token.strip() to remove space, token.lower() to make all the token to lowercase and finally separate stop words from tokens.

**Bag of word (bow) representation and Time Frequency (tf)**

Texts cannot be implemented directly into machine learning models. Bow is one of the necessary preprocessing steps in NLP which turns regular texts to ‘bag-of-words’. Bow has a dictionary like structure, and it has basically two information: (a) words in the raw text and (b) frequency of occurrence of words. Bows are created in three steps in this study. First a ‘corpus’, the combination of all raw texts, was created.

Chart

Description automatically generated

Tf vectors represent absolute numbers which a word appears in a bow and this is not very ideal. In a tf vector, it is not easy to understand weight of a word in a text. It is not the same appearing 5 times in a text with 100 words and another one with 200 words.

**TF and TF-IDF representation on ‘description’**

TF-IDF model is a quick way to acquire information about texts and documents which allows get meaning from texts and compare documents in a mathematical way. It is very easy to implement the code and the method is not computationally expensive. Besides the advantages it also has some disadvantages. It computes document (text) similarity based on only word-count (bag-of-words) and it does not take into account word’s position in the text, semantics and co-occurrences (Ahuja, 2020). TF-IDF method based on word count gives enough information about text similarity which is not always the case. It may be slow to implement this method for large text.

TF means term frequency and IDF denotes inverse document frequency. The more times a word appears in the document, the TF (and hence TF-IDF) will go up. The number of documents that contain that word goes up, the IDF (and hence the TF-IDF) for that word will go down.

**tf(t, d) = count of word t in d / number of words in d**

**idf(t, D) = log(number of documents/number of documents containing the word t)**

**tfidf(t, d, D) = tf(t, d) \* idf(t, D).**

The main function of TF-IDF is to increase value of tokens which are important in a text or document (Chakravarthy, 2020). TF-IDF model is a quick way to acquire information about texts and documents which allows get meaning from texts and compare documents in a mathematical way. It is very easy to implement the code and the method is not computationally expensive. Besides the advantages it also has some disadvantages. It computes document (text) similarity based on only word-count (bag-of-words) and it does not take into account word’s position in the text, semantics and co-occurrences (Ahuja, 2020). TF-IDF method based on word count gives enough information about text similarity which is not always the case. It may be slow to implement this method for large text.

**2.2. Practical Task: Topic modelling**

**Truncated SVD**

LSA is recommended in the course to be our first choice for topic modelling, semantic search, or content-based recommendation engines. LSA (Latent semantic analysis) uses SVD to find the combination of words, that are responsible, together, for the biggest variation in the data. SVD decomposes a matrix into three square matrices, one of which is diagonal.

We use scikit-learns’ TF-IDF vectorizer to take corpus and convert each document into a sparse matrix of TFIDF features

X is input matrix where m is the number of document (n) we have, and m is the number of terms. We decompose X into three matrices called U, S and T. When we do the decomposition, we have to pick a value p, that’s how many topics or concepts we keep.

**X --> UmxpSpxpVpxnT**

m: number of terms in vocabulary

n: number of documents

p: number of topics (same as the number of words)

U is a mxp matrix. The rows will be documents and the columns will be ‘concepts’

S is a kxk diagonal matrix. The elements will be the amount of variation captured from each concept.

V is a mxk (mind the transpose) matrix. The rows is the terms and the columns is the concepts.

Chart, bar chart

Description automatically generated

Here we see that the first topic is about man love life, second about woman love life story, third is about world war film story, forth is about new family friend in town, fifth is about family with father and son osv.

**LDiA/LDA (Laten Dirichlet Allocation)**

In some situation, LDA can give slightly better results than LSA. LDA is introduced by David Blei, Andrew Ng and Michael O. Jordan in 2003 (Edward, 2018). LDA is the most common algorithm for topic modelling. LDA assumes a Dirichlet distribution of word frequencies. It is more precise about the statistics of allocation words to topics than the linear math of LSA. LDA assumes that each document is a mixture (linear combination) of some arbitrary number of topics. It assumes that each topic can be represented by a distribution of words (term frequencies). The probability or weight for each of these topics within a document, as well as the probability of a word being assigned to a topic, is assumed to start with a Dirichlet probability distribution.

Both SVD and LDA use Bag-of-words as input matrix. However, when we use SVD, it is difficult to determine the optimal number of dimensions. In details, low dimension consumes less resources but we may not able to distinguish opposite meaning words while high dimension overcome it but consuming more resource (Edward, 2018).

Chart, bar chart, waterfall chart

Description automatically generated

Here we see that the first topic seems to be about serial killer with clue. The second topic is about beginning of new world and universe, third is about great white oscar, fourth is about man love story, fifth is vampire twin sister osv.

**Word2vec**

Word2Vec takes into account not only words but also context and semantics. Word2vec algorithm uses neural network model to learn associations in a large corpus (Wikipedia 2021). It can detect synonyms and suggest words for partial sentences. It uses word-embedding very efficiently. Therefore, it is used very efficiently by companies like Ali express, AIR BNB, Spotify, etc (Alammar, 2019).

Another efficient alternative NLP model to TF-IDF is BERT (Bidirectional Encoder Representations from Transformers). This method is released in 2018 and stirred NLP community by its highly accurate results in different NLP tasks (Horev, 2018). BERT uses a transformer, an attention mechanism, that learns contextual relationships between words. The transformer has two mechanisms – an encoder which reads texts and a decoder which produces predictions for the text. Directional models read text sequentially, from left to right or right to left. But BERT reads text in both directions; left-to-right and right-to-left. This specialty allows BERT to learn better from words surroundings, from both left and right. One of the most used methods in both industrial and research purposes is ELMo. ELMo models both complex characteristics of word use (syntax, semantics, etc.) and how words are used in linguistic contexts. The internal states of bidirectional language model (biLM), pre-trained in corpus, is used to create word vectors. These vectors can be added to existing model and improve results significantly in different NLP works such as question answering or sentiment analysis (Peters et al. 2018).

**2.3. Analysis Task: Searching for similar movies**

Table

Description automatically generated

Table

Description automatically generated

**Conclusions and Recommendations**

**References**

Jay Alammar (March 27, 2019). The illustrated Word2vec. https://jalammar.github.io/illustratedword2vec/

Madhu Sanjeevi (2017). Chapter 9.1: NLP - Word vectors. https://medium.com/deep-math-machine-learning-ai/chapter-9-1-nlp-word-vectors-d51bff9628c1.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. Deep contextualized word representations.

NAACL 2018. Pallavi Ahuja, (2020). How Good (or Bad) is Traditional TF-IDF Text Mining Technique? https://medium.com/analytics-vidhya/how-good-or-bad-is-traditional-tf-idf-text-mining-technique-304aec920009

Rani Horev (November 10, 2018). Best explained: State of the art language model for NLP. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

Srinivas Chakravarthy (2020). Simple Word Embedding for Natural Language Processing. https://towardsdatascience.com/simple-word-embedding-for-natural-language-processing5484eeb05c06.

Wikipedia (2021). Word2vec. https://en.wikipedia.org/wiki/Word2vec. Accessed October 29, 2021

<https://www.youtube.com/watch?v=BJ0MnawUpaU>

Edward (2018), [2 latent methods for dimension reduction and topic modeling | by Edward Ma | Towards Data Science](https://towardsdatascience.com/2-latent-methods-for-dimension-reduction-and-topic-modeling-20ff6d7d547)