Text Representation and Classification using TF-IDF, FastText and Naives Bayes Classifier

Ndjebayi Damaris, Abduljaleel Adejumo,Ndeye Ngone Gueye

African Master in Machine Intelligence , AMMI-Senegal

Supervised by Prof. Moutapha Cisse

April 18, 2023

Overview

1. Introduction and Methodology
   - Introduction
   - Methodology
2. Results/ Discussions
   - Implementation
   - Discussions

Overview

1. Introduction and Methodology
   - Introduction
   - Methodology
2. Results/ Discussions
   - Implementation
   - Discussions

Introduction

- Task of classifying text documents.

Introduction

- Task of classifying text documents.
- Computer understanding number

Introduction

- Task of classifying text documents.
- Computer understanding number
- e-mail spam classification,Sentiment analysis of online reviews. Topic labeling documents like research papers.

Objective

- In this report we present the textual representation methods by focusing on TF-IDF and Fast text in a first step, then,

Objective

- In this report we present the textual representation methods by focusing on TF-IDF and Fast text in a first step, then,
- In a second step, we analyze how these two approaches influence text classification using Bayes classifier

Overview

1. Introduction and Methodology
   - Introduction
   - Methodology
2. Results/ Discussions
   - Implementation
   - Discussions

Methodology

Term Frequency- Inverse Document Frequency

1. **TF-IDF:** This is done by multiplying two metrics: one which gives us information on how often a term appears in a document (Term Frequency), and another gives us information about the relative rarity of a term in the collection of documents (Inverse Document Frequency).

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$

- **Term Frequency** Term frequency works by looking at the frequency of a particular term you are concerned with relative to the document.

$$TF(t,d) = \frac{F_{t,d}}{\sum_{t' \in d} F_{t',d}}$$

- **Inverse Document Frequency** IDF is computed as follows where t is the term (word) we are looking to measure the commonness of and N is the number of documents (d) in the corpus (D). The denominator is simply the number of documents in which the term, t, appears in.

$$idf(t,D) = \log \frac{N}{\{d \in D : t \in d\}}$$

The above formula is usually implemented as follow:

$$idf(t,D) = \log \frac{N}{\{d \in D : t \in d\} + 1}$$

Methodology

Word embedding:World2Vec

FastText is extension of **World2Vec** which is an embedding technique that takes into acount the semantic and the sintactic meaning

Methodology

Word embedding:World2Vec

- FastText is extension of **World2Vec** which is an embedding technique that takes into acount the semantic and the sintactic meaning
- **Word embedding** technique is to represent words in vector form

Methodology

Word embedding:World2Vec

FastText is extension of **World2Vec** which is an embedding technique that takes into acount the semantic and the sintactic meaning

Word embedding technique is to represent words in vector form

- The word2vec model computes the vectors using two main architectures: **CBOW** and **Skip-gram**.

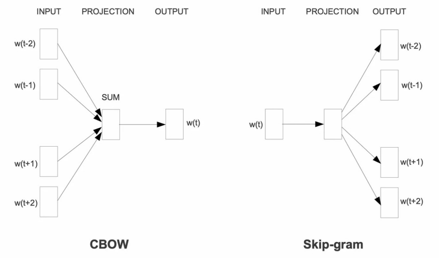

Figure: CBOW and Skip-gram

Methodology

Limitations of Word2Vec

1. **Limitations of Word2Vec:**

Methodology

Limitations of Word2Vec

1. **Limitations of Word2Vec:**
   1. Word2Vec cannot provide embeddings for out of vocabulary words.

Methodology

Limitations of Word2Vec

1. **Limitations of Word2Vec:**
    1. Word2Vec cannot provide embeddings for out of vocabulary words.
    2. Can't provide morphologically rich languages.

Methodology

Limitations of Word2Vec

1. **Limitations of Word2Vec:**
   1. Word2Vec cannot provide embeddings for out of vocabulary words.
   2. Can't provide morphologically rich languages.
   3. Assigning a distinct vector to each word.

Methodology

FastText

FastText provide embedding for **Character n-gram**

Methodology

FastText

FastText provide embedding for **Character n-gram**

- **Character n-gram**
  Is a n-grame chunk of n consecutive words for characters.
  For example: fasttext will have character n-gram ⟨fa , fas , ast , stt , tte , tex , ext , xt⟩ with n=3

Methodology

FastText

FastText provide embedding for **Character n-gram**

- **Character n-gram**
  Is a n-grame chunk of n consecutive words for characters.
  For example: fasttext will have character n-gram ⟨fa , fas , ast , stt , tte , tex , ext , xt⟩ with n=3
- **Difference with the scoring function**

$$s_F(w_t, w_c) = \sum_{g \in S_{w_t}} z_g^T v_c \text{ and } s_{SK}(w_t, w_c) = u_t^T v_c$$

Where:
- g is a character n-gram of w_t and S_{w_t} the set of character n-gram of w_t ; -z_g: the vector a character n-gram ; - vectors z_g^T and v_c , corresponding respectively to world w_t and w_c

Methodology

Naive Bayes

- **Bayes Theorem**
  "In simpler terms, Bayes' Theorem is a way of finding a probability when we know certain other probabilities."
  The formula is given:-

$$P(A \mid B) = \frac{P(A)\,P(B \mid A)}{P(B)}$$

- **Naive Bayes assumptions**
  1. The assumption made here is that the predictors/features are independent.
  2. Equal contribution the outcome.

Naive Bayes

Giving a set of feature $(x_1, x_2, x_3, ..., x_n)$ are element of X and with label y the bayes theorem can be written as:

$$P(y \mid X) = \frac{P(y)\, P(X \mid y)}{P(y)}$$

By substituting for X and expanding using the chain rule we get,

$$P(y \mid X) = \frac{P(y)\, P(x_1 \mid y)\, P(x_2 \mid y)\, P(x_3 \mid y) \ldots P(x_n \mid y)}{P(y)}$$

Naive Bayes

The denominator it can be eliminated, and proportionality can be introduced for simpler calculations.

$$P(y/x_1, , x_n) \propto P(y) \prod_{i=1}^{n} P(x_i/y)$$

$$y = \underset{y}{\operatorname{argmax}} P(y) \prod_{i=1}^{n} P(x_i/y)$$

- **Laplace Smoothing**
  Notice that some probabilities estimated by counting might be zero

$$P\left(X_j = v \mid Y = y_k\right) = \frac{\mathrm{c}_v + 1}{\sum_{v' \in \text{values}(X_j)} \mathrm{c}_{v'} + \mid \text{ values } (X_j) \mid}$$

Overview

1. Introduction and Methodology
   - Introduction
   - Methodology
2. Results/ Discussions
   - Implementation
   - Discussions

Implementation

Follow this link to check the code implementation ( )

Overview

1. Introduction and Methodology
   - Introduction
   - Methodology
2. Results/ Discussions
   - Implementation
   - Discussions

Conclusion

- The TF-IDF is faster in computation
- Using the Naive Bayes Classifier, the TF-IDF gave a better result than the FastText

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Toma s Mikolov. 2016. Enriching word vectors with subword in- formation. CoRR, abs/1607.04606.

Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. pages 1–9, 01.

Webb, G.I. (2011). Naïve Bayes. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8$_5$76

AIMS African Institute for Mathematical Sciences SENEGAL

Acknowledgements