Submission - Exercises sheet 1

MA-INF 4236 - Advanced Methods for Text Mining, So24

Yara Elwakeel 50135730

May 15, 2024

1 Preprocessing.

1.1 Concepts.

1. **Sentence tokenization**
   A tokenizer segments the The text into sentences. As once a text is divided into sentences, it becomes easier to analyze the grammatical structure and syntax of each sentence. It identifies the boundaries of sentences in a given text. Typically, sentences end with punctuation marks (eg, "?", ",", ".") as they are keys of sentence boundaries. This is crucial for tasks such as grammatical analysis and dependency parsing. Additionally, it improves efficiency as Working with individual sentences rather than large blocks of text enables a more focused analysis, as each sentence can be processed separately.

2. **Word Tokenization**
   It decomposes each text string into a sequence of words (technically tokens) for computational analysis . Where it allows for further analysis and manipulation of the text, such as lemmatization, stemming, or part-of-speech tagging. in addition, tokenization is often the first step in preparing data for training. It ensures that the data is in a structured format suitable for modeling. NLTK has module word_tokenize() which uses a default tokenizer that can handle different types of punctuation and language nuances to correctly split the text into words.

3. **Part-of-speech (POS) tagging**
   Part of Speech (POS) Tagging aims at assigning each word of a text the proper syntactic tag in its context of appearance. also called grammatical tagging, it can be seen as a grammatical classification that includes verbs, adjectives, adverbs, nouns, etc. it's an essential step in many NLP tasks as it provides grammatical structure and syntactic information about the text. as a word meaning is dependent on it's POS and it's context. especially in machine translation, information retrieval , named entity recognition as it helps identify PROPN and named entities and Text-to-speech as it also includes cues for correct pronunciation and inflection.
   Several POS tagging approaches have been proposed to automatically tag words with part-of-speech tags in a sentence. Te most familiar approaches are rule-based where uses hand-crafted rules to assign tags to words in a sentence, artificial neural network Uses neural network architectures such

as RNN to assign POS tags ,a stochastic is a statistical method that uses probability and statistics to determine the most likely POS tag for a given word based on its context(e.g hidden markov models(HMM) or conditional random fields(CRF)) . and a hybrid approach which combines different methodologies, such as rule-based and stochastic methods, to take advantage of the strengths of each method and improve performance. [2]

4. **Lemmetization**
   The goal is to normalize different inflected forms of a word Lemmatization is a natural language processing (NLP) technique that groups words based on their lemma, or base form, by removing inflectional morphemes. where an inflectional morpheme is a bound morpheme added to a word to indicate grammatical properties. In addition, Lemmatization takes into account the part of speech (POS) of a word, such as noun, verb, adjective, or adverb, so that they can be analyzed or compared more easily. In Python, the WordNetLemmatizer class from the Natural Language Toolkit (NLTK) leverages the WordNet lexical database to lemmatize each word in an input sentence according to its POS tag.The lemmatizer can also be applied to individual words, mapping them to a single, most common lemma, or providing different lemmas based on the specified POS tag. For instance, the word "leaves" can have the lemma "leave" (verb) or "leaf" (noun), Lemmatization is especially useful for parsing languages like Turkish and Arabic, where grammatical relationships such as subject, verb, and object are indicated by changes in the words themselves. [1, p. 526]

5. **Stop words removal**
   Stop words refer to the words that do not distinguish one text document from another in the corpus Examples include "the" and "or" because they are extremely common across documents, leading to little distinction among each document.so that text models can focus on the distinctive meaningful words that carry more information thus improving the model's performance , these non distinctive words with high number of occurrences can increase the data dimensionalty which may distort the results of a machine learning algorithm . you can use a predefined list of stop words from libraries such as NLTK which possess a list of 127 English stop words, SpaCy , or even create your own custom list based on the used language and applied problem . [1, p. 523]

1.2 Algebra Basics.

1.2.1 describe briefly

1. **What is a vector ?**
   a vector is a mathematical object that is used to describe both magnitude and direction, at which a vector is represented only by it's magnitude and direction vectors are elements of the vector space, at which they are used to describe both magnitude and direction.

2. **What is a matrix ?**
   A matrix is a rectangular array of numbers arranged in rows and columns, However, within the context of vectors, matrices are used to represent a

collection of vectors. they are also utilized to represent linear transformations of a vector(e.g.rotation, scaling, shearing etc..), by deploying a vector multiplication on the vector with this linear transformation matrices you are representing a transformed version of our vectors
In essence, matrices in the context of vectors are not just arrays of numbers, but dynamic tools enabling us to represent, manipulate, and transform vectors in multidimensional spaces

3. **What is a rank of a matrix ?**
the rank of a matrix is a method to represent the number of linearly independent vectors, meaning vectors that are not expressed as linear combination of other vectors. wheres a matrix possess a row rank as well as column rank, and they are always equal.
in the context of vectors and vector space, these linearly independent vectors form a basis for the vector space. This basis provides a minimal set of vectors that can be used to express any vector in the space through linear combinations. and also in the context of machine learning they represent the number of dimensions(features) present in the data that represent useful and not redundant noisy information on the data.

4. **What is outer product expansion ?**
The outer product expansion is done between two vector that involves taking each element of one vector and multiplying it by each element of the other vector. The result of this operation is a matrix where the dimensions are determined by the lengths of the input vectors. If u is an m-dimensional column vector and v is an n-dimensional row vector, then their outer product will result in an mxn matrix.When performing the outer product operation on two matrices, it involves taking the outer product of the columns or rows of one matrix with the columns or rows of the other matrix.

5. **What is cosine similarity ?**
cosine similarity is a mathematical method that is used to measure the similarity between vectors, in it's essence it is a normalized dot product, as dot product is the projection of one vector onto another scaled by the lengths of the vectors and the cosine the angle between them, the larger the dot product the more aligned the vectors are with each other

$$a \cdot b = \|a\|\|b\| \cos(\theta)$$

however this scale is highly influenced by the vector's magnitude that's why cosine similarity is used, as it's normalized by the product of vector's norms, thus taking into account also the orientation between vectors

$$\cos(\theta) = \frac{a \cdot b}{\|a\|\|b\|}$$

whereas the context of NLP, cosine similarity is commonly used to measure the similarity between word embedding, enabling tasks such as document similarity, information retrieval, and clustering.

1.2.2 gradient of a vector or a matrix

The gradient of a vector measures how much each component of the vector's output changes with respect to each input variable. It represents the slope of the tangent line to the vector's graph at a specific point, providing a precise description of the instantaneous rate of change of each component. This allows for accurate approximations of the vector's behavior near that point.

1.2.3 linear classifier

a linear classifier is a machine learning method that is used in binary classification tasks, e.g. $c_x \in \{C_1, C_2\}$, where it divides the data points in the feature space depending on a decision boundary function, as shown in the following equation.

$$c(x) = v^T x + v_0 \tag{1}$$

where v is Vector containing weights for each feature in the input data. Each element represents the importance of the corresponding feature to the classification decision. It determines the orientation of the decision boundary in the feature space. whilst V_0 is a Constant value added to the weighted sum of input features to shift the decision boundary in a linear classifier. It determines the position of the boundary in the feature space. This decision boundary splits the vector space into two parts depending on our label if $c_x = C_1$ if $c(x) > 0$ and c_x is a C_2 class label if $c(x) < 0$.
we can rephrase by simplifying equation in 1 by concatenating V_0 with V values

$$W^T = ([v_0,] \circ v)^T \tag{2}$$

and to compensate for this change we also concatenate a vector of ones to the start of our input

$$X = (1 \circ x) \tag{3}$$

from 2 and 3 we obtain:

$$c(X) = W^T X \tag{4}$$

5 TF-IDF

TF-IDF stands for The Term Frequency-Inverse Document Frequency is a statistical method that is used to process textual-data that measures how important a word is in a set of documents where it's a scaled down version of bag of words, it's the result of multiplying of two terms
Term frequency(TF), which measures the frequency of words that appeared in a document ,however to address the variation in document lengths, word frequency is often normalized by the total document length, ensuring fair comparison across documents regardless of their size.

$$tf_{term} = \frac{numberoftimesatermappearsinadocumnet}{totallnumberoftermsinthedocument} \tag{5}$$

Inverse document frequency(*IDF*), that measures the importance of a word across the entire document collection, where it assigns lower weight to frequent

words and assigns greater weight for the words that are infrequent. thus reducing the influence of common words and highlighting the importance of rare ones

$$IDF_{term} = \log(\frac{totallnumebrofdocumnets}{numberofdocumentscontainingtheterm}) \tag{6}$$

by multiplying both terms Tf in equation 5 with IDF in equation 6 :

$$TF\text{-}IDF_{term,document} = TF_{term,document} * IDF_{term,document} \tag{7}$$

yielding a score that enables the identification of terms that are both frequently occurring within a document and relatively rare across the entire document collection. [3]

The TF-IDF approach typically provides a more nuanced and informative representation of text data compared to BOW approach, particularly in tasks requiring document similarity analysis. as BOW assigns equal importance to all terms, which leads to common words taking over the representation and less informative terms being ignored.

References

[1] Christine P. Chai. Comparison of text preprocessing methods. *Natural Language Engineering*, 29(3):509â553, 2023.

[2] Alebachew Chiche and Betselot Yitagesu. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9(1):10, Jan 2022.

[3] Shahzad Qaiser and Ramsha Ali. Text mining: Use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, 181, 07 2018.