Software and Information System Engineering

Faculty of Engineering Sciences

Ben Gurion University

Cohesion Pipeline Project

Cohesion פרויקט

Project Number - 372-22-11

2022

Students: Tom Nachman, Asaf Salomon and Eden Berdugo

Academic Advisors: Dr. Rami Puzis, Dr. Aviad Elyashar

Project Mentor: Dr. Lior Rokach

Contents

Project Common Language

To have a common language in this project, we will define these definitions:

- Corpus - All documents in the dataset.
- Label - The group number to which the document belongs.
- Division - List of tuples that each tuple is a document and its label.
- Topic - List of words that represent the documents under it.

Chapter 1

Project Summary

1.1 English Summary

The topic-Detection field deals mainly with providing names to given divisions of documents and lacks a quality measurement that provides a rating for the division, that represent a human-subjective score.

Our goal is to create a new measurement, named "Cohesion", which examines the distribution quality of the documents to topics and competes with the Coherence measurement which is considered SOA in the Topic-Detection domain. Our method is to give each collection a suitable topic, and value it against the related and the foreign docs combined into a "Cohesion Formula".

As part of the process and feasibility study, we examine the Cohesion measurement of different divisions shuffled in different percentages, against NMI results (a heuristic technique to estimate the clustering quality of two groups or more) to show a higher correlation than other measurements- we perform the comparison between NMI - Coherence to NMI - Cohesion.

We give names to each group using a C-TF-IDF algorithm. To calculate the cohesion score we use a pre-trained MNLI model as a ready-made zero-shot-sequence-classifier on which calculations are performed between the groups and within each group (intra-score) to obtain an average score that presents the cohesion score.

The scoring technique is done by iterating over the topics and the documents, summing the NLP-based depending on the relation to the topic - positive if the document belongs to the topic, else we add it aa s negative score. Then, we perform a weighted average of the entire distribution of documents, between the positive and negative scores of each topic. Finally, we get the result - a Cohesion score for the whole divided corpus, a number between 0 to 1.

Contrary to Coherence which does not consider the distribution of the documents but only the corpus and the topic’s names as one unit, Cohesion considers the suitability of the docs to the topics. In this way, we reach higher accuracy that reflects the human-subjective score.

1.2 Hebrew Summary

בתחום Topic Detection, אשר מתמחה במתן שמות לחלוקות נתונות, ישנו חוסר במדד איכותי שנותן דירוג לחלוקה בצורה הקרובה ביותר לדירוג אנושי. מטרתנו היא ליצור מדד חדש לבדיקת טיב חלוקה של מסמכים לפני נושאים, אשר בנוסף לציון סובייקטיבי איכותי, יוכל גם להמליץ על שמות מתאימים לקבוצות השונות בחלוקה.

כחלק מההנחות בפרויקט המחקר שלנו, אנו מתבססים על ההבנה כי כיום מדד ה-Coherence נחשב למדד המוביל בתעשייה בתחום זה, לכן אם נוכיח כי אנו טובים יותר ממנו, כאילו הוכחנו שאנו טובים מכל השאר. למדד שלנו קראנו Cohesion, בו השיטה היא מתן שם לכל קבוצה ובדיקת לכידות המסמכים לשם של הקבוצה כנגד שמות הקבוצות האחרות.

כחלק מהתהליך ובדיקת ההיתכנות של המודל, אנו בוחנים את תוצאות המדד שלנו בחלוקות שונות ובאחוזי רנדומליות שונים, אל מול תוצאות ההיוריסטיקה NMI - היוריסטיקה אשר בודקת איכות חלוקה בין 2 קבוצות או יותר, ומראים קורלציה גבוהה יותר משאר המדדים. כלומר, נצפה כי חלוקה מושלמת תקבל ציון גבוה מאוד וככל שנכניס רנדומליות גבוהה יותר לחלוקה (אחוז שגיאה גדול יותר) ומדד ה-NMI ייתן ציונים נמוכים יותר, נרצה לראות את מדד ה-Cohesion משתנה בהתאם.

את מתן השמות לכל קבוצה אנו מבצעים באמצעות C-TF-IDF, לאחר מכן בשביל לחשב את ה-Cohesion אנו משתמשים במודל MNLI מאומן, כל שם של קבוצה נבדקת אל מול כל המסמכים אשר מקבלים ציון התאמה (ציון חיובי אם המסמך נמצא בקבוצה שלו וציון שלילי אם זה מסמך מקבוצה אחרת), לבסוף מבצעים שקלול של הנתונים ונרמול לפי גודל הקבוצות ע״מ לקבל ציון שלא מושפע מגודל הקבוצה.

בניגוד למדד ה-Coherence אשר לא מתחשב בחלוקת המסמכים, אלא רק במאגר המסמכים ובשמות שניתנו לקבוצות, במודל שלנו אנו בודקים את ההתאמה (ואת חוסר ההתאמה) של כל מסמך לכל שם קבוצה. בצורה זו אנו מגיעים בממוצע לדיוק גבוה יותר ולציון שמשקף מדד אנושי-סובייקטיבי.

Chapter 2

Introduction

2.1 General Introduction

Our Project ‘Cohesion’ is a part of Rami Puzis and Aviad Elyashar’s research. Our part in the research deals with Topic Detection, in particular Topic Modeling-Evaluation. Dealing with Big-Data such as an enormous amount of Twitter tweets might bring many obstacles with it. One of them is getting perspective and understanding the main topics the data is about, “Topic Detection”.

Even after running an NLP model, data scientists struggle with tuning and getting the best “human subjective” partition, which leads to running with many different configurations that costs a lot of frustration as well as computational and data-experts time.

Our goal is to solve this problem by suggesting a new pipeline called ‘Cohesion Pipeline’ that given a data partition into labels, the pipeline will output representative topics followed by the Cohesion-score that will reflect how much the topics are close to “Human-Evaluation”.

2.2 Problem statement - Motivation

One of the main challenges in text analysis is determining the concept a document/corpus is discussing. This information is clear to a human reading a document, but a computer program is given only the text as it was written, not the subject matter of each document or a group of documents. To accomplish this task in a computer program, data scientists utilize a method called Topic Detection.

Our project ‘Cohesion Pipeline’ deals with measuring and evaluating the division of corpus to different labels while suiting each collection a representative topic, an emerging field called “Topic Modeling Evaluation”. This section focuses on the field of “Topic Detection”, and its different derivatives of it.

2.3 Project Goals (Research Question)

The main goal of this research is to suggest a new model for scoring a division of documents. We compare our model to the top models in the Topic Detection field. Our research goals are stated as follows:

- Given a division, the 'Cohesion Pipeline' should give a cohesion score and recommend a suitable name for each group. A suitable name should suit each document of the group and be the tightest one (no entities in other groups suits the given name). We do it by using inter and intra scores.
- Create a software package available at PyPi (Python Package Index) called 'cohesion-pipeline' that accepts a Pandas data frame which represents the data divided into labels and outputs the cohesion-score and the topics for each given label.

2.4 Project Assumptions (Current State)

To help us achieve the goals we consider the following assumptions:

- Existing measurements don't consider subjective topic methods.
- The Coherence measurement is reputed as a state-of-the-art topic modeling measurement.

Chapter 3

Literature Review

3.1 Topic Modeling Techniques

Topic modeling was originally developed in the 80s and branched off from the subject area of ‘‘generative probabilistic modeling”. The first method developed for this task called TF-IDF [1].

Topic modeling is an emerged popular statistical tool for extracting latent variables from large datasets, mainly text data (corpus), but it can also be images, music, and any kind of other media. The use-cases amount is huge, one of them is constructing databases of journals and articles into groups based on a similar focus. In our case – finding the most coherent topics in a big-data corpus.

Despite its popularity, topic modeling is prone to serious issues with optimization, noise sensitivity, and instability which can result in data that is unreliable [1]. Because of that, different evaluations for topic modeling emerged. Our project suggests, a new topic modeling measurement tool, that pretends to reflect subjectively, a collective human score for Topic Model quality.

Although current topic modeling approaches perform better than early algorithms, to begin a data analysis project, it is imperative to understand the differences between models and their algorithms which there are relying on to use the most suited model for the use case.

Topic modeling methods:

- **LSA: Latent Semantic Analysis**.
  Proposed by Landauer et al. in 1998 [2], particularly distributional semantics, can be used in several areas, such as topic detection. the theoretical foundation of the method states: that terms with similar meanings are closer in terms of their contextual usage, assuming that words that are near in their meaning show in the related parts of texts.
  The Method:

  1. Generate term-by-document matrix (i.e TF-IDF matrix) (A): the term has a large weight when it occurs frequently across the document but infrequently across the corpus.

2. Applying SVD (singular value decomposition) - linear algebra technique that factorizes (A) into the product of 3 separate matrices: *A=U*S*V*, where *S* is a diagonal matrix of the singular values of *A*. SVM reduces dimensionality by selecting on the t largest singular values, and only the first t columns of U and V, in our case - Topic Modeling t is the hyper parameter we choose to describe the number of desired topics to result in keeping the t most significant topics.

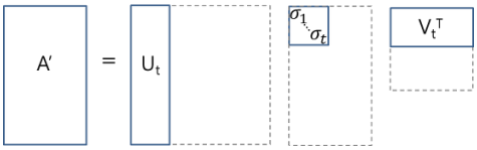

Figure 1 - Metrics product in SVD technique

In topic modeling case:

$$U_t - Doc\ X\ topics V_t - term\ X\ topics$$

3. Apply measures such as cosine similarity to evaluate the similarity of different documents or words.

- **LDA: Latent Dirichlet Allocation.**

LDA assumes that each document is generated by a statistical generative process. That means that each document is a mix of topics, and each topic is a mix of words. It's considered to be the most popular topic modeling algorithm in real-life applications, alongside (BERTopic). A significant advantage of using the LDA model is that topics can be inferred from a given collection without input from any prior knowledge. LDA ignores the order of occurrence of words and treated it as a bag of words.

The Method:

Given the K hyperparameter - the number of desired topics:

1. Generate Word - Topic distribution matrix (A^{wxt}) using Gibbs sampling.
2. Generate Document-word distribution matrix (B^{dxw}).
3. Multiply $BxA = M^{dxt}$

4. for each vector in M (doc) choose the biggest value (Topic) - to get the related topic for this document, for each column in A (topic) choose the biggest values (Words) - to get words that represent the topic the most.

- **BERTopic**

  BERTopic is a topic modeling technique that leverages transformers and C-TF-IDF to create dense clusters allowing for easily interpretable topics whilst keeping important words in the topic descriptions [3]. Comes with an active community, detailed and easy-to-implement documentation, and API supported by python.

  The Method:

  1. Embedding the Documents - using pre-trained models to convert the documents into numerical data (vectors of numbers). Although BERT is typically used for embedding documents, any embedding technique can be used.
  2. Cluster Topics into semantically similar clusters – documents with similar topics will be clustered together, so we can find the topics inside these clusters. Before we do so we need to handle the sparsity of the embedded matrix.
     2.1 Dense - Dimension Reduction Algorithm [4]: apply a dense algorithm to our data, to lower the dimensions while keeping on a significant portion. UMAP is fast, can handle large datasets and high dimensional, it suits for use as a pre-task for other machine learning tasks and it partners well with HDBSCAN.
     2.2 HDBSCAN [5] - Hierarchical Density-Based Spatial Clustering of Applications with Noise. Performs DBSCAN over varying epsilon values and integrates the result to find a clustering that gives the best stability over epsilon. This allows HDBSCAN to find clusters of varying densities (unlike DBSCAN).

     After this section, we have clustered-similar-documents together, which represent the topics they consist of.

  3. Create topic representations from clusters: in this section, we derive topics from clustered documents.
     3.1 c-TF-IDF: the regular TF-IDF compares the importance of words between documents. c-TF-IDF, treat all the documents in the same cluster as one (very long)

document, and then apply TF-IDF. The score demonstrates the important words in each cluster. Now we can look at the top 10 important words in each cluster to get a shared theme they create - Topic, this does not mean the topics are coherent, it even might overfit the documents. To improve it we do:

3.2 MMR - Maximal Marginal Relevance Model:

The maximal marginal relevance model creates a diverse ranking over the words in the same cluster (topic) with a process of sequential word selection. At each step, the word with the highest marginal relevance is selected and added to the tail of the list [6]. This results in finding the most coherent words without having much overlap between the worlds themselves.

3.2 Topic Model Evaluation

Topic model evaluation is the process of assessing how well a topic model does what it is designed for. As with any model, if you wish to know how effective it is at doing what it's designed for, you'll need to evaluate it. Therefore, topic model evaluation matters.

Evaluating a topic model can help you decide if the model has captured the internal structure of a corpus. We know probabilistic topic models, such as LDA, are popular tools for text analysis, providing both a predictive and latent topic representation of the corpus. However, there is a longstanding assumption that the latent space discovered, by these models, is generally meaningful and useful, and that evaluating such assumptions is challenging due to its unsupervised training process. Besides, there is a no-gold standard list of topics to compare against every corpus.

Nevertheless, it is equally important to identify if a trained model is objectively good or bad, as well as have the ability to compare different models/methods. To do so, one would require an objective measure of the quality.

There are several ways to evaluate the quality of topic models according to Chang [7], including:

- Human judgment - whether the topics contain words that, according to a person's subjective judgments, are representative of a single coherent concept.
    - Observation-based, e.g., observing the top N-words in a topic.
    - interpretation-based, e.g., 'word intrusion' and 'topic intrusion' to identify the words or topics that "don't belong" in a topic or document.
- Quantitative metrics - Perplexity (held out likelihood) and coherence calculations.
- Evaluating topic quality using model clustering - Silhouette Coefficient.

3.3 Quantitative metrics

3.3.1 Perplexity

Perplexity is a statistical measure of how well a probability model predicts a sample. In topic modeling, perplexity captures how surprised a model is by new data it has not seen before and is measured as the normalized log-likelihood of a held-out test set.

based on the log-likelihood idea, we can see the perplexity metric as measuring how ‘odd’ will it be to replace words with a related topic word. Same as saying - How well does the model represent or reproduce the statistics of the hidden data. However, as shown in Jonathan Chang [7] research that predictive likelihood (perplexity) and human judgment often do not goes hand in hand.

They [7] ran a large-scale experiment on the Amazon Mechanical Turk platform. For each topic, they took the top five words (ordered by frequency) of that topic and added a random sixth word. Then, they presented these lists of six words to participants asking them to identify the intruder word.

If every participant could identify the intruder, then we could conclude that the topic is good at describing an idea. On the other hand, if many people identified one of the topic’s top five words as the intruder, it means that they could not see the logic in the association of words, and we can conclude the topic was not good enough.

It is important to understand what is that experiment proves. the result proves that, given a topic, the five words that have the largest frequency within their topic are usually not good at describing one coherent idea, at least not good enough to be able to recognize an intruder word.

This limitation of perplexity measure served as a motivation for more work trying to model the human judgment, which leads directly to another measurement - Coherence.

3.3.2 Topic Coherence

As mentioned above and as mentioned in Doogan [8] one of the shortcomings of perplexity is that it does not capture context, i.e., perplexity does not capture the relationship between words in a topic or topics in a document.

Since human judgment not being correlated to perplexity (or the likelihood of unseen documents), It created the motivation for more work trying to model the human judgment. This is by itself a hard task as human judgment is not clearly defined; for example, two experts can disagree on the usefulness of a topic.

To overcome this, different approaches been developed which attempts to capture context between words in a topic. These approaches measure and assign a score to a single topic by measuring the degree of semantic similarity between high-scoring words in the topic. These measurements help distinguish between topics that are semantically interpretable topics and topics that are artifacts of statistical inference. These approaches are collectively referred to as 'coherence'.

There has been a lot of research on coherence over recent years and as a result, there are variety of methods available to calculate it. A useful way to deal with this is to set up a framework that allows you to choose the methods that you prefer. Such a framework has been proposed by Röder [9], The framework is made up of four stages:

1. Segmentation
2. Probability estimation
3. Confirmation
4. Aggregation

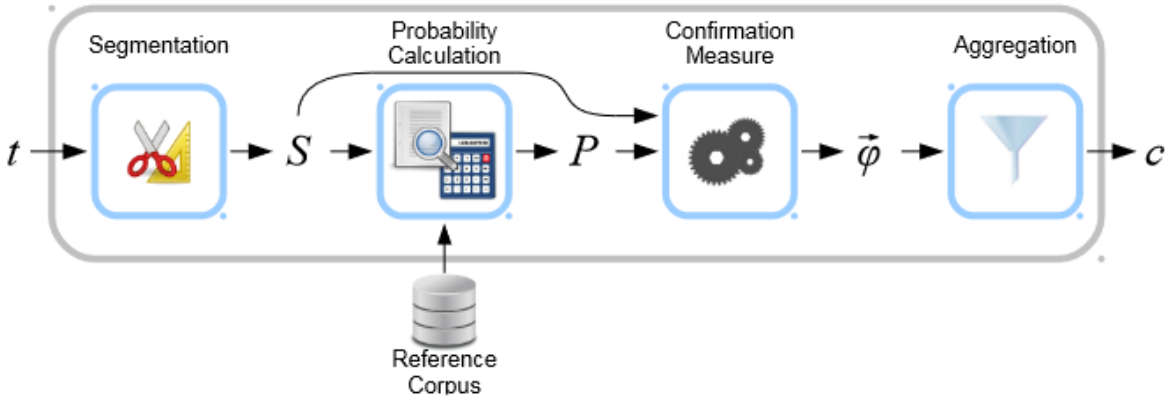

Figure 2 - Overview of the unifying coherence framework

These four stages form the basis of coherence calculations and work as follows:

1) **Segmentation** - Sets up word groupings that are used for pair-wise comparisons. To calculate the coherence of a set of topics, we start by choosing words within each topic (usually the most frequently occurring words) and comparing them with each other, one pair at a time.
   According to Röder [9] Segmentation is the process of choosing how words are grouped for these pair-wise comparisons.

   Word groupings can be made up of single words or larger groupings. For single words, each word in a topic is compared with each other word in the topic, for 2-word or 3-word groupings, each 2-word group is compared with each other 2-word group and so on. comparisons can also be made between groupings of different sizes, for instance, single words can be compared with 2-word or 3-word groups.

2) **Probability** - The method of probability estimation defines the way how probabilities are derived from the underlying data source. In [9] they presented two ways that underpin the calculation of coherence which are UCI and UMass:
   - UCI - Is based on point-wise mutual information (PMI) calculations. This is given by formula (2) for words w_i and w_j and some small number e, where $P(w_i)$ is the probability of word i occurring in a topic and $P(w_i, w_j)$ is the probability of both words i and j appearing in a topic. Here, the probabilities are based on word co-occurrence counts.

$$C_{\text{UCI}} = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \text{PMI}(w_i, w_j) \quad (1)$$

$$\text{PMI}(w_i, w_j) = \log \frac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j)} \quad (2)$$

Figure 3 - calculation of UCI score

   - UMass - Caters to the order in which words appear and is based on the calculation of: $log\ log\ (\frac{P(w_i,w_j)+e}{P(w_j)})$ with w_i, w_j, P(w_i) and $P(w_i, w_j)$ as for UCI. Here, the probabilities are conditional, since $P(w_i|w_j)$ = [$P(w_i, w_j)$ / $P(w_j)$], which we know

from Bayes' theorem. So, this approach measures how much a common word appearing within a topic is a good predictor for a less common word in the topic.

$$C_{\text{UMass}} = \frac{2}{N \cdot (N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \frac{P(w_i, w_j) + \epsilon}{P(w_j)} \quad (4)$$

Figure 4 - calculation of UMass score

3) **Confirmation** - Measures how strongly each word grouping in a topic relates to other word groupings (i.e., how similar they are). There are direct and indirect ways of doing this, it is mostly depending on the frequency and the distribution of the words in the topics.

4) **Aggregation -** This is the final step of the coherence framework, it is a summary calculation of the confirmation measures of all word groupings, resulting in a single coherence score. This is usually done by averaging the confirmation measures using the mean or median. Other calculations may also be used, such as the harmonic mean, quadratic mean, minimum or maximum.

Mid Summary:

Human judgment approaches	
Method	**Description**
Observation-based	Observe the most probable words in the topic
Interpretation-based	Word intrusion and topic intrusion

Quantitative approaches	
Method	**Description**
Perplexity	Calculate the held-out log-likelihood
Coherence	Calculate the conditional likelihood of co-occurrence

Table 1 - Methods of Topic Models Evaluation

The Coherence Problem

We can say that the coherence measurement is certainly a step in the right direction, but it does not completely solve the problem as Hoyle [10] said. For instance, it is possible that a larger topic model (i.e., 100 topics) has captured all the information a smaller model (35 topics) does. The larger model may have captured some additional information and junk.

The coherence measurement can rank the larger model as less coherent - which is not true. It would be trivial for a human to determine which junk topics are, and consequently ignore these and only use the information from the informative topics. It is therefore very difficult to get away from needing to use both metric evaluations and manual/visual inspection of the models.

We agree with Hoyle [10] that says, "topic model evaluation suffers from a validation gap: automated coherence… To the extent that our experimentation accurately represents current practice, our results do suggest that topic model evaluation - both automated and human - is overdue for a careful reconsideration...". Nevertheless, the most reliable way to evaluate topic models which serve as the gold standard for coherence evaluation is by using human judgment. Unfortunately, that will take a huge amount of time and other resources.

3.4 Evaluating topic quality using model clustering

We have shown that topic assessments based on coherence metrics do not always align well with human judgment. Although there has been progressing in evaluating the interpretability of topics, the existing intrinsic evaluation metrics do not address some of the other aspects of concern in topic modeling such as the number of topics to select [11].

Nevertheless, in the case of a topic model and evaluation, we already have clusters of topics and do not need a clustering algorithm; we need only to evaluate the clusters obtained. Once clustering is done, how well the clustering has performed can be quantified by several metrics. Ideal clustering is characterized by minimal intra-cluster distance and maximal inter-cluster distance.

There are majorly two types of measures to assess the clustering performance.

- Extrinsic measures - requires ground truth labels. Examples are Adjusted Rand index, Fowlkes-Mallows scores, Mutual information-based scores, Homogeneity, Completeness and V-measure.
- Intrinsic measures - does not require ground truth labels. Some of the clustering performance measures are Silhouette Coefficient, Calinski-Harabasz Index, Davies-Bouldin Index etc.

In our case, we are not always dealing with ground truth labels and we take into consideration inter and intra-score, that is why we will elaborate on the most related clustering measure to us, the measurement - ‘Silhouette Coefficient’.

3.4.1 Silhouette Coefficient

The silhouette value is a measure of how similar an object is to its cluster (intra) compared to other clusters (inter). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its cluster and poorly matched to neighboring clusters.

If most objects have a high-value Silhouette, then the clustering configuration is appropriate. If many points have a low or negative Silhouette value, then the clustering configuration may have too many or too few clusters [12].

The Silhouette score can be calculated with any distance metric, such as the Euclidean distance or the Manhattan distance.

The Silhouette value is defined for each sample and is composed of two scores:

- a: The mean distance between a sample and all other points in the same class.
- b: The mean distance between a sample and all other points in the next nearest cluster.

The Silhouette value for a single sample is then given as:

$$s = \frac{b - a}{max(a, b)}$$

Figure 5 - Silhouette Coefficient Formula

The Silhouette Coefficient for a set of samples is given as the mean of the Silhouette value for each sample.

Chapter 4

Research Methodology

4.1 Dataset

The dataset used in this research is provided by our facilitators Rami Puzis and Aviad Elyashar. The data is a collection of 66K Twitter tweets (documents) divided into 505 labels from year 2012, that we call - "events2012".

label	id	text	created_at
0	256292946331181000	Nobel prize in literature tc	Thu Oct 11 07:19:34 +0000 2012
0	256334302034399000	Congrats, Ateneo! Last na	Thu Oct 11 10:03:54 +0000 2012
0	256335853738160000	"@SMARTPromos: SMART	Thu Oct 11 10:10:04 +0000 2012
0	256346272506712000	CCTV invite hints at Nobel	Thu Oct 11 10:51:28 +0000 2012
0	256346650132508000	mjzone58: SIR HINDI BYAI	Thu Oct 11 10:52:58 +0000 2012

Figure 6 - Example table of 5 documents from 'events2012' dataset

As we can see in the above example table, each document in the dataset is built from:

- label - Represents the group number to which the document belongs.
- Id - A unique ID of the tweet.
- Text - The content of the document.
- Created at - The publication time of the tweet.

The average text length of each document is ~83 words, hence the time of analyzing each document with the zero-shot-classification pipeline (will expand later) is relatively short in comparison to large documents.

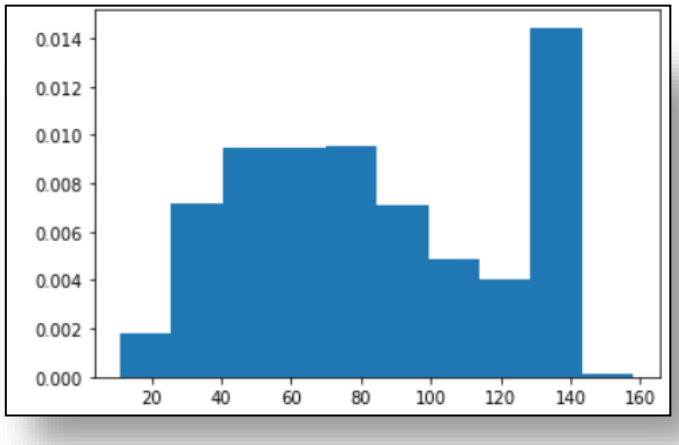

Figure 7 - Distribution of documents length

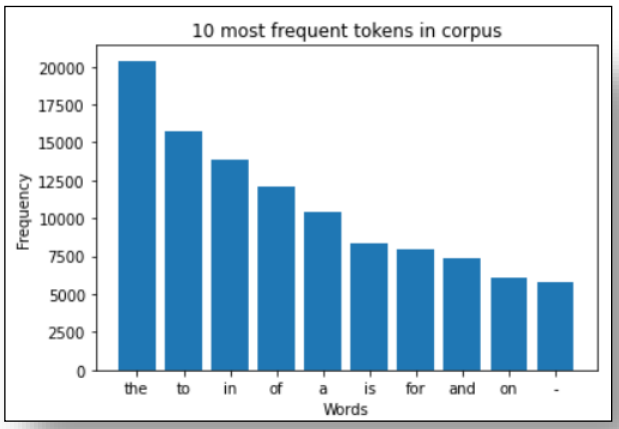

Figure 8 - 10 most frequent tokens (words) in corpus

We can easily notice that the top 10 frequent words are ‘Stop-words’ preprocessing is mandatory.

4.2 Data Preprocessing

Preprocessing the data is an important stage and can significantly influence the final results. This can be related to the "garbage in, garbage out" principle in computer science, where overloaded text brings nonsense output. Our goal is to manipulate the input data before we use it, to ensure and enhance our performance. The preprocessing of our data is defined by 3 steps: Normalization, Stop Words removal and Tokenization.

4.2.1 Normalization

In the text mining world, we define normalization as follows: *"Text normalization is the process of transforming text into a single canonical form that it might not have had before*" [13]. Normalization in normal documents is simply done by removing the punctuation marks and the lowercase of each word.

4.2.2 Stop Words

Define as follows: *"Stop words are any word in a stop list (or stop-list or negative dictionary) that are filtered out (i.e. stopped) before or after processing of natural language data (text)."* [14].

We assume that each word is not equally important which is why we remove the irrelevant words, called stop words. Words such as 'the', 'a' and 'an' are not relevant to our goals. generic English words can be removed to only keep the most distinguishable words left.

We used the Natural Language Toolkit (NLTK) lib which provides standard tokenization and remove stop words from the English vocabulary. This step leaves the relevant words in each tweet. In addition, we learned our data, and find many occurrences of hashtags as well as links such as 'https', so we removed selected words and marks.

4.2.3 Tokenization

Parsing of the documents. After the stop words have been removed, we are left with documents containing highly specific words. The tokenization process takes care of the remaining text and cut the document text into pieces called tokens while removing punctuations.

4.3 Cohesion Pipeline

Throughout our research, our pipeline included different components. These components have been replaced (in components with similar purposes), refined, and fine-tuned to improve the overall pipeline accuracy. In the following diagram you can see the final pipeline:

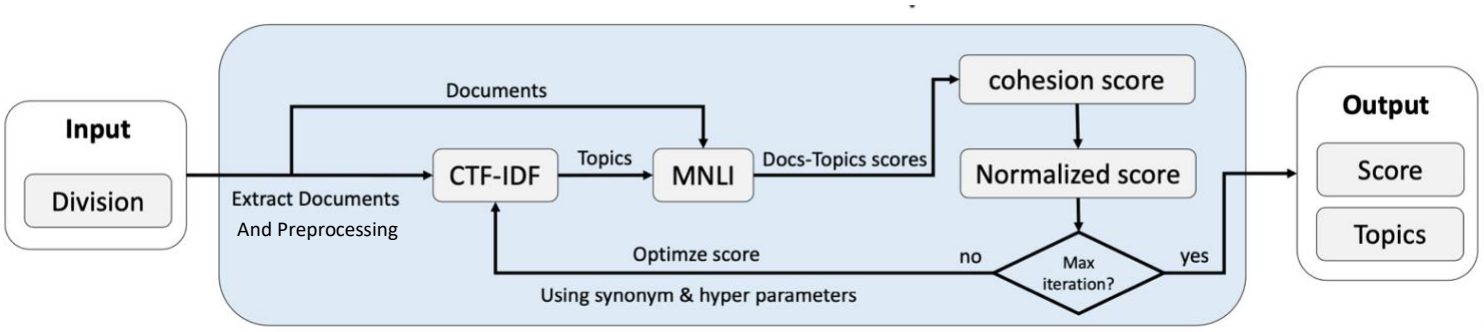

Figure 9 - Cohesion Pipeline

The pipeline gets as an input a division - documents and their labels, and generate a cohesion score and topics (list of words that represent every group). A high cohesion score indicates a good division.

Using C-TF-IDF we extract the topics (potential names) for each group. In the next step, we take those topics and documents and use the Zero-Shot text classification [15].

In zero-shot text classification, the model can classify any text between given topics without any prior data, the model allows us to get the entailment between docs and topics. Using zero-shot-classifaction based on MNLI pre-trained model we rely on a large, trained model from transformers called "facebook/bart-large-mnli" which can be found on HugginFace hub, after empiric tests was found as the most efficient to our case (accuracy, time, GPU). This classifier gets the topic and the document and gives a score from 0-1 that reflects the relation between the document to the topic.

Using the classifier score, we split the scores to positive and negative scores; positive score is a score of a document with his group (topic), while negative score is a score of a document with other groups. Because the size of the group is usually unequal, we normalize the average positive and negative score by the size of the groups ratio and get a normalized score for the cohesion pipeline.

If we reach the max iteration number, we extract as an output the best score and topics that we save in the iterations, otherwise, we return to phase one and change the topics names.

We tried different approaches to impact the topic names such as using GloVe embeddings, synonyms, different hyper-params and many more.

4.4 Cohesion Proof of Concept (POC)

To validate the pipeline's correctness, we decided to do a proof-of-concept experiment. We expect the cohesion score will be better than the most common topic-detection measurement - coherence, and that the recommended topics would be 'Human' as much as they can (words with meaning).

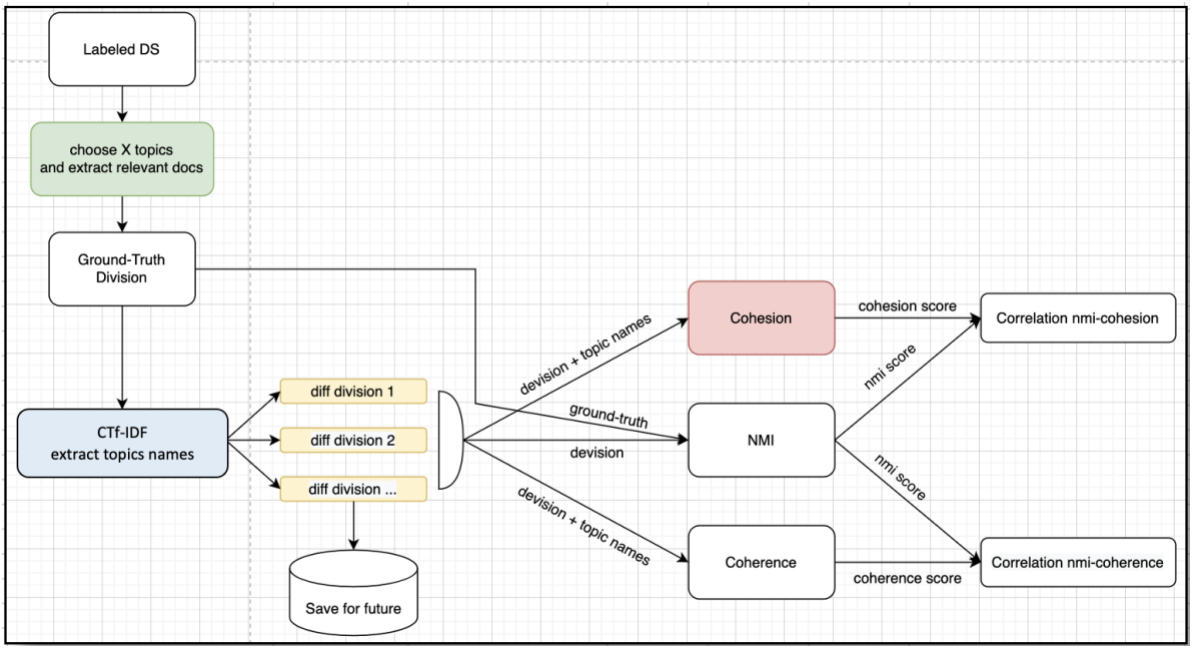

Figure 10 - Cohesion POC Pipeline

To prove cohesion is better than coherence, we show that the correlation between the cohesion score and the NMI (normalized mutual information) score is better than coherence with NMI.

Normalized Mutual Information (NMI) is a normalization of the Mutual Information (MI) score to scale the results between 0 (no mutual information) and 1 (perfect correlation).

We chose this metric since it is symmetric: switching the truth labels with the predictive labels will return the same score value.

This attribute can be useful to measure the agreement of two independent label assignment strategies on the same dataset when the real ground truth is not known.

In the experiment, our dataset includes 66k+ documents and 505 different topics that are labeled manually by humans. From this dataset we choose several topics (and the docs that belong to those topics) and set it as a "Ground Truth" division.

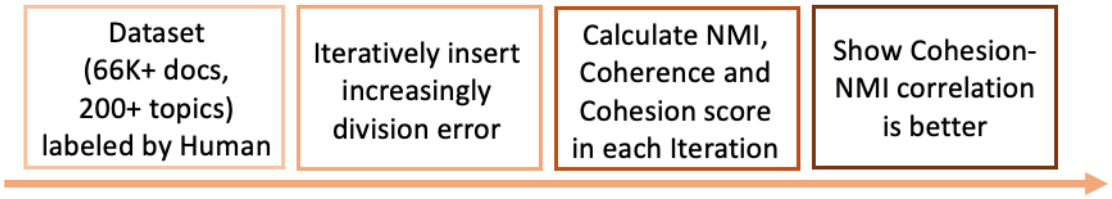

Figure 11 - Cohesion engine POC - Pipeline

We insert increasing error, from 0%-100% in 10% steps, and create several divisions, for each division we extract the topic names using C-TF-IFD (using different configurations) and calculate the cohesion, coherence, and NMI scores.

Finally, we calculate the correlation between cohesion-NMI and coherence-NMI and show that cohesion is better.

Cohesion POC Python Notebook - https://colab.research.google.com/drive/1RreFOEd5LQDaNB7kQcH44bk5gkTpSVeH?usp=sharing

Chapter 5

Results

5.1 POC Results

In this chapter, we examine the results achieved through our POC work.

For the POC, we took 60 groups with total of 6,460 documents. After inserting increasing error, from 0%-100% in 10% steps to our ground truth data we get the following results for the NMI, Coherence score, and Cohesion score:

Results 0

error %	nmi	coherence	cohesion
0%	1	0.951578	0.877106
10%	0.809273	0.921152	0.697031
20%	0.646474	0.940318	0.512486
30%	0.510531	0.908672	0.416557
40%	0.395796	0.842419	0.304715
50%	0.283144	0.693723	0.304678
60%	0.194877	0.628905	0.191405
70%	0.124539	0.518113	0.163746
80%	0.0740079	0.388406	0.171125
90%	0.0353497	0.319494	0.182303
100%	0.0526794	0.377665	0.131693

Figure 12 - NMI, Coherence, and Cohesion values for each error rate (Table)

The results in the figure are interesting and optimistic, we can see that there is a good correlation between the cohesion-score and NMI, and as excepted from the currently state-of-the-art measurement, Coherence, to be correlated to NMI as the error rate increases.

A clearer way to view that correlation, which indicates to us that we are in the right direction, is the figure below:

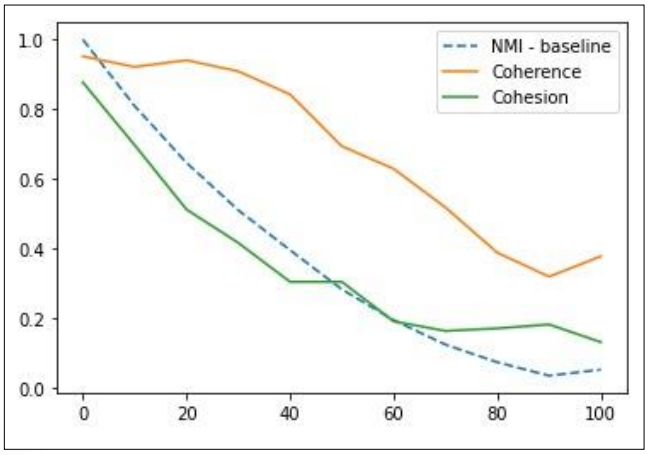

Figure 13 - NMI, Coherence and Cohesion values for each error rate (Graph)
(X-axis is the error rate, and Y-axis is the score)

To decide which measurement is more correlated, we used the Pearson correlation coefficient to measure the degree of similarity between NMI to the desired metric.

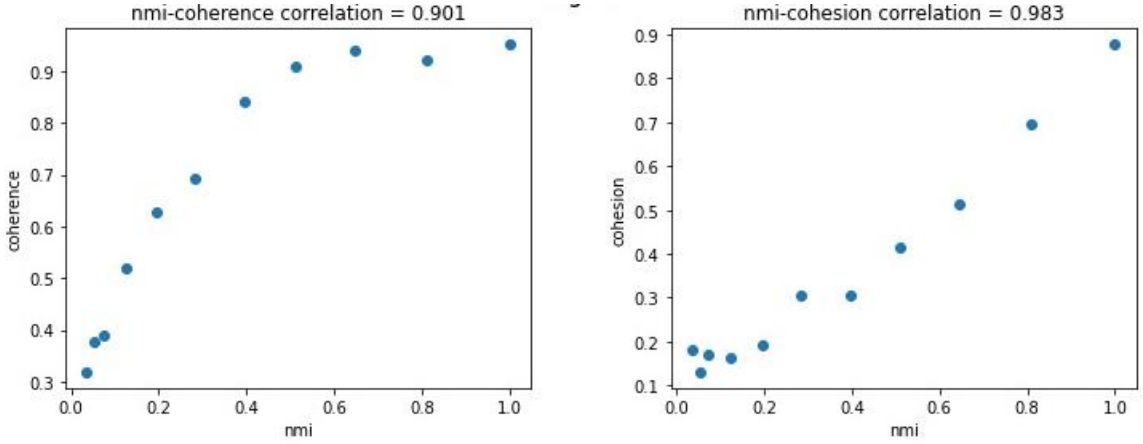

Figure 14 - Pearson correlation, Left graph: NMI-Coherence, Right graph: NMI-Cohesion

As we said before both measurements are correlated, but we can clearly see that in our experiments, our Cohesion measurement bypassed the Coherence measurement. In addition, we can see that the Cohesion result scale fits much better than the NMI scale, for example:

We will focus on the last iteration - 100% randomness which means each doc “moved” to another cluster. The Coherence score was 0.45 while Cohesion score is 0.15, much better.

Chapter 6

Conclusions

6.1 Conclusion

Topic detection is one of the interesting and complex fields. On the one hand, there is an aspiration for fast and generic algorithms that can handle different types of data from different sources. On the other hand, these algorithms must be accurate enough to give 'human' topic names.

The second requirement is difficult, because usually, these models give names that even if they suitable for the documents' groups, one would not give these names. In the current models, topic names are formal and specific, but sometimes can include signs, numbers, or insignificant words.

In our study, we showed some of the leading metrics in the fields of topic detection and clustering, such as the Coherence model and the Silhouette Coefficient, while presenting their benefits and the things that they are missing. We tried to preserve each benefits along with covering up their disadvantages by creating a quality measurement model called Cohesion Pipeline.

One of the main challenges in creating such a pipeline is maintaining the integrity of the model and not overfitting the data. Therefore, we made sure (from the very first stage) to include randomness in the documents selection and the error we put in the POC so that no run will be repeated twice. During the study, many difficulties arose: how to choose the best words for each topic in the first place? which NLP method should we use? how to optimize the topic words (Synonyms/Glove)? which method should we use to normalize the cohesion score? what is the meaning of the cohesion score? And more questions and decisions we had to make at each point.

We showed that the Cohesion pipeline is a quality model that gives a quality rating for a given division of documents, while giving names (topics) to each group. We did this by presenting a higher correlation between the cohesion scores and the results of the NMI measurement over Coherence scores by changing the affiliation (iteratively while raising the error rate) of the same documents to different groups.

Finally, we have organized and exported this pipeline into an open-source package in Python that can be consumed and used by anyone who is interested to get a cohesion score and topics to a given division.

6.2 Future work

This project leaves some open areas for optimization and further improvements. A few points come to mind when recommending future works here:

1. Since the cohesion pipeline uses pre-trained MNLI models as a ready-made zero-shot sequence classifier, we can get good results on short sentences like tweets. but zero-shot approach has its limitations:
   - Impractical inference time for more than 20 topics.
   - Probably will not beat supervised learning
   - Low performance when GPU is not available

   Above that, the MNLI corpus that we used works better on the sentence level, a good future improvement will be to adapt the pipeline to work also on the document level.

2. We find the represented topics with c-TF-IDF which can be followed by wordnet and/or Glove, one possible improvement is to analyze the use of named entity recognition (NER) to extract more "human" topics.

3. Our text pre-processing is adjusted to Twitter tweets, in case of providing different data our preprocessing task will not be efficient and can result sometimes in bad topics, in addition, we currently don't take into account irony, sarcasm, and text ambiguity which we might come across on Twitter data.

4. Our python package is not accessible as we want, and much of the functionality is still hidden from the user due to development time.

5. Since we are dealing with Bigdata and sometimes the input size to the Cohesion Pipeline will be big enough to slow down individuals' computers we need to adjust our pipeline to analyze the data in batches.

6. The measurement score depends on Hyper-Params and variables such as num of words in each topic, Cohesion Formula, pre-trained corpus, and many more. The more we will research the more we will be able to optimize these hyper-params and improve our results.

Chapter 7

Appendices

7.1 Tools and Technologies

Throughout our research we have used various tools for the development and manage our work:

First, to prove the initial feasibility of the Cohesion measurement, we took 800 documents from 20newsgroup, tagged, and score them manually. We developed a simple and intuitive tool, for the convenience of document scoring. The backend side (business logic) was written in **Python**, the information was stored in **MongoDB** as our DB, and the GUI was developed using **Tkinter** (python UI). After the labeling and scoring, we checked the results manually and were ready to move to the next step.

Then, after proving small general concept, we needed to scale it up. We used 'events2012' dataset (describe in chapter 4) and do a full POC using **Python** and manage a Python Notebook using **Google Collab.** During all our research, we used **Gitlab** (Git) as our version control, **Jira** and **GitLab** to manage our tickets and **Aha!** as our scheduler and Gantt manager.

Figure 15 - Tools and Technologies

7.2 Public Package

We developed a python package for the Cohesion Pipeline that can be installed from the Python Package Index (PyPI) as follows:

"pip install cohesion-pipeline"

A more detailed explanation of the package and a usage example can be found on the package homepage - https://github.com/TomNachman/CohesionPipeLine.

Chapter 8

References

1. **A review of topic modeling methods**
   VAYANSKY, Ike; KUMAR, Sathish AP. A review of topic modeling methods. *Information Systems*, 2020, 94: 101582.

2. **An introduction to latent semantic analysis**
   Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. Discourse processes, 25(2-3), 259-284

3. **BERTopic**
   Maarten, G. (2020). APA citation [Online]. BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics. Available at: https://maartengr.github.io/BERTopic/#citation (Accessed: 04 April 2021).

4. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction
   McInnes, Leland, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction." arXiv preprint arXiv:1802.03426 (2018).

5. **HDBSCAN: Hierarchical density-based clustering**
   McInnes, Leland, John Healy, and Steve Astels. "hdbscan: Hierarchical density based clustering." J. Open Source Softw. 2.11 (2017): 205

6. **Learning maximal marginal relevance model**
   Xia, L., Xu, J., Lan, Y., Guo, J., & Cheng, X. (2015, August). Learning maximal marginal relevance model via directly optimizing diversity evaluation measures. In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval (pp. 113-122)

7. **Reading tea leaves**
   Chang, Jonathan, et al. "Reading tea leaves: How humans interpret topic models." *Advances in neural information processing systems*. 2009

8. **Re-evaluating Semantic Interpretability Measures**
   Doogan, Caitlin, and Wray Buntine. "Topic Model or Topic Twaddle? Re-evaluating Semantic Interpretability Measures." *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2021.

9. **Exploring the space of topic coherence measures**
Röder, Michael, Andreas Both, and Alexander Hinneburg. "Exploring the space of topic coherence measures." *Proceedings of the eighth ACM international conference on Web search and data mining*. 2015.

10. **Is Automated Topic Model Evaluation Broken? The Incoherence of Coherence**
Hoyle, Alexander, et al. "Is Automated Topic Model Evaluation Broken? The Incoherence of Coherence." *Advances in Neural Information Processing Systems* 34 (2021).

11. **Evaluating topic quality using model clustering 2014**
V. Mehta, R. S. Caceres and K. M. Carter, "Evaluating topic quality using model clustering," 2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), 2014, pp. 178-185, doi: 10.1109/CIDM.2014.7008665.

12. **Silhouettes: a graphical aid to the interpretation and validation of cluster analysis**
Rousseeuw, Peter J. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis." Journal of computational and applied mathematics 20 (1987): 53-65.

13. **Text normalization - Wikipedia**
Wikipedia contributors. "Text normalization." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 31 Jan. 2022. Web. 22 Jun. 2022.

14. **Mining of massive datasets**
Rajaraman, Anand, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

15. **Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach**
Yin, Wenpeng, Jamaal Hay, and Dan Roth. "Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach." arXiv preprint arXiv:1909.00161 (2019).