Arian Rodrigo Pasquali

Automatic Coherence Evaluation Applied to Topic Models

U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
Julho de 2016

Arian Rodrigo Pasquali

Automatic Coherence Evaluation Applied to Topic Models

Dissertação submetida à Faculdade de Ciências da Universidade do Porto como parte dos requisitos para a obtenção do grau de Mestre em Ciência de Computadores

Orientador: Alípio M. Jorge

Departamento de Ciência de Computadores
Faculdade de Ciências da Universidade do Porto
Julho de 2016

To my parents,
Ariomar and Angela,
for their love and support

Abstract

Topic models are widely used to analyze large text collections. They represent a suite of algorithms whose purpose is to discover the thematic structure in a collection of documents. Originated in the field of natural language processing, they have been applied in a wide range of domains such as linguistics, bioinformatics, online advertising, political science, bibliometrics, and psychology. They were designed to infer topics from documents and provide a new way to summarize and explore archives of documents.

Once learned, these topics should correlate well with human concepts. For instance, one model analyzing articles from a newspaper might learn topics that cover ideas such as sports, movies, politics, and fashion. However, in practice, evaluating the meaning of these topics sometimes is not trivial. Eventually, learned topics are not easy and clearly interpretable by humans due to its ambiguity or just lack of coherence. Decide if a particular topic is meaningful or not is subject to subjective judgment since two experts can easily disagree about its interpretability. Moreover, in many circumstances, it is unfeasible to rely on manual evaluation because it does not scale. Therefore, the aim of this thesis is to explore automatic coherence evaluation methods applied to topic models.

We can find in the literature proposals to address this problem. To reach our goal we have implemented two approaches and compared them against human evaluation. Our aim is to reproduce results found in the literature and assess the compliance of these methods with human annotators. We expect to provide hints for the development of systems capable of dealing with eventual incoherent topics originated from unsupervised models. As a consequence, we could speed the development of more sophisticated text mining pipelines and lead to better and innovative ways to interact and explore a large amount of data.

Keywords. Data Mining, Text Mining, Natural Language Processing, Topic Models, Evaluation of Topic Models, Coherence Evaluation, Unsupervised Learning

Resumo

Modelos de tópicos são amplamente utilizados para analisar grandes coleções de textos. Eles representam um conjunto de algoritmos, cujo objectivo é descobrir a estrutura temática em uma coleção de documentos. Originado na área de processamento de linguagem natural, esses métodos têm sido aplicados em uma vasta gama de domínios, tais como linguística, bioinformática, publicidade online, ciência política, bibliometria, e psicologia. Eles foram concebidos para inferir os temas de documentos e fornecer uma nova maneira de resumir e explorar grandes arquivos de dados.

Como exemplo, um modelo analisando artigos de um jornal pode aprender tópicos que cobrem ideias como esportes, filmes, política e moda. No entanto, na prática, avaliar o significado destes temas por vezes não é trivial. Eventualmente tópicos não são fáceis e claros o bastante para nós devido à sua ambiguidade ou apenas falta de coerência. Decidir se um determinado tópico é coerente ou não está sujeito ao julgamento subjetivo, duas pessoas podem facilmente discordar sobre a sua interpretação. Além disso, em muitas circunstâncias, é inviável contar com avaliação de pessoas devido aos custos e tempo. Portanto, o objetivo desta tese é explorar métodos de avaliação da coerência automáticas aplicadas a modelos de tópicos.

Podemos encontrar propostas na literatura para abordar esta questão. Para alcançar nosso objetivo implementamos duas dessas abordagens. O nosso objectivo é reproduzir os resultados encontrados na literatura e avaliar a performance desses métodos em comparação com avaliadores humanos. Esperamos assim, fornecer dicas para o desenvolvimento de sistemas capazes de lidar com eventuais temas incoerentes originados a partir de modelos como esse. Como consequência, levar a melhores e inovadoras maneiras de interagir e explorar grande quantidade de dados.

Keywords. Mineração de Dados, Mineração de Texto, Processamento de Linguiagem Natural, Modelos de Tópicos, Avaliação de Modelos de Tópicos, Avaliação de Coerência, Aprendizagem Não Supervisionada

Acknowledgements

First, I would like to thank my supervisor, Prof. Alipio Jorge for providing guidance and invaluable insights and constant feedback throughout the development of this work.

A special word of gratitude goes to everyone that one way or another encouraged me in moments of fatigue with their unconditional support. Friends like Silvio Moreira, Luis Rei, Sergio Vasconcelos, Sergio Morgado and Manel Cruz provided critical support during these years. To Marcela Canavarro, who provided tremendous contributions (e.g. sharing the Facebook posts dataset used in this work). Also, of course, for her friendship. Many thanks to Ricardo Campos for helping me with statistics and insights as well. I don't forget to thank the voluntary collaboration by Ciro Oiticica and Lucas and Marcela Canavarro as annotators of this study.

Finally, I would like to thank my family. This work would not have taken place, and this thesis would not exist without them.

This work was supported by the Integrated Project TEC4Growth (FourEyes) - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE- 01-0145-FEDER-000020" is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

“You shall know a word by the company it keeps”

John Rupert Firth (1957)

Contents

List of Tables

List of Figures

Listings

Chapter 1

Introduction

With an ever-increasing rate, every day a massive amount of information pours into our computer networks. The computerization of our society and the fast development of data collection and storage tools made information overload one of the biggest problems of our age [1]. It has become harder to find what we are looking for, understand it and get something out of it [2].

The fast-growing amount of data available generated a need for new techniques and automated tools that could intelligently assist us in transforming data into useful information and knowledge. This environment led to the birth of *Data Mining* [1]. Jiawei Han et al. (2012) defined data mining as "the process of discovering interesting patterns and knowledge from large amounts of data".

The overall goal of data mining is to extract information from a data set and transform it into an understandable structure for further use. In this field, topic models represent a popular approach and it is also closely related to the field of machine learning. Originally designed to learn thematic topics from documents, topic models originated from the domain of natural language processing. Since then, these models have been applied in a wide range of fields, such as linguistics [3], bioinformatics [4], online advertising [5], political science [6], and psychology [7]. Topic models can provide a summary of the document collection that would be difficult, or at least very costly to obtain by hand and may yield connections between and within documents that are not obvious.

Once learned, topics should correlate well with human concepts. For example, one model might learn topics that cover ideas such as sports, movies, politics, and fashion. However, although we expect topics to be meaningful, eventually, automatically learned topics are not easy and clearly interpretable by humans due to its ambiguity or just lack of coherence. It depends sometimes on both background knowledge and familiarity with the data. Moreover, two persons can easily disagree about how to interpret a topic. To illustrate this problem, let's take as example the following topics automatically discovered within a collection of news articles:

1. space, launch, NASA, Earth, shuttle, solar, satellite, water, mission, lunar;

2. used, time, number, different, better, probably, point, using, set, large.

It seems reasonable to predict that most people would agree that topic **1** is easy to interpret. Based on the coherence of the words one can infer its meaning is related to space exploration. The subject of the second topic, however, is less clear and may confuse users because of its ambiguity and lack of coherence. It is necessary an objective method to evaluate these topics.

Applications that make use of machine learning techniques, topic models, for instance, are susceptible to errors and failures, an aspect of their success will be about how users perceive and tolerate their failures. As this kind of application becomes increasingly embedded in daily lives and used for more critical tasks, system mistakes may lead to a backlash from users and negatively affect their trust. Being able to predict if users can understand learned topics is an important issue for the adoption of topic models in a variety of applications. Frustration in carrying out functions promised by a system diminishes people's trust and reduces their willingness to use the system in the future [8].

Until recent years, standard evaluation methods for topic models didn't take into consideration topic interpretability. Chang et al. (2009) demonstrated that standard evaluation methods do not consider the semantic coherence of topics learned by a topic model, making it difficult to evaluate how well a topic model would perform in some end-user task [9]. Automatically evaluate topic coherence helps to quickly identify 'junk' topics that are hard to interpret and therefore, potentially meaningless and useless to end users. This can lead to better ways to interact and explore the data [10] [11].

In this work, the formal definition of topic coherence is a measure that scores a single topic by measuring the degree of semantic similarity between a set of words. These measures help in distinguishing topics that are semantically interpretable from topics that are a result of statistical inference. For this reason, during the last years, researchers have been working to propose solutions to address this issue. In this work, we focus on some of these proposals.

1.1 Objectives

The primary objective of this thesis is to assess the effectiveness of coherence evaluation measures applied to topic models. To achieve this, we will explore the literature, select two measures to implement and compare the performance of these measures against human annotators. Furthermore, since most of the related work in the literature deals with texts written in English and usually well-structured like scientific publication or journalistic pieces, we designed an experiment using Facebook posts in Portuguese to assess their performance in a different language and relatively badly structured texts. Consequently, the purpose of this thesis is:

- to test if the proposed measures in the literature can also be applied to topics learned from a text content from social networks where texts are often short and not well structured as editorial and/or scientific content.

1.2 Contributions

During the development of this thesis we built these software components:

- Experimental comparison of existing topic quality measures, including two user studies;
- An open source python implementation providing topic coherence evaluation;
- A web application prototype that provides exploratory analysis on topics and coherence scores;
- A web application prototype that combines social network analysis and topic modeling presenting only topics with high coherence scores to users.

1.3 Published paper

As part of this thesis a paper [12] has been published in a peer-reviewed conference[1]:

- (Pasquali et al., 2016) Arian Pasquali, Marcela Canavarro, Ricardo Campos, and Alípio Jorge. Assessing topic discovery evaluation measures on Facebook publications of political activists in Brazil. In Proceedings of the International Conference on Computer Science & Software Engineering (C3S2E 2016), Porto, Portugal, July 22, 2016.

1.4 Thesis structure

After this chapter, this thesis is divided into seven more chapters. Chapter 2 provides a background overview of text mining and topic modeling that is relevant to understand the technical context of this work. Chapter 3 presents some previous work about coherence evaluation, later it focuses on introducing a formal definition for topic coherence, two measures found in the literature and relevant implementation details. Chapter 4 focuses on reproducing literature findings. It presents an experiment using our implementation of the measures described in Chapter 3, we used external human annotators to assess the performance of the metrics. In Chapter 5 we present another experiment focused on applying these measures to explore Facebook posts and assess the compliance with manual human evaluation, this time carried out by volunteers experts in the domain of the dataset. Chapter 6 presents two applications implemented to explore different aspects of the topics and coherence measures. Chapter 7 concludes this thesis with a summary,

[1] http://confsys.encs.concordia.ca/C3S2E/c3s2e16/

the insights gained and the research gaps identified. The appendix contains resources that are too long to be described in the main chapters: a comprehensive list of relevant software and libraries that were used, relevant source code and a list of all topics and coherence scores from the experiment described in Chapter 5.

Chapter 2

Text Mining and Topic Models

In this chapter, we give a brief overview of text mining and topic models. Text mining is a subset of techniques in data mining that is concerned to handle text documents. Topic models represent a traditional text mining technique for discovering of hidden semantic structures in a text body.

2.1 Text Mining

Text mining is a highly cross-disciplinary field that can trace its roots to the theory and practice of data mining. According to Gary Miner et al. (2012), the purpose of text mining is to provide some understanding of how to extract knowledge from text without having a human to read it [13].

Radovanović and Ivanović (2008) stated [14]:

> “The field of text mining seeks to extract useful information from unstructured textual data through the identification and exploration of interesting patterns. The techniques employed usually do not involve deep linguistic analysis or parsing, but rely on simple ‘bag-of-words’ text representations based on vector space.”

This field of research studies a range of technologies for analyzing and processing semistructured and unstructured text data in order to make it accessible to statistical and machine learning algorithms. Gary Miner et al. (2012) classified some practices in the context of text mining, and here we list some of them [13]:

- **Document classification**: grouping and categorizing paragraphs or documents using data mining classification methods, based on models trained on labeled examples;
- **Information retrieval** : storage and retrieval of text documents;
- **Document clustering**: grouping documents, terms or paragraphs using data mining clustering methods;

- **Web mining** : data and text mining on the internet, with a specific focus on the scale and interconnectedness of the web;
- **Information extraction** : identification and extraction of relevant facts and relationships from unstructured text;
- **Natural language processing**: low-level language processing and understanding tasks.
- **Concept extraction**: grouping of words and phrases into semantically similar groups.
- **Sentiment analysis** : this field deals with the analysis of opinions found in documents. One of the basic tasks is sentiment classification, where texts are organized into classes which correspond to, e.g. positivity or negativity of expressed opinion [14].

We can say that the principle behind many tasks related to text mining is the need to transform text into numbers or logic representations. Converting text into these representations requires knowing how to combine techniques for handling texts, covering from words to documents to entire document databases. This data preparation phase is responsible for converting unstructured and semi-structured text into some structured representation, such as vector space model [15].

2.2 Data Preparation

The data preparation phase represents a critical role in text mining practices and applications. It is the first step in the text mining process. A variety of methods is discussed by Vijayarani et al. (2015) [16]. Here we discuss some the relevant ones in the context of this work. For some of the following tasks, we make extense use of the Python programming language and the libraries NLTK and Scikit-Learn (see Appendix A for further detail).

2.2.1 Tokenization

Frequently, an initial step is to split the input text into units called tokens where each is a word or something else like a number or a punctuation mark. This process is referred to as tokenization [17]. The right approach is highly dependent on the language. For instance, the main idea used in English is the occurrence of whitespace or the beginning of a new line between words, but even this is not necessarily reliable in every scenario.

Several languages do not put spaces in between words, and so the basic word division algorithm of breaking on whitespace will result wrong. Such languages include major East-Asian languages, such as Chinese and Japanese. Ancient Greek was also written without word spaces. Spaces were introduced recently in history. In such languages, word segmentation is a more challenging task. In German sometimes compound nouns are written as a single word, for example, *Lebensversicherungsgesellschaftsangesrellter* means life insurance company employee. In many ways, this makes linguistic sense, as compounds are a single word. However, for processing purposes, one may wish to divide it, or at least to be aware of the internal structure of the word. As Weikum (2002) mentioned in [17], ‘While not the rule, joining of compounds sometimes also happens in English, especially when they are common and have a specialized meaning. We noted above that one finds both *’data base’* and *’database”*. As another example, we can easily find *’hard disk’* instead of *‘hardisk’* in the computer press. An alternative to address the issue of compounds in English or Latin languages is to use the concept of *n-grams* or *collocations*. A n-gram groups one or more words in a single token. Tokens with one word are called uni-gram (e.g. ‘disk’), two words bi-grams (e.g ‘hard disk’), three words tri-grams (e.g. ‘life insurance company’) and so on. Choosing the maximum size of n-grams depends on the dataset and the use case. For the sake of simplicity, this work will only use uni-grams.

2.2.2 Case normalization

Most texts contain words in both upper and lowercase letters. Capitalization helps readers differentiate, for example, between nouns and proper nouns and can be useful for automated algorithms as well. In many circumstances, however, an uppercase word at the beginning of the sentence should be treated no differently than the same word in lowercase appearing elsewhere in a document. Simple text normalization converts the entire text to either to lowercase or uppercase. Let’s assume as an example the dataset described in Table 2.1.

Table 2.1: Document examples

Document	Text
1	There is no cure for curiosity
2	Curiosity killed the cat
3	My dog ate my lunch

After tokenization and case normalization we have the result presented in Table 2.2.

Table 2.2: Documents after tokenization and normalization

Document	Vector
1	[’there’,’is’,’no’,’cure’,’for’,’curiosity’]
2	[’curiosity’,’killed’,’the’,’cat’]
3	[’my’,’dog’,’ate’,’my’,’lunch’]

2.2.3 Removing common words

For many text mining tasks, it is useful to remove words such as *the* that appear in nearly every document to save storage space and speed up processing. These common words are called *stop words*. The removal of stop words is possible without loss of information because, for the most common text mining tasks and algorithms, these words have little impact on the final results. However, this technique needs caution. Too many stop words may degrade the interpretability of results and even change the meaning. A few examples of stop words for the English language are the following:

- about
- but
- by
- for
- no
- yes
- get
- my
- him
- himself
- his
- how
- the
- is

2.2.4 Part-of-speech tagging

In computational linguistics, Part-Of-Speech tagging, also called POS tagging, is the process of identifying a word in a text as corresponding to a particular grammar category, based on both its definition and its relationship with nearby and related words in a sentence. After this preprocessing step, we have the result presented in Table 2.3. Table 2.4 lists the meaning of each tag [18] found in this example. Applying this technique can demand time and computational power, but may provide good results depending on the context. POS tagging can be very useful when building the feature set for a topic model. Hinneburg et al. (2014) demonstrated in their work [19] how POS tagging analysis of terms can be used to produce more interpretable topic representations.

Table 2.3: Part-of-speech tagging examples

Document	Vector
1	('there', 'EX'), ('is', 'VBZ'), ('no', 'DT'), ('cure', 'NN'), ('for', 'IN'), ('curiosity', 'NN')
2	('curiosity', 'NN'), ('killed', 'VBD'), ('the', 'DT'), ('cat', 'NN')
3	('my', 'PRP′),(′*dog*′,′*NN*′),(′*ate*′,′*VB*′),(′*my*′,′*PRP*'), ('lunch', 'NN')

Table 2.4: Part-of-speech tags description

POS tag	Description
PRP	pronoun, possessive
EX	existential there
VBZ	verb, present tense, 3rd person singular
DT	determiner
NN	noun, common, singular or mass
IN	preposition or conjunction, subordinating
VBD	verb, past tense

2.2.5 Vector space representation

After text preprocessing has been performed, the individual word tokens commonly are transformed into a vector representation suitable for input into text mining algorithms. This vector representation can take one of three different forms: a binary representation, an integer count, or a float-valued weighted vector. Following is a simple example that highlights the difference between the three approaches. The vector space for these documents contains 15 tokens, 13 of which are distinct. The terms are sorted alphabetically by their corresponded frequency:

Table 2.5: Terms frequency

	ate	for	no	is	there	dog	cat	lunch	cure	curiosity	the	my	killed
	1	1	1	1	1	1	1	1	1	2	1	2	1

The binary and integer count vectors are straightforward to compute from a list of tokens. A binary vector stores a ‘1’ for each term that appears in a document, whereas an integer count vector stores the frenquency of that word in the document. See Table 2.6 and Table 2.7 respectively.

Table 2.6: Boolean vectors representation

	ate	for	no	is	there	dog	cat	lunch	cure	curiosity	the	my	killed
1	0	1	1	1	1	0	0	0	1	1	0	0	0
2	0	0	0	0	0	0	1	0	0	1	1	0	1
3	1	0	0	0	0	1	0	1	0	0	0	1	0

Notice that on document 3 the word ‘my’ appears twice, the binary vector still only contains a ‘1’, see Table 2.6. The integer count vectors for the three documents would look as follows in Table 2.7:

Table 2.7: Integer count vectors representation

	ate	for	no	is	there	dog	cat	lunch	cure	curiosity	the	my	killed
1	0	1	1	1	1	0	0	0	1	1	0	0	0
2	0	0	0	0	0	0	1	0	0	1	1	0	1
3	1	0	0	0	0	1	0	1	0	0	0	2	0

Now the term ‘my’ appearing twice in the document ‘3’ is reflected. The vector-space model makes an implicit assumption called bag-of-words. This assumption implies that the order of the words in the document does not matter. This may seem like a big assumption since text must be read in a specific order to be understood. For many text mining tasks, such as document classification or clustering, however, this assumption is usually not a problem. The collection of words appearing in the document is usually sufficient to differentiate between semantic concepts.

The last kind of text representation is the float-valued weighted vector. There are many variants to determine the terms weight. The most popular weighting approach is known as

$TF \cdot IDF$. It stands for **T**erm **F**requency - **I**nverse Document **F**requency. Term frequency is the number of times a term appears in a document. Document frequency is the number of documents that contains the term. The formal definition is described as following:

$$\mathrm{IDF}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}. \tag{2.1}$$

Considering that:

- where, N represents the number of documents in the collection;
- D represents all the documents;
- $|\{d \in D : t \in d\}|$ is the number of documents where the term t appears;
- t is a specific term (ngram, token, word) that occurs in d;
- d represents a specific document that belongs in D.

The product of term frequency and inverse document frequency is the term weighting $TF \cdot IDF$ [20]. It is defined as follows:

$$\mathrm{TFIDF}(t,d,D) = tf(t,d)idf(t,D). \tag{2.2}$$

The concept behind this approach is that terms with high frequency get high weight unless it also has high document frequency, meaning that the term may hold little meaningful information. For instance, the term ‘the’ in English often occurs many times within a single document and also occurs nearly every document. This will give low weight to the term and can be removed from the analysis using a predetermined threshold. A result applying TFIDF on our hypothetical dataset can be seen in Table 2.8.

Table 2.8: TF-IDF vector representation

	ate	for	no	is	there	dog	cat	lunch	cure	curiosity	the	my	killed
1	0	0.42	0.42	0.42	0.42	0	0	0	0.42	0.32	0	0	0
2	0	0	0	0	0	0	0.53	0	0	0.40	0.53	0	0.53
3	0.38	0	0	0	0	0.38	0	0.38	0	0	0	0.76	0

Karen Spärck Jones first introduced IDF in a 1972 paper [20]. It was designed to score term specificity on information retrieval context. Spärck Jones proposed an inspiration of the Zipf’s law [21] to support its information-theoretic and linguistic background. Zipf’s law illustrates that given some corpus of natural language, the frequency of a word is inversely proportional to its rank in the frequency table. Therefore the most frequent word will occur about twice as often as the second most frequent word, three times as often as the third most frequent word, and so on. From the linguistics perspective, he argues that in order to maximize the meaning of the message, words with high frequency carry less information than others with less frequency. This notion is preserved on $TF \cdot IDF$ definition penalizing the high-frequency terms with lower scores for then to be removed using a specified threshold.

2.3 Topic Models

With the unprecedented access of data we have nowadays we face the challenge of how to explore and take advantage of all this data we have at our disposal. According to one of the creators of topic modeling, David Blei, the core idea behind topic models was to provide an algorithmic approach to identify themes automatically in a collection of documents. Usually, unstructured text documents.

Topic modeling is a different vision on how we can explore documents. In this vision, we can search and explore documents based on underlying themes. For instance, let's consider a collection of all Times magazines since it was created in 1923. Typically a common method to explore this collection would be by typing keywords into search engines (e.g. Google, Yahoo) and we use those result links to start navigating through them. By applying topic models, we could access these documents in a different perspective. Topic models algorithms enable us to identify what kind of topics the magazine covered through its articles in a way we could navigate them based on their themes. We could see how a specific theme like technology might have changed over the years, how it is related to other topics and so one. So rather than relying only on keywords and links to find documents, topic models provide an alternative or a complementary technique to explore data [22].

The main point is that sometimes we have no information about a collection of documents and we do not know what the themes are. Topic models algorithms analyze collections, identify the themes automatically, annotate the documents according to those themes so we can explore, search and understand them. Of course, an activity that we do not have the human power to do. Manually annotate all the collections of documents in the world is not practical or feasible.

2.3.1 Applications

Topic models automatically infer the topics discussed in a collection of documents. These topics can be applied to summarize and organize documents, or used for extract features and dimensionality reduction in stages of a machine learning pipeline. At a high level, topic modeling tries to discover structure within an unstructured collection of documents. After discovering this "structure", a topic model can answer questions such as: What is document X discussing? How related are documents X and Y? If I am interested in topic Z, which documents should I read first?

Besides, topic models infer a distribution over topics for each document. For example, document X might be 70% about space exploration, 20% about religion and 10% about other topics. According to Bradley (2015) [23], these topic distributions can be applied in many ways:

- *Clustering*: topics are cluster and documents are related to clusters (topics). This way it can help organize or summarize document collections;
- *Feature generation*: generate features for other machine learning algorithms to use. As mentioned before, these features can then be used in algorithms for classification tasks;
- *Dimensionality reduction*: each document's distribution over topics gives a summary of the document. Comparing them in this reduced feature space can be more meaningful than comparing in the original feature space.

2.3.2 Latent Dirichlet Allocation

There are two fundamental types of topic models. The first, Probabilistic Latent Semantic Analysis (pLSA) [24] from Hofmann et al. (1999) derived from seminal works published by Dumais et al. (1990) [25][26]. The second type, Latent Dirichlet Analysis (LDA) [27] appeared in 2003.

The original LSA [26] was based on linear algebra and was designed with information retrieval goals in mind. Its approach had three basic claims, first that semantic information can be derived from a word-document co-occurrence matrix, second that dimensionality reduction is an essential part of this derivation, third that words and documents can be expressed as points in Euclidean space. PLSA is consistent with the first two of these claims, but differs in the third, instead of using points in space it is based on probabilities [28].

All these models focus on word co-occurrences to discover topics. LDA was originally inspired by the work of Hofmann et al. (1999) on pLSA, but David Blei claimed that pSLA was still hard to extend [22]. LDA made topic modeling easier to use and extendable [22] and according to Blei, this is one of the reasons for its popularity. That claim seems to be reflected by the dozens of extensions published on top of LDA along the last decade, its extensions range from time-series [29][30] to sentiment analysis [31] modeling capabilities.

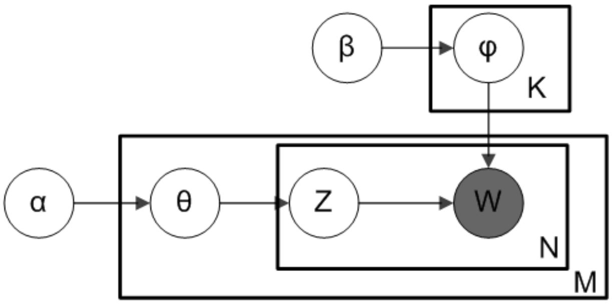


Figure 2.1: Latent Dirichlet Allocations represented in plate notation

LDA, at a very basic level, has few required parameters. One of them is the number of topics, commonly referred as K. The α and β parameters define the nature of topics and words distribution per document and per topic respectively, further explanation is provided below. Finally, the number of iterations the algorithms will perform until it stops.

Figure 2.1 is the graphical representation of LDA using plate notation, where gray circles represent the observable variables, latent (also called hidden) ones are white, and boxes represent collections of variables. Parameters of the model:

- K is the number of topics;
- N is the number of words in the document;
- M is the number of documents to analyse;
- α is the Dirichlet-prior concentration parameter of the per-document topic distribution;
- β is the same parameter of the per-topic word distribution;
- $\varphi(k)$ is the word distribution for topic k;
- $\theta(i)$ is the topic distribution for document i;
- *w(i,j)* is the j-th word in the i-th document;
- *z(i,j)* is the topic assignment for *w(i,j)*;
- φ and θ are Dirichlet distributions, z and w are multinomials.

Where a data set of documents $W = w^{(1)}, w^{(2)}, ..., w^{(M)}$ is observed, while the underlying corresponding topic assignments $Z = z^{(1)}, z^{(2)}, ..., z^{(K)}$ are not observed. Parameters φ and θ are Dirichlet distributions, z and w are multinomials. Which means, a document is a probabilistic distribution of topics and a topic is a probabilistic distribution of words. The parameter α controls per document topic distribution and β controls per topic word distribution. High α values mean that every document is likely to contain a mixture of most of the topics and not just a single topic specifically. While α low value means that a document is more likely to be represented by just a few of the topics. Similarly, high β values mean that each topic is most likely to contain a mixture of most of the words and a low value means that topics are represented by a few number of words. In practice, this models the estimated similarity between topics, same thing for documents. Depending on the use case and the dataset, we can tune these parameters to reflect our assumptions about the dataset.

LDA learns the relationships between words, topics, and documents by assuming that a particular probabilistic model generates documents. It first assumes that there are a fixed set of topics, K used at the corpus, and each topic z is associated with a distribution over the words. In this model, the distributions represent the probability of each topic appearing in each document. The generative process for a document can be described as follows:

1. for each topic: decide what words are likely.

2. for each document,
    (a) decide what proportions of topics should be in the document,
    (b) for each word,
        i. choose a topic,
        ii. given this topic, choose a likely word (generated in step 1).

There are several open source implementations of LDA, each one with pros and cons (e.g. scalability, usability, documentation, community support). Table 2.9 lists some of these projects that are worth mentioning. This thesis makes use of the *lda* library written in Python by Allen Riddell, the implementation is freely available and is licensed under Version 2.0 of the Mozilla Public License [1]. We could have used any of those listed in Table 2.9, we choose the this Riddell's implementation because of its simplicity and because it serves the purpose of this work. See Appendix A.4 for more detail about Riddell's library. Every implementation listed here is open source and can be a good starting point to study implementation details. All of them are easily downloaded from their official website.

Table 2.9: Open source implementations of LDA

Package name	Language	Url
lda	R	https://cran.r-project.org/web/packages/lda/
Apache Spark	Java/Scala	http://spark.apache.org/
Mallet	Java	http://mallet.cs.umass.edu
lda-c	C	https://www.cs.princeton.edu/$\sim$blei/lda-c/index.html
scikit-learn	Python	http://scikit-learn.org/
gensim	Python	https://pypi.python.org/pypi/gensim
Riddell's lda	Python	https://pypi.python.org/pypi/lda

2.3.3 Interpreting topics

We have a lot of knowledge about the world. We understand what a car is, what it is like to have a birthday, how does it feel to be happy. On top of that we are also good at adjusting context on the fly, we know how to choose the right level of abstraction, not too close and not too far so that everything makes sense.

A basic LDA model will find as many topics as we wish. Manually setting the number of topics is tricky because typically there is no right answer. The parameter defining the number of topics can be seen as the level of zoom in or zoom out on the data that we desire. More topics may lead to more granular topics and fewer topics may highlight only a few aspects of the dataset.

[1] https://opensource.org/licenses/MPL-2.0

There are methods that address the problem of finding the optimal number of topics in a given data set. For instance, Hierarchical Dirichlet processes [32] are topic models where the data determine the number of topics. This kind of model is out of the scope of this work.

Topics will be referred many times in this work, and the term 'topic' represents a set of words in this context. Table 2.10 serves as a hypothetical example of five topics learned using a topic model. Each line represents a topic and each topic is represented by a set of words. Just by looking at these sets of words we can, at some degree, infer the thematic of the topics. One, in particular, seems to be related to health and medical research, following religion, gun/state regulation and one about space exploration. The last one is not clear and would demand more analysis. We can interpret the first four topics without reading all the documents that originated these topics. In some use cases, we might want to use the learned topics to guide users through the collection of documents. This would require that all topics should be clear for users to explore them successfully.

Table 2.10: Example of topics learned using topic model

0	health medical disease number study drug cancer patients research
1	god jesus bible church christian christ christians man faith
2	gun state law states national public control american united
3	space earth nasa launch shuttle mission orbit moon satellite
4	good time just don like problem use make ve

Although this is natural and easy in small scale, we need computer-aided solutions to analyze hundreds or thousands of topics. The human brain can understand topics based on the words appearance in different contexts. We can easily identify topics that make sense or do not. The intuition behind word meaning according to the context has to be modeled in some way for a computer to understand. This issue will be addressed in detail in Chapter 3. Chang et al. (2009) published a relevant work [9] exploring how humans interpret topic models results, Section 2.3.4.3 explores this in detail.

2.3.4 Types of topic model evaluation

Ponweiser (2012) stated that model evaluation is needed in order to select the best possible model setup and we have to use different metrics depending on the goals and available resources[33]. For instance, a common problem in topic modeling is to choose the number of topics if this parameter is not specified a priori [34], evaluation metrics guide us when choosing the best model.

2.3.4.1 Perplexity

A comprehensive study on held-out evaluation methods applied to topic models is presented by Wallach et al. (2009). "Estimating the probability of held-out documents provides an interpretable metric for evaluating the performance of topic models relative to other topic-based models as well as to other non-topic-based generative models" [35]. One of the approaches is called *document completion*, namely to divide documents (as opposed to dividing corpora) into training and test sets [36] [35]. In this context *Perplexity* is one of the most used metrics. It is heavily used in language modeling and a lower perplexity score indicates better generalization performance [27].

One of the most common ways of evaluating a probabilistic model is to measure the log-likelihood in a test set that was left out when training the model. One starts by splitting the dataset into two sets: one for training $D_{training\,set}$ and another for testing $D_{test\,set}$.

In the case of LDA the test set is the bag of unseen documents $\bar{\boldsymbol{\omega}}_d \in D_{test\,set}$ described by the topic matrix $\boldsymbol{\Phi}$ and the hyperparameter α for distribution of topics in the documents. The LDA parameters $\boldsymbol{\Theta}$ represent, as we said, the distributions of topics for the documents of the training set in which we tunned the model, and therefore these can be ignored when computing the likelihood function for unseen documents. We can evaluate the log-likelihood of the test set $D_{test\,set}$ as

$$\begin{aligned} L\left(D_{test\,set}\right) &= \log p\left(D_{test\,set}|\boldsymbol{\Phi},\alpha\right) \\ &= \log\left[p\left(\bar{\boldsymbol{\varpi}}_1|\boldsymbol{\Phi},\alpha\right)\cdot p\left(\bar{\boldsymbol{\varpi}}_2|\boldsymbol{\Phi},\alpha\right)\cdot\ldots\cdot p\left(\bar{\boldsymbol{\varpi}}_d|\boldsymbol{\Phi},\alpha\right)\right] \\ &= \textstyle\sum_d \log p\left(\bar{\boldsymbol{\omega}}_d|\boldsymbol{\Phi},\alpha\right). \end{aligned} \tag{2.3}$$

For topic modeling it has traditionally been used the measure of *perplexity* over the documents $\bar{\boldsymbol{\omega}}_d \in D_{test\,set}$ in the held-out set. *Perplexity* is defined as

$$perplexity\left(D_{test\,set}\right) = \exp\left(-\frac{L\left(D_{test\,set}\right)}{\#tokens}\right) \tag{2.4}$$

and it is a decreasing function of the log-likelihood of the unseen documents, i.e. it should decrease as the test set increases. Therefore, a lower perplexity indicates better generalization power of the model on the words of test documents by the trained topics.

However, it is not simple to compute the likelihood of one document $\log p\left(\bar{\boldsymbol{\omega}}_d|\boldsymbol{\Phi},\alpha\right)$, let alone the sum over all the documents $L\left(D_{test\,set}\right)$. Thus, evaluating perplexity is intractable in practice.

2.3.4.2 Evaluation using secondary tasks

In some cases, a model can be evaluated by cross-validation on the error of an external task, such as document classification, information retrieval or by estimating the probability of unseen held-out documents given some training documents [35] [10].

2.3.4.3 Evaluation using human judgment

Chang et al. (2009) presented two techniques to evaluate topics using human evaluation [9]:

- *Word intrusion*: measures how semantically ‘cohesive’ the topics inferred by a model are and tests whether topics correspond to natural groupings for humans.
- *Topic intrusion*: measures how well a topic model’s decomposition of a document as a mixture of topics agrees with human associations of topics with a document.

The result of Chang et al. (2009) was that traditional measures were, negatively correlated with the measures of topic quality developed in their paper. The authors suggested that topic model developers should “focus on evaluations that depend on real-world task performance rather than optimizing likelihood-based measures”. Their work [9] was the motivation for many researchers to look for an alternative method to evaluate the quality of learned topics in terms of interpretability.

2.4 Information theory background

In LDA, documents and topics are described as distributions of probabilities. Every document is assigned to a probability distribution that tells the probability of that document been generated by each topic. The same for topics, each topic is described as a probability distribution of words, telling which words has the highest likelihood to generate that topic. That said, theory of information concepts are very helpful to evaluate and understand the model. In this section we list the most relevant for this work.

2.4.1 Entropy

Entropy is measures the uncertainty of a random variable ([37]). The notion of entropy is very important for coherence evaluation that we will see soon.

Consider a discrete random variable X over a sample space χ and probability mass function $p(x) = Pr\{X = x\}$, $x \in \chi$. The entropy of such variable, $H(X)$, is defined in base 2 to be

$$H(X) = -\sum_{x \in \chi} p(x) \log_2 p(x). \tag{2.5}$$

Intuitively, entropy is a measure of the amount of information required to describe the probability distribution of a random variable, on the average. We can calculate the entropy for a particular topic to verify if the model has uncertainty about that distribution. Lower entropy is a good indicator of good models.

2.4.2 Mutual information

We can extend the notion of entropy and relative entropy to define mutual information, which measures how much information one random variable contains about another. If two random variables, X and Y, have high mutual information then the uncertainty of one is reduced when one has knowledge of the other ([37]).

The mutual information $I(X;Y)$ of two random variables X and Y with a joint probability mass function $p(x,y)$ and individual probability mass functions $p(x)$ and $p(y)$ is the Kullback-Leibler distance between the joint distribution and the product distribution *p(x)p(y)*,

$$\begin{aligned} I(X,Y) &= D(p(x,y)\,||p(x)p(y)) \\ &= \textstyle\sum_{x\in\chi}\sum_{y\in\Upsilon} p(x,y)\log\frac{p(X,Y)}{p(X)p(Y)}. \end{aligned} \tag{2.6}$$

2.4.3 Kullback-Leibler distance

The *Kullback-Leibler distance*, also called *relative entropy*, measures the distance between two probability mass functions. It is useful to assess the degree of inefficiency of assuming that a random variable is modeled by a distribution q when the real distribution is p. It is defined, in terms of the respective probability mass functions, as

$$D(p||q) = \sum_{x\in\chi} p(x)log\frac{p(x)}{q(x)}. \tag{2.7}$$

It is important to note that, since it is not symmetric and doesn't satisfy the triangle inequality, the Kullback-Leibler distance is not a true distance between distributions, i.e. $D(p||q) \neq D(q||p)$. It would be useful to have a symmetrised metric to evaluate a true distance between distributions. This is accomplished by the symmetrised Kullback-Leibler distance, which was actually defined by Kullback and Leibler themselves,

$$sKL(p,q) = D(q||p) + D(p||q). \tag{2.8}$$

There is yet another symmetrized and smoothed version of the all important Kullback-Leibler divergence which is apparently gaining popularity among statisticians. It is the *Jensen-Shannon divergence* and it is defined as

$$JSD(p||q) = \frac{1}{2}D(p||m) + \frac{1}{2}D(q||m), \tag{2.9}$$

where $m = \frac{1}{2}(p+q)$.

Both the *sKL* and the *JSD* can be used to measure the inter-topic distance. This can estimate the quality of the topics in terms of their distinguishability: higher *sKL* divergence is usually better. Low entropy and high *sKL* also indicates high generalization power ([30]).

2.5 Summary

This chapter highlighted some relevant aspects of text mining and topic models, in particular, the basics concepts required to approach the problem addressed in this thesis. We discussed basic techniques of text mining and how to represent text in vector space model. We also covered principles of topic modeling, its applications and how we can interpret its results.

Chapter 3

Topic Coherence

To teach a computer any capability we must first be able to define it formally. A formal definition of human intuition in terms of capability to understand and recognize coherent data is a debate both in epistemology and in the philosophy of science. Fitelson, Douven and Meijs made significant contributions proposing probabilistic theories of coherence from a mathematical and philosophical perspective [38] [39]. They proved that coherence could be described as a matter of degree. For some authors their theory of coherence served as a stepping stone to formulate a set of quantitative measures of coherence.

Some researchers in the Natural Language Processing (NLP) community have proposed coherence measures to evaluate topics. While topics learned by topic models often look useful, sometimes that is not the case. Automatically quantifying topic coherence helps to quickly identify "junk" topics that may be statistically well founded, but meaningless to end users. This can lead to better ways to interact and explore the data, for instance, information retrieval applications [10] [11]. In this chapter we will cover some related work in this field and present in detail two of such measures.

3.1 Related Work

Until recently, evaluation of topic models had focused on statistical measures of perplexity or likelihood of test data. However, as demonstrated by Chang et al. (2009) [9], these measures do not consider the semantic coherence of the discovered topics, making it difficult to evaluate how well a topic model would perform in some end-user task. It was proved that sometimes perplexity could be contradictory to human evaluations in terms of interpretability of the learned topics.

Loulwah AlSumait et al. (2009) presented the first attempt of an unsupervised method to distinguish junk topics from legitimate ones. Authors argued that "topics in which the probability mass is distributed approximately equally across all words are considered likely to be difficult to interpret" [40]. Even though the results are interesting, the authors did not provide human evaluations to properly evaluate how well the method correlates with human judgments.

The first works to take human evaluations of topics into consideration were [9], [41] and [42]. Chang et al. (2009) used a false intruder detection task where humans were asked to identify intruder words on topics and intruder topics on documents. Newman et al. (2010b) published two papers about automatic evaluation of topic coherence. Their main contribution was to experiment a variety of methods to evaluate coherence. According to their results, the most promissing was based on Pointwise Mutual Information (PMI) using Wikipedia as external source, described in Chapter 3 Section 3.3.

Newman et al. (2010b) [42] experimented methods based on search engine-based similarity, term co-occurrence (PMI), WordNet similarity, etc. The method based on PMI outperformed all the alternatives and that was one of the reasons that we chose to implement the PMI based alternative in this work. The intuition behind this method is that the co-occurrence of words within documents in the corpus can indicate semantic relatedness.

Later, David Mimno et al. (2011) proposed similar method without using external source [43] and using conditional probability instead of PMI. Their method defines topic coherence as the sum of the log ratio between co-document frequency and the document frequency for the N most probable words in a topic. Their proposed measure will also be covered in this work.

K. Stevens et al. (2012) tested the coherence measures described by [43] and [42] applying different topic modeling algorithms (LDA, NMF and LSA) and compared results [44]. Their experiments explored coherence of the entire model as the average of the topic coherence from each of its learned topics.

N. Aletras and M. Stevenson (2013) enhanced the method based on PMI and proposed the construction of a semantic space to represent each topic word by making use of Wikipedia as a reference corpus to identify context features and collect frequencies [45]. Topic coherence is determined by measuring the distance between these vectors computed using a variety of metrics.

Frank Rosner et al. (2014) took the work [38] from Douven and Meijs and explored many aspects of their theory of coherence. Authors applied coherence measures from philosophy that could analyze complex word subsets and apply them to topic scoring [46]. In some experiments, they showed that their method can outperform the PMI method proposed by David Newman et al. (2010).

Michael Roder et al. (2015) were the first to propose a framework that allows constructing coherence measures by combining elementary components as we can see in Figure 3.1. Their contribution was to define and publish a framework that could help users to implement their own coherence measures [47]. Each of the boxes represents an independent aspect of the method. Segmentation is responsible for tokenization and text preprocessing, probability calculation counts the frequency of the terms in the dataset, confirmation measure is the equation for the coherence measure for each pairwise of terms and finally aggregation does the sum and normalization of the score. This picture describes virtually any coherence measure known to date and is abstract enough to accommodate new measures in the future.

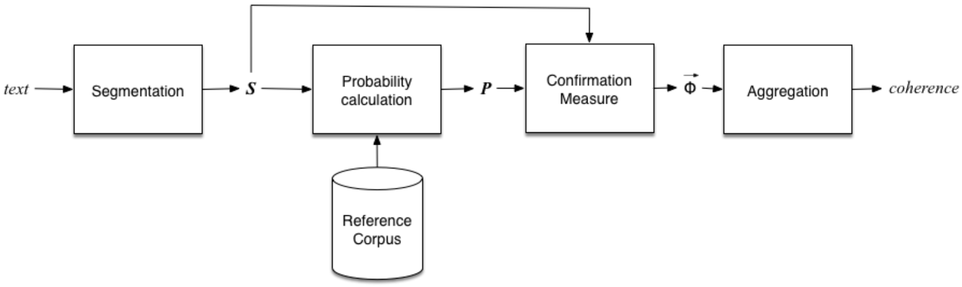


Figure 3.1: Roder et al. (2015) coherence measure framework

In this work, we implemented the measure proposed by Mimno et al. (2011), also known as UMass [43], and the one based on PMI proposed by Newman et al. (2010), also known as UCI [42]. From now on we will use these nominations to refer these measures.

3.2 Formal Definition

A topic coherence measure scores a single topic by measuring the degree of semantic similarity between high-scoring words in a particular topic given a set of documents. These measures help in distinguishing topics that are semantically interpretable from topics that are a result of statistical inference. Topic coherence is defined as *mean* of a particular coherence score for each pair of words:

$$\text{TopicCoherence}(z, D) = mean\{score(w_i, w_j, \epsilon)\}. \tag{3.1}$$

Where:

- z is a topic (i.e. a set of words describing z);
- D is a document collection (i.e. a set of documents describing D);
- and *score* is a measure of coherence between a pair of words;
- V represents the whole vocabulary present in D;
- w_i and w_j represents a pair of words that describes the topic $(w_i \in V; w_j \in V; ij \in 1 \ldots 10 \ except \ i = j)$;

- the term *epsilon* can be used as smoothing value depending on the nature of the dataset and prevents the occurrence of extreme values. This smoothing also guarantees real values as final result [44].

The smoothing was an addition proposed by Keith et al. (2012) [44]. There are several smoothing techniques that could be studied in this context. Keith et al. (2012) explored the impact of ϵ with different values [44], In this work we used $\epsilon = 1$.

Chen and Goodman (1996) described some relevant smoothing techniques used in the context of finding n-grams [48]. *Additive smoothing*, described on their work, could be eligible for an experiment focused on this aspect of topic coherence (Equation 3.1).

3.3 UCI Measure

In computational linguistics, Pointwise Mutual Information (PMI) has been used to calculate words associations and word sense disambiguation [17]. PMI measures how much one variable tells about the other and is formally defined on Equation 3.2. In our case, let variables be instances of words,

$$\mathrm{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}; \tag{3.2}$$

where the mutual information between words w_i and w_j compares the probability of observing the two words together to the probabilities of observing them independently [49].

Derived from PMI, Newman et al. (2010b) defined the UCI measure as follows:

$$\mathrm{Score}_{UCI}(w_i, w_j) = log \frac{p(w_i, w_j) + \epsilon}{p(w_i)p(w_j)}; \tag{3.3}$$

where *p(w)* represents the probability that w_i is present at a random document and *p(wi,wj)* represents the probability of both w_i and w_j being present in the same document.

To estimate probabilities we apply either the same dataset used to train the model or an external reference dataset, the first option is called *Intrinsic* and the former *Extrinsic.*

Extrinsic coherence tells us if the topics learned by the model are coherent based on external references. The external dataset can be anything related to the domain of the data used to build the topic model, in most cases Wikipedia is a good choice because it covers an immense variety of topics. According to experiments [42], authors have suggested that Wikipedia as external reference gives the best results. Intrinsic evaluation uses the original dataset rather than an external dataset to compute probabilities. It aims to confirm that the topics and words selected by the model are known to be in the data set. For instance, probabilities using Wikipedia as reference would be calculated as follows:

$$p(w_i) = \frac{D_{Wikipedia}(w_i)}{D_{Wikipedia}} \tag{3.4}$$

and

$$p(w_i, w_j) = \frac{D_{Wikipedia}(w_i, w_j)}{D_{Wikipedia}}; \tag{3.5}$$

where $D_{Wikipedia}$ counts the number of documents in the whole collection of entries at Wikipedia. $D_{Wikipedia}(w_i)$ counts entries at Wikipedia containing the word w_i and $D_{Wikipedia}(w_i, w_j)$ counts the occurrence of the words w_i and w_j at the same entry. UCI can be regarded as an external source to compare the words present in a given document with an already existent set of topics/words that gather accumulated subjective semantic evaluations. Respective code can be seen at Appendix B.

3.4 UMass Measure

UMass [43] computes the correlation of words in a given document based in *conditional probability*.

The conditional probability of an event w_i given that event w_j has occurred ($P(w_j) > 0$) is:

$$P(w_i|w_j) = \frac{p(w_i \cap w_j)}{p(w_j)}. \tag{3.6}$$

Mimno et al. (2011) [43] applied this concept to propose the UMass measure. Its equation is defined as follows:

$$\text{Score}_{UMass}(w_i, w_j) = log\frac{p(w_i, w_j) + \epsilon}{p(w_i)}; \tag{3.7}$$

where *D(w_i,w_j)* counts the number of documents that contain words w_i and w_j and *D(w_i)* counts the number of documents containing w_i. Being w_i always a word with more frequency than w_j.

The pairwise score used by the UMass is not symmetric, the order of the arguments matters. The application is that w_i must be more common than w_j. In other terms, $p(word_{rare}|word_{common})$. Respective code can be seen at Appendix B.

3.5 Discussion

An interesting aspect to note is that UCI and UMass share is that they only need a set of items as input. Such set can be the result of any model that gives sets of terms as output. Learned topics from generative models or even the list of frequent terms from clustering methods can be candidates for such evaluation. Role and Nadif (2014) successfully applied the same coherence evaluation measures discussed here to evaluate clustering labels [50]. This way is possible to compare performance from different models in terms of coherence. However, the overall coherence of the model is not in the scope of this work. Further discussion regarding this subject can be found in Stevens et al. (2012) work [44].

Intrinsic and extrinsic measures complement each other regarding topic coherence analysis. Intrinsic measures tell how much the words representing a particular topic have in common without any source beyond the original training dataset. Extrinsic measures, on the other hand, quantifies if there is any semantic meaning between the words that represent a topic using external references. Having a high extrinsic score and a low intrinsic score does not necessarily mean that one measure is better than another, they just reflect different interpretability aspects of the topic as pointed out by Omar et al. (2015) [51].

3.6 Summary

In this chapter, we covered some related work in the field of topic coherence evaluation. It also demonstrated the formal definition of coherence and two measures that aim to describe this human intuition. Using a well-known dataset in the field of text mining, in the next chapter we will test our implementation of UMass and UCI and see if they hold this claim.

Chapter 4

Topic Coherence Evaluation on 20 Newsgroups Dataset

In this chapter, we explore topic modeling and automatic coherence evaluation according to UMass, UCI measures and their extrinsic and intrinsic variations. The goal here is to test our implementation of these measures and evaluate if the scores can quantify topic's interpretability.

For this experiment, we chose the benchmark 20 Newsgroups dataset [52]. This dataset is a collection of text that consists of nearly 20,000 documents taken from a variety of newsgroups. We can think about these newsgroups as internet forums. Topics discussed in these forums range from religion to sports and politics. This dataset is famous in the text mining community for validating different types of models, such as text classification and text clustering. It comes with labels associated with documents, and these labels are informative about the content. Also, it can be used as a baseline for the number of topics we will choose to learn using LDA.

4.1 Dataset Description

The data is organized into 20 different newsgroups, each corresponding to a different theme. Some of the newsgroups are very closely related to each other (e.g. comp.sys.ibm.pc.hardware and comp.sys.mac.hardware), while others are highly unrelated (e.g misc.forsale and soc.religion.christian). Specific details about the dataset are presented in Table 4.1. Table 4.2 lists each one of the 20 newsgroups labels.

Table 4.1: Dataset features

Categories	20
Documents	18,770
Unique tokens	809,604

Table 4.2: 20 newsgroup categories

comp.graphics comp.os.mswindows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

4.2 Experimental Methodology

The dataset comprehends documents from nearly 20 different domains. We applied the same number of topics as the number of categories, $T = 20$. Extrinsic probabilities were computed using the Wikipedia dataset, considering only entries in English. After the learning phase, ten words with the highest likelihood were selected from each topic to calculate its coherence scores as described in detail in Chapter 3. The following list describes the procedures to run this experiment and Figure 4.3 illustrates them:

a) Build local Wikipedia index with entries in English. We built the index using a dump provided in September 2016 by Wikipedia at https://dumps.wikimedia.org [1]. The index was built using ElasticSearch (see Appendix A.5 for more about the tool). A total of 16,910,710 of entries were considered and it is used in this chapter to compute the extrinsic coherence scores.

b) Selection of important data within the data set. We kept only the body of each entry on the dataset. No metadata was considered (e.g. headers, footers, etc);

c) Implementation of standard procedures for text preprocessing. As described in Chapter 2.1, standard procedures to tokenize and preprocess each document were applied. In this phase, the outcome was a matrix that shows n most frequent words (unigrams) in each document in the data set. That generated a vector space representation as described in Section 2.2.5, Chapter 2.1;

d) Application of LDA. In this phase, we run LDA to learn 20 topics. We defined α as 0.1 for the Dirichlet parameter for distribution over topics and 0.01 for β the distribution over words. For more information about these parameters see Chapter 2.1 Section 2.3.2. Later we select the words that are most likely to appear on a given topic. In order to set the appropriated number of iterations we analyzed the log-likelihood along the number of iterations. As we can see on Figure 4.1 We can safely say that there is virtually no gain after running more than around 800 iterations in this particular case.

[1] Wikipedia dump downloaded from https://dumps.wikimedia.org/enwiki/20160920/enwiki-20160920-pages-articles.xml.bz2

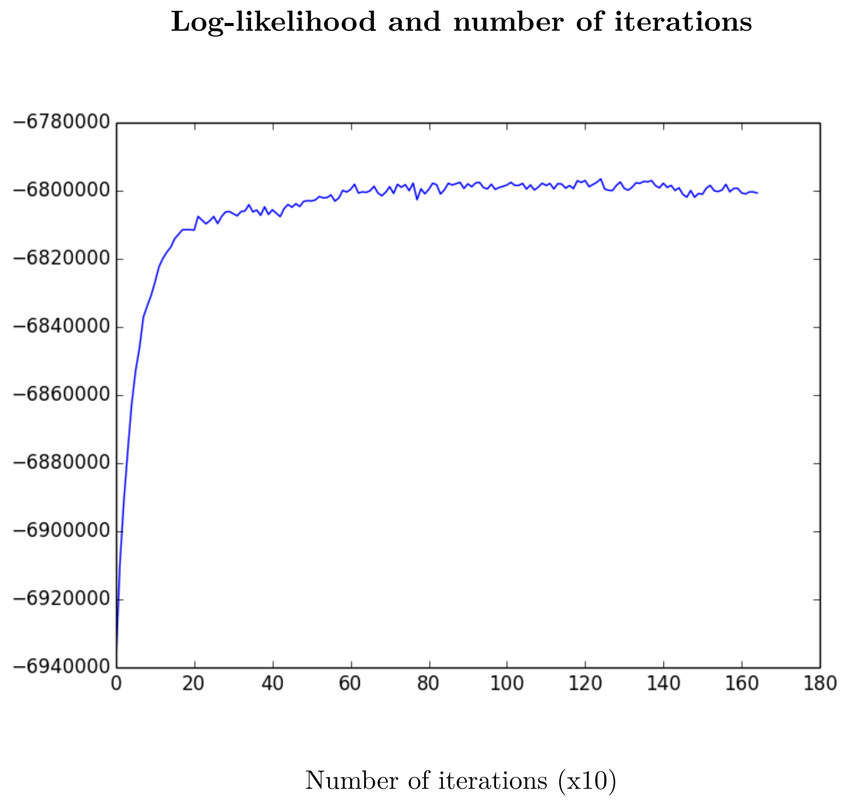


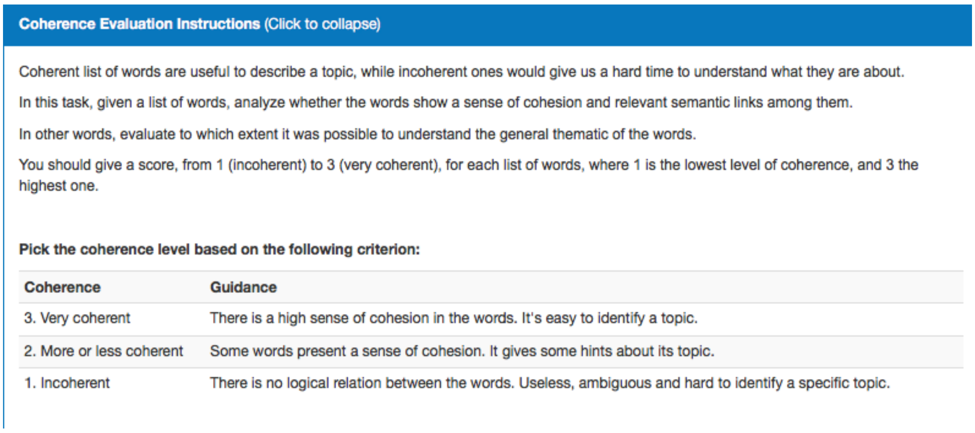
Figure 4.1: Log-likelihood evolution versus number of iterations

e) Computation of coherence scores. This phase computes the scores as described in Chapter 3. After applying LDA and detect 20 topics, for each of them we calculate extrinsic and intrinsic scores using UCI and UMass. The intrinsic scores are calculated using the original 20 newsgroup dataset while the extrinsic ones are calculated using English Wikipedia.

f) Human evaluation. In order to build a ground-truth for deeper analysis we designed a task with human annotators. We asked users to manually evaluate the coherence of topics using the platform Amazon Mechanical Turk [2]. Users had to evaluate all topics listed in Table 4.3 according the guidelines described in Figure 4.2. Annotators were not familiar with the dataset.

1. Analyze whether the words on a given topic showed some cohesion, to wich extent they were capable to understand the topic. They should give a score (from 1 to 3) where 1 for incoherence and 3 to high coherence.

[2] https://www.mturk.com/



Coherence Evaluation Instructions (Click to collapse)

Coherent list of words are useful to describe a topic, while incoherent ones would give us a hard time to understand what they are about.

In this task, given a list of words, analyze whether the words show a sense of cohesion and relevant semantic links among them.

In other words, evaluate to which extent it was possible to understand the general thematic of the words.

You should give a score, from 1 (incoherent) to 3 (very coherent), for each list of words, where 1 is the lowest level of coherence, and 3 the highest one.

Pick the coherence level based on the following criterion:

Coherence	Guidance
3. Very coherent	There is a high sense of cohesion in the words. It's easy to identify a topic.
2. More or less coherent	Some words present a sense of cohesion. It gives some hints about its topic.
1. Incoherent	There is no logical relation between the words. Useless, ambiguous and hard to identify a specific topic.

Figure 4.2: Task description for manual human evaluation

4.3 Implementations Details

Each section in Figure 4.3 corresponds to an independent phase in the data pipeline. Data preparation phase contains scripts that are responsible for indexing the documents and to perform text preprocessing tasks. At the topic modeling phase, the script loads the text features and runs LDA to learn the topics. The final part of the pipeline executes calculations to define the coherence of each topic.

Code listing 4.1 and 4.2 show how to compute UCI and UMass scores for a particular topic using our implementation. Appendix B presents inner details about our Python implementations of UCI and UMass. The complete source code for the whole experiment is accessible at https://github.com/arianpasquali/msc-thesis-code. Third-party libraries to implement this work are listed in Appendix A.

```
from topic_coherence import UCI

topic = "space launch nasa satellite mission lunar"

/* using wikipedia as external reference */
extrinsic_reference = "en_wikipedia"
extrinsic_uci = UCI(extrinsic_reference)
extrinsic_uci_score = extrinsic_uci.fit(topic)

/* using the same dataset as internal reference */
intrinsic_reference = "20newsgroups"
intrinsic_uci = UCI(intrinsic_reference)
intrinsic_uci_score = intrinsic_uci.fit(topic)

print("UCI scores")
print("    Extrinsic: ",extrinsic_uci_score)
print("    Intrinsic: ",intrinsic_uci_score)

>>> UCI scores :
    Extrinsic : 10.38
    Intrinsic : 7.86
```

Listing 4.1: Python code to calculate topic coherence using UCI

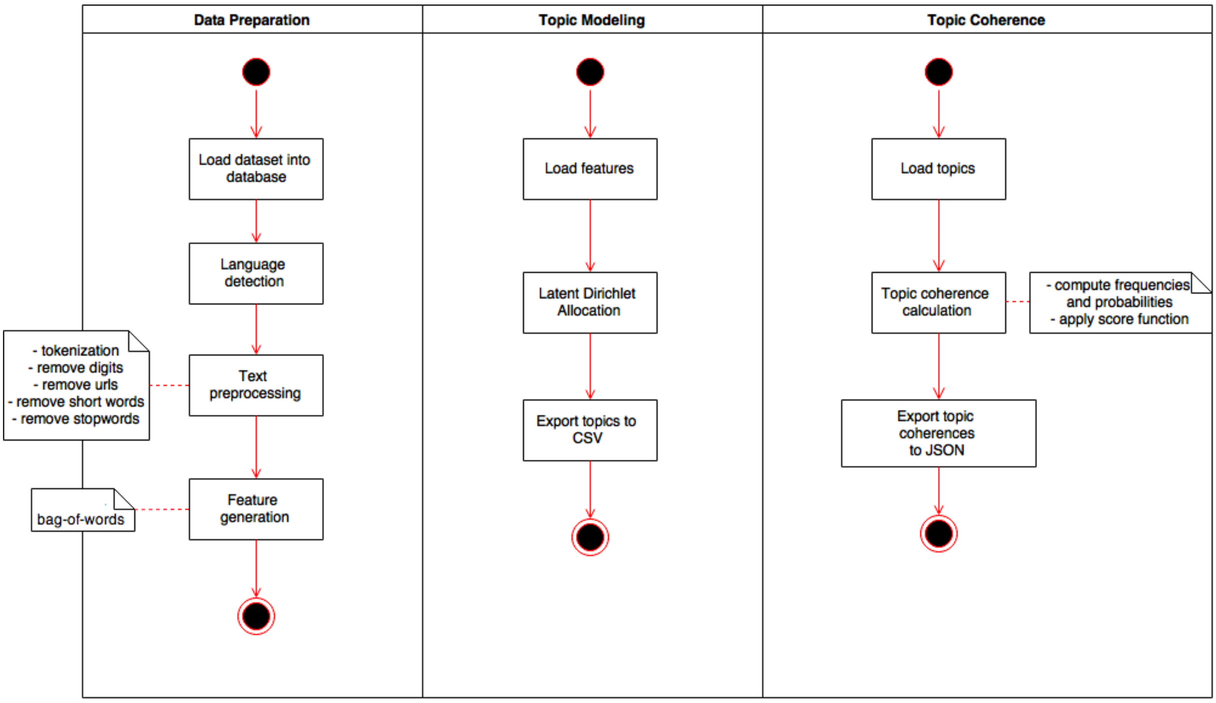


Figure 4.3: Data processing pipeline diagram

```
from topic_coherence import UMass

topic = "space launch nasa satellite mission lunar"

extrinsic_reference = "en_wikipedia"
extrinsic_umass = UMass(extrinsic_reference)
extrinsic_umass_score = extrinsic_uci.fit(topic)

intrinsic_reference = "20newsgroups"
intrinsic_umass = UMass(intrinsic_reference)
intrinsic_umass_score = intrinsic_uci.fit(topic)

print("UMass scores")
print("    Extrinsic: ",extrinsic_umass_score)
print("    Intrinsic: ",intrinsic_umass_score)

>>> UMass scores :
    Extrinsic : 4.81
    Intrinsic : 3.52
```

Listing 4.2: Python code to calculate topic coherence using UMass

4.4 Results

Table 4.3 presents the 20 topics discovered using LDA. We can manually see that some topics are easy to interpret, namely *topics 5, 7, 13* and so on. *Topic 5*, in particular, have presented the highest scores (highlighted in bold). On the other hand, *topics 9 and 3* are examples of instances that are hard to interpret. To keep different scores on the same scale we normalized them using standardization.

Table 4.5 lists the topics inversely ordered by extrinsic UCI scores, while Table 4.7 and Table 4.6 are ordered by intrinsic UCI and UMass scores, respectively. Scores were normalized using standartization, coverting a value in a normal distribution to its equivalent in a standard normal distribution.

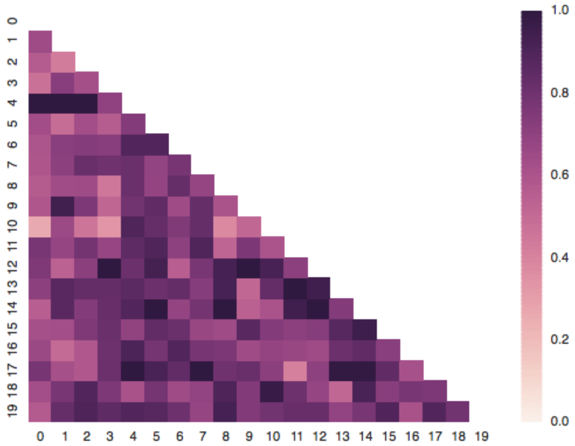


Figure 4.4: Inter-topic distance with symmetric Kullback-Leibler divergence

Using symmetric Kullback-Leibler divergence we also calculated inter-topic distances and these are represented in Figure 4.4. Calculating the distances and normalizing results we are able to see that every topic distribution seems different from each other. Topic 4, for instance, is very different from topic 0, 1 and 2. If we inspect Table 4.3 we can confirm that *topic 4* is about sports while most distance topics seem to talk about *windows operating system* and *computer graphics*. According the *sKL divergance*, topics *12* and *13* are also very distant. Inspecting Table 4.3 again, we see that the former is describing technical computer science while the last talks about arab people.

Figure 4.5 shows that all variations of extrinsic and intrinsic from UCI and UMass give similar results, with the exception of some topics. It is interesting to note that the *topic 4* related to hockey and NHL received a low score by extrinsic measures, in contrast to intrinsic results. Further analysis could be done to find an explanation. We can hypothesize that the external source does not have enough information about hockey games and NHL, this could explain the low score.

Table 4.4 presents the coherence scores inversely ordered by extrinsic UMass score. We can identify that topics on the top are in general easily understood. The first line, *topic 5* for instance, seems to represent a topic related to space explorations, the next one clearly comprehends computer hardware. At the bottom of the table, we can find less comprehensible topics.

Table 4.3: 20 Newsgroups's detected topics

Id	Topics	Extrinsic		Intrinsic	
		UMass	UCI	UMass	UCI
0	people think believe say point like evidence make way question	0.638	0.689	0.604	0.481
1	image available version file software data graphics color program images	0.769	0.806	1.034	0.914
2	windows window using file program dos set run server running	0.915	0.939	0.946	0.825
3	think good year better game hit like time got team	0.598	0.629	0.706	0.524
4	game team hockey win lost new games nhl play april	0.711	0.699	1.106	0.946
5	space launch nasa earth shuttle solar energy satellite mission data	1.069	**1.168**	**1.400**	**1.228**
6	book read article books know called reference time used written	0.571	0.604	0.845	0.646
7	medical health number study drug patients cancer disease new cause	0.863	0.878	1.267	1.027
8	car like bike new good engine used cars know really	0.759	0.704	0.782	0.540
9	aid went people told know came saw started took like	0.656	0.715	0.820	0.623
10	like think know people want going good really make say	0.648	0.711	0.509	0.434
11	like good price looking sell buy know new want interested	0.782	0.794	0.714	0.535
12	information send list mail available email computer anonymous ftp address	0.992	0.930	0.984	0.855
13	armenian turkish armenians jews israel israeli war jewish people arab	1.062	1.051	1.288	1.104
14	god jesus bible church christian christians believe man faith people	0.878	0.891	1.010	0.845
15	lower line file entry output current need used water number	0.719	0.746	0.986	0.776
16	key encryption chip government clipper security keys phone data used	1.110	1.029	1.073	0.918
17	drive disk card hard scsi mac video know bus drives	**1.154**	1.146	1.037	0.852
18	president new american national united said states general public going	0.464	0.482	0.974	0.805
19	gun government right people law police state rights fbi laws	0.752	0.700	0.993	0.809

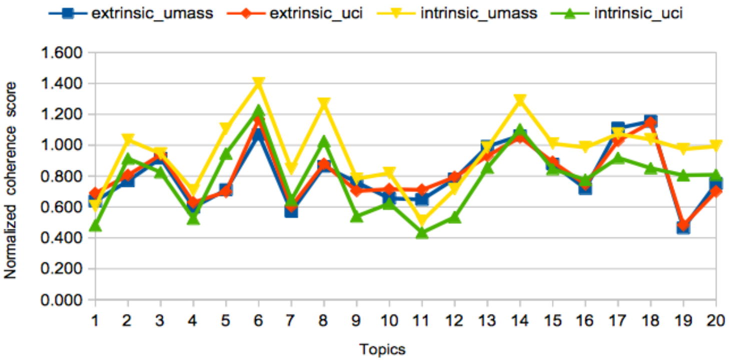


Figure 4.5: Normalized coherence scores

Table 4.4: Topics ordered by Extrinsic UMass coherence scores

Id	Topics	Extrinsic UMass
17	drive disk card hard scsi mac video know bus drives	1.154
16	key encryption chip government clipper security keys phone data used	1.110
5	space launch nasa earth shuttle solar energy satellite mission data	1.069
13	armenian turkish armenians jews israel israeli war jewish people arab	1.062
12	information send list mail available email computer anonymous ftp address	0.992
2	windows window using file program dos set run server running	0.915
14	god jesus bible church christian christians believe man faith people	0.878
7	medical health number study drug patients cancer disease new cause	0.863
11	like good price looking sell buy know new want interested	0.782
1	image available version file software data graphics color program images	0.769
8	car like bike new good engine used cars know really	0.759
19	gun government right people law police state rights fbi laws	0.752
15	power line file entry output current need used water number	0.719
4	game team hockey win lost new games nhl play april	0.711
9	said went people told know came saw started took like	0.656
10	like think know people want going good really make say	0.648
0	people think believe say point like evidence make way question	0.638
3	think good year better game hit like time got team	0.598
6	book read article books know called reference time used written	0.571
18	president new american national united said states general public going	0.464

Table 4.5: Topics ordered by Extrinsic UCI coherence scores

Id	Topics	Extrinsic UCI
5	space launch nasa earth shuttle solar energy satellite mission data	1.168
17	drive disk card hard scsi mac video know bus drives	1.146
13	armenian turkish armenians jews israel israeli war jewish people arab	1.051
16	key encryption chip government clipper security keys phone data used	1.029
2	windows window using file program dos set run server running	0.939
12	information send list mail available email computer anonymous ftp address	0.930
14	god jesus bible church christian christians believe man faith people	0.891
7	medical health number study drug patients cancer disease new cause	0.878
1	image available version file software data graphics color program images	0.806
11	like good price looking sell buy know new want interested	0.794
15	power line file entry output current need used water number	0.746
9	said went people told know came saw started took like	0.715
10	like think know people want going good really make say	0.711
8	car like bike new good engine used cars know really	0.704
19	gun government right people law police state rights fbi laws	0.700
4	game team hockey win lost new games nhl play april	0.699
0	people think believe say point like evidence make way question	0.689
3	think good year better game hit like time got team	0.629
6	book read article books know called reference time used written	0.604
18	president new american national united said states general public going	0.482

Table 4.6: Topics ordered by Intrinsic UMass coherence scores

Id	Topics	Intrinsic UMass
5	space launch nasa earth shuttle solar energy satellite mission data	1.400
13	armenian turkish armenians jews israel israeli war jewish people arab	1.288
7	medical health number study drug patients cancer disease new cause	1.267
4	game team hockey win lost new games nhl play april	1.106
16	key encryption chip government clipper security keys phone data used	1.073
17	drive disk card hard scsi mac video know bus drives	1.037
1	image available version file software data graphics color program images	1.034
14	god jesus bible church christian christians believe man faith people	1.010
19	gun government right people law police state rights fbi laws	0.993
15	power line file entry output current need used water number	0.986
12	information send list mail available email computer anonymous ftp address	0.984
18	president new american national united said states general public going	0.974
2	windows window using file program dos set run server running	0.946
6	book read article books know called reference time used written	0.845
9	said went people told know came saw started took like	0.820
8	car like bike new good engine used cars know really	0.782
11	like good price looking sell buy know new want interested	0.714
3	think good year better game hit like time got team	0.706
0	people think believe say point like evidence make way question	0.604
10	like think know people want going good really make say	0.509

Table 4.7: Topics ordered by Intrinsic UCI coherence scores

Id	Topics	Intrinsic UCI
5	space launch nasa earth shuttle solar energy satellite mission data	1.228
13	armenian turkish armenians jews israel israeli war jewish people arab	1.104
7	medical health number study drug patients cancer disease new cause	1.027
4	game team hockey win lost new games nhl play april	0.946
16	key encryption chip government clipper security keys phone data used	0.918
1	image available version file software data graphics color program images	0.914
12	information send list mail available email computer anonymous ftp address	0.855
17	drive disk card hard scsi mac video know bus drives	0.852
14	god jesus bible church christian christians believe man faith people	0.845
2	windows window using file program dos set run server running	0.825
19	gun government right people law police state rights fbi laws	0.809
18	president new american national united said states general public going	0.805
15	power line file entry output current need used water number	0.776
6	book read article books know called reference time used written	0.646
9	said went people told know came saw started took like	0.623
8	car like bike new good engine used cars know really	0.540
11	like good price looking sell buy know new want interested	0.535
3	think good year better game hit like time got team	0.524
0	people think believe say point like evidence make way question	0.481
10	like think know people want going good really make say	0.434

4.5 Human evaluation

In order to build a ground-truth we asked users to manually evaluate the coherence of topics using the platform Amazon Mechanical Turk [3]. Users had to evaluate all topics listed in Table 4.3 according to given guidelines (see Figure 4.2).

We selected a total of 10 people that answered our task, each one evaluated all of the 20 topics. We selected only users that evaluated all 20 topics to avoid having to deal with missing values. In order to assess the agreement of the annotators, we calculated *Fleiss' Kappa* [53], a usual measure of inter-raters agreement. Before any analysis we calculated the inter-raters agreement to identify potential outliers. We calculate Kappa Fleiss for the answers leaving one user out each time to see which one had the most negative impact. As we can see in Figure 4.6, when we leave *user 2* out the Kappa value increases considerably giving us an inter-rater agreement score of 0.22, while keeping that user Kappa value decreases to values between 0.15 to 0.17. We decided to ignore user 2 for this evaluation based on this claim.

We then calculate the correlation of their answers against our calculated scores presented in Table 4.3. We can see in Figure 4.7 that human coherence evaluation correlation with our four measures are strong. In most scenarios, intrinsic UMass and UCI presented better correlation than their extrinsic version. We can conclude that with the exception of *user 3* and *user 10*, all users presented strong correlation with our measures, specially *user 4*.

In order to have more insight over correlations for each topic we built the plot at Figure 4.8. Each point in the plot represents a particular topic and we can see how they were evaluated comparing with the average score that topic receive from users. The most points we have in the grey area, the better. We can confirm again that intrinsic measures performed better than their extrinsic counterpart in this experiment, both UCI and UMass presented a very similar behavior.

[3] https://www.mturk.com/

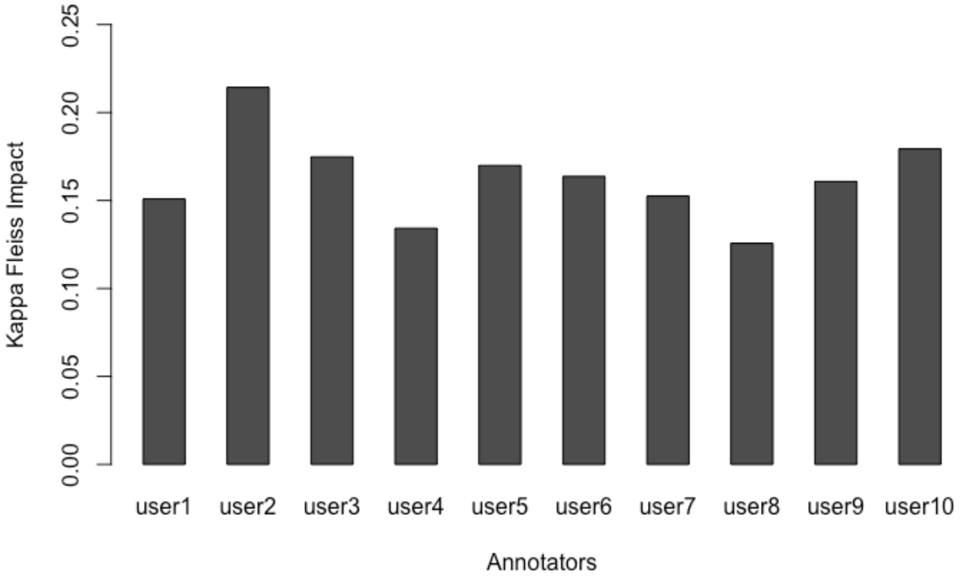


Figure 4.6: Users negative impact on inter-raters agreement

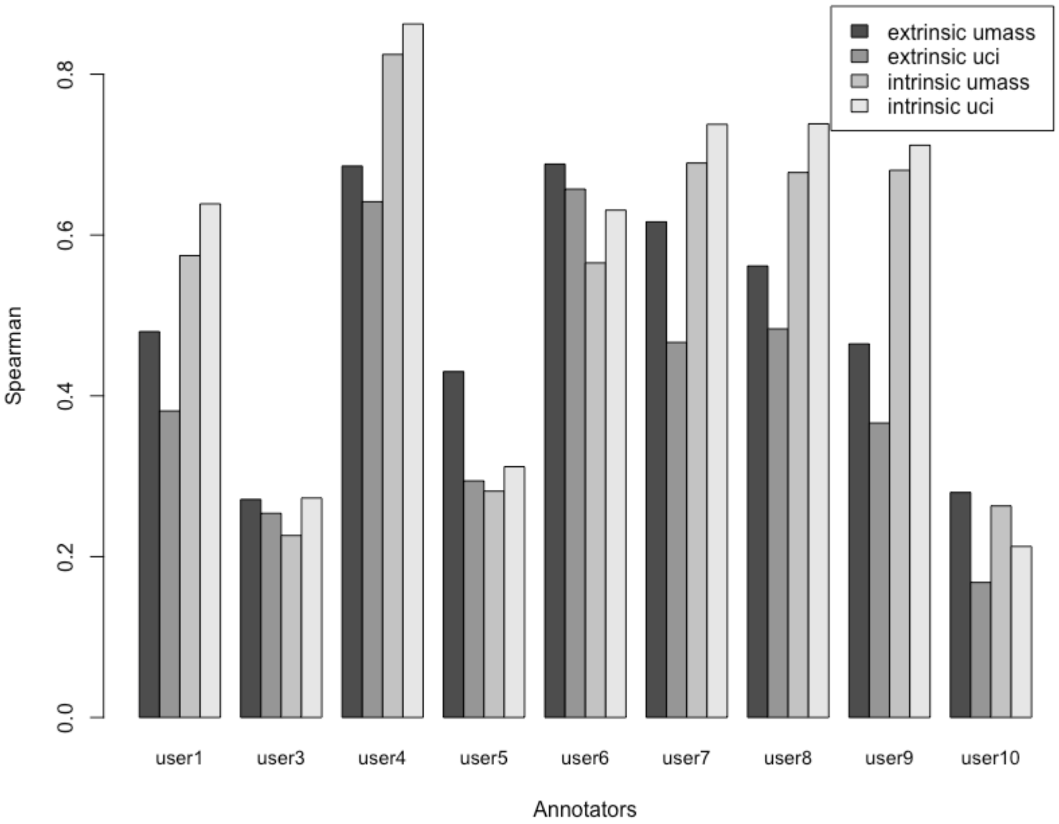


Figure 4.7: Correlations: user evaluation versus automatic measures

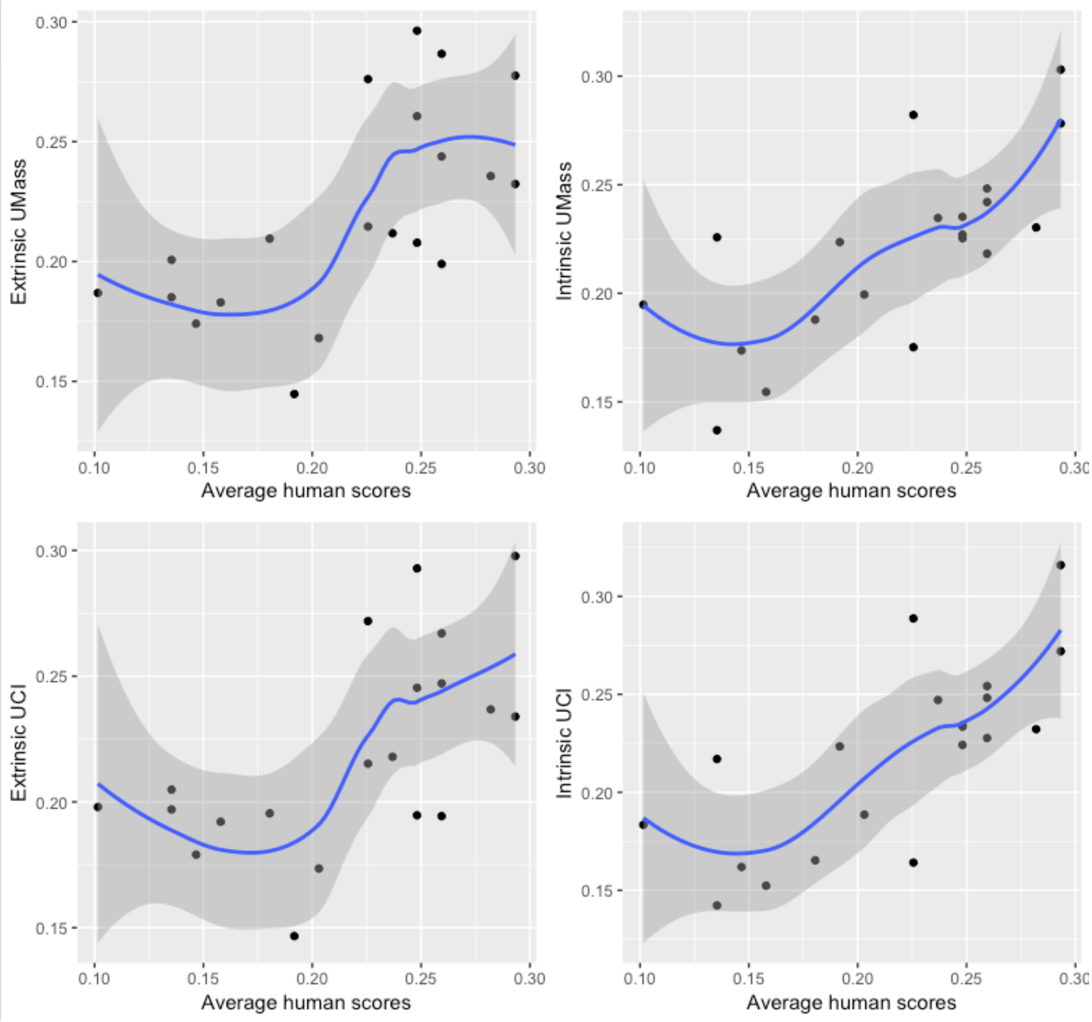


Figure 4.8: Average human scores versus automatic scores

4.6 Performance Analysis

We made a simple performance analysis counting the duration of function calls for each coherence measure. For each of the 20 topics we calculated the time spent to compute the intrinsic and extrinsic scores using UCI and UMass, we then calculated the average time spent on each operation. Figures 4.9 and Figure 4.10 present the performance in milliseconds for UMass and UCI respectively. We can see that intrinsic measures performed 7 times faster than extrinsic ones. One hypothesis for extrinsic worst performance is disk usage, the Wikipedia index used as external index occupies 27Gb of disk space and considerably more CPU power, in contrast to only 38M of 20newsgroup text index.

Comparing our implementations of UCI and UMass algorithms alone, we conclude that UCI performed 6% to 8% faster in comparison to UMass. There is, however, room for optimization since we didn't put additional effort for optimizing the algorithm and the index.

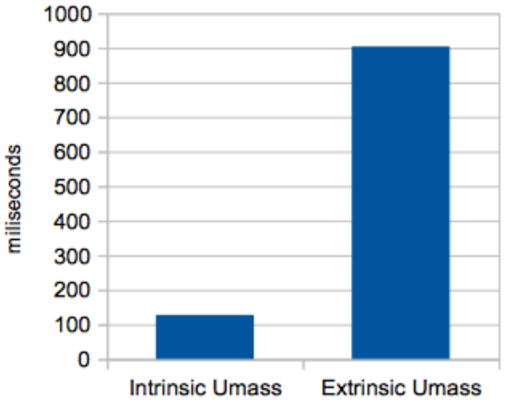


Figure 4.9: Intrinsic versus Extrinsic UMass

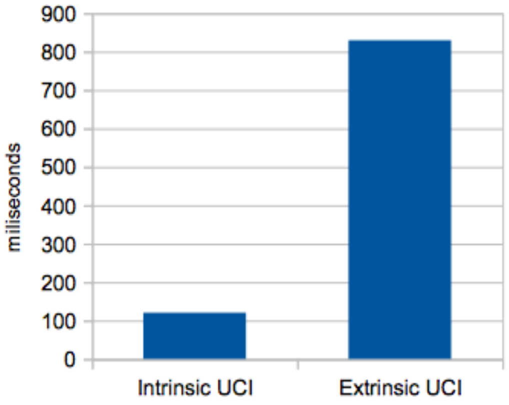


Figure 4.10: Intrinsic versus Extrinsic UCI

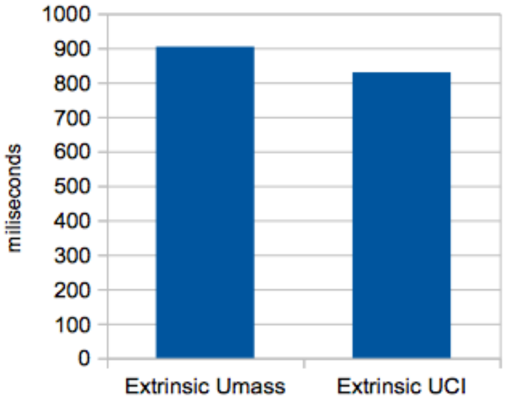


Figure 4.11: Extrinsic UMass versus UCI

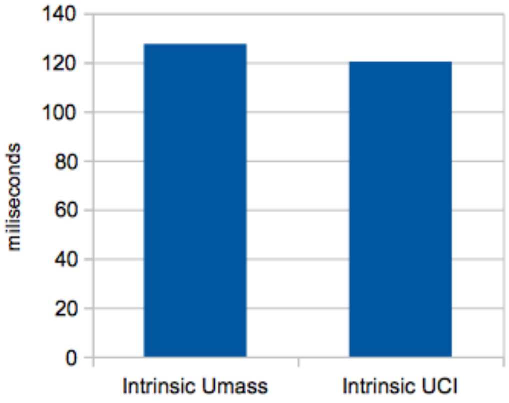


Figure 4.12: Intrinsic UMass versus UCI

4.7 Discussion

This experiment shows that we can highlight the differences between topics that are easier to interpret than others using these measures. According to our experiments in this chapter we saw that intrinsic evaluation tends to have better performance than extrinsic using Wikipedia as an external resource, mainly because of the index size and disk usage. Further optimization and strategies are necessary for that area. The difference in terms of performance between our implementations of UMass and UCI are about 5% to 8%. This 5% to 8% difference is almost irrelevant in this particular scenario, however this small difference maybe important for large scale scenarios and became relevant.

Regarding coherence evaluation, the score results in this experiment give us confidence to further investigate other applications. For instance, let's imagine a hypothetical use case where these measures are applied to support end-users tasks. One practical application would be to consider only topics with high scores based on a predefined threshold, discarding topics with lower interpretability score. Another approach could group all topics with scores lower than a predefined threshold and label it as 'others' or 'miscellaneous'. See Table 4.8 for a hypothetical example (topic labels were manually defined).

Table 4.8: Application example

Relevant Topics
Space exploration
Jews and Arab situation
Medical research
Computer hardware
Religion
others

4.8 Summary

This chapter provided a use case to test our implementations of UCI and UMass coherence measures. According to our first experiment, we could see that UMass and UCI performed reasonably well with both providing satisfactory results when assessing the correlation with human evaluation scores. In the next next chapter we designed a similar experiment, this time with a dataset of Facebook posts in Portuguese covering topics about politics and social movements, human evaluation will be done by experts in the domain of politics. Finally, in Chapter 6 will describe a user interface implemented to explore the results found in that experiment.

Chapter 5

Topic Coherence Evaluation on Facebook Posts Dataset

Experiments with data collected from social networks are well covered by the literature in the field of text mining and topic modeling [54] [55] [56]. However, we didn't find any studies about topic models and automatic coherence evaluation applied to social networks data or in Portuguese. Regarding coherence evaluation, authors have been publishing their results using exclusively well-stablished datasets like 'The 20 Newsgroup Dataset' [52] and 'The New York Times Annotated Corpus' [57]. The last, comprehends articles written and published by the New York Times between January 1, 1987, and June 19, 2007.

The interest in mining the Web data for political insights has increased since the booming of popular upheavals around the world, in the 2000's, especially after the Arab Spring. A number of authors [58] [59] [60] [61] agree that recent uprisings have been a result of a complex network of interactions both on social networks and live political demonstrations (sometimes simultaneously). On these grounds, many researchers have begun to explore open social data to study topics on Social Science [62] [63] [55] [6]. A currently relevant example is the automatic analysis of streams of posts issued by different political activist groups in Brazil, through the analysis of the generated streams of texts made available on the web.

Due to recent historical events [64], we chose to explore messages published on Facebook about political events in Brazil. In this chapter, we apply topic modeling on Facebook posts in Portuguese related to political movements and verify whether two automatic evaluation measures can model human judgment when working with short and poorly structured texts. In particular, our aim is to assess the compliance of the measures with manual human evaluation using a domain specific dataset and domain experts to create our ground-truth.

We applied topic modeling on political messages published on 36 Facebook pages and then asked three annotators to analyze the relevance of each learned topic. Their scores were compared to the UMass and UCI scores. Portuguese Wikipedia was used as an external reference to calculate extrinsic coherence scores.

The 36 pages were selected by a domain expert based on her political views and categorized into six different classes based on previous knowledge about their general features/profile, and according to the following criteria:

- show some relevance in the production and/or dissemination of content on Brazilian contentious politics;
- not being a social network official page of any corporate media organization;
- be identified as a group instead of a singular individual (a recurring feature in political actors in social media);
- be active from March 1, 2015, and February 29, 2016 (data collection time range).

5.1 Dataset Description

All data was collected in mid-March, 2016, using the application Netvizz 1.25 [65], which retrieved 314,973 posts for the 36 pages. See Appendix A.6 for more detail about this tool.

Netvizz lets the researcher choose between the last n posts and all posts published in a window of time. We opted to collect data from March 1, 2015, to February 29, 2016, because that was an intensive period in the Brazilian political context, generating lots of relevant content in social networks. Then we run the application to retrieve the data automatically.

The generated dataset aggregates 313,514 posts, considering each status update, photo, video, note and link share on a page as a *document*. We split the pages into 6 political orientation classes. Each class' features are described below (Table 5.2 lists all pages considered and Table 5.1 refers general features of each class data set).

Table 5.1: Number of posts per class of page

Class	Features	Posts
1	Particular cause (Social Movement)	7,367
2	Grassroots news (Leftist)	14,591
3	Pro-Governism news (Center) [1]	47,080
4	Pro-impeachment news (Rightist)	37,433
5	Pro-impeachment virals (Rightist)	196,641
6	Progressist virals	10,333
Total	-	313,514

5.1.1 Page classes

This section describes in detail the characteristics of each page class.

Class 1 - social movement with a singular main cause: page focused on a specific kind of Right, disseminating topics related to its main cause. It is managed by activists who maintain actions on the streets and digital social networks.

Class 2 - grassroots media: leftist groups that disseminate own-produced and third-party news pieces, mainly about social movements, popular demonstrations, and other related topics. Many of them were born from massive popular protests in Brazil in 2013 and tended to be neither pro-President Dilma Rousseff nor pro-impeachment. They are frequently confronting mass-media outlets' versions on political topics.

Class 3 - Pro-President Dilma Rousseff administration: news outlets that disseminate own-produced pieces. They also tend to share lots of content from each other and are frequently confronting mass-media outlets' versions on political topics.

Class 4 - Rightist news outlets that disseminate own-produced and third-party pieces that demand President Dilma Rousseff impeachment. They are also consistently against left-wing administrations in other Latin American countries and adopt a strong discourse against corruption.

Class 5 - Rightist pages that spread viral *memes* and third-party links demanding President Dilma Rousseff impeachment. They are frequently against left-wing administrations in other Latin American countries and are more focused on easy-to-turn-viral content than analytical or descriptive news pieces.

Class 6 - Pages with a progressist view of political themes. They are more focused on easy-to-turn-viral content than analytical or descriptive news pieces although sometimes they publish third-party news, usually with sarcastic comments.

Table 5.2: Facebook pages

Class	Description	Facebook Pages
Class 1	Particular cause (Social Movement)	- Aliados do Parque Augusta - Comitê Popular Rio Copa - Das Lutas - Garis do Rio de Janeiro em Luta - Movimento Passe Livre - Ocupe Estelita
Class 2	Grassroots news (Leftist)	- A Nova Democracia - Guerrilha GRR - Mariachi - Midia Independente Coletiva - Papo Reto - Vírus-Planetário
Class 3	Pro-Governism news (Center)	- Brasil 247 - Diario do Centro do Mundo - Favela 247 - Revista Forum - Jornal GGN - Pragmatismo Político
Class 4	Pro-impeachment news (Rightist)	- Correio do Poder - Folha Política - Implicante - O Antagonista - O Reacionário - Vem Pra Rua Brasil
Class 5	Pro-impeachment virals (Rightist)	- Humor 13 - Movimento Brasil Livre - Movimento Contra a Corrupção - Movimento Endireita Brasil - TV Revolta - Revoltados Online
Class 6	Progressist virals	- Acorda Meu Povo - Deboas na Revolução - Movimento Pro-Corrupção - O Badernista - Porque Eu Quis - Rede Esgoto de Televisão

5.2 Experimental Methodology

The experimental methodology applied in this use case combines the application of LDA, the computation of intrinsic and extrinsic coherence using UMass and UCI, and human evaluation.

Inspired by D. Newman (2010) methodology, we defined an experiment to evaluate the correlation between human judgment regarding observed coherence against our coherence measures. The basic procedures are the same as described in the previous chapter Section 4.3, with an addition that we calculated their correlation with human judgment from domain experts, all the steps are described bellow:

a) Build local Wikipedia index with entries in Portuguese. We built the index using a dump provided in March 2016 by Wikipedia at https://dumps.wikimedia.org [2]. A total of 2,065,963 of documents in Portuguese were considered. This index is used to compute the extrinsic coherence scores.

b) Selection of relevant data within the Facebook posts data set. We kept only the original text of each publication and the type of post (status update, link, photo and video). All other data is not considered (e.g.: users unique number ID; number of likes, comments and shares; post ID; date of publication).

c) Implementation of standard preprocessing procedures. As described on Chapter 2.1, standard procedures to tokenize and preprocess each document were applied. In this phase, the outcome was a matrix that shows n most frequent words (uni-grams) in each document in the data set. That generated a vector space representation as described in Section 2.2.5, Chapter 2.1;

d) Application of LDA. Application of LDA to learn 15 topics from each class and to select the top 9 words (uni-grams) from each topic. We defined α as 0.1 for the Dirichlet parameter for distribution over topics and 0.01 for β the distribution over words. For more information about these parameters see Chapter 2.1 Section 2.3.2. This phase selected the nine words that were most likely to appear on a given topic. The number of topics was arbitrarily defined, and we use the same number of topics for all six classes. Finding the optimal number of topics for each class was out of the scope of this work.

e) Computation of topic coherence. For each of the 15 topics from each of the 6 classes, this phase computes the extrinsic and intrinsic coherence scores using UMass and UCI, as described in Chapter 3.

f) Human evaluation. All annotators were familiar with the general thematic on the pages. They are professionals in the Communications field and are personally involved in the Brazilian political scenario to which the pages' content relates to. Each annotator has analyzed all the 15 learned topics with nine words for each class of pages (see Table 5.3). Annotators were asked to evaluate each topic according to this guideline:

1. Analyze whether the words on a given topic showed relevant semantic links among those words. In other words, as domain experts, to which extent it was possible to understand the general thematic of the topics. They should give a score (from 1 to 5) for each topic, where 1 is the lowest level of coherence, and 5 the highest one. Later we simplified the scores to 1-3 scale as we did in the previous chapter.

g) Inter-rates agreement.

[2] Wikipedia dump downloaded from https://dumps.wikimedia.org/ptwiki/20160920/ptwiki-20160920-pages-articles.xml.bz2

Table 5.3: Annotators tasks example

Topics	Score(1-5)
rio, esquerda, professor, janeiro, paulo, carlos, universidade, partir, centro	2
brasil, governo, povo, presidente, federal, direitos, direito, poder, caso	5
garis, greve, trabalhadores, luta, comlurb, sindicato, rio, gari, chapa	5
transporte, aumento, copa, movimento, mundo, governo, passe, livre, tarifa	4
povo, negro, marcha, reaja, campanha, internacional, anos, luta, dia	5
ato, dia, policiais, rio, pessoas, protesto, frente, apoio, rua	3
bem, pessoas, coisa, cidade, sempre, poder, fazendo, anos, bom	1
direitos, rio, dia, humanos, janeiro, ativistas, mil, caso, segundo	3
movimento, dia, coletivo, popular, movimentos, rede, social, luta, coletiva	2
apoio, moradores, prefeitura, vila, luta, hoje, solidariedade, novas, praia	4
povo, anos, pior, pessoas, banco, hoje, dias, brasileiro, infelizmente	1
parque, pic, nic, circulo, dia, poder, cidade, gente, podemos	4
rio, vila, moradores, prefeitura, projeto, comunidade, prefeito, eduardo, copa	5
parque, augusta, cidade, municipal, prefeitura, dia, luta, rua, guarda	5
mulheres, pessoas, sociedade, forma, vida, mulher, nunca, grupo, homens	2

In statistics, the inter-rater agreement is the degree of agreement among raters. It gives a score of how much consensus there is in the ratings given by the annotators. We ranked topics from each of the 6 classes according to the ratings. We then analyze the correlation from these manual evaluations with the extrinsic and intrinsic coherence using UCI and UMass.

5.3 Results

In this section, we present the outcomes in terms of the correlation between human evaluations and our automated measures. We applied Spearman correlation for this task. All learned topics from the model are listed in Appendix section C.1.

In order to assess the agreement of the annotators, we calculated *Fleiss' Kappa* [53], a usual measure of inter-raters agreement. In our experiment, values of *Kappa* range between 0.209 and 0.53 for all the classes but 5, where it is negative. Being 1 maximum agreement and -1 maximum disagreement, we see that there is a moderate agreement between raters in all classes but one. The low p values indicate that the value of *Kappa* for that class is significantly different from zero.

All scores manually given by annotators are listed in Appendix Section C.2.

Table 5.4: Inter-raters agreement

Class	1	2	3	4	5	6
kappa	0.265	0.43	0.53	0.453	-0.009	0.209
p value	0.015	0.00	0.00	0.000	0.938	0.053

5.3.1 Correlation between UCI and manual evaluation

Figure 5.1 shows the correlation between manual evaluations and UCI considering extrinsic and intrinsic variations. We can see that annotators have very similar correlations with UCI for all classes, with the exception of *Class 5* for extrinsic and intrinsic scores and *Class 1* for intrinsic scores. Bars represent the three different annotators. Considering the extrinsic scores for *Class 5*, only one annotator agreed with UCI evaluation. It is interesting to note that, with exception of annotator 3 in *Class 5*, all annotators tend to agree with each other, while in *Class 1* and 5 they have the opposite opinion than our automatic coherence measures,

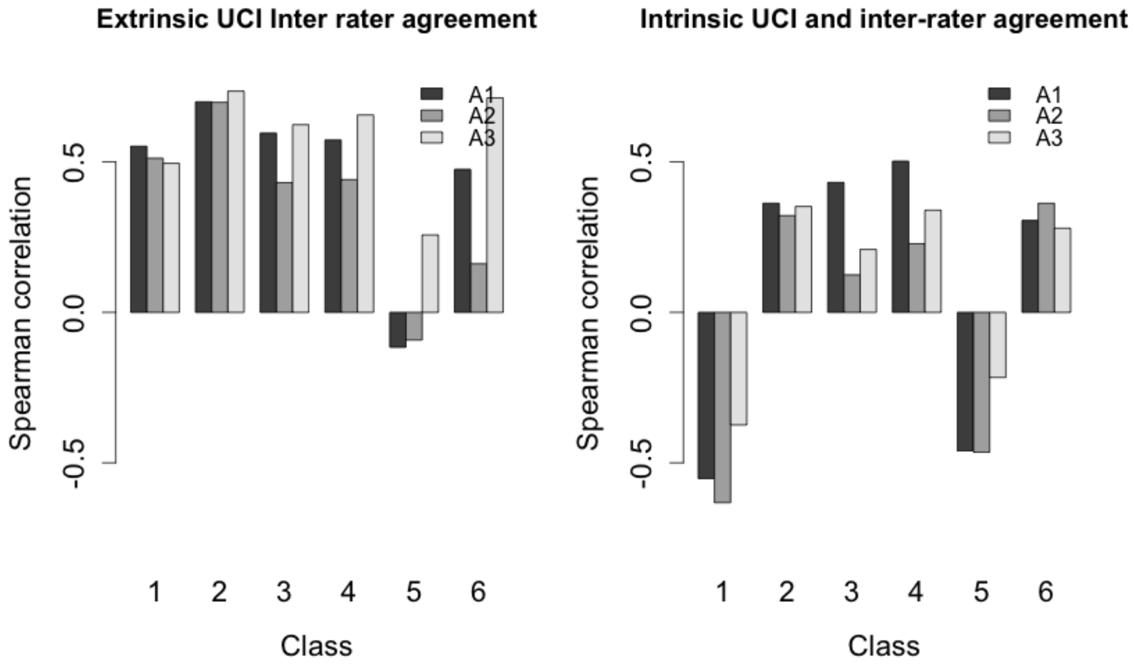


Figure 5.1: Correlation between UCI and annotators

5.3.2 Correlation between UMass and manual evaluation

Figure 5.2 shows that the correlation between annotators' rates and UMass always go in the same direction class-wise. This tendency for agreement is confirmed by the positive values of *Kappa*.

As well as the intrinsic UCI, intrinsic UMass scores show the same disagreements between annotators and the automatic measure for classes 1 and 5. This indicates that all annotators had trouble to interpret those topics.

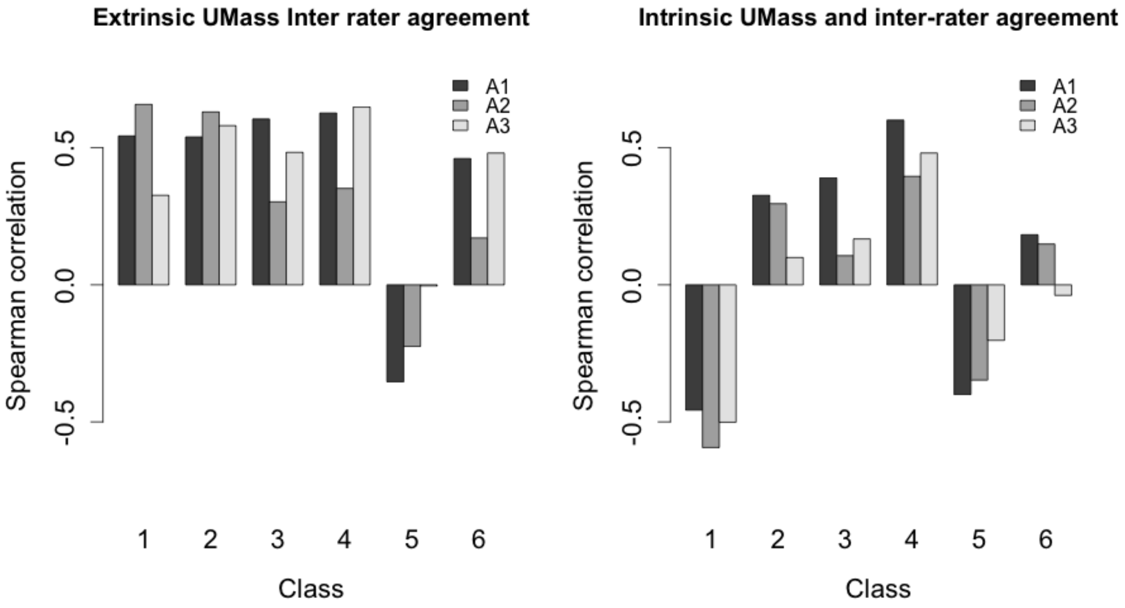


Figure 5.2: Correlation between UMass and annotators

5.4 Discussion

It is important to note that the authors who proposed UCI [41] and UMass [43] have tested their results against datasets with well-structured English text, such as news and academic papers. In this work, we faced a variety of texts from Facebook posts which are usually short and not necessarily well written or structured.

When comparing UCI and UMass scores we don't see significant differences, both presented similar results in all classes. However, in this particular case, extrinsic scores presented better correlation with manual evaluation than intrinsic scores. Intrinsic UMass and intrinsic UCI had trouble with *Class 1*, while *Class 5* was troublesome in all scenarios. For these classes in particular, intrinsic and extrinsic scores using UCI or UMass performed poorly in comparison to manual evaluation.

We can hypothesize some possible explanations for this *Class 5*. As explained in Section 5.1, *Class 5* represents Facebook posts from pages related to viral content and 'memes'. The poor agreement between human and automated scores could be explained by the lack of textual description on shared pictures and videos, but more exploratory analysis should be made in that area to confirm this hypothesis. Deeper analysis should be done to understand why *Class 1* performed so bad.

Regarding extrinsic coherence evaluation, an important aspect noted in this experiment was that we need to careful choose the dataset used as the external reference. In Chapter 4 we chose Wikipedia as the external reference. In that case, it was a reasonable choice, mainly because topics covered in the 20 Newsgroup dataset are old and very likely to be covered by an encyclopedia. On the other hand, content published on social networks like Twitter and Facebook are, by nature, mainly related to recent events. Some topics raised by the experiment carried out in this chapter are so fresh and time sensitive that we should not expect to find them in an encyclopedia. This raises a reasonable doubt if a source like Wikipedia was the best choice for this particular case. We believe that further research should be done to explore extrinsic evaluation and different reference corpus applied to social networks data streams.

5.5 Summary

In this chapter, the goal was to assess the compliance of the implemented coherence measures with external human annotators. We accomplished the task and we were able to analyze the correlation between humans and the automatic measures. We found that UCI performed better for this particular use case, but that doesn't necessarily mean that one measure is better than another. As mentioned in Chapter 3, they reflect different interpretability aspects of the topic.

Chapter 6

Applications and Prototypes

In this chapter, we present two applications. The first was develop to assist our analysis during the experiments carried in Chapter 4. The second application applies the results from the experiment demonstrated in Chapter 5 and proposes a combination of Social Network Analysis and Topic Modeling.

Many authors have already proposed different and innovative ways to visually explore topics, [66], [67], [68] and [69] are among the most relevant works. In this section, we present two prototypes implemented during the work of this thesis. Each of them focused on different aspects of topic coherence analysis and its application. The first is a user interface to visually interact with different topic models learned using the 20 Newsgroup dataset described in Chapter 4. The second user interface explores an application combining social network analysis and topics learned from the Facebook Posts dataset discussed in Chapter 5.

6.1 Topics Coherence Explorer

To explore results of the experiments carried in Chapter 4 we implemented a web user interface. We ran a series of different experiments with LDA using a range of a number of topics on the 20 newsgroup dataset previously described in Chapter 4. All these tests generated hundreds of topics and a proper way to browse and compare the performance of the implemented coherence measures was necessary. We propose in this section a user interface with that in mind. Our work here focused mainly on the aspect of topic coherence. We put the effort to produce a usable interface where we could see how the implemented coherence measures performed in our experiments. Figure 6.1 shows how the data flows through the components, from loading the data set until be presented to the user. Each of these components is independent and corresponds to a self-contained script. This way they can be maintained separately. For a comprehensive list of libraries and tools used to build this prototype see Appendix A.

Figure 6.2 presents the interface implemented to explore the topics learned in the experiments

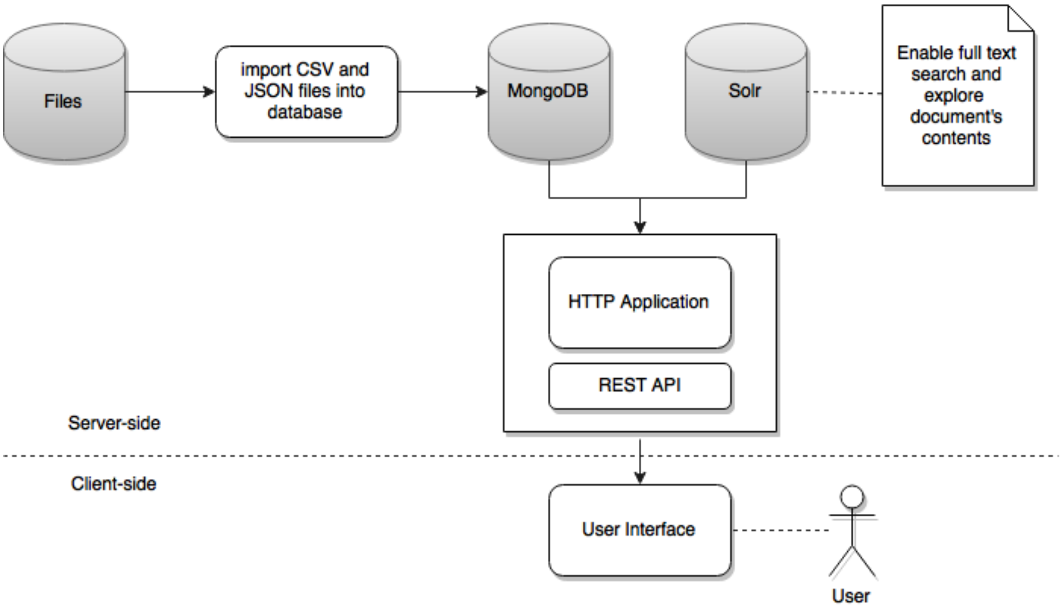
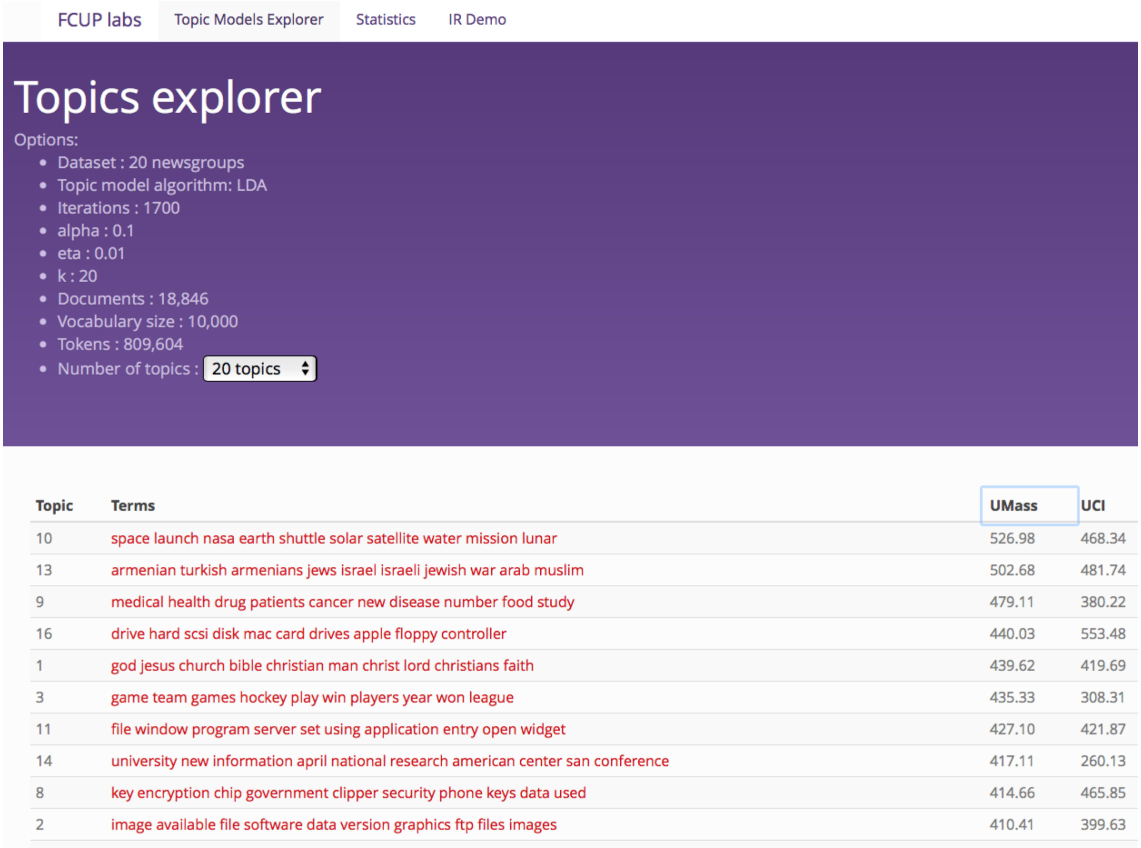

Files
import CSV and JSON files into database
MongoDB
Solr
Enable full text search and explore document's contents
HTTP Application
REST API
Server-side
Client-side
User Interface
User

Figure 6.1: Topics coherence explorer architecture

and its coherence scores. It is possible to sort the topics by UCI and UMass scores just clicking on the respective measure on table's header. With the interface is possible to compare topics that are easy to interpret and topics that are not.

Figure 6.3 is a screenshot of the user interface. On the top, it presents the parameters like the number of topics, as well as the alpha and beta value to run LDA. It also shows some characteristics of the dataset like the number of tokens and size of the vocabulary, it is also possible to change the number of topics and explore different granularity ranging from 5 to 100 topics. The user interface demonstrated on Figure 6.4 was designed to explore the documents and its contents. The web application is backed by a search engine to navigate through the document collection.



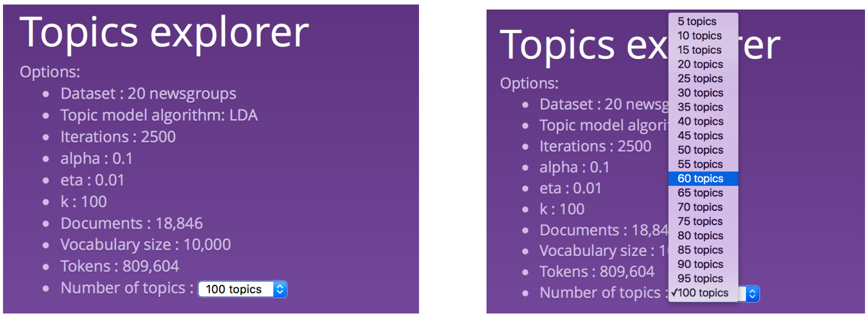
Topics explorer

Options:

- Dataset : 20 newsgroups
- Topic model algorithm: LDA
- Iterations : 1700
- alpha : 0.1
- eta : 0.01
- k : 20
- Documents : 18,846
- Vocabulary size : 10,000
- Tokens : 809,604
- Number of topics : 20 topics

Topic	Terms	UMass	UCI
10	space launch nasa earth shuttle solar satellite water mission lunar	526.98	468.34
13	armenian turkish armenians jews israel israeli jewish war arab muslim	502.68	481.74
9	medical health drug patients cancer new disease number food study	479.11	380.22
16	drive hard scsi disk mac card drives apple floppy controller	440.03	553.48
1	god jesus church bible christian man christ lord christians faith	439.62	419.69
3	game team games hockey play win players year won league	435.33	308.31
11	file window program server set using application entry open widget	427.10	421.87
14	university new information april national research american center san conference	417.11	260.13
8	key encryption chip government clipper security phone keys data used	414.66	465.85
2	image available file software data version graphics ftp files images	410.41	399.63

Figure 6.2: User interface to explore topics and coherence scores



Topics explorer

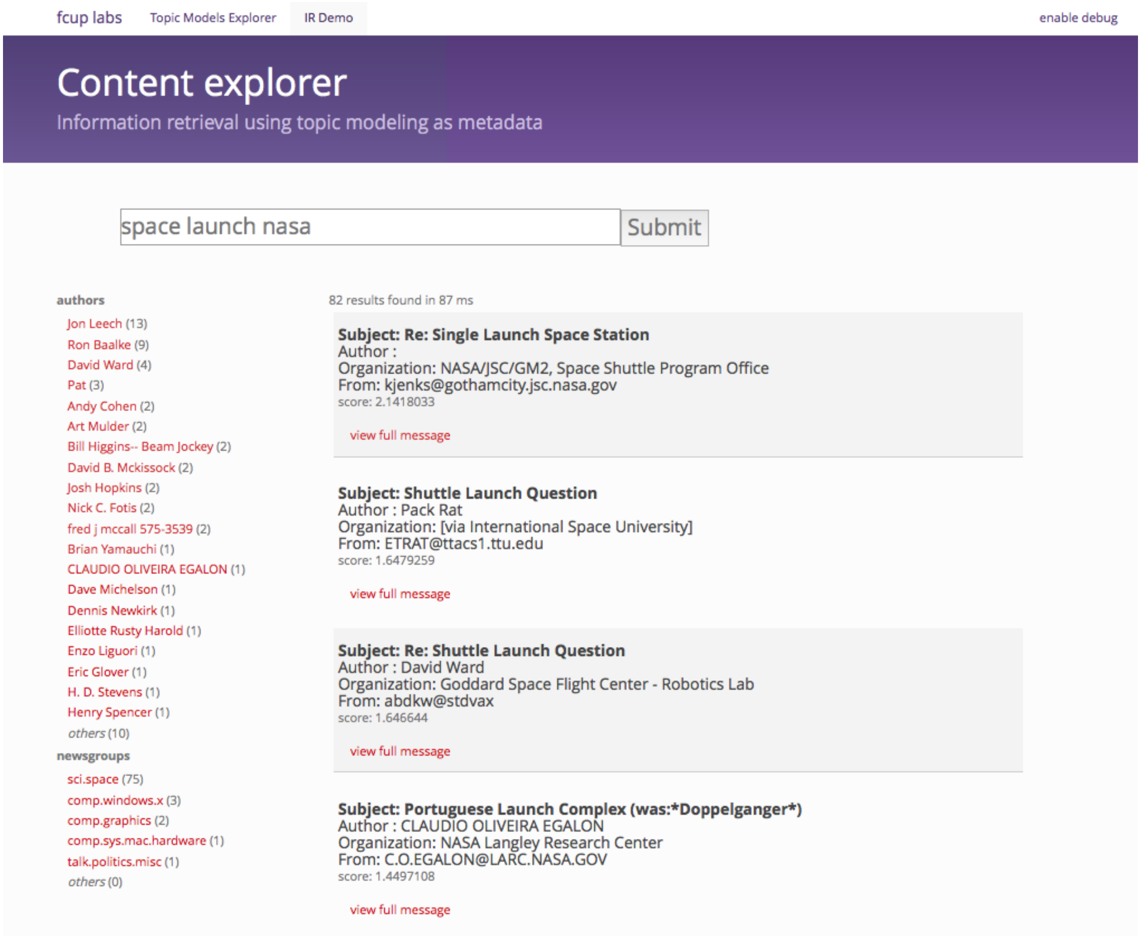
Options:

- Dataset : 20 newsgroups
- Topic model algorithm: LDA
- Iterations : 2500
- alpha : 0.1
- eta : 0.01
- k : 100
- Documents : 18,846
- Vocabulary size : 10,000
- Tokens : 809,604
- Number of topics : 100 topics

(a) Detailed parameters

(b) Changing number of topics

Figure 6.3: Topics explorer interface's header



Content explorer

Information retrieval using topic modeling as metadata

space launch nasa

authors

- Jon Leech (13)
- Ron Baalke (9)
- David Ward (4)
- Pat (3)
- Andy Cohen (2)
- Art Mulder (2)
- Bill Higgins-- Beam Jockey (2)
- David B. Mckissock (2)
- Josh Hopkins (2)
- Nick C. Fotis (2)
- fred j mccall 575-3539 (2)
- Brian Yamauchi (1)
- CLAUDIO OLIVEIRA EGALON (1)
- Dave Michelson (1)
- Dennis Newkirk (1)
- Elliotte Rusty Harold (1)
- Enzo Liguori (1)
- Eric Glover (1)
- H. D. Stevens (1)
- Henry Spencer (1)
- *others* (10)

newsgroups

- sci.space (75)
- comp.windows.x (3)
- comp.graphics (2)
- comp.sys.mac.hardware (1)
- talk.politics.misc (1)
- *others* (0)

82 results found in 87 ms

Subject: Re: Single Launch Space Station
Author :
Organization: NASA/JSC/GM2, Space Shuttle Program Office
From: kjenks@gothamcity.jsc.nasa.gov
score: 2.1418033

view full message

Subject: Shuttle Launch Question
Author : Pack Rat
Organization: [via International Space University]
From: ETRAT@ttacs1.ttu.edu
score: 1.6479259

view full message

Subject: Re: Shuttle Launch Question
Author : David Ward
Organization: Goddard Space Flight Center - Robotics Lab
From: abdkw@stdvax
score: 1.646644

view full message

Subject: Portuguese Launch Complex (was:*Doppelganger*)
Author : CLAUDIO OLIVEIRA EGALON
Organization: NASA Langley Research Center
From: C.O.EGALON@LARC.NASA.GOV
score: 1.4497108

view full message

Figure 6.4: User interface to explore documents

6.2 Social Network Analysis and Topics Visualization

Social network analysis (SNA) is the study of mathematical models for interactions among people, organizations, and groups [70]. With the availability of large data sets of human interactions, the popularity of services like Facebook and Twitter, there has been growing interest in social network analysis. Although the research in this field has a long history, it gained momentum after 9/11 hijackers, even more after social movements worldwide related to the financial crisis in 2008 and later the Arab Spring. We have covered in detail the motivations in this area in Chapter 5.

In Chapter 5 we applied topic modeling on Facebook posts in Portuguese related to political movements. Our objective with that experiment was to assess the compliance of the automatic evaluation measures with the external human evaluation. Once we have assessed the reliability of such measures through empirical experiments we developed a user interface prototype that combines SNA with topic modeling.

The prototype presented in this section combines social network analysis and the richness of the language content of the interactions using topic modeling, namely the topics discovered during the experiments in Chapter 5. Using Netvizz [65] and Gephi [71], we generate the network of relationships between the 36 selected pages described in Chapter 5. The overview architecture can be seen in Figure 6.5. See Appendix A.8 for more detail about Gephi. The interactive version is available at https://arianpasquali.github.io/sna-topicmodeling-facebook-pages. For a comprehensive list of libraries and tools used to build this prototype see Appendix A.

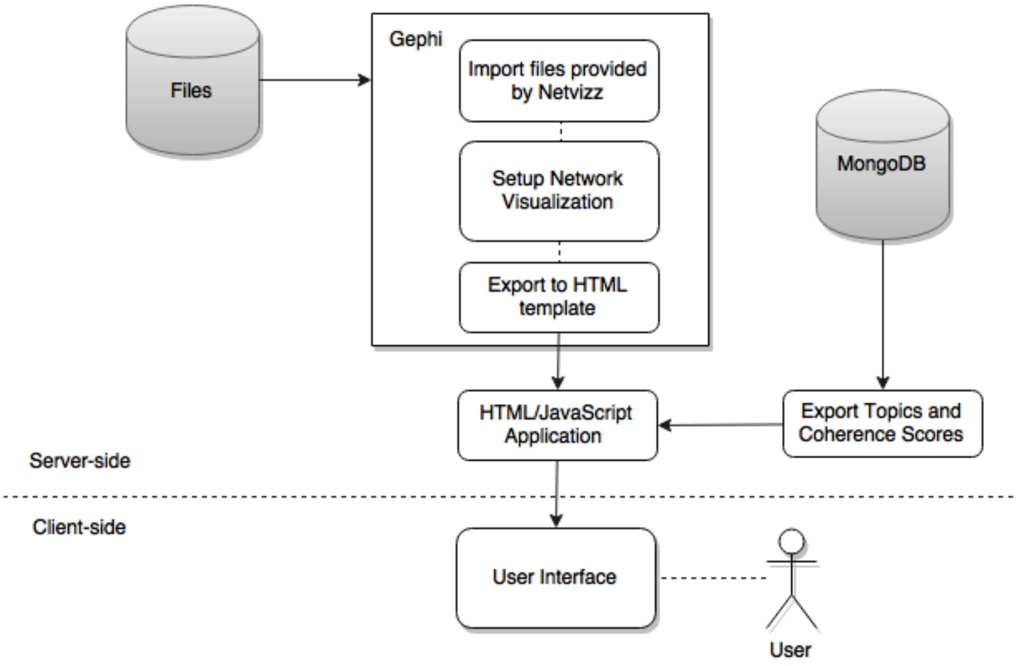
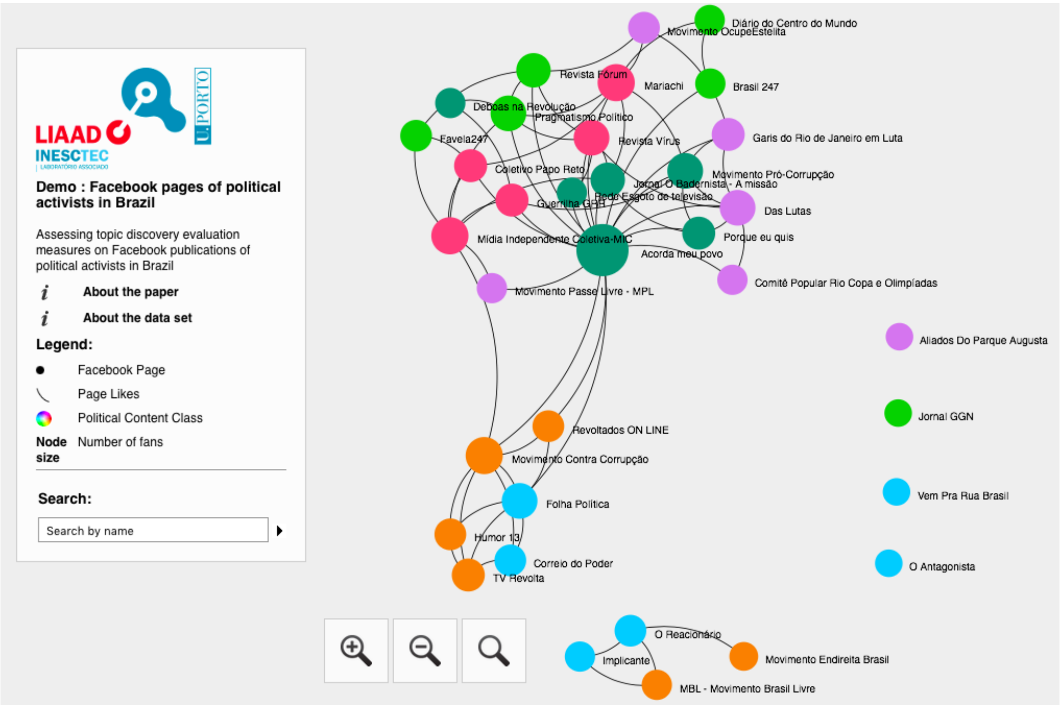

Files
Gephi
Import files provided by Netvizz
Setup Network Visualization
Export to HTML template
MongoDB
HTML/JavaScript Application
Export Topics and Coherence Scores
Server-side
Client-side
User Interface
User

Figure 6.5: Network and Topics visualization architecture

This network represents the connections between the pages in terms of "Page Like" - if a Facebook Page profile "likes" another Facebook Page, we consider that an interaction. In Facebook, the action of "like" is analogous to the action of "follow", so that is an important aspect that tells ous how much one Facebook Page profile is interested on another. Our social network representation considers Facebook Pages as *nodes* and "pages likes" as *edges*. Our use case didn't take into consideration the direction of the interactions. The colors represent the 6 political orientation described in Chapter 5 Section 5.1. The user interface can be seen in Figure 6.6. The size of each node is represented by its number of fans from the respective page.

The user interface is interactive and by selecting a single node is possible to see its details in the panel located on the right side of the screen. As we can see in Figure 6.7, page name, political orientation, facebook page link and the topics that received the higher scores using the coherence measures. The selected node at Figure 6.7 represents the Facebook Page called *"Midia Indipendente Coletiva"*, an influent group of journalists. According to its interactions we can see that this page has connections with other pages considered progressist or leftist in terms of political orientation, which is perfectly expected from the Social Science perspective.



Demo : Facebook pages of political activists in Brazil

Assessing topic discovery evaluation measures on Facebook publications of political activists in Brazil

i **About the paper**

i **About the data set**

Legend:

● Facebook Page

Page Likes

Political Content Class

Node size Number of fans

Search:

Search by name

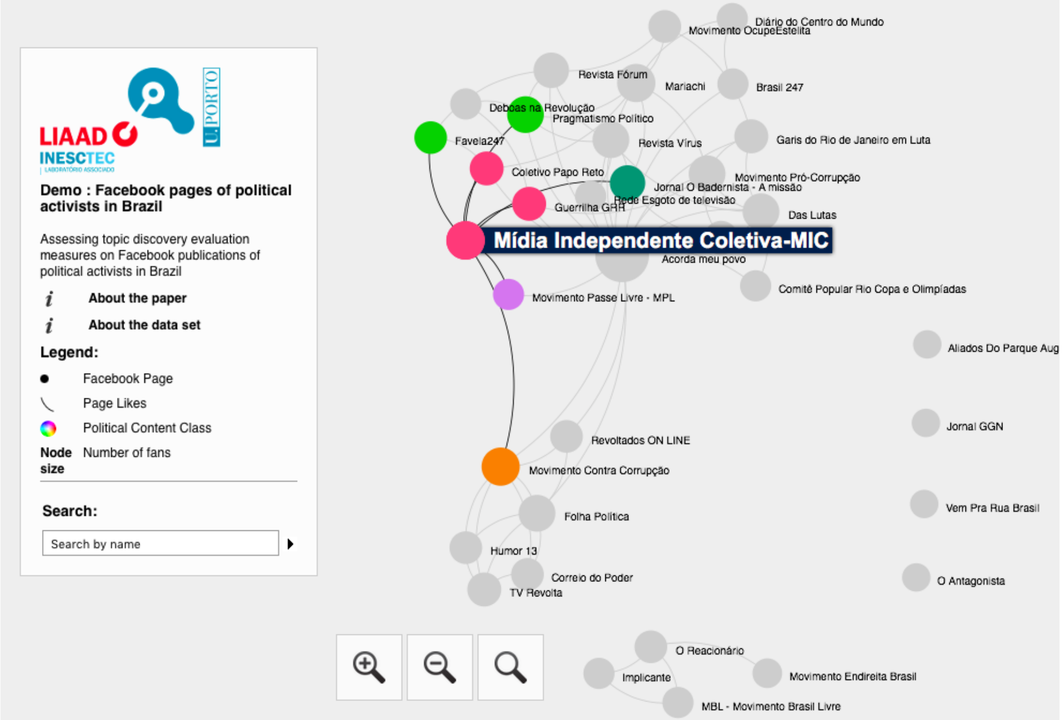
Figure 6.6: Network of Facebook Pages relationships

It is interesting to note that leftists pages are very connected while conservatives tend to be more isolated according to Figure 6.6 (check Table 6.1 for color legends). The nodes represented in blue and orange are on the bottom of the graph and have less interconnections when compared to other classes of political orientation. This points to a curious social behavior and deserves deeper analysis from a Political Social Scientist.

Although Social Network Analysis gave us interesting insights further analysis can be done combining it with Topic Modeling. For instance, it would be interesting to browse the topics that these pages are discussing. Using the topics extracted in Chapter 5 we list the top-n topics in the interface when a particular node is selected, see Figure 6.8. In this use case, we show the top 7 topics according to the UCI measure and we ignored the low scored topics.

Topics listed in the picture elucidate the nature of debate in those pages. The most coherent topics learned regarding pages from that political orientation highlight themes like human rights, women rights, labor day, prisoner rights and etc. From the Social Science perspective, these topics say a lot about concerns raised by leftists and progressists pages in our study.

This prototype served as an experiment to explore pages relationship and its linguistic content, further research has to take place to provide deeper and innovative solutions in topics and social data visualization.



Demo : Facebook pages of political activists in Brazil

Assessing topic discovery evaluation measures on Facebook publications of political activists in Brazil

i About the paper

i About the data set

Legend:

●	Facebook Page
╲	Page Likes
◍	Political Content Class
Node size	Number of fans

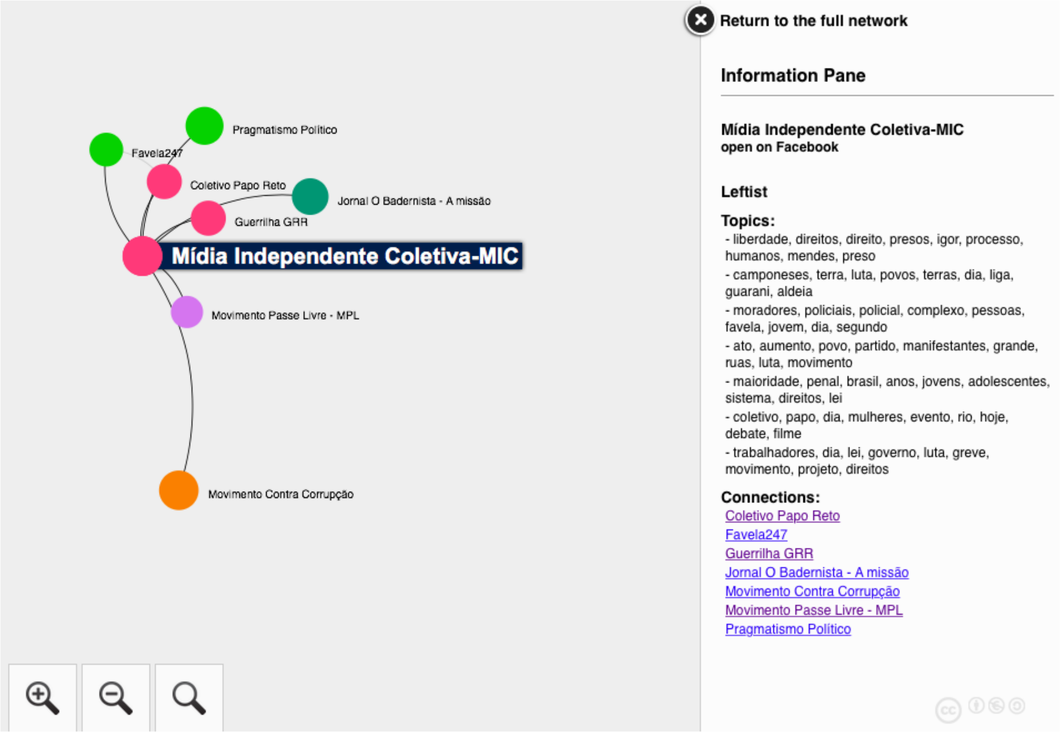
Search:

Search by name ▸

Figure 6.7: Page selected and its relationships

Table 6.1: Page classes legend

Class	*Political orientation*	*Color*
1	Particular cause (Social Movement)	🟪
2	Grassroots news (Leftist)	🟥
3	Pro-Governism news (Center)	🟩
4	Pro-impeachment news (Rightist)	🟦
5	Pro-impeachment virals (Rightist)	🟧
6	Progressist virals	🟩


Pragmatismo Politico
Favela247
Coletivo Papo Reto
Jornal O Badernista - A missão
Guerrilha GRR
Mídia Independente Coletiva-MIC
Movimento Passe Livre - MPL
Movimento Contra Corrupção

Return to the full network

Information Pane

Mídia Independente Coletiva-MIC
open on Facebook

Leftist

Topics:
- liberdade, direitos, direito, presos, igor, processo, humanos, mendes, preso
- camponeses, terra, luta, povos, terras, dia, liga, guarani, aldeia
- moradores, policiais, policial, complexo, pessoas, favela, jovem, dia, segundo
- ato, aumento, povo, partido, manifestantes, grande, ruas, luta, movimento
- maioridade, penal, brasil, anos, jovens, adolescentes, sistema, direitos, lei
- coletivo, papo, dia, mulheres, evento, rio, hoje, debate, filme
- trabalhadores, dia, lei, governo, luta, greve, movimento, projeto, direitos

Connections:
Coletivo Papo Reto
Favela247
Guerrilha GRR
Jornal O Badernista - A missão
Movimento Contra Corrupção
Movimento Passe Livre - MPL
Pragmatismo Político

Figure 6.8: Network and topics visualization

6.3 Summary

This chapter presented an overview of the data processing pipeline implemented to process the topics and calculate their coherence, along with two visualization tools. The first, a topic coherence explorer that assisted us to analyze results from the experience in Chapter 4. The second prototype focused on exploring social network relationships among the Facebook Pages used in the experiment carried in Chapter 5. This prototype is available at https://arianpasquali.github.io/sna-topicmodeling-facebook-pages. Future versions could focus on others features like supporting uploading different datasets, better visualization of topics words distributions and topic document assignments for example.

Chapter 7

Conclusion and Future Work

The main goal of this thesis was to assess the effectiveness of automatic coherence evaluation measures applied to topic models. We covered implementation aspects and experiments to reproduce findings from related works [42] [43]. We carried out experiments to validate their compliance against external human annotators and we confirmed that the proposed measures could produce interesting results when compared to manual human evaluation. Although the results were not perfect, these measures represent a important tool in the field.

We conclude that UMass and UCI produce similar scores and similar performance. The most relevant differences in terms of correlation with human annotations and performance are related to automatic extrinsic and intrinsic variations. Ideally, both techniques should be used together to have a good picture of the model in terms of coherence because they reflect different aspects of interpretability and tend to produce different results.

However, if we take into consideration computational performance and resources, our implementation of intrinsic measures was up to 7 times faster than its extrinsic counterpart. Mainly because of disk usage and the difference in size of the Wikipedia indexes in our training indexes, as discussed in Chapter 4. We summarize in Table 7.1 our conclusions after the experiments with the 20 newsgroup dataset and the Facebook dataset.

Regarding extrinsic coherence evaluation alone, we need to choose the dataset used as an external reference carefully. In Chapter 4 we chose Wikipedia. In that case, it was a reasonable choice, mainly because topics covered in the 20 Newsgroup dataset were old and very likely to be covered by an encyclopedia. In contrast, content published on social networks like Twitter and Facebook are, by nature, mainly related to recent events. Some topics raised by the experiment in Chapter 5 are so fresh and time sensitive that we should not expect to find them in an encyclopedia. We believe that further research should be done to explore extrinsic evaluation and different reference corpus applied to social networks data streams. This can be seen as a research gap since we couldn't find in the literature anything related to this issue.

In Chapter 6 we implemented two applications. It is worth to mention the application that

Table 7.1: Lessons learned

Coherence Type	Pros	Cons
Intrinsic	- Tends to use less computational resources since we only need to build the index for the training dataset. - In our experiments, intrinsic evaluations were about 7 times faster than extrinsic evaluation	- Original dataset might not be available.
Extrinsic	- Using a general purpose reference like Wikipedia we can develop generic purpose coherence evaluation service for different domains. - We don't need access to the training dataset.	- Tends to use more computational resources since it requires an specific index. English Wikipedia alone demands dozens of gigabytes of disk space. - In our experiments, extrinsic evaluations were about 7 times slower than intrinsic evaluation

combined topic models and social network analysis where users can browse Facebook Pages, their relations, and respective topics (see Chapter 6 Section 6.2). In that prototype, we applied a threshold to ignore topics that could represent noise and poor labels, showing to the user only topics with high coherence score.

Future work

The results of this work give us the confidence to investigate other applications of automatic coherence evaluation. The scope of the current work can be extended in several other directions in the future. For instance optimization, the code provided in this work is not production-ready and optimization regarding its performance is necessary.

Another interesting path is to study how coherence measures could be applied to help summarization methods to produce and evaluate real document summaries. Evaluating not only the coherence among isolated terms, but whole sentences to produce complete and coherent texts.

Alternative smoothing methods like the ones described by Chen and Goodman [48] could also be explored in the context of this work. Although some authors have addressed this subject [44], there is still room for further investigation to understand how different smoothing methods could impact coherence scores.

Automatic coherence evaluation measures like those presented in this work represent important progress in the field of text mining. This kind of evaluation is an important one to text miners have in their toolkit. We hope to see improvements in applications that make use of topic models thanks to automatic evaluation in terms of semantic coherence from unsupervised learning methods like topic models and clustering.

Appendices

Appendix A

Used Tools and Libraries

In order to build our implementation, we applied several technologies. The following list briefly introduces the most relevant ones for this particular work.

A.1 Python

Python is a high-level, general-purpose and interpreted programming language. It supports multiple programming paradigms, including object-oriented, imperative, functional or procedural styles. It was the language of choice because it supports different programming paradigms and features a large standard library, especially easy-to-use data mining libraries such as scikit-learn [72] and NLTK [73]. It has a special license based on BSD and GNU, but with no Copyleft, all the details of the Python Software Foundation License (PSFL) [1]. Its official website can be found at https://www.python.org.

A.2 Scikit-learn

Scikit-learn is an open source machine learning library written in Python [72]. It provides features such as classification, regression and clustering algorithms including support vector machines, random forests, k-means and etc. It is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. We used this library mainly for text preprocessing. Its source code is available at https://github.com/scikit-learn/scikit-learn and it is distributed under the 3-Clause BSD license [2].

[1] https://docs.python.org/3/license.html

[2] https://opensource.org/licenses/BSD-3-Clause

A.3 NLTK

The Natural Language Toolkit (NLTK) [73], is a suite of libraries and programs for statistical natural language processing (NLP) written in the Python programming language. The source code is distributed under the terms of the Apache License Version 2.0. The source code is available at https://github.com/nltk/nltk. This particular package was used in this work especially to implement text pre-processing tasks such as tokenization and stopword removal.

A.4 Allen Riddell's LDA Python implementation

Allen B. Riddell is an Assistant Professor in the School of Informatics and Computing at Indiana University Bloomington. His LDA implementation is used in this work and implements Latent Dirichlet Allocation (LDA) [27] using collapsed Gibbs sampling proposed by [74]. The library was chosen because is easy-to-use and follows conventions found in Scikit-learn library [72]. The source code can be found at https://github.com/ariddell/lda/ and it is licensed under Version 2.0 of the Mozilla Public License [3]. The author's website can be found at https://ariddell.org.

A.5 Elasticsearch

Elasticsearch is an open source search engine. Its major features include full-text search and HTTP web interface. Elasticsearch's REST API makes it usable from most popular programming languages including Java and Python. It was used to build our local full-text index for Wikipedia-PT, Wikipedia-EN, as well as the index for our Facebook Posts and 20 Newsgroups datasets. The official source repository can be accessed at https://github.com/elastic/elasticsearch. It is licensed under Apache License version 2.0 [4].

A.6 Netvizz

Netvizz 1.25 [65] is an application developed by Bernhard Rieder to collect data from Facebook. It provides features to extract information from pages, posts and comments and Page Like network. This application was used to collect and build our Facebook Posts dataset mentioned in Chapter 5. It is licensed under GNU license [5]. The application is available using Firefox or Chrome at https://apps.facebook.com/netvizz/.

[3] https://opensource.org/licenses/MPL-2.0

[4] https://www.apache.org/licenses/LICENSE-2.0

[5] https://www.gnu.org/licenses/agpl-3.0.html

A.7 Python Flask

Flask is a micro-framework that helps to implement simple web applications. To provide some interactivity with a database where we stored the topics we needed some server-side code. The library of choice was Python Flask since the majority of the code used in this thesis used Python language. Flask is licensed under BSD license [6]. Documentation can be accessed at http://flask.pocoo.org.

A.8 Gephi

Gephi is an open-source platform for visualizing and manipulating graphs. Gephi was the tool of choice to handle the graph files collected by Netvizz. It provides visualization and exploration features for graphs and social networks analysis [71]. It is open-source, free can be downloaded at https://gephi.org. The main source code is distributed under the dual license CDDL 1.0 and GNU General Public License v3[7].

A.9 Gephi Sigmajs Exporter Plugin

Sigmajs Exporter Plugin exports the network from Gephi to a predefined HTML interactive template. Choose to include search, group selection, explanatory text, etc. without having to do any HTML/JavaScript coding. Upload the output to any webserver and enjoy a rich HTML5 interactive visualization of your network. Available at https://marketplace.gephi.org/plugin/sigmajs-exporter/.

[6] https://opensource.org/licenses/BSD-3-Clause

[7] https://www.gnu.org/licenses/agpl-3.0.html

Appendix B

Code listings

This section lists some relevant code implemented for this work. The complete source code is available at https://github.com/arianpasquali/msc-thesis-code.

```
class UCI(TopicCoherence):

  def fit(self, words):
    self.words = words

    for word_i in self.words:
        for word_j in self.words:
            if(word_i is not word_j):
                pairwise_key = "_".join(sorted([word_i,word_j]))
                if(pairwise_key not in self.pairwise_probability.keys()):
                  self.pairwise_probability = 0
                    self.pairwise.append(pairwise_key)

    pool = ThreadPool(N_CPUS)
    pool.map(self.compute_word_hits, self.words)
    pool.map(self.compute_pairwise_hits, self.pairwise)

    return np.mean(self.coherence_scores.values())

  def compute_pairwise_hits(self,pairwise_key):
    word_i = pairwise_key.split("_")[0]
    word_j = pairwise_key.split("_")[1]

    self.pairwise_hits[pairwise_key] =
      self.get_hit_count_for_terms(self.index_name,"text",[word_i,word_j])

    self.pairwise_probability[pairwise_key] =
      self.compute_probability(self.pairwise_hits[pairwise_key])

    self.coherence_scores[pairwise_key] = self.compute_coherence(
                                                       self.pairwise_probability[pairwise_key],
                                                       self.word_probability[word_i],
                                                       self.word_probability[word_j])

  def compute_coherence(self, prob_ngrams, prob_ngram_a, prob_ngram_b):
    1 if prob_ngram_a == 0 else prob_ngram_a
    1 if prob_ngram_b == 0 else prob_ngram_b

    prob_product = (prob_ngrams + self.epsilon) / (prob_ngram_a * prob_ngram_b)
    1 if prob_product == 0.0 else prob_product

    return math.log(prob_product)



  def __init__(self, index_name, doc_type="page",es_address="localhost:9200"):
      super(UCI,self).__init__(index_name, doc_type,es_address)
```

Listing B.1: UCI Coherence code

```

class UMass(TopicCoherence):

  def fit(self, words):
    self.words = words

    pool = ThreadPool(N_CPUS)
    pool.map(self.compute_word_hits, self.words)

    for word_i in self.words:

      sorted_desc = sorted(self.word_hits.items(), key=operator.itemgetter(1),reverse=True)
      sorted_asc = sorted(self.word_hits.items(), key=operator.itemgetter(1))

      for most_common in sorted_desc:
        most_common_ngram = most_common[0]
        most_common_hits = most_common[1]

        for most_rare in sorted_asc:
          most_rare_ngram = most_rare[0]
          most_rare_hits = most_rare[1]

          if(most_common_ngram is not most_rare_ngram):
            if(most_rare_hits < most_common_hits):
              pairwise_key = most_rare_ngram + "_" + most_common_ngram

              if(pairwise_key not in self.pairwise_probability.keys()):
                self.pairwise_probability = 0
                self.pairwise[pairwise_key]= {
                  "most_rare_ngram":most_rare_ngram,
                  "most_rare_hits":most_rare_hits,
                  "most_common_ngram":most_common_ngram,
                  "most_common_hits":most_common_hits,
                  }

    pool.map(self.compute_pairwise_hits, self.pairwise.keys())

    return np.mean(self.coherence_scores.values())

  def compute_pairwise_hits(self,pairwise_key):

    most_rare_hits = self.pairwise[pairwise_key]["most_rare_hits"]
    most_rare_ngram = self.pairwise[pairwise_key]["most_rare_ngram"]
    most_common_hits = self.pairwise[pairwise_key]["most_common_hits"]
    most_common_ngram = self.pairwise[pairwise_key]["most_common_ngram"]

    self.pairwise_hits[pairwise_key] = self.get_hit_count_for_terms(self.index_name,
                                 "text",
                                 [most_rare_ngram,most_common_ngram])

    self.pairwise_probability[pairwise_key] =
      self.compute_probability(self.pairwise_hits[pairwise_key])

    self.word_probability[most_rare_ngram] =
      self.compute_probability(most_rare_hits)

    self.word_probability[most_common_ngram] =
      self.compute_probability(most_common_hits)

    self.coherence_scores[pairwise_key] = self.compute_coherence(
                        self.pairwise_probability[pairwise_key],
                        self.word_probability[most_common_ngram],
                        self.word_probability[most_rare_ngram])

  def compute_coherence(self, prob_ngrams, prob_ngram_a, prob_ngram_b):
    1 if prob_ngram_a == 0 else prob_ngram_a
    1 if prob_ngram_b == 0 else prob_ngram_b

    prob_product = (prob_ngrams + self.epsilon) / (prob_ngram_a)
    1 if prob_product == 0.0 else prob_product

    return math.log(prob_product)

  def __init__(self, index_name, doc_type="page",es_address="localhost:9200"):
      super(UMass,self).__init__(index_name, doc_type,es_address)
```

Listing B.2: UMass Coherence code

```python

from __future__ import division
import math
import operator
from scipy import stats
from pathos.pools import ProcessPool, ThreadPool
from elasticsearch import Elasticsearch
from sklearn import preprocessing

class TopicCoherence(object):

  epsilon = 1

  def __init__(self, index_name, doc_type="page",es_address="localhost:9200"):
    self.index_name = index_name
    self.doc_type = doc_type

    self.es = Elasticsearch(es_address)
    self.collection_size = self.get_collection_size(index_name, doc_type)

  def get_collection_size(self,index_name, _doc_type):
    res = self.es.search(index=index_name,
                           doc_type=_doc_type,
                           body={"query": {"match_all": {}}})

    return res['hits']['total']

  def get_hit_count_for_terms(self, es_index_name, field, terms):
    must_query = []
    exact_terms = ["\"" + x + "\"" for x in terms]
    lucene_query = " AND ".join(exact_terms)

    res = self.es.search(index=es_index_name,
                         q=lucene_query,
                         doc_type=self.doc_type)

    return res['hits']['total']

  def compute_probability(self, hits):
    0 if hits is None else hits
    return float(hits) / float(self.collection_size)

  def compute_word_hits(self,word):
    self.word_hits[word] =
      self.get_hit_count_for_terms(self.index_name,"text",[word])

    self.word_probability[word] =
      self.compute_probability(self.word_hits[word])

  @abstractmethod
  def compute_pairwise_hits(self,pairwise_key): pass

  @abstractmethod
  def compute_coherence(self, prob_ngrams, prob_ngram_a, prob_ngram_b): pass

  @abstractmethod
  def fit(self, words): pass

  @staticmethod
  def normalize(scores):
    scores = np.array(scores)
    scores = scores.reshape(1,-1)

    min_max_scaler = preprocessing.Normalizer()
    normalized_scores = min_max_scaler.fit_transform(scores)

    return normalized_scores[0].tolist()

  @staticmethod
  def entropy(scores):
    return stats.entropy(scores)
```

Listing B.3: Topic Coherence abstract class code

Appendix C

Facebook Experiment Results

C.1 Automatic Topic Coherence Scores

This section lists in detail all topics learned at the experiment described in Chapter 5. Topics are inversely ordered by Extrinsic UMass coherence score.

Table C.1: Topics from Class 1: Particular cause (Social Movement)

Id	Topics	Extrinsic UMass	Extrinsic UCI
2	garis, greve, trabalhadores, luta, comlurb, sindicato, rio, gari, chapa	2.950	2.931
9	apoio, moradores, prefeitura, vila, luta, hoje, solidariedade, novas, praia	0.706	0.239
11	parque, pic, nic, circulo, dia, poder, cidade, gente, podemos	0.573	1.087
8	movimento, dia, coletivo, popular, movimentos, rede, social, luta, coletiva	0.373	0.145
3	transporte, aumento, copa, movimento, mundo, governo, passe, livre, tarifa	0.366	0.298
12	rio, vila, moradores, prefeitura, projeto, comunidade, prefeito, eduardo, copa	0.074	-0.289
13	parque, augusta, cidade, municipal, prefeitura, dia, luta, rua, guarda	-0.027	-0.108
5	ato, dia, policiais, rio, pessoas, protesto, frente, apoio, rua	-0.130	-0.070
4	povo, negro, marcha, reaja, campanha, internacional, anos, luta, dia	-0.162	0.381
1	brasil, governo, povo, presidente, federal, direitos, direito, poder, caso	-0.404	-0.697
14	mulheres, pessoas, sociedade, forma, vida, mulher, nunca, grupo, homens	-0.528	-0.835
10	povo, anos, pior, pessoas, banco, hoje, dias, brasileiro, infelizmente	-0.779	-0.525
7	direitos, rio, dia, humanos, janeiro, ativistas, mil, caso, segundo	-0.883	-0.508
0	rio, esquerda, professor, janeiro, paulo, carlos, universidade, partir, centro	-0.905	-0.993
6	bem, pessoas, coisa, cidade, sempre, poder, fazendo, anos, bom	-1.223	-1.056

Table C.2: Topics from Class 2: Grassroots news (Leftist)

Id	Topics	Extrinsic Umass	Extrinsic UCI
0	liberdade, direitos, direito, presos, igor, processo, humanos, mendes, preso	2.121	1.427
5	camponeses, terra, luta, povos, terras, dia, liga, guarani, aldeia	1.089	1.051
14	trabalhadores, dia, lei, governo, luta, greve, movimento, projeto, direitos	0.710	0.534
6	moradores, policiais, policial, complexo, pessoas, favela, jovem, dia, segundo	0.662	0.971
8	ato, aumento, povo, partido, manifestantes, grande, ruas, luta, movimento	0.647	0.740
3	governo, federal, presidente, globo, cunha, deputado, segundo, eduardo, dinheiro	0.592	0.243
11	coletivo, papo, dia, mulheres, evento, rio, hoje, debate, filme	0.250	0.686
1	maioridade, penal, brasil, anos, jovens, adolescentes, sistema, direitos, lei	0.082	0.883
10	estudantes, dia, rio, professores, escola, universidade, professor, escolas, paulo	-0.160	-0.206
7	rio, rua, prefeitura, janeiro, revista, centro, ano, eduardo, frente	-0.708	-0.685
9	gente, pessoas, vida, nunca, sempre, tempo, bem, casa, mundo	-0.765	-1.066
4	disse, maria, mulher, sido, anos, dois, homem, policial, casa	-0.886	-0.757
2	nova, jornal, democracia, anos, dia, rio, popular, luta, apoio	-0.913	-0.658
12	brasil, forma, sempre, classe, poder, sociedade, governo, grande, fato	-1.137	-1.344
13	pessoas, mundo, menos, guerra, grande, brasil, grupo, cidade, grandes	-1.584	-1.820

Table C.3: Topics from Class 3: Pro-Governism news (Center)

Id	Topics	Extrinsic Umass	Extrinsic UCI
11	lava, luis, nassif, federal, presidente, jato, juiz, dinheiro, lula	2.353	2.027
10	governo, presidente, dilma, cunha, eduardo, federal, lei, processo, penal	0.881	0.594
7	dilma, direitos, brasil, partido, pena, lei, presidenta, direito, crime	0.717	0.966
9	governador, movimento, nacional, financiamento, reforma, paulo, professores, governo	0.593	0.695
0	povo, brasil, brasileiro, ruas, golpe, divida, brasileiros, auditoria, presidente	0.454	0.799
5	artigo, paulo, eduardo, carlos, entrevista, luiz, deputado, antonio, silva	0.414	-0.151
12	brasil, povo, governo, dilma, lula, militar, disse, verdade, guerra	0.162	0.188
6	anos, mulheres, jovens, redes, negros, releituras, direitos, brasil, universidade	-0.072	0.477
14	deus, pessoas, anos, homem, brasileiros, verdade, presente, identidade, pais	-0.140	-0.438
3	brasil, presidente, povo, dinheiro, lula, maior, partido, comunista, poder	-0.315	-0.241
2	rio, favelas, moradores, projeto, favela, zona, cidade, dia, cidades	-0.363	0.199
8	sempre, bem, vida, gente, dia, nunca, pessoas, coisa, tempo	-0.791	-0.987
1	forma, sociedade, poder, sistema, crise, grande, bem, maior, economia	-0.930	-1.023
13	mulheres, pessoas, mundo, deus, bem, vida, brasil, dia, filme	-1.106	-1.359
4	anos, globo, eua, grande, grupo, dias, maior, ano, governo	-1.857	-1.744

Table C.4: Topics from Class 4: Pro-impeachment news (Rightist)

Id	Topics	Extrinsic UMass	Extrinsic UCI
5	lava, folha, jato, lula, veja, moro, dilma, juiz, sergio	1.810	1.307
3	folha, dilma, sociais, redes, cunha, impeachment, folhapress, eduardo, stf	1.482	1.727
13	folha, sociais, redes, dilma, lava, ministro, stf, gilmar, contas	1.436	1.143
12	folha, lula, redes, filho, sociais, cpi, dilma, petrobras, campanha	0.887	0.848
6	folha, odebrecht, marcelo, paulo, propina, globo, campanha, vaccari, veja	0.709	1.129
11	dilma, lula, folha, veja, governo, quer, petista, partido, presidente	0.084	0.344
1	dilma, vem, impeachment, dia, presidente, rua, eduardo, processo, contas	-0.243	-0.022
0	brasil, eric, ano, balbinus, crise, movimento, governo, pior, dilma	-0.541	-0.957
7	brasil, lula, dilma, carta, sempre, homem, passo, bem, palavras	-0.560	-0.361
4	adicionar, brasil, lei, foto, boa, capa, bom, poder, direito	-0.582	-0.837
14	dilma, ministro, governo, presidente, rousseff, segundo, queda, caso, conta	-0.611	-0.369
8	brasil, rua, avenida, paulo, presidente, vargas, paulista, santo, carlos	-0.841	-0.920
9	dia, vem, rua, brasil, hoje, ruas, pessoas, hora, amigos	-0.925	-1.082
10	dinheiro, folha, governo, dilma, mil, anos, maior, conta, federal	-0.974	-0.764
2	frente, rio, matriz, prefeitura, santa, sul, centro, bandeira, parque	-1.132	-1.187

Table C.5: Topics from Class 5: Pro-impeachment virals (Rightist)

Id	Topics	Extrinsic Umass	Extrinsic UCI
0	movimento, veja, dilma, lula, folha, juventude, greve, caminhoneiros, convoca	1.829	1.723
6	impeachment, dia, ruas, apoio, mariana, camargo, mascarenhas, coragem, beatriz	0.979	0.715
5	impeachment, ruas, dia, beatriz, oliveira, mariana, camargo, cara, mascarenhas	0.846	0.598
8	dilma, lula, movimento, presidente, ministro, lava, governo, stf, jato	0.609	0.608
14	impeachment, dia, ruas, beatriz, oliveira, povo, mariana, camargo, mascarenhas	0.581	0.439
4	impeachment, camargo, mariana, povo, beatriz, oliveira, mascarenhas, brasil, ruas	0.432	0.374
3	impeachment, camargo, mariana, povo, brasileiros, beatriz, oliveira, ruas, dia	0.280	0.087
12	impeachment, ruas, beatriz, oliveira, povo, dia, hora, camargo, mariana	0.258	0.072
9	movimento, brasil, livre, ajude, mbl, impeachment, construir, luta, povo	-0.064	0.852
7	line, revoltados, reis, marcello, brasil, banco, deus, equipe, futuro	-0.250	-0.170
11	caso, governo, crise, compra, grave, ministro, brasil, ministros, dinheiro	-0.527	-0.904
13	governo, dinheiro, brasil, dilma, patricia, quer, pagar, pessoas, melo	-0.547	-0.499
10	povo, brasil, governo, chega, greve, apoio, pessoas, podemos, impostos	-0.848	-0.926
1	dia, brasil, frente, dilma, movimento, rio, paulo, mbl, congresso	-1.490	-0.397
2	governo, brasil, poder, povo, lei, partido, anos, pessoas, bem	-2.088	-2.571

Table C.6: Topics from Class 6: Progressist virals

Id	Topics	Extrinsic Umass	Extrinsic UCI
5	cara, gente, muita, dinheiro, oque, coisa, foto, mal, fazendo	1.956	1.699
12	povo, esquerda, jornal, bem, badernista, acorda, gente, fim, deixar	1.046	0.029
0	luta, coletivo, estudantes, apoio, governo, professores, escolas, nova, professor	0.878	0.741
6	governo, dilma, presidente, caso, empresa, eduardo, cunha, dinheiro, banco	0.834	0.805
9	direitos, contra, liberdade, direito, humanos, drogas, jovens, crime, dias	0.797	0.510
7	bom, maioridade, bandido, contra, dia, fica, deus, pobre, mal	0.767	1.228
14	contra, estado, ato, hoje, guerra, manifestantes, protesto, maconha, aumento	0.164	1.091
3	contra, governo, povo, brasil, ruas, protestos, coisa, fim, menos	-0.152	-0.027
2	policiais, policial, caso, militar, dois, jovem, pessoas, casa, rio	-0.386	-0.253
8	dia, bem, pessoas, vida, gente, medo, grande, nenhum, menos	-0.693	-0.645
1	pessoas, mundo, sempre, vida, nunca, tanto, melhor, quanto, brasil	-0.715	-1.154
10	dia, livre, eduardo, brasil, passe, segundo, lado, direito, ano	-0.923	-0.598
13	lei, brasil, mil, menos, sistema, grande, segundo, conta, maior	-1.046	-1.320
4	sistema, rio, janeiro, anos, bem, fhc, dinheiro, brasileira, hoje	-1.090	-0.411
11	rio, anos, brasil, grupo, paulo, mil, acordo, segundo, ficou	-1.437	-1.697

C.2 Human Coherence Evaluation Scores

This section lists in human evaluation scores and all automatic coherence scores.

Table C.7: Human evaluation from Class 1: Particular cause (Social Movement)

Topic Id	A1	A2	A3	Extrinsic Umass	Extrinsic UCI	Intrinsic UMass	Intrinsic UCI
14	3	2	5	-0.528	-0.835	2.019	1.467
10	1	1	2	-0.779	-0.525	1.204	1.741
3	4	5	5	0.366	0.298	1.114	0.309
6	1	1	3	-1.223	-1.056	0.687	0.169
11	3	3	5	0.573	1.087	0.261	0.563
9	2	4	4	0.706	0.239	0.123	0.455
0	2	3	5	-0.905	-0.993	-0.004	1.294
4	3	3	5	-0.162	0.381	-0.224	0.147
12	2	5	5	0.074	-0.289	-0.315	-0.960
8	1	3	5	0.373	0.145	-0.426	-0.379
7	2	4	4	-0.883	-0.508	-0.994	-0.657
5	2	4	5	-0.130	-0.070	-1.107	-1.293
13	4	5	5	-0.027	-0.108	-1.379	-1.339
1	4	2	3	-0.404	-0.697	0.461	-0.334
2	5	5	5	2.950	2.931	-1.421	-1.183

Table C.8: Human evaluation from Class 2: Grassroots news (Leftist)

Topic Id	A1	A2	A3	Extrinsic Umass	Extrinsic UCI	Intrinsic UMass	Intrinsic UCI
0	4	5	5	2.121	1.427	1.470	1.103
5	4	4	5	1.089	1.051	1.450	1.807
3	2	5	5	0.592	0.243	1.345	1.159
4	1	1	2	-0.886	-0.757	1.019	0.846
1	4	5	5	0.082	0.883	0.287	0.713
9	3	1	1	-0.765	-1.066	0.282	-0.193
8	4	4	5	0.647	0.740	-0.027	-0.179
12	3	2	3	-1.137	-1.344	-0.059	-0.187
10	4	3	5	-0.160	-0.206	-0.112	0.353
13	2	3	2	-1.584	-1.820	-0.522	-0.675
7	2	3	3	-0.708	-0.685	-0.709	-0.805
6	2	4	5	0.662	0.971	-0.738	-0.859
14	3	3	5	0.710	0.534	-0.753	-0.674
11	2	2	5	0.250	0.686	-0.954	-0.335
2	2	2	4	-0.913	-0.658	-1.978	-2.076

Table C.9: Human evaluation from Class 3: Pro-Governism news (Center)

Topic Id	A1	A2	A3	Extrinsic Umass	Extrinsic UCI	Intrinsic UMass	Intrinsic UCI
0	4	5	5	0.454	0.799	0.115	1.247
9	3	5	5	0.593	0.695	0.181	0.541
5	3	2	2	0.414	-0.151	1.141	1.338
14	2	1	1	-0.140	-0.438	0.947	1.109
2	4	5	5	-0.363	0.199	-1.568	-1.445
1	2	5	5	-0.930	-1.023	-0.168	-0.453
6	3	3	2	-0.072	0.477	-0.017	-1.268
11	3	4	5	2.353	2.027	2.489	1.295
7	2	2	5	0.717	0.966	-0.360	-0.004
4	2	3	3	-1.857	-1.744	-0.394	-0.469
8	3	2	1	-0.791	-0.987	-0.997	-1.433
3	3	5	5	-0.315	-0.241	-0.284	0.232
10	4	5	5	0.881	0.594	0.351	0.352
12	4	3	3	0.162	0.188	-0.189	0.009
13	1	1	1	-1.106	-1.359	-1.247	-1.051

Table C.10: Human evaluation from Class 4: Pro-impeachment news (Rightist)

Topic Id	A1	A2	A3	Extrinsic Umass	Extrinsic UCI	Intrinsic UMass	Intrinsic UCI
2	4	1	3	-1.132	-1.187	-1.272	-0.584
4	1	2	3	-0.582	-0.837	-0.617	-0.276
8	3	2	3	-0.841	-0.920	-0.255	-0.121
7	2	2	3	-0.560	-0.361	-0.211	-0.255
9	2	5	5	-0.925	-1.082	-1.618	-1.880
14	4	5	5	-0.611	-0.369	-0.097	-0.094
0	3	4	5	-0.541	-0.957	-0.452	-0.845
6	3	5	5	0.709	1.129	0.822	-0.048
1	2	4	5	-0.243	-0.022	-0.826	-0.761
10	2	5	5	-0.974	-0.764	-0.718	-0.877
13	4	5	5	1.436	1.143	1.455	1.300
5	3	5	5	1.810	1.307	1.865	1.871
12	4	5	5	0.887	0.848	0.719	0.639
3	4	4	5	1.482	1.727	1.019	1.432
11	4	4	5	0.084	0.344	0.184	0.499

Table C.11: Human evaluation from Class 5: Pro-impeachment virals (Rightist)

Topic Id	A1	A2	A3	Extrinsic Umass	Extrinsic UCI	Intrinsic UMass	Intrinsic UCI
11	4	5	5	-0.527	-0.904	0.028	-0.095
0	3	5	5	1.829	1.723	0.433	0.184
10	4	5	5	-0.848	-0.926	-1.292	-1.732
7	1	3	5	-0.250	-0.170	1.545	1.279
13	2	5	4	-0.547	-0.499	-0.721	-0.670
2	4	5	5	-2.088	-2.571	-1.299	-1.880
8	4	5	5	0.609	0.608	0.583	0.369
9	4	5	5	-0.064	0.852	-1.121	-0.457
1	2	3	5	-1.490	-0.397	-1.735	-1.214
6	1	3	3	0.979	0.715	0.810	0.917
5	1	3	3	0.846	0.598	1.141	1.068
12	1	3	3	0.258	0.072	0.045	0.230
14	1	3	3	0.581	0.439	0.607	0.799
3	1	3	3	0.280	0.087	0.225	0.345
4	1	3	3	0.432	0.374	0.751	0.856

Table C.12: Human evaluation from Class 6: Progressist virals

Topic Id	A1	A2	A3	Extrinsic Umass	Extrinsic UCI	Intrinsic UMass	Intrinsic UCI
0	4	5	5	0.878	0.741	0.027	-0.094
5	1	1	2	1.956	1.699	2.277	1.683
6	5	4	5	0.834	0.805	1.215	0.776
7	3	1	4	0.767	1.228	0.941	1.615
9	3	2	5	0.797	0.510	-0.300	-0.097
4	1	2	3	-1.090	-0.411	-1.373	-0.461
14	3	4	5	0.164	1.091	-0.667	0.109
10	1	2	4	-0.923	-0.598	-0.574	-0.992
2	3	3	5	-0.386	-0.253	-0.230	-0.530
13	2	2	3	-1.046	-1.320	0.157	-0.430
12	3	3	4	1.046	0.029	1.208	1.821
1	2	2	2	-0.715	-1.154	-0.181	-0.685
11	1	3	1	-1.437	-1.697	-0.886	-0.834
8	2	1	2	-0.693	-0.645	-0.652	-0.861
3	2	1	4	-0.152	-0.027	-0.962	-1.019

Bibliography

[1] Jian Pei Jiawei Han, Micheline Kamber. *Data Mining : Concepts and Techniques*. Elsevier, 3 edition, 2012.

[2] David Blei, Lawrence Carin, and David Dunson. Probabilistic topic models. *IEEE Signal Processing Magazine*, 27(6):55–65, 2010.

[3] Ilana Heintz, Ryan Gabbard, Mahesh Srinivasan, David Barner, Donald S Black, Marjorie Freedman, and Ralph Weischedel. Automatic Extraction of Linguistic Metaphor with LDA Topic Modeling. (June):58–66, 2013.

[4] Lin Liu, Lin Tang, Wen Dong, Shaowen Yao, and Wei Zhou. An overview of topic modeling and its current applications in bioinformatics. *SpringerPlus*, 2016.

[5] Songgao Tu and Chaojun Lu. Topic-based user segmentation for online advertising with latent dirichlet allocation. In *Proceedings of the 6th International Conference on Advanced Data Mining and Applications - Volume Part II*, ADMA'10, pages 259–269, Berlin, Heidelberg, 2010. Springer-Verlag.

[6] Yi Fang, Luo Si, Naveen Somasundaram, and Zhengtao Yu. Mining contrastive opinions on political texts using cross-perspective topic model. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 63–72, New York, NY, USA, 2012. ACM.

[7] Richard Socher, Kenneth A Norman, and David M Blei. A Bayesian Analysis of Dynamics in Free Recall. pages 1–9, 2010.

[8] Russ Altman, Deirdre Mulligan, and Yoav Shoham. Artificial intelligence and life in 2030. Report of the 2015 study panel. *Stanford*, 2016.

[9] Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*, 2009.

[10] Xing Wei and W. Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 178–185, New York, NY, USA, 2006. ACM.

[11] Romain Deveaud, Eric SanJuan, and Patrice Bellot. Are semantically coherent topic models useful for ad hoc information retrieval? In *ACL (2)*, pages 148–152. The Association for Computer Linguistics, 2013.

[12] Arian Pasquali, Marcela Canavarro, Ricardo Campos, and Alípio M. Jorge. Assessing topic discovery evaluation measures on facebook publications of political activists in brazil. In *In Proceedings of the International Conference on Computer Science & Software Engineering*, C3S2E, 2016.

[13] Thomas Hill Robert Nisbet Dursun Delen Andrew Fast Gary Miner, John Elder IV. *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications 1st Edition*. Academic Press, 2012.

[14] Miloš Radovanović and Mirjana Ivanović. Text mining: Approaches and applications. *Novi Sad J. Math*, 38(3):227–234, 2008.

[15] Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January 2010.

[16] S Vijayarani, Ms J Ilamathi, and Ms Nithya. Preprocessing techniques for text mining - an overview. *vol*, 5:7–16, 2015.

[17] Christopher Manning and Hinrich Schutz. *Foundations of Statistical Natural Language Processing*. 1999.

[18] Ewan Klein Steven Bird and Edward Loper. Categorizing and Tagging Words. http://www.nltk.org/book/ch05.html, 2009. [Online; accessed 01-May-2016].

[19] Alexander Hinneburg, Frank Rosner, Stefan Pessler, and Christian Oberländer. Exploring document collections with topic frames. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 2084–2086, New York, NY, USA, 2014. ACM.

[20] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[21] G. Zipf. Human behaviour and the principle of least-effort. Addison-Wesley, Cambridge, MA, 1949.

[22] Ben Lorica and David Blei. O'Reilly Data Podcast. Topic models: Past, present, and future., February 2010.

[23] Joseph Bradley. Topic modeling with LDA: MLlib meets GraphX. https://databricks.com/blog/2015/03/25/topic-modeling-with-lda-mllib-meets-graphx.html, 2015. [Online; accessed 01-June-2016].

[24] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[25] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '88, pages 281–285, New York, NY, USA, 1988. ACM.

[26] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[27] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[28] Mark Steyvers and Tom Griffiths. *Probabilistic topic models*, volume 427. 2007.

[29] Xuerui Wang and Andrew McCallum. Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 424–433, New York, NY, USA, 2006. ACM.

[30] Ali Daud, Juanzi Li, Lizhu Zhou, and Faqir Muhammad. Knowledge discovery through directed probabilistic topic models: a survey. *Frontiers of Computer Science in China*, 4(2): 280–301, 2010.

[31] Z. Nanli, Z. Ping, L. Weiguo, and C. Meng. Sentiment analysis: A literature review. In *Management of Technology (ISMOT), 2012 International Symposium on*, pages 572–576, Nov 2012.

[32] Matthew J. Beal David M. Blei Yee Whye Teh, Michael I. Jordan. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[33] Martin Ponweiser. Latent dirichlet allocation in r, 2012. Institute for Statistics and Mathematics, WU (Wirtschaftsuniversitat Wien), Austria.

[34] David M Blei and John D Lafferty. Topic Models. *Text Mining: Classification, Clustering, and Applications*, pages 71–89, 2009.

[35] Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. *Proceedings of the 26th Annual International Conference on Machine Learning*, (4):1105–1112, 2009.

[36] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press.

[37] Thomas M. Cover and Joy A. Thomas. Elements of information theory. 2006.

[38] Igor Douven and Wouter Meijs. Measuring coherence. pages 405–425, 2007.

[39] Branden Fitelson. A probabilistic theory of coherence. *Analysis*, 63(3):194–199, 2003.

[40] Loulwah Alsumait, Daniel Barbará, James Gentle, and Carlotta Domeniconi. Topic significance ranking of lda generative models. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part I*, ECML PKDD '09, pages 67–82, Berlin, Heidelberg, 2009. Springer-Verlag.

[41] David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. Evaluating topic models for digital libraries. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, JCDL '10, pages 215–224, New York, NY, USA, 2010. ACM.

[42] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 100–108, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[43] David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 262–272, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[44] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. Exploring Topic Coherence over many models and many topics. *Emnlp2012*, (July):952–961, 2012.

[45] Nikolaos Aletras and Regent Court. Evaluating topic coherence using distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics*, IWCS 2013, pages 13–22. Association for Computational Linguistics, 2013.

[46] Frank Rosner, Alexander Hinneburg, Michael Röder, Martin Nettling, and Andreas Both. Evaluating topic coherence measures. *CoRR*, abs/1403.6397:1–4, 2014.

[47] M. Röder, A. Both, and A. Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 399–408, New York, NY, USA, 2015. ACM.

[48] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 310–318, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.

[49] Colette Joubarne and Diana Inkpen. Comparison of semantic similarity for different languages using the google n-gram corpus and second- order co-occurrence measures. In *Proceedings of the 24th Canadian Conference on Advances in Artificial Intelligence*, Canadian AI'11, pages 216–221. Springer-Verlag, 2011.

[50] François Role and Mohamed Nadif. Beyond cluster labeling: Semantic interpretation of clusters' contents using a graph representation. *Knowledge-Based Systems*, 56:141 – 155, 2014.

[51] Muhammad Omar, Byung-Won On, Ingyu Lee, and Gyu Sang Choi. Lda topics: Representation and evaluation. *Journal of Information Science*, 41(5):662–675, 2015.

[52] T. Mitchell. Uci machine learning repository - twenty newsgroups data set. 1997.

[53] Joseph L. Fleiss. Measuring nominal scale agreement among many raters. In *Psychological Bulletin*, Vol 76(5), pages 378–382, 1971.

[54] Liangjie Hong and Brian D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 80–88, New York, NY, USA, 2010. ACM.

[55] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR'11, pages 338–349, Berlin, Heidelberg, 2011. Springer-Verlag.

[56] Anat Ben-David and Ariadna Matamoros Fernández. Hate speech and covert discrimination on social media: Monitoring the facebook pages of extreme-right political parties in spain. *International Journal of Communication*, 10(0), 2016.

[57] Evan Sandhaus. The New York Times Annotated Corpus LDC2008T19. Philadelphia: Linguistic Data Consortium. https://catalog.ldc.upenn.edu/LDC2008T19, 2008. [Online; accessed 01-Sept-2016].

[58] Manuel Castells. *Redes de Indignacion y Esperanza.* Alianza Editorial, 2012.

[59] Javier Toret (coord.). *Tecnopolítica y 15M: La potencia de las multitudes conectadas.* 2013.

[60] Katrina Kimport Jennifer Earl. *Digitally Enabled Social Change: Activism in the Internet Age.* MIT Press, 2011.

[61] W. Lance Bennett and Alexandra Segerberg. *The Logic of Connective Action: Digital Media and the Personalization of Contentious Politics.* Cambridge University Press, 2013.

[62] Justin Grimmer. A bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. In *In Proceedings of the First Workshop on Social Media Analytics, SOMA '10*, 2010.

[63] Margaret E. Roberts, Brandon M. Stewart, Dustin Tingley, and Edoardo M Airoldi. The structural topic model and applied social science. In *Advances in Neural Information Processing Systems Workshop on Topic Models: Computation, Application, and Evaluation*, Lake Tahoe, Nevada, 2013.

[64] Catherine Shoichet and Euan McKirdy. Brazil's Senate ousts Dilma Rousseff in impeachment vote. CNN. ̀http://edition.cnn.com/2016/08/31/americas/brazil-rousseff-impeachment-vote/index.html, 2016. [Online; accessed September 5, 2016].

[65] Bernhard Rieder. Studying facebook via data extraction: The netvizz application. In *Proceedings of the 5th Annual ACM Web Science Conference*, WebSci '13, pages 346–355, New York, NY, USA, 2013. ACM.

[66] Allison J. B. Chaney and David M. Blei. Visualizing topic models, 2012.

[67] Carson Sievert and Kenneth E. Shirley. Ldavis: A method for visualizing and interpreting topics. In *ACL Workshop on Interactive Language Learning, Visualization, and Interfaces*, 2014.

[68] Jaimie Murdock and Colin Allen. Visualization techniques for topic model checking. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 4284–4285. AAAI Press, 2015.

[69] Jason Chuang, Christopher D. Manning, and Jeffrey Heer. Termite: Visualization techniques for assessing textual topic models. In *Advanced Visual Interfaces*, 2012.

[70] a McCallum, a Corrada-Emmanuel, and X Wang. Topic and role discovery in social networks. *Computer Science Department Faculty Publication Series*, 30:3, 2005.

[71] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks, 2009.

[72] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, October 2011.

[73] Edward Loper Steven Bird, Ewan Klein. *Natural Language Processing with Python : Analyzing Text with the Natural Language Toolkit.* O'Reilly Media, 2009.

[74] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101 Suppl:5228–35, 2004.