LUISS Guido Carli
LIBERA UNIVERSITÀ INTERNAZIONALE DEGLI STUDI SOCIALI

Department of Business and Management

Bachelor's Degree in Management and Computer Science

Comparison of Topic Analysis: an exploratory research on the variations of topic modelling algorithms

Supervisor:
Dr. Francisco Javier Villarroel Ordenes

Student:
Davide Rosatelli

ACADEMIC YEAR 2020/2021

1 - Introduction

1.1 Practical phenomenon

Nowadays, companies have a lot of data available regarding their products or services, which obviously represent a great potential to exploit. These data are not all generated by the companies themselves, a large part of these come from the network created with their users: all the customer interactions, brand mentions, social media posts, online feedback and reviews, are part of a huge set of information that an organization sends and receives.

These data are not limited to numerical aspects: most of these are in the form of unstructured text. Given the huge quantity, it is simply impossible for companies to analyze all this material manually: it would be time-consuming, expensive, and inaccurate. For this reason, one of the aspects on which today's marketing is focusing on is the application of various techniques, based on artificial intelligence, to interact with this vast range of material.

Among the techniques of Natural language processing, that are the applications of all those algorithms capable of working on unstructured text data, Topic Analysis is one of the most valid and useful business applications.

1.2 Managerial relevance and problems

Also known as Topic Modelling, this unsupervised machine learning technique organizes large collections of text data, by assigning "tags" or categories according to the topic or theme of each individual text. The application of this technique to the massive amount of data collected by companies will help them to identify trends, make better decisions, optimize some processes, and become more efficient and productive. "*Topic detection allows us to easily scan large documents and find out what customers are talking about, to carry out a consistent, at scale and in real time analysis*." (MonkeyLearn, "Topic Analysis: a comprehensive guide to detecting topics from text")

Using this type of algorithms is particularly useful in the analysis of customer reviews. Identifying which aspects of the products or services offered by the companies are most discussed, positively or negatively, is a critical step of brand monitoring, and allows us to get closer to the actual needs of customers. Nearly 90% of consumers read at least 10 online reviews before forming an opinion about a business, and 79% of shoppers say they trust online reviews as much as personal recommendations (Oberlo, "10 Online Review Statistics You Need to Know in 2021"). Therefore, companies cannot absolutely ignore information from forums, blogs, or review sites.

For the correct use of these techniques, however, it is necessary to know how to apply them in the best way. As a matter of fact, there are several topic modeling algorithms, and it is not easy to find indications on which one is more suitable in certain situations or on a particular type of data; most of these are unsupervised techniques, and must therefore be able to assign the topics without knowing what they might be. A good level of accuracy is not easy to achieve, and large amounts of quality data are usually required.

1.3 Previous research and research GAPS

The use of topic models in marketing has been the subject of many research articles, which have made highlighted the usefulness and effectiveness of these methods to extract accurate information from the text. Many of these articles have a common starting point: the most used algorithm is definitely the Latent Dirichlet Allocation (LDA), although with many variations in the methods and applications with which it is used.

A perfect example of how the LDA can be modified to get as close as possible to our goal is reported in an article (Tirunnilai and Tellis 2014) in which a strategic brand analysis is carried out by identifying the dimensions of quality typical of the product examined and quantifying the valence of them. Most marketing research using topic models works on user generated content, not just reviews, but also social media posts; one of the aims here may be to monitor conversations about brands, as in the article in which, through changepoints integrated into the basic algorithm, were studied shifts in conversations caused by some event (Zhong and Scweidel 2020). Other very interesting researches concern the study of the costumer journey, such as the article that analyzes the search phrases used in the Online Search Engines (Li and Ma 2020); in this case, among the algorithms used there is not only the LDA, but also different types such as the Correlated Topic Model, equally valid if not even better in some situations. Other papers also focus on the topic dependency and the correlation between the extracted topics, using other variations of the LDA (Buschken and Allenby 2020).

We have therefore seen that there are several topic model algorithms, with many interesting applications. Each of these is based on different mathematical and statistical bases, and, as often happens among the various machine learning techniques, we cannot state “a priori” that one of these is the best in all cases.

1.4 The approach of this thesis to study the aforementioned Research GAPS

The purpose of this thesis is to analyze various topic models, testing each of them on two datasets. Each of these algorithms will be analyzed at a theoretical and practical level, studying the formulas

on which it is based and then applying it in the most optimized way on the data; the results will then be evaluated and compared with the appropriate measures.

The work will be carried out on customer reviews, divided into two datasets, concerning respectively a category of goods and one of services.

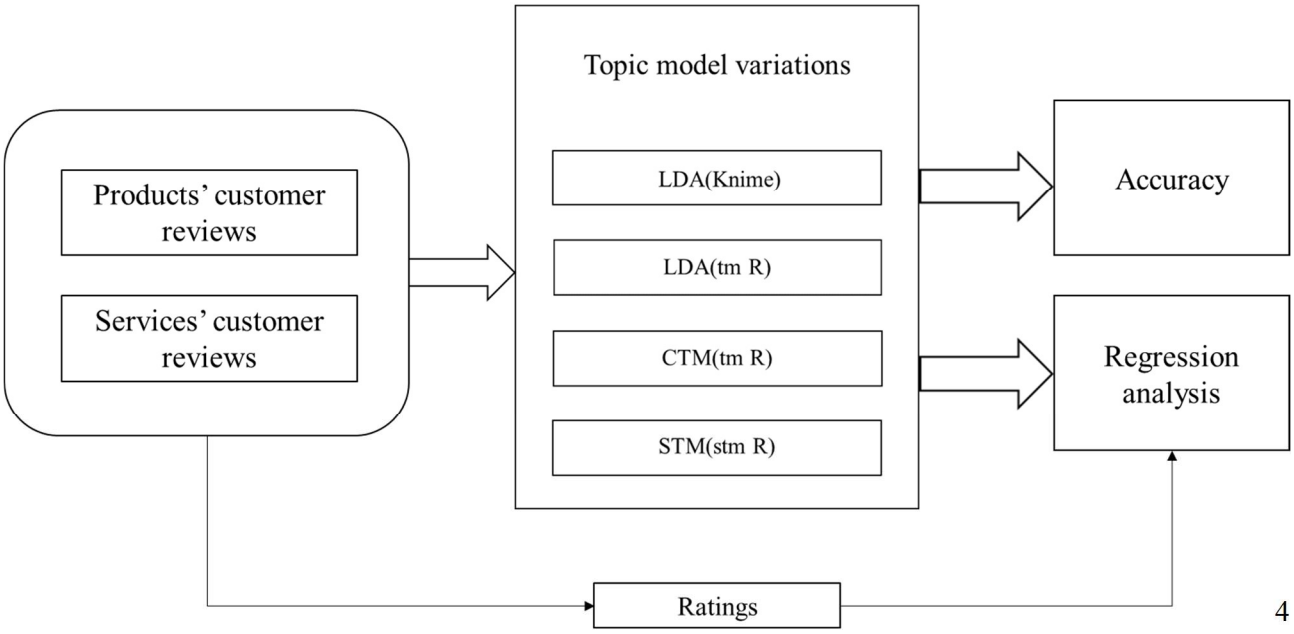
1.5 Contributions

This thesis, working on different models to be tested on two datasets, and hoping for good results, will allow to compare them accurately, and to highlight the differences between the various topic analysis.

The ultimate purpose of this thesis is to find the most suitable topic model variation for a given situation to be analyzed, creating a Knime workflow that allows a comparison between different topic models. Through appropriate measures I will determine which analysis obtained the most accurate results for each of the two datasets; I will then observe whether the different type of customer review, coming in one case from products and in the other from services, has had an impact on the models or not, that is, if the models with the most accurate results are the same in both cases.

To assess the model fit, and try to find the reason for the differences in the results if there are any, we will use the results given by the topic models as independent variable in a supervised regression model, predicting the reviews star rating.

Figure 1. Conceptual framework



2 - Literature Review

In the area of marketing research, we can find many articles regarding the use of topic detention algorithms. The best-known models, that are the ones that I will analyze in this work and whose technical details we will see later, can be implemented through various functions already defined in the various programming languages or analytics software. To have a complete picture of the most interesting applications of these models, in this section I will analyze the four works already mentioned in the introduction, that represent the state of art of topic models usage in marketing; these articles describe very advanced ways to use the models, combining them with other interesting aspects of natural language processing or marketing in general.

The oldest, but still one of the most relevant of these articles, is *Tirunillai and Tellis (2014), "Mining Marketing Meaning from Online Chatter: Strategic Brand Analysis of Big Data Using Latent Dirichlet Allocation"*; the authors of the paper propose a framework to extract the key latent dimensions of consumer satisfaction. The main technical part of the work is based on a latent Dirichlet allocation (LDA), integrated with a valence dictionary: in this way the authors were able to exploit topic modeling to extract the valence and its dimensions in the text. Everything is set up on user generated data coming from firms; the work represents a perfect application of topic modeling with the purpose of brand monitoring, as the result is an analysis to map the competitive brand position on those valence dimensions over time. The contributions resulting from this article are many and very interesting, their analysis has succeeded in defining some valence dimensions that apply perfectly to brand mapping, and shows how these are discussed between various companies and markets.

The second paper is *Buschken and Allenby (2020), "Improving Text Analysis Using Sentence Conjunctions and Punctuation"*. The main purpose of the authors is to propose a topic model that allows for serial dependency of topic, obtained through some unusual procedures, both in the preprocessing and in the model. As can be seen from the title of the article, one of its characteristics is to include the conjunctions and punctuations of the text in the algorithm, arguing that this can improve the accuracy of the topic detection; this is very different from the way texts are normally treated in machine learning, in which the classic bag-of-words considered relevant for analysis is created, discarding everything else. It is therefore very interesting to observe how preprocessing changes radically in this work, and how this leads to excellent results, giving more context and details to the data inserted. As for the models, more than one were used, with remarkable variations: first a classic latent Dirichlet allocation (LDA), then some models of local topic chunking as the sentence-constrained LDA (SC-LDA) and the conjunctions-and-punctuation-constrained (CPC-LDA), and

finally topic models with autocorrelated topics across words (AT-LDA); for the last two categories different variations have been tried, and the final part of the work analyzes and compares their results.

Another very relevant work is certainly *Li and Ma (2020), "Charting the Path to Purchase Using Topic Models"*. This research was performed on a dataset of consumer search phrases, aiming to find the best model to identify latent topics in them, and using them to understand the customers' position in the path to purchase. Analyzing the previous research in the fields of consumer's path to purchase, of search keywords, and of topic models applications in marketing, the authors were able to create a very specific and perfect model for their goal: it is a joint model, created combining a correlated topic model and a hidden Markow model (CTM-HMM); this algorithm is able to perform an accurate analysis at consumers level, based on a joint modeling of topics and consumers' choices; the analysis captures the latent states in the search phrases and, at the same time, is connected to purchase decisions. The work demonstrates how to obtain a great improvement in this specific application, working with variations of already existing algorithms.

The last article analyzed is *Zhong and Scweidel (2020), "Capturing Changes in Social Media Content: A Multiple Latent Changepoint Topic Model"*; here the goal of the authors is to create a model that can identify and emphasize the changes of subject in social media conversations. This is a more advanced task than that of a normal topic model, which is possible to achieve through a dataset that includes conversations in which there will be topic changes and a model that consider these changes over time. The dataset used includes social media posts from two brand crises (Volkswagen's 2015 emissions testing scandal and Under Armor's 2018 data breach), and from a brand promoting a new product (Burger King's 2016 launch of the Angriest Whopper). The model is based on an LDA, in which has been embedded a multiple latent changepoint model (LDA-MLC) through a Dirichlet process hidden Markov model; this allows to better deal with the temporal nature of social media posts, changing the prevalence of topics before and after each changepoint, without requiring prior knowledge about the number of changepoints. The results obtained are very precise, and the article brings a great contribution regarding conversation monitoring about brands.

These articles, summarized in *Table 1*, are proof of the innumerable research applications given by this kind of algorithms, obviously combined with the technical skills of the authors who programmed these complex variations.

Table 1. Literature review summary

Article	Model used	Dataset	Main purpose	Marketing contributions
Tirunillai, Tellis (2014)	LDA + Valence dictionary	Approximately 350,000 consumer reviews from firms in the markets of personal computing, cellular phones, footwear toys and data storage.	Use topic modelling to extract valence and its dimensions in the text.	Defining valence dimensions, Improving brand mapping and brand monitoring.
Buschken, Allenby (2020)	SC-LDA, CPC-LDA, AT-LDA	Four customer reviews datasets, respectively of restaurants, camping tents, luxury hotels and dog food.	Create a topic model that improves using sentence conjunctions and punctuation, and that allows for serial dependency of topic.	Improving text analysis.
Li, Ma (2020)	CTM-HMM	Consumers' search phrases, provided by a leading global hospitality company.	Identify latent topic and use them to identify the customers' position in the path to purchase.	Improving analysis of path to purchase.
Zhong, Scweidel (2020)	LDA-MLC	Social media posts from two brand crises and from a brand promoting a new product.	Create a topic model that can underline shifts in conversation.	Improving brand and conversation monitoring.

For this thesis, these papers have been a source of inspiration to understand how far the quality of these techniques can be pushed. As mentioned, this work will examine the most common models, which we can find already preset in some packages. In addition to the models, a fundamental part of the work will be based on how to evaluate their results; also for this activity, sources have been found in many previous researches, mentioned in the references.

3 - Research questions

To trace the path of the analysis that we will do and identify the objectives, let us now define what questions this thesis will try to answer.

As mentioned, the work will mainly rely on applying various topic modeling algorithms on two review datasets; these models operate with different formulas, but their goal is the same: to find common words in the text and to group them into topics, which can provide an overview of what is discussed in the reviews. Consequently, the first analysis will certainly be based on the observation of these topics, trying to highlight the similarities and differences between the results. The topics extracted from the various models will then be analyzed, assessing how representative they are of the complete dataset. In addition to the observation of the topics, the models will be evaluated through appropriate quantitative measures, indicative of how accurately the algorithms have worked. The first objective can therefore be summarized as follows:

> *By using different topic models on the same texts, are the words and topics extracted the same? which model is the most accurate?*

Another fundamental aspect of this work was the choice of the two datasets on which to test the models: they are two large collections of customer reviews, one relating to mobile phones and the other to hotels. Thanks to these datasets, it was possible to carry out another important analysis on the results: a comparison between the topics extracted respectively from product and service reviews. From a marketing research point of view, it can be very interesting to be able to identify which differences in the results are dictated by the type of asset analyzed. Surely the topics extracted from the two datasets will be different, but will the algorithms work in the same way in both cases? Through the evaluation measures, we will be able to observe if the accuracy of the models present greater differences by operating on the two datasets, and if the worst and best models remain the same in the two cases. This type of analysis can be a great contribution, as it can give an extra indication of which model is the best depending on the type of asset. The second objective that the work aims to analyze is therefore:

> *Do the best and worst topic models remain the same in both analyzed datasets? Does their accuracy depend on the asset covered in the reviews?*

These first two questions refer to the observation and evaluation of the topic models and their results taken alone; for the last analysis these will be the starting point for another type of machine learning model. As output of the topic models, in addition to the extracted words, we can obtain a probability for each topic, proportional to how much it is treated, in each specific row; in this way a variable will

be created in each review for each topic identified by the models. The two datasets also have the rating given by the customer to the product as a variable for each review. What I will then do is use this kind of topic models' output as independent variables in a supervised regression, which will predict the rating. With this analysis, obviously the goal is not to get a perfect prediction, but rather to find out how relevant are the topics extracted and how descriptive they really are of the review. The more accurate the prediction and the more statistically significant the coefficients, the more each topic will be a good variable. The last question I will try to answer is therefore:

> *Are the topics extracted statistically relevant as a description of the reviews, and good indicators for the quality of the product or service?*

4 – Method

4.1 Data description

In this section the technical work performed will be accurately described, starting with an overview of the data used.

The complete Knime workflow, both executed and not, and the two datasets, can be downloaded from this link:

https://drive.google.com/drive/folders/194c1gar_f2fuA4n3AZ8mFSAOsrWaVsXF?usp=sharing

The first dataset is a collection of mobile phone reviews on Amazon and is readily available, as it is available on Kaggle under the name "Amazon reviews: Unlocked mobile phones", where it can be downloaded as a csv. The dataset contains 413839 rows, and is divided into the following fields:

> *Product name, Brand, Price, Rating, Review text, Review votes*.

The second dataset was provided in an excel file by Professor Villaroel Ordenes, supervisor of this thesis. This dataset was scraped from TripAdvisor using Python and it includes all the hotels reviews of the city of Philadelphia from 2010 to 2019. It contains 79070 lines, with the following variables:

> *Hotel Name, Hotel URL, Hotel Location, Ranking of things to do, Number of reviews, Number of English reviews, Phone, Author Name, Author Location, Number of Reviews from the Author, Number of helpful votes for the author, Review Star Rating, Date of Experience, Review Date, Review Title, Review text, Helpful votes for review, Review ID, Has Images*.

Both datasets are perfectly representative for this type of work. For this thesis the necessary fields were only the text of the reviews and the rating; the others fields have only been aggregated to these two main columns. Both collections with all their fields can therefore have many more applications in machine learning projects. The amount of data used is a good compromise, given that the more quality data we have, the more accurate the models will be, but already with this amount the work on Knime has been extremely slow: for algorithms of this kind a great computing power is required, and some steps took hours to complete on a normal laptop.

4.2 Data processing

As in any machine learning project, in this thesis a large part of the workflow concerns the data preparation before they are inserted into the models; this preparation is particularly important when it concerns natural language processing, and even more when dealing with texts generated by generic

users, as in our case. In this section I will describe all the steps performed on Knime for data cleaning and preparing, which were the same for both datasets.

I started by inserting the files, which are read directly on Knime. The first steps are those necessary to perform a basic cleaning of the reviews: checking that there were no missing values in any line, correcting the spelling of the text using the "Spell checker" node, and eliminating all special characters. The reviews column is then transformed into a "document" object, to be able to treat it as a text on Knime.

After having transformed all the text into lowercase letters, a very important step that can help to obtain more accurate results was identifying and creating the relevant "ngrams". Through a node, we can automatically identify the words that appear together in the text more frequently, in this work the identification was limited to the pairs of words, or the “bigrams”, but this process can also be applied to longer sentences. Looking at the most frequent pairs of words, I then manually identified the relevant ones, i.e. those that together made more sense than alone (for example - in smartphone dataset: battery-life, sim-card, sd-card; in hotel dataset: check-in, front-desk, room-service, and so on). To make the models consider these bigrams as single words, I saved them externally, then through a loop I identified and highlighted them in the text, and finally I made them unmodifiable by using them as elements of a dictionary in the text and assigning them an identification tag. For this identification I used the "POS tagger" node, which also has the function of tokenizing the text, another step always necessary in this type of work.

After the bigrams, I continued with cleaning and preprocessing operations of the documents, through the specific nodes. I have filtered all the stop words, that are those used to form correct sentences but that taken by themselves have no relevance (articles, propositions, conjunctions or the like). I deleted from the text all words less than 4 characters of length, I deleted all numbers or words containing numbers, and I removed all punctuation. Another important step is stemming: using an algorithm that identifies all the words derived from others and transforms them into their root word; for this operation I used the "Kuhlen stemmer" node, the most used algorithm for English texts.

Finally, all the words that appeared in the dataset only rarely (less than 50 appearances for smartphones and less than 25 for hotels), which would not have affected the models, were deleted; to do this I first saved them externally and then used this file as a reference column in a dictionary filter.

The last step of the preprocessing is to create the Bag of Words, that is to extract the relevant terms from the text, and give them more or less relevance according to the frequency with which they

appear. This is a crucial step in these topic models, since it is an assumption that make word order in sentences irrelevant.

Having kept the most frequent words, all these data must now be grouped, so through a node I have aggregated all the columns keeping only the rating and the preprocessed documents. In the topic models only the document column will be used as input, while the ratings will be used later as a variable to be predicted with the results of the models.

4.3 Topic models

The models used in this work are all derivations of a main model: the Latent Dirichlet Allocation or LDA; by modifying some assumptions or some steps of the algorithms we can obtain models that work differently, focusing on different aspects. For a complete technical description and statistical explanation of the LDA, you can refer to *Blei 2012, "Probabilistic topic models"* while the articles I used as main source for understanding and writing the models on R are *Grün and Hornik (2011), "topicmodels: An R Package for Fitting Topic Models"* and *Roberts et al. (2019) "stm: An R Package for Structural Topic Models".*

In a nutshell, LDA is a generative model that describes how text documents could be generated probabilistically from a mixture of topics, where each topic follows a distribution over the most common words. The process defines a joint probability distribution over both the observed & hidden random variables, which are respectively the words and topics extracted. The model is therefore based this particular random variable, the Dirichlet distribution, that is a derivation of a multinomial distribution with discrete outcomes, with the following probability density function:

Dirichlet distribution PDF: $$\frac{1}{\mathrm{B}(\boldsymbol{\alpha})}\prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

where: $$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K}\Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}$$ where: $$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$$

the first model I used is the classic Latent Dirichlet Allocation, in particular the Knime version, a simple parallel threaded implementation of it, that is the "topic extractor (parallel LDA)" node. The configuration of this model was simple as any node on Knime, and for the hyperparameters I kept standard values:

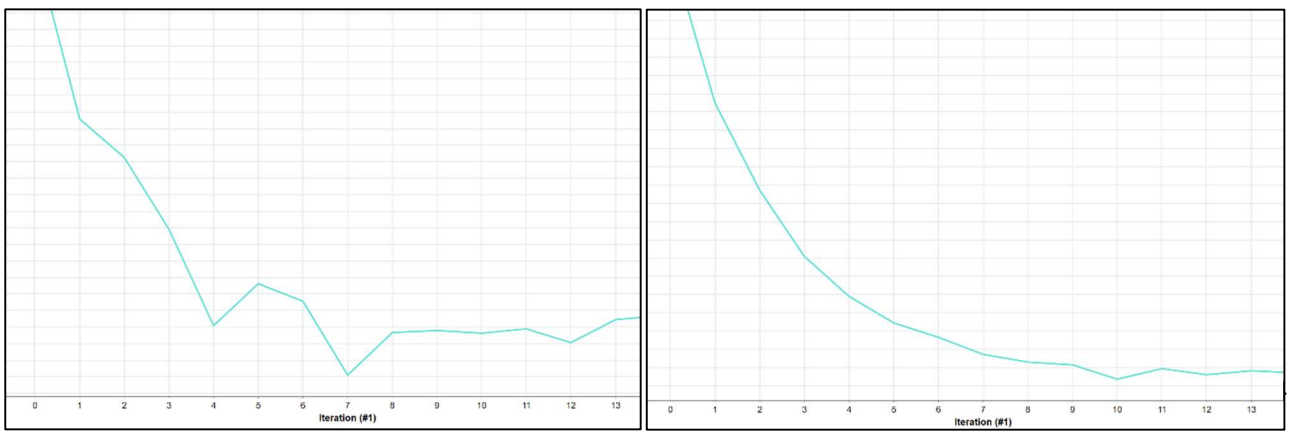
- 16 threads working in parallel
- Alpha (how much a document is allowed to be associated with more than one topic) 0.1
- Beta (how much one word is allowed to be associated with one topic) 0.01.

In the configurations of all the models there were also two other fundamental parameters to set before proceeding with the analysis: the first was the number of words per topic, which I decided to keep at 10 in all the cases I analyzed. The second is the number of topics that the model must extract, that is, the k parameter, which I have selected in this way: for both datasets, I took a random sample and on it through a loop I applied various LDAs of Knime with various possibilities of k, observing in each the perplexity, that is an evaluation measure which I will explain in section 4.4; I then plotted this measure for each iteration, and through the "elbow method" I selected the optimal k, that is the point where the lines start to smooth. In *Figure 2* you can see the plots for both datasets, and I therefore decided to use 4 topics for each model in both.

The second and third models used are both written in R through the “topicmodels” package; they therefore have an almost identical configuration, in which the method to be used for estimation must be specified (Gibbs samplig or VEM algorithm) and take as input a document-term-matrix object (a matrix that has terms and documents as rows and columns, respectively, and whose values indicate how many times these terms appear in documents). The first of these two models is an LDA, like the previous one, but implemented without threads, using the Gibbs sampling algorithm, and with the hyperparameters automatically optimized by the package.

The third model is a Correlated topic model, or CTM, also from the “topicmodels” package; it is a variation of the LDA that modifies the proportions of the topic distribution for the documents, thus allowing the model to have flexible correlations between topics. This model is particularly useful and more accurate when the most common words in the text can be associated with various similar topics. This model too was automatically optimized by the package, while it used VEM as estimation algorithm.

Figure 2. perplexity at different k, left for smartphone dataset and right for hotel dataset



The fourth and last model used is a Structural topic model or STM, implemented in R through the "stm" package. This type of model is the one that differs most from the previous three, but is still derived from the LDA. The essential difference is that the topic prevalence and content do not depend only on the Dirichlet distribution, but also on the metadata in the documents; the estimation method is so modified again at the distribution step, as the STM takes into consideration the document level structure information, investigating how covariates affect text content. Even on a practical level, the implementation was different, the function in fact required three inputs: the texts, the terms, and a metadata matrix. The objects used in the code were different from the previous models, and for this reason it was very difficult to use the package in the best possible way.

4.4 Evaluation measures

After defining the models, a crucial and difficult part of this work was finding and implementing the appropriate evaluation measures to compare their results. The two common measures chosen are *perplexity* and *topic coherence*, even if implemented for each model in a very different way, as there is no common method compatible with the different methods with which the models were written.

Perplexity is a measurement of how accurately a probability distribution or probability model predicts a sample; it is often used to compare different models, and is very common in natural language processing. The perplexity PP of a discrete probability distribution p is defined as:

$$PP(p) := 2^{H(p)} = 2^{-\sum_x p(x)\log_2 p(x)} = \prod_x p(x)^{-p(x)}$$

In the LDA of Knime, for each iteration there is as output the *log likelihood* of the model, another commonly used measure of fitting accuracy for models; in this model, the perplexity was computed slightly differently respect to the previous formula, that is with the log likelihood as negative exponential instead of the classical entropy (the *H(p)* in the formula is -*Loglikelihood*). For the LDA and CTM written in R, the package topicmodel had a function to calculate perplexity already implemented; unfortunately, for the STM model, in this work it was not possible to find a way to calculate this measure.

Topic coherence is an evaluation measure created specifically to evaluate topic models, introduced and explained in the article *Mimno et al. (2011) "Optimizing Semantic Coherence in Topic Models"*. Letting *D(v)* be the document frequency of word *v*, *D(v, v')* be co-document frequency of words *v* and *v'*, and *V(t)* as the list of the *M* most probable words in topic t, we define topic coherence as:

$$C(t; V^{(t)}) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{D(v_m^{(t)}, v_l^{(t)}) + 1}{D(v_l^{(t)})}$$

In short, it measures how often the main terms in each topic appear together in the same document. For all the models written in R, calculating this measure was quite easy, since both packages used already had functions set up to do so; for Knime's LDA, on the other hand, it was not possible to find a way to get topic coherence.

4.5 Regression analysis

Each of the models seen above, was used to create 4 topics. The outputs obtained, which will be fundamental for the final analysis of this work, are the probabilities for each document to include these topics. We will therefore have four new variables, which we can call topic0, topic1, topic2, topic3, the sum of which for each line will be 1. For simplicity we can say that each document is assigned to the topic that has this value higher than the others.

The last part of the work is a supervised predictive model: taking up the variable Rating that we had previously set aside, we will observe how it correlates with these new variables generated by the models, and how precisely is predicted by them.

As model I used the Negative Binomial Regression, similar to the Poisson but with a parameter to manage the over dispersed data, that is, with a high conditional variance. This regression is particularly useful for observing how much the various variables are statistically relevant in the prediction, through the coefficients produced. For both datasets, and using the the results of each of the four models, I trained these regressions, with this generic model specification:

$$\text{Rating}i = exp\,(\propto + \beta 1 * \text{topic0}i + \beta 2 * \text{topic1}i + \beta 3 * \text{topic2}i + \beta 4 * \text{topic3}i + \mu i)$$

5 – Results

After several hours for training, all 4 topic models worked correctly on both datasets. The main result obtained are summarized in this section.

The first thing we obviously observe are the topics produced, that are the groups of words that the models have produced; *Table 2* shows the 4 topics produced by each model, of which the 10 most relevant words are reported. Looking at the fields of this table, we can see many small similarities and differences: considering the two datasets separately, the models have actually produced different topics, but the words are often repeated, both from model to model, and in the groups produced by the same algorithm. The most frequent words in reviews (such as the obvious "phone" or "hotel") are present in most of the topics, while each model has grouped the other relevant words in various ways and combinations.

Analyzing each of the groups individually, the way in which the words have been grouped is not very clear: in theory the topics should be easily identifiable (i.e. by observing the words you should easily be able to assign a name to each topic), but clearly this is not the case; in almost none of the topics produced all the words have a common thread, through which we can identify what topic it is (almost always in the 10 words of each topic there are some that seem to be disconnected from the others). This particularity is perhaps more consistent in the CTM, where the topics are designed to intersect each other. In any case, judging by just looking at the words, it is very difficult to say which of the models produced the best topics.

To analyze the performance of the models from a more technical point of view, let's focus on the evaluation metrics described above for each of them. *Table 3* and *Table 4* show the perplexity and topic coherence scores, respectively for the smartphone and hotel datasets.

In general, a lower perplexity indicates a better performance. As first thing, from the data reported we can see that there is one model that has a value very distant from the others: in the Knime's LDA the perplexity was in fact calculated manually, while in the models of the topicmodels package it was used an already set function; looking at the results, this function certainly calculate perplexity in a different way from the other.

What we can conclude for both datasets is that between the two models in the topicmodels package, LDA has better perplexity than CTM, and the first LDA also performs well.

Table 2. Topics and words extracted by each model in both dataset

	Smartphones	Hotels
LDA (Knime)	Topic0: phone, unlock, sim-card, service, call, receive, purchase, amazon, verizon, bought	Topic0: hotel, stay, location, staff, clean, park, restaurant, walk, city, nice
	Topic1: phone, battery, screen, month, time, charge, return, charger, dont, bought	Topic1: stay, hotel, staff, friend, love, philadelphia, service, helpful, location, time
	Topic2: phone, excellent, love, product, perfect, price, recommend, very-good, thank, fast	Topic2: hotel, stay, call, check, night, front-desk, told, time, check-in, arrive
	Topic3: phone, screen, camera, android, phones, price, quality, dont, battery, battery-life	Topic3: hotel, bathroom, stay, floor, shower, breakfast, night, water, nice, coffee
LDA (topicmodels)	Topic0: phone, battery, month, time, return, didnt, amazon, receive, charger, money	Topic0: stay, hotel, service, love, time, experience, Philadelphia, beautiful, wonderful, help
	Topic1: screen, camera, phones, android, quality, battery-life, device, look, Samsung, size	Topic1: park, breakfast, walk, city, lobby, street, free, nice, hotel, restaurant
	Topic2: phone, call, sim-card, watch, text, dont, able, easy, wifi, data	Topic2: hotel, stay, staff, clean, location, friend, comfortable, helpful, nice, restaurant
	Topic3: phone, love, excellent, product, price, perfect, recommend, unlock, purchase, bought	Topic3: night, check, front-desk, call, didnt, door, people, floor, check-in, time
CTM	Topic0: excellent, product, love, perfect, recommend, unlock, time, receive, seller, very-good	Topic0: hotel, stay, clean, location, breakfast, staff, friend, front-desk, city, comfortable
	Topic1: phone, price, battery, month, dont, watch, charge, cant, quality, purchase	Topic1: stay, nice, time, night, location, definite, hotel, love, Philadelphia, staff
	Topic2: screen, camera, android, phones, device, update, battery-life, button, feel, video	Topic2: hotel, stay, location, walk, floor, excellent, night, locate, water, call
	Topic3: phone, call, screen, sim-card, nice, love, doesnt, start, fine, phones	Topic3: staff, hotel, stay, friend, park, restaurant, helpful, service, city, view
STM	Topic0: love, camera, bought, quality, expect, issue, sim-card, cant, service, amazon	Topic0: breakfast, nice, hotel, floor, bathroom, free, park, coffee, lobby, street
	Topic1: phone, screen, time, product, battery, charge, perfect, fast, feature, seller	Topic1: hotel, stay, night, call, check, time, front-desk, didnt, told, check-in
	Topic2: price, look, unlock, device, doesnt, battery-life, connect, pretty, start, function	Topic2: hotel, stay, locate, staff, clean, friend, restaurant, city, Philadelphia, comfort
	Topic3: Excel, dont, call, purchase, iphone, month, nice, receive, easy, recommend	Topic3: stay, staff, hotel, help, service, love, time, friend, wonder, thank

Regarding the topic coherence, each topic produced by the models has its own score; in general the higher this number is, the more the topic is considered correct. From the tables we can see that in the same models the 4 topics have coherence scores quite similar, and all fairly good. Among the 3 models in which this measure is present, for both datasets the STM is the one with the best coherence values.

Let's now analyze the other type of topic models' output, that is the probability for each document to contain the topics produced. To get an idea of which of the topics was the most relevant and discussed in the reviews, I have approximated these scores, assigning each document to the topic most discussed in it; then these results were visualized through pie charts, in *Figure 3*. We can see that only in some of the models the distribution of topics per document is uniform. In all the LDAs the 4 topics are well distributed, while in others, such as the two CTMs or the STM for smartphones, some topics are practically absent. This does not mean that they are wrong or useless, but the models attribute greater importance to the topics most present in the graphs, which are therefore the most discussed.

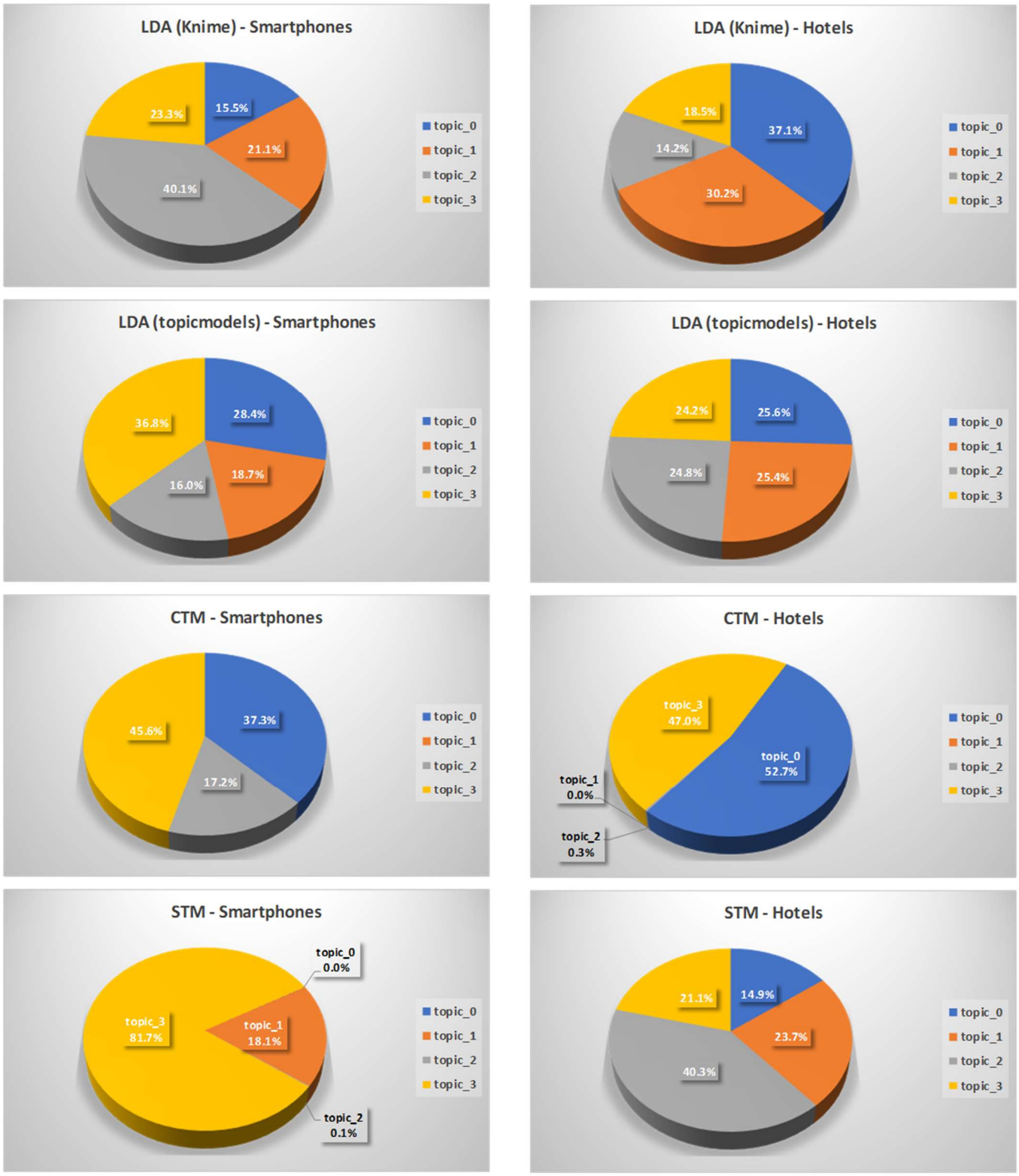
Table 3. Perplexity and Topic coherence scores for the models in the Smartphone dataset

	LDA (Knime)	**LDA (tm)**	**CTM**	**STM**
Perplexity	116.763	719.4104	757.1935	-
Topic coherence	-	-177.8005	-160.0358	-109.5752
		-155.1351	-137.1285	-106.9239
		-140.9074	-158.1452	-110.0150
		-158.4573	-171.5871	-108.4815

Table 4. Perplexity and Topic coherence scores for the models in the Hotel dataset

	LDA (Knime)	**LDA (tm)**	**CTM**	**STM**
Perplexity	150.815	884.3921	994.2428	-
Topic coherence	-	-126.15783	-84.12935	-76.0215
		-98.74233	-92.36959	-75.87660
		-77.89023	-123.02498	-56.74962
		-117.03756	-95.42542	-79.05873

Figure 3. Pie charts of the topic distribution in the documents for each model. Respectively from top to bottom we have LDA (Knime), LDA (tm), CTM and STM; the column on the left is the smartphone dataset, the one on the right the hotel dataset.



The latest results to be analyzed are related to the regressions. In each of the negative binomial regressions, we can analyze in detail the coefficients of the variables, to know how much they are correlated with the prediction. In all the cases analyzed the 4 topics of each model were all highly statistically relevant to predict the rating of the reviews, as shown in each of the R outputs, where each variable is characterized by 3 asterisks.

To measure the accuracy of the regressions, *Table 5* shows the Akaike information criterion (AIC) and the Mean squared error (MSE); as known, both of these measures are estimators of the prediction error, so the lower the value, the more accurate the regression.

The scores of the various models in the table are not very far from each other, but we can still use them to find the most accurate prediction. In the smartphone dataset, unfortunately the STM's MSE was not calculated, as in one of the last steps some lines were automatically deleted; in this dataset the model that produced the best regression is the Knime's LDA, for both measures. In the hotel dataset, on the other hand, the model with the best values in both measures is the STM.

Table 5. accuracy measures for the regressions, for each of the topic models

	AIC	**MSE**
LDA (Knime) – Smartphones	1132450	8.167184
LDA (tm) – Smartphones	1157375	8.253221
CTM – Smartphones	1202866	8.406275
STM – Smartphones	1217073	-
LDA (Knime) – Hotel	272605	8.300287
LDA (tm) – Hotel	272047	8.295247
CTM – Hotel	279002	8.391239
STM – Hotel	269240	8.256647

6 – Discussion

6.1 Contributions

This work, with the results produced, can, at least partially, provide answers to all the research questions initially set. Let's summarize the main findings.

First, looking again at *Table 2*, we can conclude that the various models have grouped the same words in the texts, and topics are only slightly different from each other.

As already mentioned, it is very difficult to establish by looking just at these outputs which model is better, but we can do it instead by looking at the quantitative measures as a whole: if we use as a reference the accuracy of the final regression (the scores of the *Table 5*), the best models, albeit slightly, are the Knime's LDA for the Smartphone dataset and the STM for the Hotel dataset.

A very interesting result is therefore the fact that in the two datasets the best model is not the same. One of the purposes of the analysis was to find out if using reviews of products and services had changed the results, and as a matter of fact it did. What this work demonstrates is that among those seen there is not a better topic model than the others, that will lead to more precise results in any situation. Indeed, it would be risky to claim that STM is best suited for service reviews and LDA for good reviews, as no direct correlation has been demonstrated, and there are many other factors to be considered. Evaluation measures are very useful for comparing models on the same dataset, but they are obviously very far apart if we look at them for both datasets, and much of this difference is due to the different sizes.

The part of the analysis concerning the final regressions produced valid results. Obviously if the aim had been to obtain a good accuracy in the predictions, we would never have used the results of the topic models as variables, but observing the results of each one individually, and in particular the coefficients related to these new variables, we can state that all the generated topics are statistically relevant and so related to the rating.

The Knime workflow used can be a basis to build any type of topic model, and the measures explained in the method are the most suitable for comparing this type of model.

This work has therefore contributed to the topic model literature, highlighting the aspects to keep in mind when choosing the most suitable variation.

6.2 Managerial implications

This study allowed us to identify areas on which to focus in selecting and using a Topic model, which remains an extremely useful tool from a managerial point of view. This work can be a reference for a basic comparison among these algorithms, considering what companies want to identify through the application of a thematic model. Furthermore, the workflow on Knime shows the standard process of how to prepare this type of data and can be easily modified and adapted to other data sets or natural language processing techniques.

The results demonstrate the effectiveness of these algorithms in being able to synthesize a huge number of texts in a few groups of words, especially when applied to reviews or other types of user-generated data; furthermore, the regression showed that these arguments are indeed descriptive of the text, given their statistical significance.

6.3 Further research directions

As seen, this study has led to some interesting conclusions; this despite many limitations encountered.

First of all, since these algorithms are complex and the amount of data used was considerable, the work has required a high computational effort: with a normal PC, many steps took many a very long time to complete; without this limitation, it would have been possible to try different combinations of the various parameters in each model, in order to make each of them as optimized as possible and suitable for the data entered. The most problematic parts of the work were probably the evaluation measures; I was unable to calculate the perplexity and coherence for all the models, and even among the values obtained, the methods were different. This problem is due to the fact that the models used came from different packages or software, with different behaviors and different objects as inputs and outputs; it is therefore impossible to apply the same functions for one of these measures to their results. Finding a way to evaluate a single model would not be complicated, but it could be ineffective: I believe that it is often important (as in this case) to try various algorithms that do the same job in different ways; as at the moment there is no a simple quantitative way to compare any type of topic model with each other, this is certainly a starting point for further studies.

Finally, an aspect that would be interesting to investigate deeply is the correlation between the type of model to be used and the category of asset the text deals with. In this study I tried to do this by examining reviews of a product and a service, but further research could analyze this relationship in much more depth, perhaps by adding variables that indicate this correlation, or trying to identify other possible variations of the models that are as suitable as possible for the type of data.

References

Netzer et al. (2012), “Mine Your Own Business: Market-Structure Surveillance Through Text Mining”, Marketing Science 31(3), 521-543

Tirunillai, Tellis (2014), “Mining Marketing Meaning from Online Chatter: Strategic Brand Analysis of Big Data Using Latent Dirichlet Allocation”, Journal of Marketing Research Vol. LI (August 2014), 463–479

Levy, Franklin (2014), “Driving Regulation: Using Topic Models to Examine Political Contention in the U.S. Trucking Industry”, Social Science Computer Review 2014, Vol. 32(2) 182-194

Li, Ma (2020), “Charting the Path to Purchase Using Topic Models”, American Marketing Association, Journal of Marketing Research 2020, Vol. 57(6) 1019-1036

Buschken, Allenby (2020), “Improving Text Analysis Using Sentence Conjunctions and Punctuation”, Marketing Science 39(4), 727-742

Zhong, Scweidel (2020), “Capturing Changes in Social Media Content: A Multiple Latent Changepoint Topic Model”, Marketing Science 39(4), 827-846

Blei (2012), “Probabilistic Topic Models”, Communications of the acm, April 2012 vol. 55 no. 4

Mimno et al. (2011), "Optimizing Semantic Coherence in Topic Models", Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, 262–272

Roberts et al. (2019), “stm: An R Package for Structural Topic Models”, Journal of Statistical Software, October 2019, Volume 91, Issue 2

Grün, Hornik (2011), “topicmodels: An R Package for Fitting Topic Models”, Journal of Statistical Software, 40(13), 1–30

Villarroel Ordenes, Zhang (2019), “From words to pixels: text and image mining methods for service research”, Journal of Service Management

Berger et al. (2020), “Uniting the Tribes: Using Text for Marketing Insight”, American Marketing Association, Journal of Marketing, 1-25

MonkeyLearn, “Topic Analysis: a comprehensive guide to detecting topics from text”, https://monkeylearn.com/topic-analysis/

Oberlo, "10 Online Review Statistics You Need to Know in 2021", https://www.oberlo.com/blog/online-review-statistics

Towards data science, "Introduction to The Structural Topic Model (STM) | by Theo Lebryk", https://towardsdatascience.com/introduction-to-the-structural-topic-model-stm-34ec4bd5383

Towards data science, "Evaluate Topic Models: Latent Dirichlet Allocation (LDA) | by Shashank Kapadia", https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0

Towards data science, "Using LDA Topic Models as a Classification Model Input | by Marc Kelechava", https://towardsdatascience.com/unsupervised-nlp-topic-models-as-a-supervised-learning-input-cf8ee9e5cf28

Towards data science, "A Beginner's Guide to Latent Dirichlet Allocation(LDA) | by Ria Kulshrestha", https://towardsdatascience.com/latent-dirichlet-allocation-lda-9d1cd064ffa2

Towards data science, "Intuitive Guide to Correlated Topic Models | by Theo Lebryk", https://towardsdatascience.com/intuitive-guide-to-correlated-topic-models-76d5baef03d3

Rpubs, "Topic modeling with quanteda and STM", https://api.rpubs.com/cbpuschmann/un-stm

GitHub, "Learning structural topic modeling using the stm R package", https://github.com/dondealban/learning-stm