Sentiment Analysis

Gaging Opinions of Large Groups

Andrei P. Kirilenko, Luyu Wang, and Svetlana O. Stepchenkova

Learning Objectives
- Define sentiment analysis goals
- Describe variety of data for sentiment analysis
- Explain main approaches used in text sentiment analysis
- Apply sentiment analysis to tourism domain data
- Indicate popular software used for sentiment analysis

1 Introduction

"Sentiment analysis or opinion mining is the computational study of people's opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes" (Liu & Zhang, 2012, p. 215). The word "sentiment" represents peoples' feelings such as joy, sadness, anger, and similar. With the explosive popularity of social media leading to the necessity of fast processing of huge volumes of data, e.g., from customer reviews, the traditional methodologies of manual estimation of people's opinion about topics of products of interest are being increasingly replaced with the automated sentiment analysis (Liu, 2012). Consequently, the scholarship on the methodologies and practices of the computer-based sentiment analysis is in demand and exhibits fast growth. For example, the paper by Bakshi et al. (2016) on using sentiment analysis of tweets to predict changes in stock prices was cited over 10,000 times.

A. P. Kirilenko (✉) · L. Wang · S. O. Stepchenkova
Department of Tourism, Hospitality and Event Management, University of Florida, Gainesville, United States
e-mail: Andrei.kirilenko@ufl.edu; luyuw.93@ufl.edu; Svetlana.step@ufl.edu

R. Egger (ed.), *Applied Data Science in Tourism*, Tourism on the Verge,
https://doi.org/10.1007/978-3-030-88389-8_17

Sentiment analysis attempts to measure the emotional valence of the text using a one-dimensional numerical scale from positive to negative sentiment. Depending on the goal of the analysis, it can be applied to the entire document, separate sentences, or the aspects of interest (e.g., different aspects of a consumer product). In addition, the comparative sentiment analysis attempts to compare different sentiment estimates, e.g., "Bonaire is better than Aruba." Liu (2012) and Hu and Bing (2004) postulate the following essential elements of an expressed opinion: the sentiment (s), the opinion target (g), the aspect of the target on which the opinion is expressed (a), the opinion holders (those who holds the opinion) (h), and the time when the opinion is expressed (t). The opinion then can be formally written as a 5D vector (g, a, s, h, t). For example, a hotel review sentence "I hated beddings in the hotel, but liked the view" written on May 5, 2020, by user cat1967 could be expressed as two vectors (hotel, bed, negative, cat1967, 05052020) and (hotel, view, positive, cat1967, 05052020). This makes possible a variety of probes such as aspect sentiment analysis, comparative sentiment analysis, evolution of sentiment over time, and so on.

The document-level analysis is the simplest one. Its goal is finding the sentiment of an entire document; thus, it assumes that opinions expressed in an analyzed document are coming from a single person and related to a single event or product (Liu, 2020). This assumption mainly holds for the review-type documents since they are typically authored by one person and express an opinion on one product and for microblogs such as Twitter, but generally, it is too restrictive.

The sentence-level analysis is free of the abovementioned restriction and hence can be applied to many more types of documents. The drawback is that the amount of information used to determine the sentiment is much smaller compared to the document-level classification, making the problem more complex. In addition, while it is generally possible to classify documents into two classes, positive and negative, many sentences contain no sentiment. Hence, instead of the two-class classification of a document, a three-class sentence-level classification is a must. The latter drastically reduces accuracy of the classification algorithms (for comparison, see Ribeiro et al., 2016).

The sentence-level analysis, however, does not assign the sentiment to a specific target. For example, in the sentence "I liked Disneyland but driving there was terrible" the sentiment "terrible" relates only to driving experience, but not to Disneyland. Complicating the analysis, in a sentence "Bonaire diving was excellent" the sentiment "excellent" relates only to the target "diving," but not to Bonaire as a whole. The approaches aiding in finding the target or an entity of the sentiment are described in detail by Liu (2020).

The sentiment itself may be characterized by its orientation (also called polarity or valence) and intensity. In terms of orientation, the sentiment can be positive or negative, with some researchers also including neutral sentiment. The intensity can be measured using a variety of scales; however, for practical purposes Liu (2020) advises no more than five levels, with two levels frequently being adequate. For example, the sentiment of the statement "I hated beddings" could be -4 intensity on a one-dimensional scale $[-5, 5]$ due to the presence of word "hate" with negative valence. This method was accepted by the authors of the popular software

SentiStrength (Thelwall et al., 2010) that is based on a large dictionary containing word stems rated according to their sentiment scores (Thelwall, 2016).

The lexicon-based approach is based on a list of words and phrases together with their sentiment orientation and strength; this list is referred to as a sentiment (opinion) lexicon. In the most simplistic implementation, the software performs sentiment analysis by matching each word with the lexicon, thus, extracting the sentiment score. This sentiment score would then be reversed if negation words (such as "not") are present. In addition, modifiers may weaken or strengthen the sentiment: compare "suspicious person" with "deeply suspicious person" (Polanyi & Zaenen, 2006). The document's sentiment is then defined as a sum of sentiment scores for all words in the document or as two separate sums of positive and negative sentiments.

A specific problem in the lexicon-based approach is how to generate the sentiment lexicon (for detail, see Liu, 2020). The most straightforward approach is dictionary based. In this approach, a small manually collected seed set of sentiment carrying words with known orientation is used to search a dictionary in order to extract the synonyms, which in turn are used as new seeds. When no new candidate sentiment words are found, the generated list is manually cleaned. The shortcoming of this approach is, however, that the obtained sentiment list is generic and lacks the context. For example, the word "cold" in phrases "cold beer" and "cold person" carry opposite sentiment. This problem is tackled with the corpus-based approach, which applies a variety of approaches to extract sentiment from a collection of representative texts from the field of interest (a corpus). For example, provided that the corpus contains the phrase "he is a cold and greedy person" and that the sentiment of "greedy" is negative, we could conclude that the word "cold" is also negative. A better approach further enhances specificity by including the context in which the adjective "cold" is used.

Essentially, sentiment analysis is a classification problem. The dictionary-based approach is frequently described as an unsupervised classification, that is, classification performed without providing additional external information regarding classification patterns. A competing supervised classification approach is based on machine learning (Liu, 2020). Here, a sample of documents from the same domain is manually classified according to the sentiment expressed (those documents are called "labeled"). This sample is then used to train and validate a machine learning algorithm which is finally applied to the rest of the documents (which are called "unlabeled"). Notice that this approach does not require a list of sentiment carrying words or phrases. Instead, the sentiment is learned by the algorithm during the training process on a pre-processed huge dataset of representative documents. In terms of the machine learning models, many papers apply Naïve Bayes or SVM (Alpaydin, 2020). Recently, a new crop of machine learning models optimized for natural language processing are being successfully used to improve sentiment analysis process; among these models, the most visible is BERT (Bidirectional Representation for Transformers) developed by a Google team. The idea of BERT is to simplify learning process by introducing a new pre-training step which uses a model that is already pre-trained on generic texts. The results can then be fine-tuned

using the field-specific data, resulting in lower training data requirements and faster training process.

The comparative analysis of two approaches typically demonstrates that the machine learning algorithms outperform the lexicon-based ones when the formers are properly trained (Hailong et al., 2014). The lexicon-based methods, however, have distinct advantage by being transparent compared to the "black-box" machine learning algorithms; they require no human- and computer-intensive model training and, therefore, are not sensitive to training quality (Ibid.). The latter point illustrates critical dependency of the machine learning approach on high-quality labeling of a sample of documents by human raters; when a model is pre-trained on documents from a somewhat different domain, the advantage of the machine learning approach disappears (Kirilenko et al., 2018). Even though the lexicon-based methods rely on generic language dictionaries and, hence, are less effective in recognizing emotions in specialized texts such as tweets, they are easier to use and more robust, which frequently make them preferable.

Recently, a new crop of semi-supervised methods has appeared that radically reduces demands of the machine learning methods by injecting the unlabeled documents into the algorithm training process (Van Engelen & Hoos, 2020). These algorithms can be applied to sentiment analysis as well (Lee et al., 2019) and make the machine learning methods more user-friendly. Finally, machine-based methods can be used to improve outcomes of the lexicon-based approach (Zhang et al., 2011).

As a final note, one area closely related to sentiment analysis is emotion detection. While sentiment can be expressed in a single "negative to neutral to positive" dimension, emotion recognition involves classification into multiple emotion classes, for example, Happiness, Sadness, Fear, Disgust, Anger and Surprise (Eckman, 1992). Some researchers experiment with lexicon-based approaches, similar to those used in sentiment analysis; for example, Mohammad and Turney (2013) developed a large multi-language emotion dictionary based on Plutchik (1980) "wheel of emotions." Nevertheless, it seems that currently emotion detection is better progressing in image and audio analysis (Gajarla & Gupta, 2015), as opposed to text analysis. Indeed, one could imagine the difficulties in recognizing the emotion in a sentence "I work mostly over Zoom nowadays," which could express happiness, sadness, or be just neutral. For this reason the emoticons and emojis are frequently used in social media to aid convening writer's emotions. In a distinct line of research, the emoticons as indicators of emotions are used to successfully train emotion recognition models (Felbo et al., 2017).

In tourism and hospitality, sentiment analysis is an emerging field. The existing reviews found only 26 (Ma et al., 2018), 24 (Alaei et al., 2019), and 68 (Jain & Pamula, 2021) articles; the latter review mostly included papers published in non-tourism journals. The most comprehensive upcoming publication by Mehraliyev et al. (2021) used a systematic search and uncovered 70 articles published in hospitality and tourism journals that used sentiment analysis up to June 2020. The main venues include *Tourism Management*, followed by the *International Journal of Hospitality Management*, *International Journal of*

Contemporary Hospitality Management and *Journal of Travel Research*. Notably, ¼ of all articles was published in the first half of 2020, indicating that the interest toward sentiment analysis in tourism scholarship is very recent. Further, the absolute majority of scholarship was focused on market intelligence, with very few papers dealing with other fields such as destination management, strategic management, or social media management. Methodologically, the majority (72%) of the papers used the lexicon approach; half of those papers employed one of the four most popular packages: SentiStrength (Thelwall et al., 2010), AFINN (Nielsen, 2011), LIWC (Pennebaker et al., 2001), or SentiWordNet (Baccianella et al., 2010). Overall, it seems that tourism and hospitality academics only recently discovered sentiment analysis and methodology is mainly based on the most accessible and widely available approaches and packages.

2 Theoretical Foundations

The problem of unearthing sentiment in texts was recognized as a distinct aspect of content analysis in the first half of the twentieth century. To differentiate on people's evaluative judgments and affective responses to stimuli (issues, topics, etc.) conveyed in texts, Osgood et al., 1957) identified three aspects of meaning: Evaluation, Potency, and Activity (EPA system) which, taken together, make three-dimensional space where the meaning of each word can be located. Evaluation dimension represents cognitive appraisals on the good-bad continuum. Potency reflects the intensity of the evaluative judgments on the strong-weak continuum. The last dimension, Activity, is represented by the active-passive pair of anchors. The EPA three-factor system was determined through a factor analysis of a large collection of semantic-differential scales and provided the foundation to the attitude research, and numerous studies supported validity of the approach (Heise 1970). Research has also found the stability of EPA structure across various cultures (Osgood 1964; Jakobovits 1966). Not only adjectives but also concepts can be tagged with the meaning along the EPA dimensions. For example, the concept of "war" would score very high on bad, strong, and active dimensions, while the word "baby" would likely score as highly positive, highly weak, and somewhat passive.

Currently, a large amount of works on sentiment analysis involves determining valence, which can be roughly equated with the evaluative EPA dimension; that is, where the sentiment is identified as good/bad; positive/negative or favorable/unfavorable (e.g., Pang and Lee 2008; Liu 2015). Valence, arousal, and dominance are the three dimensions of the Russell's (1980) core affect framework for study of emotions, where valence is associated with pleasure and is also placed on the positive/negative scale. For example, joy is considered as carrying positive valence and, thus, indicates positive sentiment, while anger is indicative of a negative sentiment. The intensity of the emotion can be measured by how far from a neutral point on the positive–negative scale it is located: e.g., wrath is judged as a stronger emotion than anger. This idea that various concepts, descriptors, and affective states

have valence and, thus, can be assigned a score on a positive–negative dimension, lies at the foundation of the automated sentiment analysis (e.g., Pang & Lee, 2008).

3 Practical Demonstration

This section provides a brief explanation of the methodology steps, while detailed implementation will be covered in the case study discussed in the next section. Generally, the analysis starts with data cleaning and normalization. The goal of this step is broadly described as increasing data quality and cohesiveness. That may include the following:

- Removal of noise and artifacts such as HTML tags, pictograms, and unwanted characters.
- Tokenization and decapitalization, which breaks textual data into the atomic analysis units, for example, lower-case words.
- Stopword removal: examples include the words like "in," "of," "are," "the," and "it"; One popular list of stopwords comes from the Natural Language Toolkit (nltk.org).
- Resolving the attached words such as encountered in hashtags, e.g., "#AwesomeDay".
- Spelling and grammar correction.
- Resolving negations (e.g., "no good").
- Part-of-speech (POS) tagging with retaining the words of interest (e.g., adjectives and nouns only).
- Lemmatization or stemming. This step reduces the inflectional and derivational forms of words to a common base form, which in turn increases data cohesiveness. This step is especially important when the machine learning approach is used but may be skipped otherwise.

In no way those steps should be applied without validation. For example, multiple recommendations of a popular tourist guide Mr. Luck or Ms. Grim may dramatically skew distribution of park visitors' sentiment. Spell checking reviews of Manuel Antonio National Park may replace "Manuel" for "manual." As a solution, customization of data normalization algorithms is a must.

When the rule-based lexicon approach is used, the next step includes matching the tokens with one of the sentiment or emotion dictionaries, as discussed in Software section. The machine learning approach will include manual processing of a sample of documents classifying them according to expressed emotions. To improve reliability, it is recommended to attract multiple raters. The classified ("labeled") data then are used to train a classifier such as Naïve Bayes, SVM, or many others, followed by algorithm validation. Finally, the trained algorithm is used to process the unlabeled data.

During the final step, the outcomes are validated, analyzed, and interpreted. The following two sections present a case study demonstrating how those steps are realized in practice.

4 Research Case 1: Lexicon-Based Sentiment Analysis[1]

In this section, we demonstrate how the lexicon-based sentiment analysis is used to understand the sentiment expressed by visitors to Manuel Antonio National Park, Costa Rica. Manuel Antonio is the smallest Costa Rica national park (land area 6.8 km^2) famous for its beaches, wildlife viewing opportunities, beauty of landscapes, and hiking opportunities. Owing to the park's proximity to the national capital (130 km), the park is visited by 150,000 tourists annually, making it the busiest park in the country (govisitcostarica.com).

The following case study shows how sentiment analysis was applied to TripAdvisor data to measure the polarity of tourists' reviews covering personal opinions and real travel events. To demonstrate both approaches covered in this article, this section covers both the supervised feature-based machine learning and the rule-based lexicon approaches. The data includes 2700 TripAdvisor park reviews from February 2016 to September 2020 in all languages. All non-English reviews were translated to English by Goggle Cloud Translate. Then, reviews were normalized following the steps discussed in the previous section.

The scope of the project did not allow us to do the manual classification of sentiment reflected in customer reviews as required by the machine learning approach; hence, the decision was made to use the lexicon-based approach. Specifically, two widely used lexicon methods, SentiWordnet and VADER (Valence Aware Dictionary for Sentiment Reasoning) were applied to extract tourists' sentiment about the park. Both methods are based on opinion (sentiment) lexicons which contain the words with positive sentiment such as happy or enjoyable and negative sentiment such as terrible or bad. The sentiment is then defined by mapping the text into the respective lexicon (Al-Shabi, 2020). SentiWordNet (Baccianella et al., 2010) is based on the WordNet (wordnet.princeton.edu) lexical database of English language (Bonta & Janardhan, 2019). The algorithm assigns each text three scores: objectivity, positivity, and negativity, which range from 0 to 1.

As opposed to SentiWordNet, optimized for texts written in general English, VADER (Valence Aware Dictionary for Sentiment Reasoning) is specifically optimized for microblogs (Gilbert & Hutto, 2014). For each review, Vader generates four sentiment scores: text neutrality, positivity, negativity score, and a compound summary score. The compound score ranges between −1 for the most negative sentiment and 1 for the most positive. A typical sentence with positive sentiment

[1]The sentiment analysis code used in this article is publicly available at https://github.com/luyuwang1993/Sentiment-Analysis/tree/dev-sentiment

Table 1 Sentiment analysis validation for SentiWordNet and Vader algorithms

	SentiWordNet	Vader
Accuracy	0.681	0.681
Precision	0.711	0.710
Recall	0.872	0.990
F1 measure	0.783	0.827

would have a compound score greater than 0.05, and a negative sentiment sentence would have a compound score lesser than −0.05.

In order to validate sentiment predictions, we manually labeled 300 reviews (Table 1). Notice a slightly better performance of VADER; this is to be expected since this algorithm is optimized for social media as opposed to SentiWordNet, which would be preferable for texts written in standard English. Also, notice multiple metrics used for performance evaluation; the data distribution and intended application of the sentiment analysis indicate which metrics is the most useful. In our case, the sentiments were highly imbalanced with many more positive reviews than the negative ones, which makes F1 measure a preferable indicator of model quality. Another good choice of classification quality is Cohen's kappa.

Finally, the reviews carrying negative sentiment were manually processed to find the main topics of dissatisfaction shared by park visitors. The analysis revealed five shared areas of complaint: overcrowding, unprofessional staff, trail condition, opportunistic locals selling parking tickets, and monkeys thieving personal belongings.

5 Research Case 2: Machine Learning Sentiment Analysis

In this section, we demonstrate how the machine learning sentiment analysis is used to understand the sentiment expressed by the airline travelers. The dataset[2] represents scraped Twitter data representing six US airlines, subsequently processed by volunteers who classified the tweets into three categories: positive, negative, and neutral, together with the volunteer's confidence score. For this case study, we selected only the tweets with 0.6 or better confidence scores, which removed 1.6% of tweets. Together, that constituted 14,402 airline reviews.

The reviews were pre-processed as described in the How-to section and then vectorized using the Term Frequency-Inverse Document Frequency (TF-IDF) metrics (Liu, 2020). Further, the data was split into training and testing sets with 90% of tweets used for training and 10% reserved for testing. Three models, Bernoulli and Multinomial Naïve Bayes and SVM, were trained on the training dataset; then, models were validated on the testing dataset.

[2] https://www.kaggle.com/crowdflower/twitter-airline-sentiment

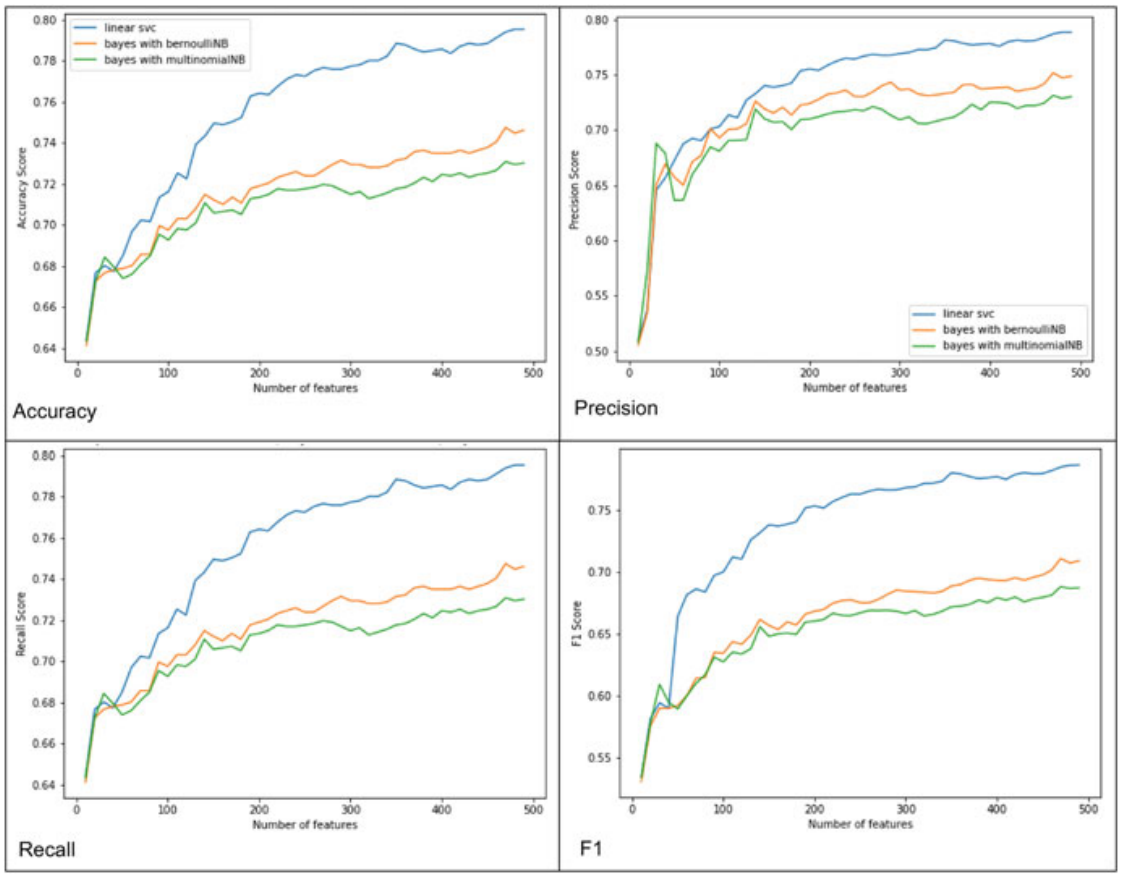


Fig. 1 Performance comparison of algorithms

An important step in the machine learning approach is feature selection. The higher the number of features (e.g., words) selected for model training, the better model predictions on the training data are. However, model performance on the testing dataset follows the bell shape and is reduced when the number of features is too high ("model overfitting"). In addition, a large number of features negatively affect model complexity, requiring expensive computer resources. A performance comparison of models utilizing a progressively increasing number of features (Fig. 1) was used to make decision on the optimal number of features. Notice that after the initial fast growth the performance curve eventually flattens down as more and more features (words) are taken into account by the machine learning algorithm. Hence, the decision was made to limit the number of features at $N = 300$.

Similar to the lexicon-based approach, the final decision on satisfactory model implementation was made based on the analysis of multiple indicators of model performance on an independent dataset selected for model testing (Table 2). Similarly to the Manuel Antonio park, the dataset is highly unbalanced: while the natural park reviews are predominantly positive, the airline reviews are predominantly negative. In our case, 61% of the tweets were negative, 22% neutral, and only 17% positive, which makes F1 measure preferable for judging model performance. Overall, SVM model was selected over two other.

Table 2 Overall performance of all tested algorithms (at 300 features for machine learning approaches)

	Bernoulli NB	Multinomial NB	SVM
Accuracy	72.94%	71.48%	77.72%
Precision	73.63%	70.92%	76.90%
Recall	72.94%	71.48%	77.72%
F1 measure	68.47%	66.65%	76.85%

Service Section

Main Application Fields: Computational study of people's emotions, attitudes, and opinions, usually expressed in a written text. In tourism, the primary area of application is the analysis of visitors' reviews of the hotels, destinations, points of interest, and similar.

Limitations and Pitfalls: Uncritical use of the computational sentiment analysis without deep understanding of the methods results in unwarranted predictions. For the lexicon-based approach, the dictionary used by the algorithm and the analyzed data much originate from similar domains (e.g., social media). For the machine learning approach, manual classification of a sample of data from same domain is a must. Both approaches require accurate validation on an independent manually classified dataset using multiple performance indices; the latter should account for data distribution and the purpose of analysis.

Similar Methods and Methods to Combine with: The sentiment analysis is frequently used together with content analysis and share many approaches and methods.

Code: The Python code is available at: https://github.com/DataScience-in-Tourism/Chapter-17-Sentiment-Analysis

Further Readings and Other Sources

Books: "Sentiment Analysis: Mining Opinions, Sentiments, and Emotions" by Bing Liu (2020) is a good introductory text covering all important aspects of computational analysis of sentiment and emotions as well as the most popular algorithmic approaches and major developments in the field.

Videos: "Sentiment Analysis: extracting emotion through machine learning" by Andy Kim. A 10-minutes TED talk introducing sentiment analysis. https://www.youtube.com/watch?v=n4L5hHFcGVk

Web sites: Medium.com, towardsdatascience.com, and KDnuggets.com sites have an excellent set of AI articles including those covering sentiment analysis.

References

Alpaydin, E. (2020). *Introduction to machine learning*. MIT Press.

Alaei, A. R., Becken, S., & Stantic, B. (2019). Sentiment analysis in tourism: Capitalizing on big data. *Journal of Travel Research, 58*(2), 175–191.

Al-Shabi, M. A. (2020). Evaluating the performance of the most important lexicons used to sentiment analysis and opinions mining. *IJCSNS, 20*(1), 1.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010, May). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Lrec* (Vol. 10, no. 2010, pp. 2200–2204).

Bakshi, R. K., Kaur, N., Kaur, R., & Kaur, G. (2016, March). Opinion mining and sentiment analysis. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 452–455). IEEE.

Bonta, V., & Janardhan, N. K. N. (2019). A comprehensive study on lexicon based approaches for sentiment analysis. *Asian Journal of Computer Science and Technology, 8*(S2), 1–6.

Eckman, P. (1992). An argument for basic emotions. *Cognitive Emotions, 6*, 169–200.

Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., & Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524*.

Gajarla, V., & Gupta, A. (2015). *Emotion detection and sentiment analysis of images*. Georgia Institute of Technology.

Gilbert, C. H. E., & Hutto, E. (2014, June). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14)* (Vol. 81, p. 82). *Available at (20/04/16)* http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf.

Hailong, Z., Wenyan, G., & Bo, J. (2014, September). Machine learning and lexicon based methods for sentiment classification: A survey. In *2014 11th web information system and application conference* (pp. 262–265). IEEE.

Heise, D. R. (1970). The semantic differential and attitude research. *Attitude Measurement*, 235–253.

Hu, M., & Bing, L. (2004). Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*.

Jain, P. K., & Pamula, R. (2021). A systematic literature review on machine learning applications for consumer sentiment analysis using online reviews. *Computer Science Review, 41*. Available at: https://arxiv.org/pdf/2008.10282.pdf

Jakobovits, L. A. (1966). Comparative psycholinguistics in the study of cultures. *International Journal of Psychology, 1*(1), 15–37.

Kirilenko, A. P., Stepchenkova, S. O., Kim, H., & Li, X. (2018). Automated sentiment analysis in tourism: Comparison of approaches. *Journal of Travel Research, 57*(8), 1012–1025.

Lee, V. L. S., Gan, K. H., Tan, T. P., & Abdullah, R. (2019). Semi-supervised learning for sentiment classification using small number of labeled data. *Procedia Computer Science, 161*, 577–584.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies, 5*(1), 1–167.

Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.

Liu, B. (2020). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.

Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining text data* (pp. 415–463). Springer.

International Journal of Contemporary Hospitality Management, In second review.

Ma, E., Cheng, M., & Hsiao, A. (2018). Sentiment analysis – A review and agenda for future research in hospitality contexts. *International Journal of Contemporary Hospitality Management, 30*(11), 3287–3308.

Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence, 29*(3), 436–465.

Mehraliyev, F., Chan, I. C. C., & Kirilenko, A. P. (2021). Sentiment analysis in hospitality and tourism: A thematic and methodological review. *International Journal of Contemporary Hospitality Management*.

Nielsen, F. Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.

Osgood, C. E. (1964). Semantic differential technique in the comparative study of cultures. *American Anthropologist, 66*(3), 171–200.

Osgood, C. E., Suci, G. J., & Tannenbaum, P. H. (1957). *The measurement of meaning* (No. 47). University of Illinois Press.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Found Trends Inf Retr, 2*(1–2), 1–135.

Pennebaker, J. W., Francis, M. E., & Booth, R. J. (2001). Linguistic inquiry and word count: LIWC 2001. *Mahway: Lawrence Erlbaum Associates, 71*(2001), 2001.

Plutchik, R. (1980). A general psychoevolutionary theory of emotion. In R. Plutchik & H. Kellerman (Eds.), *Emotion: Theory, research and experience, theories of emotion* (Vol. v. 1, pp. 3–33). Academic Press.

Polanyi, L., & Zaenen, A. (2006). Contextual valence shifters. In *Computing attitude and affect in text: Theory and applications* (pp. 1–10). Springer.

Ribeiro, F. N., Araújo, M., Gonçalves, P., Gonçalves, M. A., & Benevenuto, F. (2016). Sentibench – a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science, 5*(1), 1–29.

Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology, 39*(6), 1161–1178.

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., & Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology, 61*(12), 2544–2558.

Thelwall, M. (2016). *Sentiment analysis for small and big data. The SAGE handbook of online research methods* (pp. 344–355).

Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning, 109*(2), 373–440.

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., & Liu, B. (2011). *Combining lexicon-based and learning-based methods for twitter sentiment analysis*. HP Laboratories, Technical Report HPL-2011, 89.

Topic Modelling

Modelling Hidden Semantic Structures in Textual Data

Roman Egger

Learning Objectives

- Understand the main intuition behind the most relevant topic modelling techniques
- Appreciate the application of topic modelling in the tourism industry
- Identify various hurdles and pitfalls causing bad topic quality
- Apply LDA, NMF, CorEx, Top2Vec, and BERTopic in Python to a dataset about Airbnb experiences

1 Introduction and Theoretical Foundations

It has been estimated that approximately 80% of the Internet's data is available in text format (Anandarajan et al., 2019). In particular, the large amount of user-generated content (UGC) produced every day fosters the increase of unstructured text data (Rossetti et al., 2015), especially in regard to the tourism sector (Li et al., 2019). As a result, the unstructured and unorganised nature of information in the digital sphere complicates the process of traditional quantitative and qualitative analytics. Although topic modelling is based on quantitative methods, it still serves as a means for qualitative research (Evans, 2014; Nikolenko et al., 2017). It can be described as an inductive approach with quantitative measurements and is therefore suitable for descriptive and explorative analyses (Banks et al., 2018).

R. Egger (✉)
Salzburg University of Applied Sciences, Innovation and Management in Tourism, Urstein (Puch), Salzburg, Austria
e-mail: Roman.egger@fh-salzburg.ac.at

R. Egger (ed.), *Applied Data Science in Tourism*, Tourism on the Verge,
https://doi.org/10.1007/978-3-030-88389-8_18

To gain a deeper understanding of a large volume of text, topic modelling has long been considered an effective technique in the lead disciplines of tourism, including marketing and management (Hannigan et al., 2019; Reisenbichler & Reutterer, 2019). Rooted in machine learning and natural language processing (NLP), topic modelling is a method that attempts to efficiently structure large amounts of text, based on co-occurrences of terms in similar texts (Daenekindt & Huisman, 2020). For example, words like "snow", "ski", and "snowboard" have a semantic relationship to one another; therefore, in a text document, they could be expected to form a topic called "winter sports". If only one topic appears in a document, it is called a single-membership model. In most cases, however, mixed-membership models, in which documents consist of a mix of numerous different topics (Maier et al., 2018), are to be found. Thus, this chapter will concentrate exclusively on this variant. In summary, topic modelling refers to a group of methods that attempts to identify topics and their prevalence within a corpus (a collection of documents) in an automated way (Sotomayor & Bellono, 2019).

Many different topic modelling approaches have been developed in recent years, with Latent Dirichlet Allocation (LDA) being the best known and most widely used algorithm (Jockers & Thalken, 2020). Other less known approaches such as Latent Semantic Analysis (LSA) (Landauer et al., 1998), Structural Topic Modelling (STA) (Lindstedt, 2019), Non-Negative Matrix Factorisation (NMF) (Wei et al., 2003), Correlation Explanation (CorEX) (Gallagher et al., 2017), Top2Vec (Angelov, 2020a), or BERTopic (Grootendorst, 2021), amongst others, often outperform and can still be considered exciting alternatives, depending on the requirements. As social media posts are often the data basis for topic modelling projects, in which hotel and restaurant reviews and/or Facebook, Twitter, or Instagram posts are analysed, the fact that these texts are usually short text sections is a particular challenge. The evaluation of topic modelling methods is, therefore, of particular importance. For example, Albalawi et al. (2020) evaluated different topic modelling methods for short-text data like Facebook or Twitter posts.

2 Topic Modelling Approaches

As introduced in the previous paragraph, a multitude of different topic modelling approaches exist. In the following section, five methods particularly worth mentioning will be presented and discussed. First, we will turn to the Latent Dirichlet Allocation (LDA), which is typically viewed as the standard approach. Thereafter, Non-Negative Matrix Factorisation (NMF), also popular in the social sciences, will be explained, followed by CorEx. Although much less known, the latter rival algorithm will be discussed in addition to being part of the practical demonstration as it, according to the author's experience, often achieves better results than LDA and also offers a possibility to define anchor words that enable a "seeded" or "guided" topic modelling process. The fourth approach, Top2Vec, which is a new and, therefore, unfamiliar approach, can be seen as having high potential as well.

Lastly, BERTopic, an approach that makes use of BERT generated word embeddings, will conclude this section.

2.1 Latent Dirichlet Allocation (LDA)

LDA, developed by Blei et al. (2003), is currently the most popular topic model algorithm and is implemented in numerous toolkits such as Gensim,[1] Stanford TM toolbox,[2] Machine Learning for Language Toolkit (MALLET[3]) (Albalawi et al., 2020), or Promoss.[4] The basic assumption of the LDA approach is that documents with similar topics also use similar word groups. By searching for groups of words that often co-occur in documents within a certain corpus, latent topics can be found (Evans, 2014). It is also assumed that documents have probability distributions (Dirichlet distribution) over latent topics, and topics have probability distributions over words (Blei, 2012b) (Fig. 1).

On the left side, one can see certain numbers of topics reflecting the distribution of individual terms across the entire document. The assumption is that each

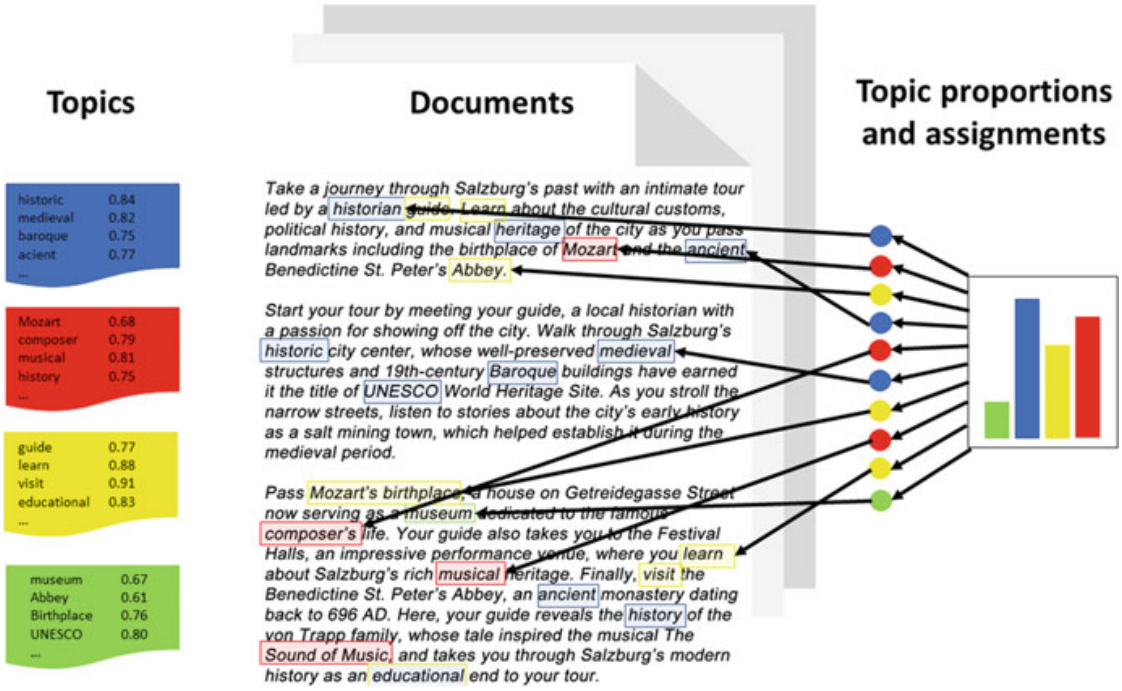


Fig. 1 The intuition behind LDA. Source: Author's presentation based on Blei (2012b)

[1] https://radimrehurek.com/gensim/

[2] https://nlp.stanford.edu/software/tmt/tmt-0.4/

[3] http://mallet.cs.umass.edu/

[4] https://github.com/ckling/promoss

document is generated by choosing a distribution over the topics (topic proportions and assignments/histogram). Thereafter, a topic assignment is chosen for each word (coloured circles), and the word is selected from the corresponding topic. As such, each document is composed of several topics, where each topic consists of words that characterise it and is treated as a "bag-of-words" with no importance attached to word order. The algorithm finds related terms by randomly assigning each word to a topic. This assumes that the number of topics has been defined in advance. In a second, iterative step, each word is reassigned based on the probability of a certain word belonging to a topic as well as the probability of a topic determining the document. The algorithm then calculates these probabilities and reassigns keywords until the model converges.[5]

It is essential to understand that these are not hard clusters like in a K-means clustering. In contrast, the result is an allocation of different topics with different weights per document. For example, a document may be 65% dominated by topic A and only contain 20% of topic B and 15% of topic C. For each topic, the words with the highest probability of belonging to this specific cluster are obtained. It is therefore the user's task to find a meaningful topic title/umbrella term for the words. The output could appear as follows: document A is assigned to Topic #3, where the words with the highest probability for this topic include "ski, snowboard, sledge, . . . ice skating". Topic #3 could thus be labelled "winter sports". Since a topic is formed by finding terms that belong together in terms of probability, it should be noted that the number of terms belonging together largely depends on the size of the bag-of-words. Therefore, it is advisable to divide large text units (e.g. whole papers or even whole books) into segments in order to extract topics that might otherwise disappear.

Due to the popularity of LDA, numerous extensions and adaptations have been developed over time. For instance, some examples include labelled LDA (supervised) (Ramage, Hall, et al., 2009), polylingual LDA topic models (Mimno et al., 2009), LDA models discovering topics over time (Hu et al., 2015; Ungar et al., 2006). Other researchers have extended LDA to a joint sentiment-topic model, enabling simultaneous extraction of both topics and sentiments to improve rating prediction and tourist recommendation applications (Rossetti et al., 2016). Moreover, when constructing the latent topics, a correlated topic model (CTM) was proposed. The CTM incorporates the correlation between words to optimise data fitting, which is particularly helpful when a large amount of corpus-based data is involved (Loureiro et al., 2020). Stemming from this CTM idea, a recent study advanced LDA to a structural topic model (STM) in order to assist hoteliers in identifying various aspects of consumers' dissatisfaction in online reviews (Hu et al., 2019). Specifically speaking, the STM distinguishes itself from LDA in that it considers document-level metadata when estimating a topic model (Roberts et al.,

[5] Assuming we want to extract 8 topics (T) with 1000 tokens (n) from our corpus, the computation of the Bayesian probability formula would be 81,000 (i.e. it would require an enormous amount of computing power). Therefore, Gibbs sampling is used at this point to estimate the result. For a detailed explanation, it is recommended to read Sotomayor and Bellono (2019).

2019). Similarly, research conducted by Park et al. (2018) was one of the first in the hospitality domain to incorporate a more holistic application of topical content analysis through the use of STM.

2.1.1 LDA Hyperparameters

The quality of the extracted topics depends not only on the input text but also on the hyperparameters of the LDA model. Depending on which hyperparameters are chosen, the extracted topics can be very general or very specific (Blair et al., 2020). Essentially, there are three crucial hyperparameters that must be defined. The first value is the number of topics (**num_topics**) to be extracted. LDA generates the number of topics based purely on quantitative calculations, which is why social scientists often rely on a "middle-ground" (Lesnikowski et al., 2019) approach to select the **K** topics. This combines statistical parameters of topic stability with an assessment of interpretability done by experts. It is recommended to run several experiments and ultimately select the number of topics with the highest coherence score (Mohammed & Al-augby, 2020).

The other two hyperparameters are alpha and eta (sometimes also referred to as beta). The **alpha** value controls the prior distribution over the topic weights in each document and, thus, the mix of topics for a document. A higher alpha smooths out the document preference over topics and results in documents with a greater mix of topics. **Eta** (a.k.a. **beta**), on the other hand, controls prior distribution across word weights in each topic. A higher eta smooths out the topic preference across words and leads to topics that are likely to receive more words. It is recommended to perform a grid-search[6] in order to identify the optimal LDA hyperparameter values.

2.2 Non-negative Matrix Factorisation (NMF)

Non-negative Matrix Factorisation (NMF) is a decompositional method and belongs to the family of linear algebra algorithms. Therefore, NMF, unlike LDA, is a non-probabilistic algorithm. NMF attempts to decompose a term-document matrix A (input) as a product of two matrices W and H in such a way that both have rank k (and the total size of W and H is significantly smaller than A) (Papilloud & Hinneburg, 2018) (Fig. 2). Another condition is that all entries of W and H must be non-negative (Lee & Seung, 1999). This helps to prevent difficulties that may arise when interpreting topics with negative entries for certain words (Wang & Zhang, 2013).

Preprocessing of the documents involving steps such as lowercasing, removing stopwords, lemmatising/stemming, RegEx with removing brackets, punctuation,

[6] See the Jupyter Notebook at the books Github-Profile for more details.

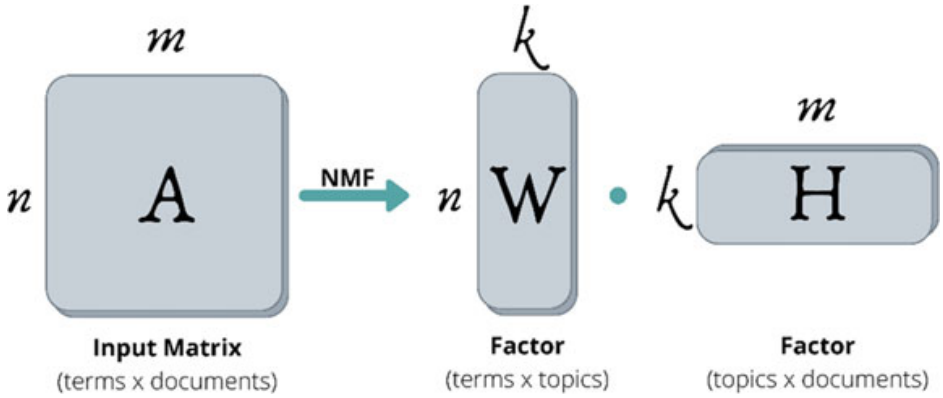


Fig. 2 Non-negative matrix factorisation. Source: Adapted from Kuang et al. (2017)

etc., should be performed. Term-document matrix should typically be TF-IDF normalised, and the number of topics k needs to be defined beforehand.

Let's take three sentences (documents) as an example to visualise NMF's intuition and select $k = 2$ topics:

Information is the lifeblood of tourism
The internet has altered the tourism industry
Covid-19 challenges the hospitality industry

A - Term Document Matrix

	Document 1	Document 2	Document 3
Information			
lifeblood			
tourism			
internet			
altered			
industry			
Covid-19			
challenges			
hospitality			

W - Weights for terms

	Topic A	Topic B
Information		
lifeblood		
tourism		
internet		
altered		
industry		
Covid-19		
challenges		
hospitality		

t

	Document 1	Document 2	Document 3
Topic A			
Topic B			

After decomposition of the original n tokens by k topics *(W)* and k topics by the original m documents *(H)*, we get two non-negative matrices as a result.

2.3 Correlation Explanation (CorEX)

Gallagher et al. (2017) proposed CorEx as a hierarchical topic modelling approach with minimal domain knowledge. It can be used for large corpora and allows for the integration of domain knowledge by seeding anchor words, making it a semi-supervised approach (Greg Ver Steeg, 2016). The inclusion of anchor words is an optional extension that allows the user to interact with the corpus and to explore the content in an innovative way. When compared with LDA, Gallagher et al. (2017) identified numerous advantages of CorEx and emphasised that higher homogeneity can be reached.

Topics are considered latent factors that may or may not appear in a document. By adding this binary information as input to another layer, hierarchical topic modelling can be achieved. Another advantage over other topic modelling methods is that the number of topics that need to be selected can be easily estimated. Since each topic explains a certain proportion of the total correlation (TC), further latent topics can be added until an increase in the total correlation is only insignificant (Greg Ver Steeg, 2016).

The optional anchoring strategy helps to reveal a topic that, had it been unsupervised, may not have appeared in the first place (Reing et al., 2016). If domain knowledge exists, anchor words such as "Covid", "hotel", and "cancellation" can be used to extract a topic around these terms. Anchor words can be assigned to a single topic or to several topics (Cai et al., 2018). Additionally, the strength of anchor words can also be defined; nevertheless, the rule of thumb is that values between 1 and 3 gently nudge a topic, whereas values higher than 5 strongly encourage a topic (Greg Ver Steeg, 2016). With regard to any preprocessing steps for this approach, a binarisation of documents is suggested.

2.4 Top2Vec

An approach, still largely unknown due to its recent introduction, is Top2Vec. Although new topic modelling algorithms are appearing constantly, this approach along with the corresponding Jupyter notebook was chosen to be presented in this chapter since Top2Vec eliminates some of the weaknesses that arise in the LDA and NMF approaches. Top2Vec uses the joint semantic embedding of documents and words to find topic vectors and does not require the specification of the number of topics to be extracted or any preprocessing. Thus, steps such as the removal of stopwords, stemming, or lemmatisation are rendered unnecessary (Angelov, 2020a). It additionally contains a built-in search function to search for topics by keywords and for documents by topics as well as to find similar words and similar documents (Angelov, 2020b).

Top2Vec uses Doc2Vec or a pre-trained model to find the numeric representation of the given document. By doing so, the semantic relationships between similar

documents and similar words are preserved (Egger, 2022). UMAP is then used to perform a dimensionality reduction and to identify dense areas with HDBSCAN. For these dense areas, a topic vector is ultimately created by taking the arithmetic mean of all the document vectors of a cluster and assigning each document a topic number (Weng, 2020). Finally, the resulting topic vectors are embedded together with the word and document vectors. The distance between the vectors can then be interpreted as semantic similarity (Angelov, 2020a).

2.5 BERTopic

The last topic model method in this section is BERTopic, which also follows an embedding approach. It uses state-of-the-art sentence-transformers and a class-based version of TF-IDF to create interpretable topic clusters (Grootendorst, 2020). BERTopic can be used for any language if an embedding model already exists for it.[7] As a result, this even allows for the performance of multilingual topic modelling in cases where the documents contain several different languages. Overall, the possibility of using self-created embedding models instead of pre-trained embeddings may prove to be particularly useful. In numerous experiments, it seems that tourism domain-specific models provide much more accurate results than publicly-available pre-trained models.[8] According to Grootendorst (2021), no preprocessing is needed unless the used transformer model requires it. The algorithm allows the manual or automatic merging of similar topics according to the cosine similarity between the vectors. As with other topic modelling approaches, topics are shown as a set of words, and the researcher can define the n-gram range. BERTopic generates a value for the topics and a probability score as output, demonstrating the likelihood that a document belongs to any specific topic. Additionally, BERTopic comes with its own topic visualisation, similar to LDAvis.

3 Topic Modelling Limitations and Challenges

We have now learned about different topic modelling approaches, each of which comes with its own set of advantages and disadvantages and is suitable for particular application scenarios. However, for all approaches, it is up to the researcher to interpret the topic clusters and make sense of them (Mimno et al., 2011). Most topic modelling techniques require domain knowledge to correctly interpret the connections and relationships between the extracted terms of a latent topic. Although the identified topics make sense from a mathematical perspective, they often do not

[7] Pretrained models available: https://www.sbert.net/docs/pretrained_models.html

[8] A tourism-domain specific embedding is available at:

match human judgement (Chang et al., 2009; Nikolenko et al., 2017). As such, the poor quality of identified topics seems to stunt the acceptance of statistical topic models outside the machine learning community (David Mimno et al., 2011).

Cai et al. (2018) mention the following reasons as to why extracted topics may appear suspicious:

1. Two or more themes are merged into one theme.
2. Two themes are extracted that, to humans, look like duplicates.
3. Extracted keywords of topics do not seem to make sense.
4. Topics contain too many generic terms.
5. Topics that are based on seemingly unrelated terms are extracted.
6. Topics do not match human judgement.
7. Topics appear irrelevant.
8. The relationship between topics and documents is not apparent.
9. Several similar topics are extracted.

However, in most cases, the poor quality of topic models does not lie in the algorithms themselves but, rather, in an insufficient understanding of how to apply them. In many studies, there is insufficient documentation on how the input text data was preprocessed, how hyperparameters were defined, how models were evaluated, how reliability and interpretability were increased, and how the results were validated. All of these suggest that default values are used quite often and results are published without further processing. Yet, this is problematic as DiMaggio et al. (2013) note that "producing an interpretable solution is the beginning, not the end, of an analysis" (p. 586).

For most algorithms, preprocessing data is a prerequisite to achieving good results, and the saying "garbage in–garbage out" most certainly applies. Lim and Buntine (2014) stress the importance of preprocessing as the basis of a successful topic modelling process in which a number of standard steps and also the order of performing them is emphasised (Denny & Spirling, 2018). The standard steps include language recognition, if necessary, and correction of misspellings, tokenisation, lowercasing of words, removing non-informative features such as punctuation and special characters like emojis, numbers, HTML-codes, and URLs, filtering out stopwords and highly frequent and infrequent terms, and stemming or lemmatising (Kadhim et al., 2014; Maier et al., 2018). For further information on these steps, refer to Chapter 15 ("Introduction: Natural Language Processing").

When it comes to social media posts in particular, the underlying text elements may be too short to achieve meaningful results. In this case, it is advisable to either aggregate texts (Liangjie & Davison, 2010; Zhao et al., 2011) or apply special algorithms for short-text analysis (Qiu & Shen, 2017; Vo & Ock, 2015). Ultimately, the quality of the topics also depends on hyperparameter tuning. Depending on the modelling approach, numerous settings can be adjusted, which, in turn, greatly impact the results. For example, LDA uses random initialisation and stochastic inference, making the results non-deterministic, which is why Maier et al. (2018) recommend always using reliability checks to verify the robustness of the results.

One of the biggest challenges in topic modelling, however, is to determine the optimal number of topics. This important step is often only estimated or determined by trial and error without paying much attention to quantitative key figures or qualitative insights. It has been observed that researchers tend to choose a larger number of topics in order to be able to make fine-grained delimitations. Yet, as a result, this may lead to entities that can no longer be meaningfully distinguished because they are too similar (Grimmer & Stewart, 2013). Too few topics, on the other hand, may lead to entities that are too broad (Evans, 2014), and revealed insights can no longer be isolated and interpreted. As David Mimno et al. (2011) show in their study, there is a strong correlation between the number of topics and the judgement of nonsense topics by domain experts. However, it is also possible to determine key figures that allow an optimal number of topics to be derived. For example, Greene et al. (2014) developed a stability analysis[9] tool for topic models, which measures the robustness to perturbations in the data so as to identify an appropriate number of topics.

3.1 Evaluating and Interpreting Topics

Interpreting and evaluating topic models can often be challenging (Wallach et al., 2009) and frustrating since, with unsupervised approaches like LDA, the interpretability of the results is not always guaranteed (Röder et al., 2015). The result is a list of words ranked by relevance, and naming the topic based on these words can indeed be difficult (Hindle et al., 2013).

There are numerous approaches to measuring the quality of topic models. The evaluation should correctly assess the generalisation capability of a topic model, being computationally efficient and independent of a specific use case (Wallach et al., 2009). Since there is no gold standard for the data, the question of how to quantify the quality of topics becomes the first challenge. Statistical topic models are mostly evaluated either by extrinsic methods or quantitative intrinsic methods (Mimno et al., 2011; Wallach et al., 2009). Extrinsic methods test for associations between topics using data that was not used while learning the topic model. This can include annotations or metadata of documents. If a connection between these external data points and the topics exists, it should be possible to interpret them meaningfully (Papilloud & Hinneburg, 2018). On the other hand, for intrinsic evaluation, the existing text corpus is used to make a quantitative statement about the quality of the topics based on word probabilities estimated during the learning of the model (Blair et al., 2020).

[9] https://github.com/derekgreene/topic-stability

An established method for latent variable models like LDA is the calculation of topic coherence, where the notion coherent refers to a set of statements that support each other (Röder et al., 2015). Thus, a coherent set of facts can be interpreted in a context that includes all or most of the facts. In very simplified terms, topic coherence measures how often the top words co-occur in a topic

$$\text{Topic Coherence (T)} = \Sigma_{w1,\ w2}\ \log\ \frac{\text{docs with both words} + 1}{\text{docs with word } w2.}$$

Since the overall goal should be to generate a well interpretable topic result, it is recommended to create several candidate models (Maier et al., 2018) by changing the hyperparameters (how to perform this is shown in the Jupyter notebook for this chapter). The comparison of the topic coherence between the different models already provides a good hint as to which model to use. Additionally, word intrusion can be used as a form of human judgement in which a person is shown high probability words for a specific topic without knowing the word list of the topic (Fig. 3). The assumption is that a topic model is good if the person can correctly guess which word should not belong. The model precision is then calculated by dividing the number of the true intruder's correct guesses by all guesses. The higher the score, the better the model (Chang et al., 2009).

Intertopic distance maps are an additional option that can be used to support a visual decision. These can be created as scatterplots (after performing dimensionality reduction); in the case of LDA, the Python module LDAvis (Sievert & Shirley, 2014) is available for visualising topics. Based on multidimensional scaling of the topics, their size and the most relevant keywords per topic are presented. This visualisation especially helps in identifying overlapping topics and getting a better feeling in terms of selecting the optimal number of K topics (Fig. 4).

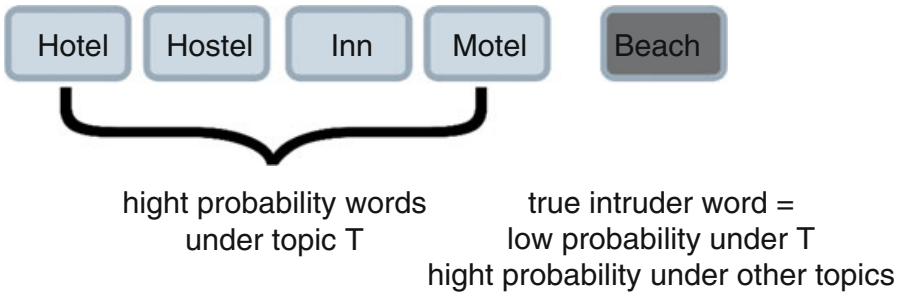


Fig. 3 Word intrusion. Source: Adapted from Dietz (2016)

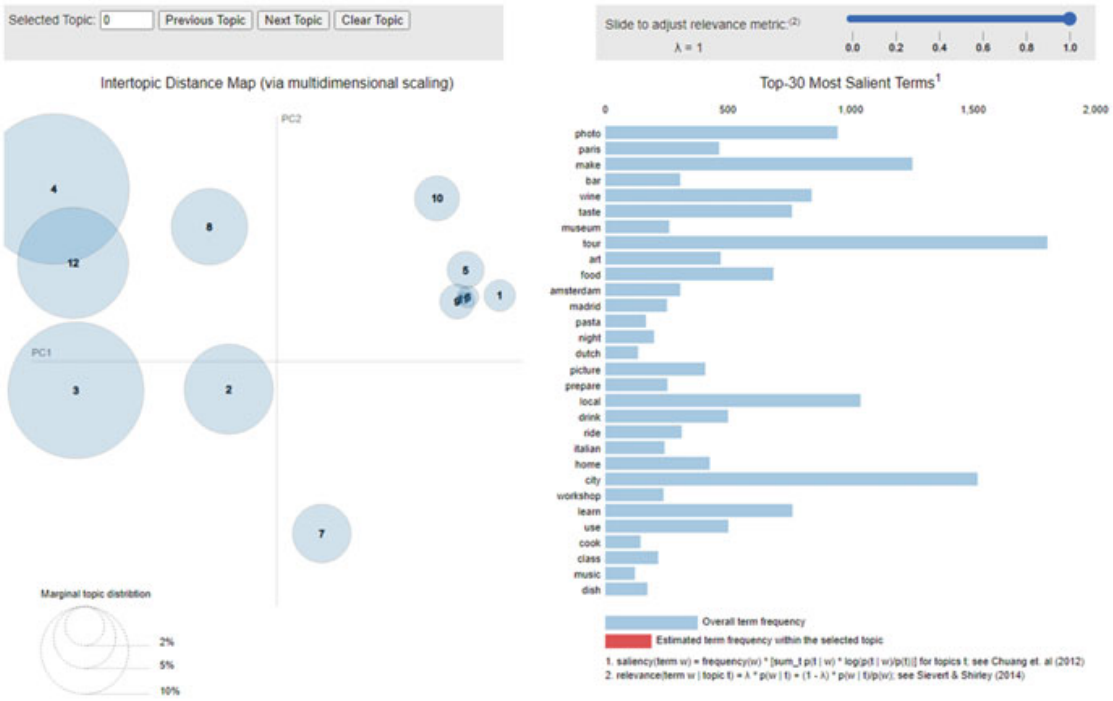


Fig. 4 LDAvis

4 Topic Modelling in Tourism Studies

In a nutshell, topic models assume that meanings are relational (Saussure, 1959), and each document is composed of diverse topics that comprise a collection of highly related words. In a practicable sense, topic modelling is often applied in exploratory studies (Guo et al., 2017) as it can disclose tourism experiences from a bottom-up approach, thereby providing insights that are often neglected by marketers (Shafqat & Byun, 2020). When it comes to tourism-related studies, user-generated content from various social media channels is mainly analysed in order to gain insights into the users' characteristics, attitudes, and opinions (Cai et al., 2018).

Compared to other methods of social research, the literature surrounding topic modelling in the field of tourism is quite limited; the main reasons being that there is a learning gap when it comes to effectively applying topic modelling to tourism research, especially regarding statistical and mathematical basics as well as programming skills (Papilloud & Hinneburg, 2018) in coding languages like Python or R. Only recently has topic modelling been integrated into more studies, most likely because more and more tools that do not require programming skills are becoming readily available on the market. However, such "Topic Modelling Toolkits", which can be applied without any programming knowledge, often have a disadvantage as they contain insufficient data preprocessing and lack hyperparameter tuning, ultimately leading to, as previously noted, bad quality results.

Existing studies suggest that LDA is the most widely accepted approach (Calheiros et al., 2017; Park et al., 2018) because of its capability to assign a

probability composition of the document to a latent theme through Bayesian inference (Blei, 2012a; Park et al., 2018). This suggests the iterative process of topic models, where documents are first assigned via random probability, but the performance of the algorithm becomes more accurate once more data has been processed (Vu et al., 2019). However, as can be seen from existing research with LDA methods, a strong emphasis has been placed on extracting the perceptions of tourism experiences based on online reviews and comments (Bi et al., 2019; Dickinger et al., 2017; Kim et al., 2019; Taecharungroj & Mathayomchan, 2019). One study (Vu et al., 2019) even took this further by using LDA to uncover tourist activity preferences in the context of travel itineraries. On the other hand, Wang et al. (2020) adopted LDA to uncover tourists' spatial and psychological involvement in multiphasic travel stages, while Shafqat and Byun (2020) used LDA and sentiment analysis to recommend under-emphasised locations. The following table lists some of the most recent research projects in the tourism domain, highlighting their research objectives, the applied methodology and tools/software used for topic modelling (Table 1).

Table 1 Tourism-related research projects using topic modelling

Name	Year	Title	Objectives	Methodology *(Software)*
Egger et al.	2022	Topic modeling of tourists dining experiences based on the GLOBE model	Identifying the dining preferences per cultural dimension	**LDA**, MDS, FastText *(Python, Orange)*
Yu & Egger	2021	Tourist experiences at overcrowded attractions: A text analytics approach	To explore the perception and feelings of tourists when visiting overcrowded attractions	**LDA**, sentiment analysis *(Python, Orange)*
Egger & Yu	2021	Identifying hidden semantic structures in Instagram data: a topic modelling comparison	Evaluating the effectiveness of different topic modelling algorithms	**LDA, NMF, CorEX** (Python)
Luo, He, Mou, Wang, & Liu,	2021	Exploring China's 5A global geoparks through online tourism reviews: A mining model based on machine learning approach	To provide valuable suggestions for managers by increasing the understanding of the psychological cognition of tourists	**LDA**; SVM; IPA *(Python)*
Shafqat & Byun	2020	A recommendation mechanism for under-emphasized tourist spots using topic modelling and sentiment analysis	To help the tourism industry in designing effective promotional activities for under-emphasised locations	**LDA**; SVM; cross mappings *(Python)*
Wang, Li, Wu, & Wang	2020	Tourism destination image based on tourism user generated content on internet	To study tourists' spatial and psychological involvement reflected through a tourism destination image	**LDA**; SNA *(Unknown)*

(continued)

Table 1 (continued)

Name	Year	Title	Objectives	Methodology *(Software)*
Zou	2020	National park entrance fee increase: A conceptual framework	To understand the public acceptance and opposition of a fee increase in the context of public park tourism	**LDA** *(R)*
Sun, Liang, & Chang	2020	Online social construction of Taiwan's rural image: Comparison between Taiwanese self-representation and Chinese perception	To determine how the objective discourse concerning Taiwanese rurality represented on online media is constructed and maintained	**Word embeddings** (Word2Vec) Keyword analysis; correspondence analysis; *(R)*
Wen, Park, Tao, Chae, Li, & Kwon	2020	Exploring user-generated content related to dining experiences of consumers with food allergies	To explore factors influencing perceptions of consumers with food allergies towards restaurants when accommodating allergen-free requests	**Structural topic model** *(Python)*
Han, Zejnilovic, & Novais	2019	Tourism2vec: An adaptation of Word2Vec to investigate tourism spatio-temporal behaviour	To propose tourism2vec for the investigation of tourism spatio-temporal behaviour	**Word embedding** *(Python)*
Hu, Zhang, Gao, & Bose	2019	What do hotel customers complain about? Text analysis using structural topic model	To identify the antecedents of hotel customers' dissatisfaction across different classes of hotels	**Structural topic model** *(R)*
Kim, Park, Barr, & Yun	2019	Tourists' shifting perceptions of UNESCO heritage sites: Lessons from Jeju Island-South Korea	To analyse the shifting perceptions of international tourists to Jeju Island	**LDA** *(R)*
Taecharungroj & Mathayomchan	2019	Analyzing TripAdvisor reviews of tourist attractions in Phuket, Thailand	To develop a methodology that can analyse online reviews using machine learning techniques for tourism practitioners to improve their attractions	**LDA**; naïve Bayes modelling *(KNIME)*
Vu, Li, & Law	2019	Discovering implicit activity preferences in travel itineraries by topic modelling	To introduce a framework for travel itinerary analysis that can reveal the underlying activity preferences of tourists	**LDA** *(Unknown)*
Hayashi	2019	Applying the document vector model to tour recommendation	To propose a system for recommending tours and their customer reviews	**Word embedding** (Word2Vec) *(Python)*

(continued)

Table 1 (continued)

Name	Year	Title	Objectives	Methodology *(Software)*
Huang, Wang, Yang, & Xu	2018	Topic mining of tourist attractions based on a seasonal context aware LDA model	To detect the representative and comprehensive attributes corresponding to various seasonal contexts for each attraction	**Season topic model LDA** *(Unknown)*
Li, Li, Hu, Zhang, & Hu	2018	Tourism review sentiment classification using a bidirectional recurrent neural network with an attention mechanism and topic-enriched word vectors	To analyse the emotions, preferences, feelings, and opinions expressed by visitors based on hotel review comments	**Lda2vec;** Bidirectional gated recurrent unit neural network model *(Python)*
Li, Zhu, Guo, Shi, & Zheng	2018	Build a tourism-specific sentiment lexicon via Word2Vec	To mine useful knowledge which can help tourism websites make decisions and improve their travel products	**Word embedding** (Word2Vec) *(HowNet)*
Calheiros, Moro, & Rita	2017	Sentiment classification of consumer-generated online reviews using topic modelling	To gather relevant topics that characterise a given hospitality issue through a sentiment	**LDA**; Sentiment analysis; *(R)*
Xu & Li	2016	The antecedents of customer satisfaction and dissatisfaction toward various types of hotels: A text mining approach	To discover and compare the determinants of customer satisfaction and dissatisfaction towards different hotels	**LSA** *(RapidMiner)*

5 Topic Model Toolkits and Software Solutions

Although an overview of available topic modelling solutions is given in Chapter 26 (Software & Tools), some further information is given here. For those working with Python, *Gensim*, presented by Rehurek and Sojka (2010) is a widely used and accepted topic modelling toolkit. Alternatively, *MALLET* (McCallum, 2002) or *tmtoolkit* (Konrad, 2017) can be used. For users who prefer R, *MALLET* is also available along with numerous packages such as *topicmodels* (Hornik & Grün, 2011) or *LDA* (Chang et al., 2009). A graphical user interface is provided by *Orange 3* offering the "Topic Modelling" widget in the text mining module. Three algorithms, namely LSI (Latent Semantic Indexing), LDA, and HDP (Hierarchical Dirichlet Process) can be used here. Unfortunately, for LDA and LSI, only the number of topics can be defined, which leads to obtaining results that must be accepted without having a closer look at the quality criteria. *Rapidminer* also offers topic modelling with the *LDA operator*. Stand-alone solutions include the *Stanford*

Topic Modelling Toolbox (Ramage, Rosen, et al., 2009), the *Topic Modelling Tool* by Scott Enderle or *Serendip*, which is a system for visually exploring topic models (Alexander et al., 2014), or *Topics Explorer (*Simmler et al., 2019*)*. Alternatively, browser-based solutions like *jsLDA* (David Mimno, 2013), or *topix.io* can be used. At this point, however, it must be pointed out, once again, that solely the tools that allow for the preprocessing of input texts as well as extensive control of the hyperparameters and fine-tuning of the results are to be taken seriously. It is therefore recommended to perform topic modelling either in Python or R.

6 Practical Demonstration

In this section, we will complete a walkthrough with a dataset from Airbnb, applying LDA (using Gensim) and CorEX as two distinct topic modelling approaches. This dataset was crawled by the author and contains 2890 descriptions of Airbnb experiences from the following European cities: Amsterdam, Athens, Berlin, Brussels, Copenhagen, Helsinki, London, Madrid, Oslo, Paris, Prague, Rome, Stockholm, Vienna, and Warsaw. The complete code and the dataset are available as a Jupyter notebook, together with an NMF exercise. For Top2Vec and BERTopic, two separate Jupyter notebooks are provided in the book's Github-profile.

6.1 LDA: Data Preparation & Preprocessing

Since we start with LDA in our example, we need to preprocess the text accordingly. As already mentioned, however, not every method requires this exact form of preprocessing. To prepare the text data for the application of LDA, the text is first converted to lowercase. It is then followed by some regex operations, the removal of punctuation, special characters and numbers (note: depending on the objective, this may or may not be necessary), the removal of stopwords, and tokenisation and lemmatisation of the text. These steps are described in detail in Chapter 15 (Introduction: Natural Language Processing) and are therefore disregarded here.

If you use the Gensim library, you can use the integrated preprocessing module (preprocess_string), which strips punctuation, removes stopwords, cleans HTML tags and non-alphabetical characters, stems the text, and much more. Table 2 shows an excerpt of the original text and what the text looks like after preprocessing or lemmatising.

To get a first impression of the dataset, generating a wordcloud for better visualisation is recommended and worthwhile (Fig. 5).

In order to avoid looking at individual tokens in isolation and, consequently, drawing the wrong conclusions in the analysis, one should also have a look at bi- and tri-grams. In this way, instead of the three words "red", "light", and "district", the

Table 2 Preprocessing input data

Original text	Preprocessed text	Lemmatised text
First of all we want to thank you all for choosing our experience. We are proud to announce that we have won an award from Airbnb for the most popular experience! Besides that we received more than 75 different nationalities on board. This would never have been possible without you ❤\n\n \nDo you want to experience Amsterdam in a more authentic way?	First want thank choosing experience proud announce award airbnb popular experience besides received different nationalities board would never possible without want experience Amsterdam authentic way	First want thank choose experience proud announce award airbnb popular experience besides receive different nationality board would never possible without want experience Amsterdam authentic way

Fig. 5 Wordcloud

tri-gram "red light district" will be identified (Fig. 6). The computation of n-grams can, however, be very computationally expensive if large datasets are involved.

```
def get_top_n_trigram(corpus, n=None):
 vec = CountVectorizer(ngram_range=(3, 3), stop_words='english').fit
(corpus)
 bag_of_words = vec.transform(corpus)
 sum_words = bag_of_words.sum(axis=0)
 words_freq = [(word, sum_words[0, idx]) for word, idx in vec.
vocabulary_.items()]
 words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
 return words_freq[:n]
```

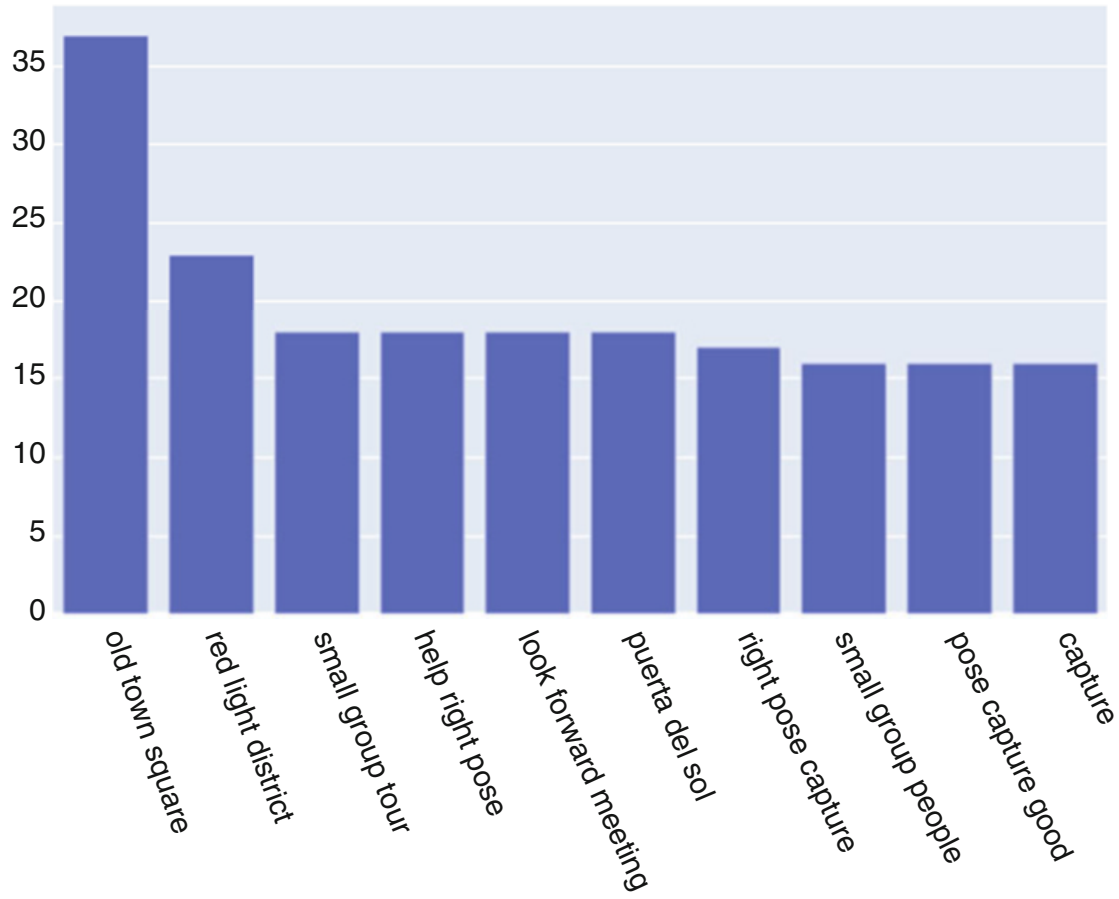


Fig. 6 Histogram of the top 20 tri-grams

```
common_words = get_top_n_trigram(df['lemmatized'], 20)
df_toptri = pd.DataFrame(common_words, columns = ['trigram' ,
'count'])

fig = go.Figure([go.Bar(x=df_toptri['trigram'], y=df_toptri
['count'])])
fig.update_layout(title=go.layout.Title(text="Top 20 trigrams in the
airbnb documents"))
fig.show()
```

Once the data has been preprocessed, it is then time to generate the first baseline topic model. We start by using the class gensim.corpra.Dictionary to set a unique ID for each word in our corpus. Thus, we have a dictionary defining all the words that we are currently working with. In the next step, a document can be represented either as a vector or as a bag-of-words (BOW); in this case, the latter will be used.

```
#Create Dictionary
df["lemmatized_split"] = df["lemmatized"].map(lambda x: x.split())
id2word = corpora.Dictionary(df["lemmatized_split"])

#Keep top n words ordered by term frequency across the corpus
n=5000
```

```
id2word.filter_extremes(no_below=1, no_above=1, keep_n=n)
#Get bag of words representation (word_id, frequency)
corpus = [id2word.doc2bow(doc) for doc in df['lemmatized_split'].
tolist()]
```

Now, we can fit a baseline model with three topics to get an overall idea of the data. This model will be fine-tuned in one of the next steps.

```
lda = LdaModel(corpus, num_topics = 3, id2word=id2word, passes=50)
```

After visualising the three topics with the interactive LDAvis tool (see Jupyter notebook), we have a rough idea of the most relevant words per topic. In the next step, hyperparameters are tuned so as to optimise the model and measure the coherence score between candidate models. We use standard values for the chunksize, which defines how many documents are processed in the training at a time, the passes value (or epochs), which controls how often the model is trained on the entire corpus (Kapadia, 2019), the topic range, and the step size. All three hyperparameters (number of Topics (K), Document-Topic Density (α/a), and Word-Topic Density (η/β/b)) are tested step by step, where one parameter always varies and the other parameters are kept constant. This procedure is applied to two different corpus validation sets. After that, we can use the coherence score ***C_v*** as a performance measure to find the best constellation.

```
def compute_coherence_values(mycor, mydic, k, a, b):
 lda_model = gensim.models.LdaMulticore(corpus=mycor,
 #id2word=dictionary,
 id2word=mydic,
 num_topics=k,
 random_state=100,
 chunksize=100,
 passes=10,
 alpha=a,
 eta=b)

 coherence_model_lda = CoherenceModel(model=lda_model, texts=df
['lemmatized_split'].tolist(),
 dictionary=mydic, coherence='c_v')

 return coherence_model_lda.get_coherence()

def main_hyperparameters_search():
 grid = {}
 grid['Validation_Set'] = {}

 # Topics range
 min_topics = 2
 max_topics = 15
 step_size = 1
 topics_range = range(min_topics, max_topics, step_size)
```

```
# Alpha parameter
alpha = list(np.arange(0.01, 1, 0.3))
alpha.append('symmetric')
alpha.append('asymmetric')

# Beta parameter
beta = list(np.arange(0.01, 1, 0.3))
beta.append('symmetric')

# Validation sets
num_of_docs = len(corpus)
corpus_sets = [# gensim.utils.ClippedCorpus(corpus,
num_of_docs*0.25),
# gensim.utils.ClippedCorpus(corpus, num_of_docs*0.5),
# gensim.utils.ClippedCorpus(corpus, num_of_docs*0.75),
corpus]

corpus_title = ['100% Corpus']

model_results = {'Validation_Set': [],
'Topics': [],
'Alpha': [],
'Beta': [],
'Coherence': []
}

# Can take a long time to run
if 1 == 1:
pbar = tqdm.tqdm(total=(len(beta)*len(alpha)*len(topics_range)*len
(corpus_title)))

# iterate through validation corpuses
for i in range(len(corpus_sets)):
# iterate through number of topics
for k in topics_range:
# iterate through alpha values
for a in alpha:
# iterare through beta values
for b in beta:
# get the coherence score for the given parameters
cv = compute_coherence_values(mycor=corpus_sets[i], mydic=id2word,
k=k, a=a, b=b)
# Save the model results
model_results['Validation_Set'].append(corpus_title[i])
model_results['Topics'].append(k)
model_results['Alpha'].append(a)
model_results['Beta'].append(b)
model_results['Coherence'].append(cv)

pbar.update(1)
df = pd.DataFrame(model_results)
df.to_csv('lda_tuning_results.csv', index=False)
```

```
 pbar.close()
 return df

if os.path.exists(os.path.join(os.getcwd(), 'lda_tuning_results.
csv')):
 print('Results loaded from {}'.format(os.path.join(os.getcwd(),
'lda_tuning_results.csv')))
 df_hyper = pd.read_csv('lda_tuning_results.csv')
else:
 print('Hyperparameters search, it will take some time...')
 df_hyper = main_hyperparameters_search()
df_hyper.head()
```

This will provide us with the coherence score for all different combinations of K, α, and β (only the first five combinations are listed below) (Table 3).

Next, let's see the highest coherence score archived.

```
best = df_hyper[df_hyper.Coherence == df_hyper.Coherence.max()]
print('The highest coherence score is {}'.format(best.Coherence.
values[0]))
```

Output: The highest coherence score is 0.601.

Lastly, we also want to see the combination producing the highest score.

```
print('And the corresponding parameters are following:')
best[['Topics', 'Alpha', 'Beta']].reset_index(drop=True)
```

Output:

	Topics	Alpha	Beta
0	13	0.91	0.61

To get a better overview of the grid-search results, we can then visualise them in an interactive 3D-scatterplot (Fig. 7).

Finally, let's visualise the results with the interactive pyLDAvis tool.

```
# Visualize LDA after optimalization
lda_optim = LdaModel(corpus, num_topics = ntopics, id2word=id2word,
passes=50, alpha=alpha, eta=beta)
```

Table 3 Coherence scores

	Validation_Set	Topics	Alpha	Beta	Coherence
0	100% corpus	2	0.01	0.01	0.42945
1	100% corpus	2	0.01	0.31	0.425401
2	100% corpus	2	0.01	0.61	0.431354
3	100% corpus	2	0.01	0.91	0.41898
4	100% corpus	2	0.01	Symmetric	0.425401

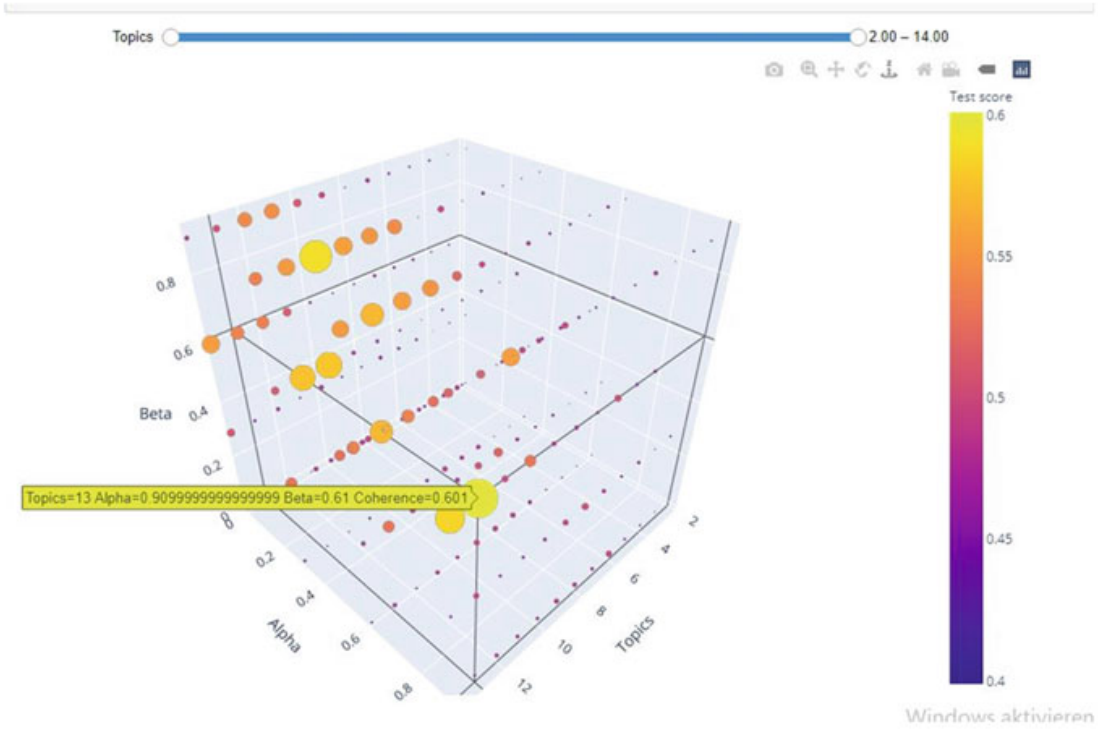
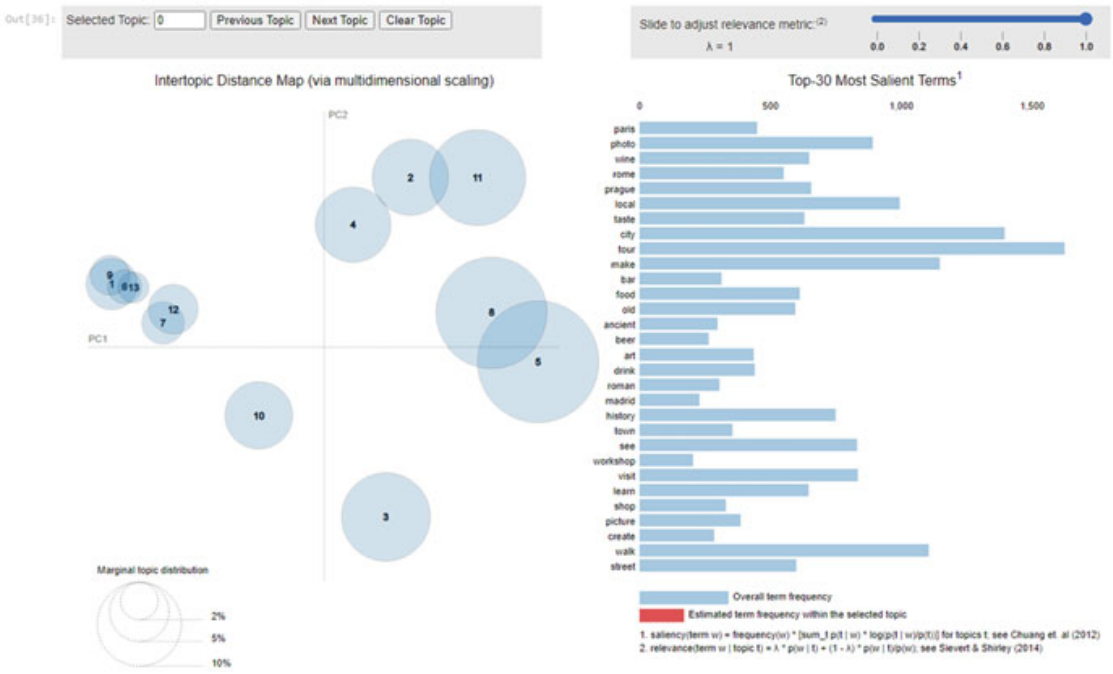


Fig. 7 3D-Scatterplot of results

```
lda_visualization = pyLDAvis.gensim.prepare(lda_optim, corpus,
id2word, sort_topics=False)
pyLDAvis.display(lda_visualization)
```



As we can see, topics 1, 6, 9, and 13 are overlapping; we should therefore go into more detail by inspecting the most salient terms. If necessary, word intrusion or topic intrusion can be used to further evaluate the results with a qualitative human judgement approach (Chang et al., 2009). Based on this, we can decide whether further changes need to be made or if the results can be accepted as the final solution.

6.2 Topic Modelling with CorEx

After having loaded the preprocessed dataset, we will again create a dictionary a dictionary and a apply TF-IDF transformation

```
df["lemmatized_split"] = df["lemmatized"].map(lambda x: x.split())
id2word = corpora.Dictionary(df["lemmatized_split"])

text_string = [' '.join(d) for d in df['lemmatized_split'].tolist()]
np.random.seed(42)
n_features=n_features

tfidf_vectorizer = TfidfVectorizer(max_df=0.95, min_df=2,
max_features=n_features, ngram_range=(1,2), stop_words='english')
tfidf = tfidf_vectorizer.fit_transform(text_string)
vocab =tfidf_vectorizer.get_feature_names()
```

Next, we can define the anchors to nudge the model towards specific terms. A certain minimum domain knowledge is necessary and it must be justifiable why and how the anchoring is applied in the research context.

```
anchors = [
 ["sight"],
 ["activity"],
 ["photo"],
]

anchors = [
 [a for a in topic if a in vocab]
 for topic in anchors
]

anchors_dict = {w[0] : a for a,w in enumerate(anchors)}

model = ct.Corex(n_hidden=len(anchors), seed=42)
model = model.fit(
 tfidf,
 words=vocab,
 anchors=anchors,
 anchor_strength=6 # Tell the model how much it should rely on the anchors
)
```

Finally, lets generate the topics and preview the most relevant ten terms of each topic

```
topic_words = []
for i, topic_ngrams in enumerate(model.get_topics(n_words=10)):
 topic_ngrams = [ngram[0] for ngram in topic_ngrams if ngram[1] > 0]
 print("Topic #{}: {}".format(i+1, ", ".join(topic_ngrams)))
 topic_words.append(topic_ngrams)
```

```
Output:
Topic #1: paris, pilot, notre dame, dame, notre, maisonslaffitte, pont
neuf, neuf, louvre, saintgermain
Topic #2: workshop, material, shopping, instawalk, perfume, leather,
fashion, boating, climbing
Topic #3: photo, shoot, photoshoot, picture, capture, photography,
edit, camera, pose, session
```

Service Section

Main Application Fields: There are numerous different topic modelling approaches, all of which have their respective advantages and disadvantages. Basically, all methods try to extract latent topics from texts. They therefore have an inductive and explorative character.

Limitations and Pitfalls: For most topic modelling approaches, the number of topics to be extracted must be determined in advance. This is a task that requires both sensitivity and knowledge of appropriate hyperparameter tuning. If default settings are used, it is all too easy to extract nonsense topics that are too broad, too fine granular or simply have no meaningfulness.

Similar Methods and Methods to Combine with: Topic Modelling is particularly suitable in combination with semantic analysis. Text clustering could be seen as a similar procedure.

Code: The Python Code is available at: https://github.com/DataScience-in-Tourism/Chapter-18-Topic-Modeling

Further Readings and Other Sources

A great tutorial about LDA - "An approachable explanation of how Topic Modelling works" – can be found at https://topix.io/tutorial/tutorial.html. This tutorial also provides an interactive explanation of the Gibbs Sampling process.

Greene and Cross (2017) present a paper together with the code for a dynamic topic modeling approach, allowing to track how topics envolve over time. https://github.com/derekgreene/dynamic-nmf

As further literature for R-users the book by Jockers and Thalken (2020) can be suggested

A very good overview on the application of topic models by social scientists is given in the book by Papilloud and Hinneburg (2018) "Qualitative Textanalyse mit Topic Modellen, eine Einführung für Sozialwissenschaftler" (only in German langage available)

Furthermore, the following online articles are recommended:

https://towardsdatascience.com/topic-modelling-of-2019-hr-tech-conference-twitter-d16cf75895b6

https://towardsdatascience.com/topic-modeling-with-nlp-on-amazon-reviews-an-application-of-latent-dirichlet-allocation-lda-ae42a4c8b369

https://medium.com/@kurtsenol21/topic-modeling-lda-mallet-implementation-in-python-part-1-c493a5297ad2

https://towardsdatascience.com/short-text-topic-modeling-70e50a57c883

https://towardsdatascience.com/introduction-to-nlp-part-5b-unsupervised-topic-model-in-python-ab04c186f295

https://sagarpanwar249.medium.com/guide-to-topic-modeling-eac693c9d3e0

References

Albalawi, R., Yeap, T. H., & Benyoucef, M. (2020). Using topic modeling methods for short-text data: A comparative analysis. *Frontiers in Artificial Intelligence, 3*, 42. https://doi.org/10.3389/frai.2020.00042

Alexander, E., Kohlmann, J., Valenza, R., Witmore, M., & Gleicher, M. (2014). Serendip: Topic model-driven visual exploration of text corpora. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)* (pp. 173–182). IEEE. https://doi.org/10.1109/VAST.2014.7042493

Angelov, D. (2020a). *Top2Vec: Distributed Representations of Topics*. Retrieved from http://arxiv.org/pdf/2008.09470v1

Angelov, D. (2020b, April 16). *COVID-19: Topic modeling and search with Top2Vec: Kaggle – Jupyter Notebook*. Retrieved from https://www.kaggle.com/dangelov/covid-19-topic-modeling-and-search-with-top2vec

Banks, G. C., Woznyj, H. M., Wesslen, R. S., & Ross, R. L. (2018). A review of best practice recommendations for text analysis in R (and a user-friendly app). *Journal of Business and Psychology, 33*(4), 445–459. https://doi.org/10.1007/s10869-017-9528-3

Bi, J.-W., Liu, Y., Fan, Z.-P., & Cambria, E. (2019). Modelling customer satisfaction from online reviews using ensemble neural network and effect-based Kano model. *International Journal of Production Research, 57*(22), 7068–7088. https://doi.org/10.1080/00207543.2019.1574989

Blair, S. J., Bi, Y., & Mulvenna, M. D. (2020). Aggregated topic models for increasing social media topic coherence. *Applied Intelligence, 50*(1), 138–156. https://doi.org/10.1007/s10489-019-01438-z

Blei, D. M. (2012a). Probabilistic topic models. *Communications of the ACM, 55*(4), 77–84.

Blei, D. M. (2012b). Probabilistic topic models. *Communications of the ACM, 55*(4), 77–84. https://doi.org/10.1145/2133806.2133826

Blei, D., Ng, A., & Jordan, M. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3*(Jan), 993–1022.

Cai, G., Sun, F., & Sha, Y. (2018). Interactive visualization for topic model curation. *IUI Workshops*.

Calheiros, A. C., Moro, S., & Rita, P. (2017). Sentiment classification of consumer-generated online reviews using topic modeling. *Journal of Hospitality Marketing & Management, 26*(7), 675–693. https://doi.org/10.1080/19368623.2017.1310075

Chang, J., Gerrish, S., Wang, C., Boyd-graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. *Advances in Neural Information Processing Systems*. Retrieved from http://papers.nips.cc/paper/3700-reading-tea-leaves-how-humans-interpret-topic-models.pdf

Daenekindt, S., & Huisman, J. (2020). Mapping the scattered field of research on higher education. A correlated topic model of 17,000 articles, 1991–2018. *Higher Education, 80*(3), 571–587. https://doi.org/10.1007/s10734-020-00500-x

Denny, M. J., & Spirling, A. (2018). Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it. *Political Analysis, 26*(2), 168–189. https://doi.org/10.1017/pan.2017.44

Dickinger, A., Lalicic, L., & Mazanec, J. (2017). Exploring the generalizability of discriminant word items and latent topics in online tourist reviews. *International Journal of Contemporary Hospitality Management, 29*(2), 803–816. https://doi.org/10.1108/IJCHM-10-2015-0597

Dietz, L. (2016). Topic model evaluation: How much does it help? WebSci2016, University Mannheim.

DiMaggio, P., Nag, M., & Blei, D. (2013). Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of U.S. government arts funding. *Poetics, 41*(6), 570–606. https://doi.org/10.1016/j.poetic.2013.08.004

Egger, R., & Yu, J. (2021). Identifying hidden semantic structures in Instagram data: A topic modelling comparison. *Tourism Review*.

Egger, R. (2022). Machine learning in tourism – a brief overview. In R. Egger (Ed.), *Tourism on the verge. Applied data science in tourism: Interdisciplinary approaches, methodologies and applications (n.a)*. Springer.

Egger, R., Pagiri, A., Prodinger, B., Liu, R., & Wettinger, F. (2022, January). Topic modelling of tourist dining experiences based on the GLOBE Model. In *ENTER22 e-tourism conference* (pp. 356–368). Springer.

Evans, M. S. (2014). A computational approach to qualitative analysis in large textual datasets. *PLoS One, 9*(2), e87908. https://doi.org/10.1371/journal.pone.0087908

Gallagher, R. J., Reing, K., Kale, D., & Ver Steeg, G. (2017). Anchored correlation explanation: Topic modeling with minimal domain knowledge. *Transactions of the Association for Computational Linguistics, 5*, 529–542. https://doi.org/10.1162/tacl_a_00078

Greene, D., & Cross, J. P. (2017). Exploring the political agenda of the European Parliament using a dynamic topic modeling approach. *Political Analysis, 25*(1), 77–94. https://doi.org/10.1017/pan.2016.7

Greene, D., O'Callaghan, D., & Cunningham, P. (2014, April 16). *How many topics? Stability analysis for topic models*. Retrieved from http://arxiv.org/pdf/1404.4606v3

Grimmer, J., & Stewart, B. M. (2013). Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis, 21*(3), 267–297. https://doi.org/10.1093/pan/mps028

Grootendorst, M. (2020, May 10). *Topic modeling with BERT. | Towards data science*. Retrieved from https://towardsdatascience.com/topic-modeling-with-bert-779f7db187e6

Grootendorst, M. (2021, June 1). *Interactive topic modeling with BERTopic | Towards data science*. Retrieved from https://towardsdatascience.com/interactive-topic-modeling-with-bertopic-1ea55e7d73d8

Guo, Y., Barnes, S. J., & Jia, Q. (2017). Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation. *Tourism Management, 59*, 467–483. https://doi.org/10.1016/j.tourman.2016.09.009

Hannigan, T. R., Haans, R. F. J., Vakili, K., Tchalian, H., Glaser, V. L., Wang, M. S., . . . Jennings, P. D. (2019). Topic modeling in management research: Rendering new theory from textual data. *Academy of Management Annals, 13*(2), 586–632. https://doi.org/10.5465/annals.2017.0099

Hindle, A., Ernst, N. A., Godfrey, M. W., & Mylopoulos, J. (2013). Automated topic naming. *Empirical Software Engineering, 18*(6), 1125–1155. https://doi.org/10.1007/s10664-012-9209-9

Hornik, K., & Grün, B. (2011). Topicmodels: An R package for fitting topic models. *Journal of Statistical Software, 40*(13), 1–30. Retrieved from https://epub.wu.ac.at/3987/

Hu, J., Sun, X., & Li, B. (2015). Explore the evolution of development topics via on-line LDA. In *IEEE 22nd international conference* (pp. 555–559). IEEE. https://doi.org/10.1109/SANER.2015.7081876

Hu, N., Zhang, T., Gao, B., & Bose, I. (2019). What do hotel customers complain about? Text analysis using structural topic model. *Tourism Management, 72*, 417–426. https://doi.org/10.1016/j.tourman.2019.01.002

Jockers, M. L., & Thalken, R. (2020). Topic modeling. In M. L. Jockers & R. Thalken (Eds.), *Quantitative methods in the humanities and social sciences. Text analysis with R* (pp. 211–235). Springer International Publishing. https://doi.org/10.1007/978-3-030-39643-5_17

Kadhim, A. I., Cheah, Y.-N., & Ahamed, N. H. (2014). Text document preprocessing and dimension reduction techniques for text document clustering. In *2014 4th international conference on artificial intelligence with applications in engineering and technology* (pp. 69–73). IEEE. https://doi.org/10.1109/ICAIET.2014.21

Kapadia, S. (2019, August 19). *Evaluate topic models: Latent Dirichlet Allocation (LDA). Towards data science*. Retrieved from https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0

Kim, K., Park, O., Barr, J., & Yun, H. (2019). Tourists' shifting perceptions of UNESCO heritage sites: Lessons from Jeju Island-South Korea. *Tourism Review, 74*(1), 20–29. https://doi.org/10.1108/TR-09-2017-0140

Konrad, M. (2017). Tmtoolkit [computer software].

Kuang, D., Brantingham, P. J., & Bertozzi, A. L. (2017). Crime topic modeling. *Crime Science, 6*(1), 12. https://doi.org/10.1186/s40163-017-0074-0

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes, 25*(2–3), 259–284. https://doi.org/10.1080/01638539809545028

Lee, D., & Seung, H. S. (1999). Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems, 13*, 556–562.

Lesnikowski, A., Belfer, E., Rodman, E., Smith, J., Biesbroek, R., Wilkerson, J. D., . . . Berrang-Ford, L. (2019). Frontiers in data analytics for adaptation research: Topic modeling. *Wiley Interdisciplinary Reviews: Climate Change, 10*(3), e576. https://doi.org/10.1002/wcc.576

Liangjie, H., & Davison, B. D. (2010). Empirical study of topic modeling in twitter. In P. Melville (Ed.), *Proceedings of the first workshop on social media analytics* (pp. 80–88). ACM.

Lim, K. W., & Buntine, W. (2014). Twitter opinion topic model. In J. Li, X. S. Wang, M. Garofalakis, I. Soboroff, T. Suel, & M. Wang (Eds.), *Proceedings of the 23rd ACM international conference on conference on information and knowledge management* (pp. 1319–1328). ACM. https://doi.org/10.1145/2661829.2662005

Lindstedt, N. C. (2019). Structural topic modeling for social scientists: A brief case study with social movement studies literature, 2005–2017. *Social Currents, 6*(4), 307–318. https://doi.org/10.1177/2329496519846505

Loureiro, S. M. C., Guerreiro, J., & Ali, F. (2020). 20 years of research on virtual reality and augmented reality in tourism context: A text-mining approach. *Tourism Management, 77*, 104028. https://doi.org/10.1016/j.tourman.2019.104028

Maier, D., Waldherr, A., Miltner, P., Wiedemann, G., Niekler, A., Keinert, A., . . . Adam, S. (2018). Applying LDA topic modeling in communication research: Toward a valid and reliable methodology. *Communication Methods and Measures, 12*(2–3), 93–118. https://doi.org/10.1080/19312458.2018.1430754

McCallum, A. (2002). MALLET: A machine learning for language Toolkitefrom https://ci.nii.ac.jp/naid/20001704926/

Mimno, D. (2013). jsLDA [Computer software].

Mimno, D., Hanna, W., Edmund, T., Miriam, L., & Andrew, M. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 262–272. Retrieved from https://www.aclweb.org/anthology/D11-1024.pdf

Mimno, D., Wallach, H., Naradowsky, J., Smith, D. A., & McCallum, A. (2009). Polylingual topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.

Mohammed, S. H., & Al-augby, S. (2020). LSA & LDA topic modeling classification: comparison study on e-books. *Indonesian Journal of Electrical Engineering and Computer Science, 19*(1), 353. https://doi.org/10.11591/ijeecs.v19.i1.pp353-362

Murugan, A., Chelsey, H., & Thomas, N. (2019). *Practical text analytics*. Springer International Publishing.

Nikolenko, S. I., Koltcov, S., & Koltsova, O. (2017). Topic modelling for qualitative studies. *Journal of Information Science, 43*(1), 88–102. https://doi.org/10.1177/0165551515617393

Papilloud, C., & Hinneburg, A. (2018). *Qualitative Textanalyse mit topic-Modellen*. Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-21980-2

Park, E., Chae, B., & Kwon, J. (2018). The structural topic model for online review analysis. *Journal of Hospitality and Tourism Technology, 11*(1), 1–17. https://doi.org/10.1108/JHTT-08-2017-0075

Qin, L., Shaobo, L., Sen, Z., Jie, H., & Jianjun, H. (2019). A Review of text corpus-based tourism big data mining. *In Applied Sciences, 9*(16), 3300. https://doi.org/10.3390/app9163300

Qiu, Z., & Shen, H. (2017). User clustering in a dynamic social network topic model for short text streams. *Information Sciences, 414*, 102–116. https://doi.org/10.1016/j.ins.2017.05.018

Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (Ed.) (2009). *A supervised topic model for credit attribution in multi-labeled corpora*.

Ramage, D., Rosen, E., Chuang, J., Manning, C. D., & McFarland, D. A. (2009). *Topic modeling for the social sciences: Topic modeling for the social sciences.* NIPS. NIPS 2009 workshop on applications for topic models: Text and beyond. Retrieved from http://nlp.stanford.edu/dramage/papers/tmt-nips09.pdf

Rehurek, R., & Sojka, P. (2010). *Software framework for topic modelling with large corpora: Rehurek, Radim, and Petr Sojka. "Software"*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.695.4595

Reing, K., Kale, D. C., Steeg, G. V., & Galstyan, A. (2016). *Toward interpretable topic discovery via anchored correlation explanation*. arXiv preprint arXiv:1606.07043.

Reisenbichler, M., & Reutterer, T. (2019). Topic modeling in marketing: Recent advances and research opportunities. *Journal of Business Economics, 89*(3), 327–356. https://doi.org/10.1007/s11573-018-0915-7

Roberts, M. E., Stewart, B. M., & Tingley, D. (2019). Stm: An R package for structural topic models. *Journal of Statistical Software, 91*(2), 1–40. https://doi.org/10.18637/jss.v091.i02

Röder, M., Both, A., & Hinneburg, A. (Eds.) (2015). *Exploring the space of topic coherence measures*.

Rossetti, M., Stella, F., Cao, L., & Zanker, M. (2015). Analysing User Reviews in Tourism with Topic Models. In I. Tussyadiah & A. Inversini (Eds.), Information and Communication Technologies in Tourism 2015 (pp. 47–58). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-14343-9_4

Rossetti, M., Stella, F., & Zanker, M. (2016). Analyzing user reviews in tourism with topic models. *Information Technology & Tourism, 16*(1), 5–21. https://doi.org/10.1007/s40558-015-0035-y

Saussure, F. D. (1959). *Course in general linguistics (W. Baskin, Trans.)*. Philosophical Library.

Shafqat, W., & Byun, Y.-C. (2020). A recommendation mechanism for under-emphasized tourist spots using topic modeling and sentiment analysis. *Sustainability, 12*(1), 320. https://doi.org/10.3390/su12010320

Sievert, C., & Shirley, K. (2014). LDAvis: A method for visualizing and interpreting topics. In J. Chuang, S. Green, M. Hearst, J. Heer, & P. Koehn (Eds.), *Proceedings of the workshop on interactive language learning, visualization, and interfaces* (pp. 63–70). Association for Computational Linguistics. https://doi.org/10.3115/v1/W14-3110

Simmler, S., Vitt, T., & Pielström, S. (2019). Topic modeling with interactive visualizations in a GUI tool. In *Proceedings of the Digital Humanities Conference*.

Sotomayor O. D., & Bellono G. (2019, January 3). *Automated topic discovery: An approachable explanation*. Retrieved from https://topix.io/tutorial/tutorial.html

Taecharungroj, V., & Mathayomchan, B. (2019). Analysing TripAdvisor reviews of tourist attractions in Phuket, Thailand. *Tourism Management, 75*, 550–568. https://doi.org/10.1016/j.tourman.2019.06.020

Ungar, L., Craven, M., Gunopulos, D., & Eliassi-Rad, T. (2006). Topics over time: A non-Markov continuous-time model of topical trends: Proceedings of the twelfth ACM SIGKDD International Conference on Knowledge Discovery and data mining August 20–23, 2006, Philadelphia, PA, USA, 424–433.

Ver Steeg, G. (2016). *Open source project implementing hierarchical topic models on sparse data*. Retrieved from https://github.com/gregversteeg/corex_topic

Vo, D.-T., & Ock, C.-Y. (2015). Learning to classify short text from scientific documents using topic models with various types of knowledge. *Expert Systems with Applications, 42*(3), 1684–1698. https://doi.org/10.1016/j.eswa.2014.09.031

Vu, H. Q., Li, G., & Law, R. (2019). Discovering implicit activity preferences in travel itineraries by topic modeling. *Tourism Management, 75*, 435–446. https://doi.org/10.1016/j.tourman.2019.06.011

Wallach, H. M., Murray, I., Salakhutdinov, R., & Mimno, D. (2009). Evaluation methods for topic models. In A. Danyluk, L. Bottou, & M. Littman (Eds.), *Proceedings of the 26th Annual International Conference on Machine Learning – ICML '09* (pp. 1–8). ACM Press. https://doi.org/10.1145/1553374.1553515

Wang, J., Li, Y., Wu, B., & Wang, Y. (2020). Tourism destination image based on tourism user generated content on internet. *Tourism Review*. https://doi.org/10.1108/TR-04-2019-0132. (ahead-of-print).

Wang, Y.-X., & Zhang, Y.-J. (2013). Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering, 25*(6), 1336–1353. https://doi.org/10.1109/TKDE.2012.51

Wei, X., Xin, L., & Yinhong, G. (2003). Document clustering based on non-negative matrix factorization. In J. Callan (Ed.), *Special issue of the SIGIR forum, Sigir 2003: Proceedings of the twenty-sixth annual international ACM SIGIR conference on Research and Development in information retrieval, Toronto, Canada, July 28 to august 1, 2003* (pp. 267–273). ACM Press.

Weng, J. (2020, December 21). Topic modeling in one line with Top2Vec – towards data science. *Towards data science*. Retrieved from https://towardsdatascience.com/topic-modeling-in-one-line-with-top2vec-a413991aa0ef

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., & Li, X. (2011). Comparing twitter and traditional media using topic models. In P. Clough, C. Foley, C. Gurrin, G. J. F. Jones, W. Kraaij, H. Lee, & V. Mudoch (Eds.), *Lecture notes in computer science, advances in information retrieval* (pp. 338–349). Springer. https://doi.org/10.1007/978-3-642-20161-5_34