Natural Language Processing (NLP): An Introduction

Making Sense of Textual Data

Roman Egger and Enes Gokce

Learning Objectives
- Illustrate the foundations of Natural Language Processing
- Appreciate where NLP can be applied in tourism research
- Demonstrate how to pre-process texts
- Explain the logic behind each data-cleaning and pre-processing step

1 Introduction and Theoretical Foundations

Within both academia and the tourism industry, it is vital to make proper use of textual data. According to a commonly reported statistic, 80% of the data produced every day consists of text (Wennker, 2020). This number brings about numerous challenges as text data is fundamentally unstructured and must be processed appropriately before it can be used for further analysis. The statement "information is the lifeblood of tourism" (Poon, 1993) has never been more appropriate than it is today, especially due to social media channels contributing greatly to the rising importance of user-generated content (UGC) (Conti & Lexhagen, 2020; Aicher et al., 2016). The analysis of customer reviews and posts from social media channels such as Twitter, Facebook, or Instagram has opened up previously unimagined opportunities for companies to better understand customer wishes, needs, and feelings, ultimately

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

E. Gokce
Pennsylvania State University, Pennsylvania, USA

R. Egger (ed.), *Applied Data Science in Tourism*, Tourism on the Verge,
https://doi.org/10.1007/978-3-030-88389-8_15

helping businesses to improve their services accordingly (Aicher et al., 2016; Egger, 2010).

The history of text analysis can be traced back to the twelfth century, when the first biblical concordances were written by monks (Ignatow & Mihalcea, 2017). It has also been reported that the first qualitative text analysis was performed in Sweden in the seventeenth century. Thereafter, systematic text analysis experienced a rapid upswing in the twentieth century as numerous methodological approaches for text analysis and associated procedures relating to the qualitative interpretation of texts and documents were developed in the social sciences (Bussière, 2018). More recently, a change towards the digital analysis of texts has been observed (Rockwell, 2003), and the enormous amount of data makes digital processing and analysis inevitable. Yet, text, unlike numerical data, poses particular challenges for computer-aided analyses. Take the following description of a dish on a restaurant menu as an example:

> *Italian dish made of stacked layers of thin flat pasta, alternating with fillings such as ragù and other vegetables, cheese, and seasonings and spices such as garlic, oregano and basil, topped with melted grated mozzarella cheese.*

The thought of lasagna should instantly come to mind; for a computer, however, the context is difficult to grasp, and/or making a connection to the term *lasagna* is challenging.

Driven by advances in machine learning, Natural Language Processing (NLP) is now a well-established field in artificial intelligence, linguistics, computer science, and information technology. NLP uses machines to read, understand, and make sense of human language in order to streamline data mining, data analysis, and business operations (Han et al., 2016). Alongside the improvements in NLP, there has been a shift from traditional manual coding to the automation of data collection, data cleaning, and statistical analysis (Li et al., 2019).

NLP's general task is to make a computer understand written language in the form of words, sentences, or paragraphs. Thus, the overall aim of NLP is for machines and computers to be able to successfully accomplish tasks concerning natural language. Natural language refers to any human language that was developed through natural circumstances and follows a specific syntactic and semantic system (Sarkar, 2019). Hapke et al. (2019) defines NLP as:

> *an area of research in computer science and artificial intelligence (AI) concerned with processing natural languages such as English or Mandarin. This processing generally involves translating natural language into data (numbers) that a computer can use to learn about the world. And this understanding of the world is sometimes used to generate natural language text that reflects that understanding (p. 4).*

Natural Language Processing (NLP) and Natural Language Understanding (NLU) are two closely related concepts in which confusion often arises between the demarcations of the two terms as well as misinterpretations of other subfields of artificial intelligence. NLU first came about in the 1960s, approximately ten years after NLP, out of the need to understand increasingly complex language input. While NLP covers all areas of communication between humans and computers, from input to processing to response, NLU seeks to understand content (MacCartney, 2014).

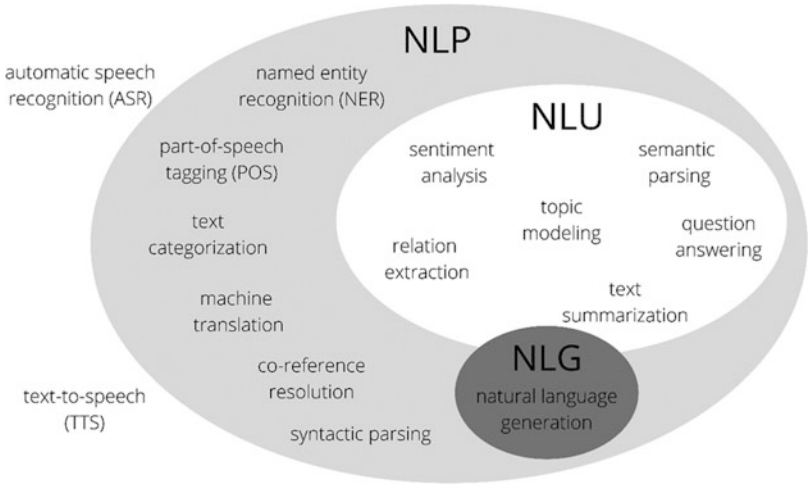


Fig. 1 Terminology relating to NLP, NLU, and NLG. Source: adapted from MacCartney (2014)

NLP is thus an overarching concept, and Natural Language Understanding (NLU) and Natural Language Generation (NLG) are sub-disciplines thereof. For the sake of simplicity, however, in this chapter, we will only refer to NLP, which includes NLU and NLG (Fig. 1).

NLP is considered to be very multifaceted and has already been integrated into various everyday situations. Machine translation, such as Google Translate or DeepL, allows machines to successfully translate natural language from one language to another (Sarkar, 2019), while text prediction (i.e. auto-correction) can aid in quickly solving typos and grammatical errors (Naik, 2020). Text summarisation, on the other hand, takes a text or collection of texts and expels a reduced summary based on the most prominent keywords, phrases, or sentences that appear. If, however, a large corpus of texts is presented, topic modelling would be a more potent way to extract and summarise essential concepts and themes (Kao & Poteet, 2007). Lastly, when it comes to texts with subjective content (e.g. feedback surveys, reviews, etc.), sentiment analysis or opinion mining would be considered the best fit (Ignatow & Mihalcea, 2017).

2 Text Analysis in Tourism

With the use of social media and user-generated content, a flood of unstructured data in the form of text, images, videos, and audio has become the dominant source of data in tourism research (Xiang, 2018). In order to be able to analyse the vast amounts of user-generated content available, automated processes that prepare unstructured texts for further processing and interpretation are needed. As such, in

the field of NLP applications, topic modelling (compare Chapter "Topic Modeling") and sentiment analysis (compare Chapter "Sentiment Analysis") seem to be the most common techniques implemented by tourism researchers (Yu & Egger, 2021; Alaei et al., 2017; Munezero et al., 2014; Guerreiro & Rita, 2020; Li et al., 2020; Chang et al., 2020). Sentiment analyses make it possible to capture tourists' feelings based on a given text (Markopoulos et al., 2015). By using TripAdvisor, Expedia, and Yelp datasets, Xiang et al. (2017), for example, identified the semantic features and sentiment scores of online reviews. Their study also highlights the importance of social media analytics in tourism and hospitality as insights from text data can promote smart tourism development (Li et al., 2019). Another recent study applied text mining techniques to reveal the antecedents of tourism recommendations on the Yelp platform (Guerreiro & Rita, 2020). Supported by previous literature, Han et al. (2016) reinforce the suggestion that tourism researchers should go beyond numerical ratings and interpret quantitative results with text data to reveal tourists' true feelings. In topic modelling (e.g. Latent Dirichlet Allocation), on the other hand, different topics are extracted from an existing text. In this way, by structuring them thematically and preparing them for further analysis, an overview of the text can be obtained. Topic modelling methods are often also combined with sentiment analysis via looking at the sentiments within the extracted topics (Calheiros et al., 2017). The results generated by most topic modelling algorithms must be interpreted by individual identified keywords, which is often a difficult task (Hannigan et al., 2019).

Recent literature has introduced Deep Learning approaches to NLP as a way to minimise existing limitations in text analysis (Chang et al., 2020; Ma et al., 2018). For example, a study by Chang et al. (2020) processed tourist reviews and comments by integrating visual analytics and Deep Learning-based NLP to assist hoteliers with strategic planning. Furthermore, Named Entity Recognition (NER) (Chantrapornchai & Tunsakul, 2019) and co-reference resolution (García-Pablos et al., 2016) have gradually gained popularity within the tourism domain as well. The former is based on information extraction in which predefined codes are automatically assigned to text elements, while the latter identifies expressions that refer to the same entity in natural language (Hannigan et al., 2019). Examples of such entities include the names of hotels, restaurants, people, countries, cities, or destinations (García-Pablos et al., 2016). More details on NER will be presented below. Beyond the field of tourism marketing and management, the use of NLP can also help to better understand socio-cultural aspects in the field of tourism. For instance, Li et al. (2020) examined online comments to conceptualise the impact of global racism on tourism experiences.

In addition, the possibilities of text summarisation have been explored further in the context of tourism. Tsai, Chen, Hu, & Chen (2020) presented approaches on how to create summaries of online hotel reviews, while Yang et al. (2020), for example, dealt with question-answering systems and built such a system based on a tourism knowledge graph. Knowledge-based reasoning systems are often used for recommendation systems in which these techniques use domain-specific knowledge to rate items and predict how useful they are for a specific user (Ricci, 2020).

Table 1 provides an overview of selected articles applying a text analytics approach within the tourism domain.

Table 1 Text analysis in tourism

Name	Year	Title	Objectives	Methodology	Software
Yu & Egger	2021	Tourist experiences at overcrowded attractions: A text analytics approach	To explore the perception and feelings of tourists when visiting overcrowded attractions	Topic modelling; 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	Topic modelling	Python
Chang, Ku, & Chen	2020	Using deep learning and visual analytics to explore hotel reviews and responses	To analyse hotel reviews and responses collected on TripAdvisor and to identify response strategies	Visual analytics; deep learning-based natural language processing	Selenium; TripCollective; geocoding API; tableau
Chen, Wang, Zhu, & Lian	2020	Will you miss me if I am leaving? Unexpected market withdrawal of Norwegian Joy and customer satisfaction	To investigate how customer satisfaction changes as a response to a cruise's market-withdrawal decision	Sentiment analysis	Baidu Senta system
Li, Li, Law, & Paradies	2020	Racism in tourism reviews	To validate the impact of racial discrimination on tourists' experience	Semantic analysis; sentiment analysis	Python
Andreu, Bigne, Amaro, & Palomo	2020	Airbnb research: An analysis in tourism and hospitality journals	To examine Airbnb research using bibliometric methods	Stemming; n-grams detection	R
Guerreiro & Rita	2020	How to predict explicit recommendations in online reviews using text mining and sentiment analysis	To explore what may drive reviewers to make direct endorsements in text	Text mining; sentiment analysis	SPSS modeler text analytics
Zhang, Yang, Zhang, & Zhang	2020	Designing tourist experiences amidst air pollution: A spatial analytical approach using social media	To propose a spatial analytical framework from geotagged social media data in Beijing for a better understanding of tourist experiences	Sentiment analysis; deep learning classifiers; econometric analysis	Linguistic inquiry and word count (LIWC) program

(continued)

Table 1 (continued)

Name	Year	Title	Objectives	Methodology	Software
Tsai, Chen, Hu, & Chen	2020	Improving text summarisation of online hotel reviews with review helpfulness and sentiment	To evaluate how review helpfulness and hotel features improve the results of hotel review summarisation	Segmentation; stemming; part of speech	Stanford CoreNLP
Chantrapornchai & Tunsakul	2019	Information extraction based on named entity for tourism corpus	To extract information in order to help with ontology data acquisition within the tourism domain	Named entity recognition	Python; spaCy
Deng, Liu, Dai, & Li	2019	Different cultures, different photos: A comparison of Shanghai's pictorial destination image between east and west	To propose a new method to compare destination image differences among inbound tourists	Part of speech; sentiment analysis	Python; TextBlob
Ban, Joung, & Kim	2019	The text mining approach to understand seat comfort experience of airline passengers through online review	To explore the seat comfort experience of airline passengers via online reviews	Text mining; semantic network analysis	Python;NetDraw
Yadav & Roychoudhury	2019	Effect of trip mode on opinion about hotel aspects: A social media analysis approach	To uncover the aspects of hotels discussed by people from online reviews	Aspect-based sentiment analysis; part of speech	R
Fazzolari & Petrocchi	2018	A study on online travel reviews through intelligent data analysis	To assist providers in adapting their services to help customers improve their decision processes based on review data	Tokenisation; language-specific and domain-specific stop words; stemming	Python; Scikit-learn
Antonio, de Almeida, Nunes, Batista, & Ribeiro	2018	Hotel online reviews: Creating a multi-source aggregated index	To obtain a prediction model for hotel review ratings	Sentiment analysis	C#; R

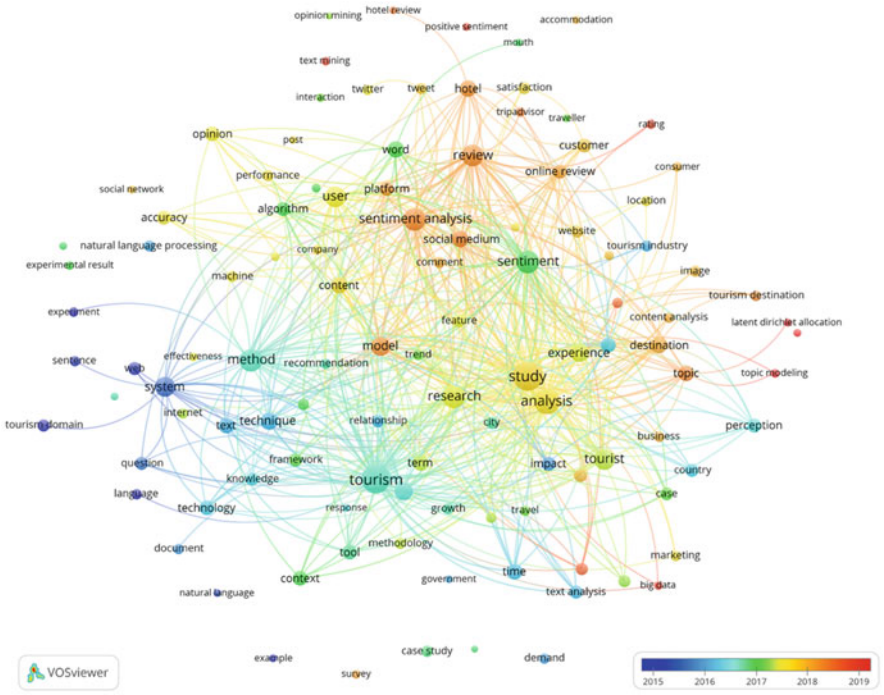


Fig. 2 NLP development in tourism

Figure 2 shows the development of Natural Language Processing in the field of tourism based on 356 identified articles. Relevant terms such as "Text Analysis", "Natural Language Processing", "Topic Modeling", "Sentiment Analysis", "Text Summarization", etc., were used in combination with "Tourism" to obtain articles from Web of Science and Scopus. One can clearly see that sentiment analysis has been one of the most frequently applied techniques since 2018. Contrarily, topic modelling, mostly by applying the Latent Dirichlet Allocation approach, only first developed in 2019, whereby, as already indicated, there is a strong connection to sentiment analysis. The graph also clearly shows that online reviews are the main source for text analyses.

3 NLP Techniques

Driven by advances in machine learning, NLP has been a well-established field under the realm of artificial intelligence, linguistics, computer science, and information engineering. NLP uses machines to read, understand, and make sense of human languages so as to streamline data mining, data analysis, and business operations (Han et al., 2016). Owing to improvements in NLP, a sea change from conventional

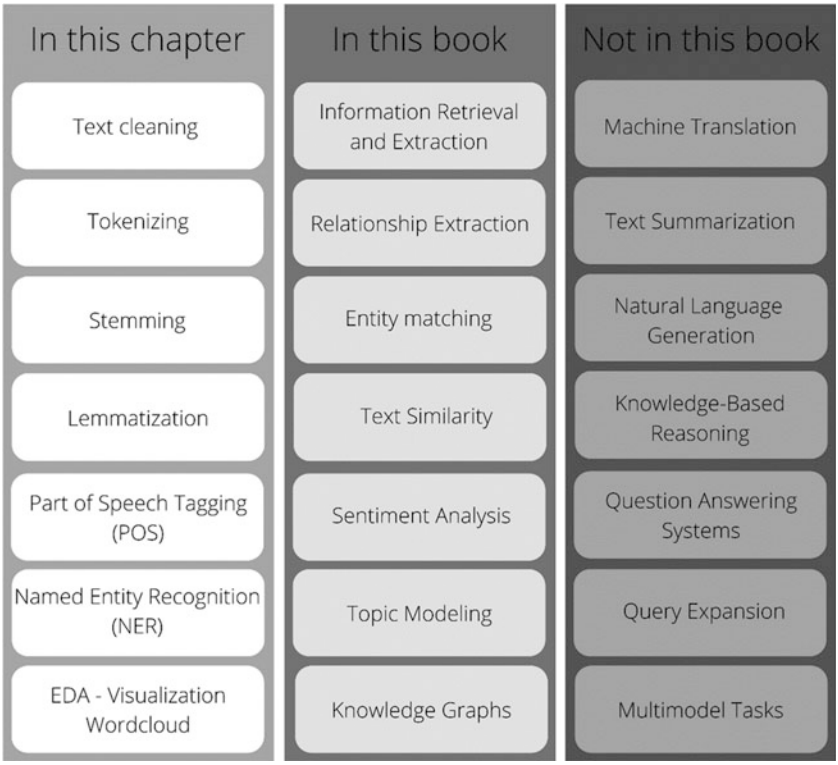


Fig. 3 NLP techniques covered in this book

manual coding to the automation of data collection, data cleaning, and statistical analysis has occurred. Essentially, NLP covers diverse tools and techniques such as named entity recognition, information extraction, topic classification, text analysis, sentiment analysis, and word embedding, amongst many others (Li et al., 2019).

Figure 3 shows the techniques presented in this chapter/book, respectively, as well as additional advanced tools that are not discussed here.

4 Text Preparation and Pre-processing

This chapter is mainly dedicated to the pre-processing of texts. "Garbage in, garbage out" is an accurate statement that can also be applied to the analysis of texts since if the pre-processing stage is not carried out properly, then the algorithms used will subsequently have problems processing the content correctly. Text pre-processing can be understood as a series of operations that are necessary for a machine to automatically process texts (Kannan & Gurusamy, 2014). Thus, the main purpose of text cleaning is to systematise the text data (Albishre et al., 2015). Uncleaned data

can contain many potential issues, such as misspelt words, incorrect punctuation, improper spacing, etc., and using uncleaned data can even distort the document's linguistics and inhibit information extraction processes (Saralegi & Leturia, 2007).

Moreover, in NLP methods, each word is viewed as a variable (dimension). On the one hand, the aim is trying to keep the vocabulary and, thus, the dimensions as small as possible, while, on the other hand, also attempting to keep them large enough so that no critical information is lost. Removing noise from the document can reduce computational costs and increase NLP models' performance (Kumar & Babu, 2019).

It must be noted that text cleaning is an intensive process. It is estimated that text pre-processing takes about 80% of the time, whereas only 20% of the effort is required for data analysis (IDC, 2018). During text cleaning, the researcher makes subjective decisions that can lead to biased results of the analysis; therefore, text cleaning has a crucial impact on text analysis and the final results. In the following, the different steps needed for text data pre-processing are described in detail.

4.1 Language Detection

When it comes to texts, in the context of tourism especially, corpora often consist of documents in several languages. Until recently, most algorithms have required one specific language to be processed; however, multilingual algorithms have now become more common (Keung et al., 2020). At the same time, research often focuses on analysing texts in a defined language. As such, language detection algorithms like *spacy-langdetect* allow the language of documents to be identified. It makes sense to perform the language detection at the beginning of the text wrangling process in order to filter just those documents from the entire corpus that correspond to the target language and need to be processed further.

4.2 Tokenisation

To perform NLP tasks, a robust vocabulary is needed. Each corpus exists of a number of documents, which represent one instance each and serve as a unique identifier. This means that each document collection (corpus) consists of individual documents that are formed by individual tokens. To receive the required vocabulary, a document needs to be split into distinct tokens of meaning (Hapke et al., 2019). Languages such as English, German, or French are space-delimited, meaning that most terms are separated by white spaces. For languages like Thai or Chinese, however, this is not the case; as these languages do not provide clear boundaries between words, tokenisation becomes a challenging task (Kannan et al., 2014). Tokenisation is often the first step in a classic NLP pipeline (Cook et al., 2016),

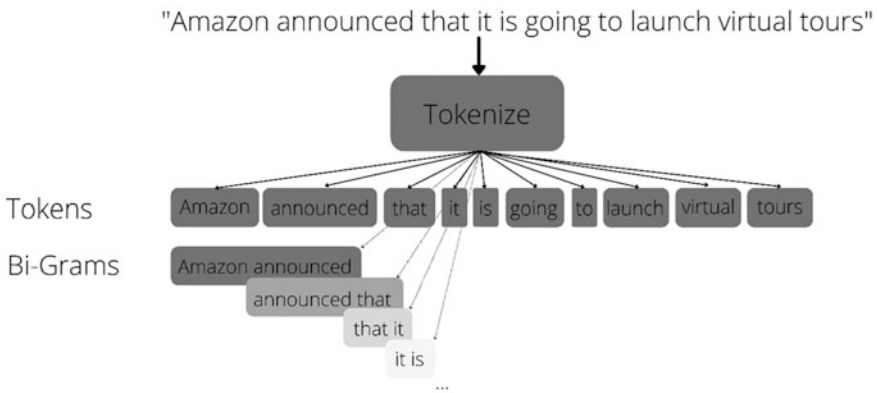


Fig. 4 The tokenisation process

but, if required for specific reasons, it can also be done at a later point in time, as will be shown in our practical demonstration.

The following document in Fig. 4 exists of 10 individual words. By tokenising this text, it is split into more useable tokens.

Often, however, one may not only want to represent a single word as a token, but also successive words. N-grams are, therefore, alternatives to single words. A bi-gram, for example, consists of two words in a row, a tri-gram of three words in a row, and so on (Anandarajan et al., 2019; Sarkar, 2019).

4.3 Lowercasing and Removal of Punctuation

Usually, the transformation of text into lowercase letters and the removal of punctuation are considered the first pre-processing steps. Lowercase text data is particularly relevant so as to avoid word redundancy. For example, when counting words, the terms "Tourism" and "tourism" would be counted as two separate words, leading to an undesired increase in the dimensions of the data. At the same time, punctuation marks create noise in the data and do not add value to the analysis, explaining why they should be removed. However, in certain situations, it makes sense to analyse individual sentences; in such cases, the corpus should be separated into individual sentences before punctuation marks are removed.

4.4 Expand Contractions

A contraction is "a shortening of a word, syllable, or word group by omission of a sound or letter" (Merriam-Webster, 2021); for instance, a contraction like "we'll" is

split into two words, “we” and “will”. Again, the idea is to normalise the text, i.e. to turn tokens that mean the same thing into a normalised form. Thus, the vocabulary does not unnecessarily increase, which leads to the fact that the association of the different spellings of a token has decayed. At the same time, the likelihood of overfitting can be reduced (Hapke et al., 2019).

4.5 Removal of Stop Words

Stop words are common words in English such as “a, in, the, can, may”, and so on. These words are not useful in NLP analyses because they can be part of any sentence; in other words, stop words are not considered keywords in text mining methods (Porter, 1980). Furthermore, they increase vocabulary and hardly provide useful information (Vijayarani et al., 2015).

There are numerous Python packages that can be used to load stop words, with each package varying in size. For example, the NLTK (Natural Language Toolkit) package stores a list of stop words for 16 different languages, with 127 stop words specified for English (Bird et al., 2009), while the Stanford NLP package contains 257 English stop words (Qi et al., 2018). In most cases, one will use standard stop words but can add individually defined stop words as well.

4.6 Removal of URLs, HTML Tags, and Emotions/Emojis

Regarding social media posts from Twitter, Facebook, or Instagram, URLs are often a part of documents. However, these limit NLP algorithms from recognising the actual meaning of a sentence. Thus, they are merely noise within a text and should be deleted. (Baldwin et al., 2013; Sarker & Gonzalez, 2016; Kumar & Babu, 2019). The same applies to HTML tags as well as emojis and emoticons. Emojis are symbols such as, while an emoticon is a character such as:-). Even though they convey information about feelings, we might not want to analyse them and are better off being removed from the data.

4.7 Correction of Spelling

Spelling errors can be seen as another example of increasing the number of features in a vocabulary due to corresponding words being counted twice. For example, the words “tourism” and “turism” would be considered two different words. To correct spelling errors, many different Python packages using different approaches are available, such as “*pyspellchecker*”, “*SymSpell*”, “*TextBlob* Spell Checker”, etc. However, their execution time and performance differ significantly. Spelling

correction algorithms use two main approaches: context-sensitive spelling correction and isolated term spelling correction. In the latter, the algorithm focuses on a single document at a time, such as a single tweet, and attempts to correct it (Schütze et al., 2008). Here is an example using *TextBlob*, illustrating the power of spelling correction packages:

Original sentence: "Minesota is briming with natual and cltural bauty."

After spelling correction: "Minnesota is brimming with natural and cultural beauty."

4.8 Stemming and Lemmatisation

Stemming and lemmatisation are two very similar concepts but should not be used together. In both cases, the overall goal is to break down the words until only the root is remaining (Mendez et al., 2005). As with the other pre-processing steps, the aim is to avoid redundancies between terms (in this case, the same root) in order to reduce the vocabulary.

Take the following as an example: *"recreation"* is reduced to *"recreat"*.

Since terms with the same root often contain similar meanings, they can be combined into a single token. Yet, although there are terms in which a reduction to the root word leads to different meanings (e.g. booking → book), the incorrect assignment of some words is accepted for the dimension reduction gained (Anandarajan et al., 2019). Popular stemmers include the *Porter Stemmer* (Porter, 1980) and the *Snowball Stemmer* (Porter, 2001).

In contrast to stemming, lemmatising takes the term's part of speech into consideration. When a document is lemmatised, each word in that document is replaced with its lemma, which means that the verbs in the output become the uninflected form of the verb, or the base form, and all nouns are presented in the singular form (Siemens, 1996). For instance, with lemmatisation, "caring" can be transformed to "care" (Table 2).

Depending on the needs of the researcher, different Python libraries are available for the lemmatisation process. Some of them reduce the number of words more than

Table 2 Comparison of a word's original, stemmed, and lemmatised versions

Original word	Stemmed	Lemmatised
tourist	tourist	tourist
booking	book	book
rating	rate	rat
itinerary	itinerari	itinerary
recreation	recreat	recreation
amenities	amen	amenities
attractions	attract	attractions
sightseeing	sightse	sightsee
eating	eat	eat

others, but two of the most popular lemmatisation packages include *WordNet*[1] Lemmatizer and *UDPipe*[2] Lemmatizer.

4.9 Part of Speech Tagging (POS)

With part of speech tagging, individual tokens are labelled based on their parts of speech (Anandarajan et al., 2019). This can be achieved using language models that include dictionaries of terms with all their possible parts of speech (Hapke et al., 2019); for instance, both *NLTK*[3] and *spaCy* are Python modules that enable extensive POS. The example below shows the parts of speech for the sentence "information is the lifeblood of tourism" by using *spaCy*[4] and visualising the dependencies with *displaCy* (Fig. 5).

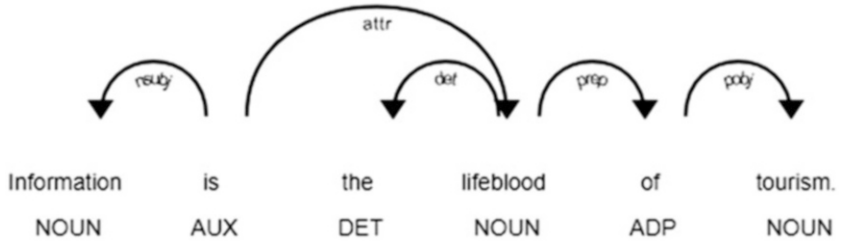


Fig. 5 Part of Speech tagging example

4.10 Named Entity Recognition (NER)

Named entity recognition, also known as entity identification or entity chunking, is an advanced statistical procedure that can no longer be assigned to the preprocessing stage but, rather, to the field of information extraction. Via an entity recognition system, labels (metainformation) can be assigned to contiguous spans of tokens (spaCy, 2021). For example, from the following sentence,

> *Mr. Egger has booked a room from Monday to Friday at the Sheraton in Vienna for €130 per night*

[1] https://wordnet.princeton.edu/
[2] https://ufal.mff.cuni.cz/udpipe/1
[3] https://www.nltk.org/
[4] https://spacy.io/

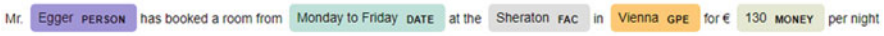


Fig. 6 Named entity recognition example

it can be concluded that "Egger" is a person, "Monday" and "Friday" represent a date unit, "Sheraton" represents an organisation, "Salzburg" is a place, and "130" stands for an amount of money. For Fig. 6, *spaCy*'s entity visualiser was used.

By extracting the identified labels, new features can be generated and used for further analysis, which brings us to the final technique presented in this chapter.

4.11 Feature Extraction

Feature extraction is another essential part of text pre-processing. In the context of machine learning, features represent a variable, and in feature extraction, new variables are extracted from text data to be used for further analysis. These features could be the number of stop words, hashtags or numerical characters, the average word length, etc. When further NLP analyses are carried out, new features emerge that can also be constantly extracted, for example, sentiment analysis results in sentiment scores, term weights in topic modelling or entities (extracted by performing NER), and so on.

4.12 Visual EDA

Wordclouds are a helpful way to visualise text data and get an overview of word frequencies and the importance placed upon them. Typically, the maximum number of words to be displayed as well as parameters such as font size, colour, etc. can be defined. Additionally, a word frequency analysis, sentence length analysis, etc. can be performed and visualised, mainly using bar charts for categorical data and histograms for continuous features.

With the methods presented here, text data can be prepared and pre-processed in a suitable fashion. To be able to carry out text analysis approaches presented in the following chapters, the texts, however, must first be converted into an appropriate format. This can be done by converting the text data that resulted from pre-processing into frequencies (Hapke et al., 2019). The following chapter (16) on "Text Representation and Word Embeddings" is dedicated to these approaches.

5 Challenges of Working with Text

Although NLP is continuously increasing in popularity and can be applied to many scenarios, it comes with a number of challenges as well. Moreover, due to the fact that different languages involve various grammatical constructions, which NLP may be incapable of identifying and/or deciphering, working with textual data can easily turn into a complex task. Challenges of working with textile data stem from five main reasons:

Synonymy: Most NLP approaches pay attention to the frequency of words occurring in the text whilst analysing them. Due to the presence of synonyms, these results may be biased, and the similarity of sentences may be incorrectly evaluated. NLP methods that deal with synonyms usually focus on unigrams (single-word terms) (Blondel & Senellart, 2002), while some other tasks such as machine translation and text simplification consider the similarity between multi-word terms (Hazem & Daille, 2018).

Lexical ambiguity: Lexical ambiguity exists in situations where a term or phrase is considered a homonym (Fielding et al., 2017); in other words, certain words possess multiple meanings. For example, the word "watch" can be used as a noun such as in the case of "I bought a new watch", or it can also be used as a word in a sentence such as "watch your back".

Language-related issues: Different languages have different grammatical structures and rules. For example, in Turkish, there is only one pronoun for he/she/it. These types of cases are challenging for text data analysis.

Referential ambiguity: In many situations, it is not clear what an entity may be referring to (Boyarskaya, 2019). In the example of "Jennifer met her friend at the restaurant before she went back to the hotel". Is she referring to Jennifer or her friend? As such, many different types of reference-ambiguity resolution systems (Manning, 2019) have been rapidly optimised in recent years thanks to the use of neural networks. Referential ambiguity, however, still poses a problem in automatic text analysis.

Out-of-vocabulary problem: Computers can only work with terms if they have seen them before, making new and previously unknown words difficult to process correctly. Therefore, manually created lexical databases are of high significance for text analysis. A commonly used lexical corpus and database with an extensive list of different entities is WordNet.[5] Here, English nouns, verbs, adjectives, and adverbs are grouped into synsets based on semantic similarities, resulting in semantic relations between words already being recorded (Sarkar, 2019). WordNet can therefore be understood as a semantic network of meaningfully related words and concepts.

[5] https://wordnet.princeton.edu/

6 Practical Demonstration

In this section, we will use a Twitter dataset to perform and showcase the most relevant pre-processing steps in text analytics. The corpus used for this demonstration contains 6027 tweets that we crawled for the hashtag #travelsomeday. This very popular hashtag during the COVID-19 pandemic has been used by individuals to tag a post and express their desire to travel again. The code provided below highlights the most relevant elements only, but the dataset, together with the full code and detailed markdowns with a stepwise explanation, will be shared as a Jupyter Notebook and can be downloaded from the books' Github profile. It must be noted that if you are using another Python IDE, minor changes might be required when it comes to the code, installations, and requirements.

After importing the necessary modules and loading the text data into a Pandas data frame, it is now time for data cleaning. We will start by converting all the text into lowercase and expanding the contractions, allowing us to turn terms like "we're" into "we are" or "I'd" into "I would".

```
## Lowercase
df['tweet'] = df['tweet'].apply(lambda x: " ".join(x.lower() for x in
x.split()))
## For contradictions
df['tweet'] = df['tweet'].apply(lambda x: contractions.fix(x))
df.tweet.sample(3)

Output:
4080 share a moment that makes you smile. tell us w...
4192 great american summer road trips: https://t.co...
5168 7 reasons to visit #romania #travelsomeday #ar...
Name: tweet, dtype: object
```

Language detection is another useful tool for data cleaning. Especially for texts relating to tourism, being able to filter out solely the target language is an important step towards acquiring useful data. When the Language Detection algorithm is applied, the existing languages in the corpus can be seen (see output table below). After detecting all the languages in the documents, the desired language can be kept, and the rest of the tweets in the other languages can be dropped.

```
## Language detection

import os, re, nltk, spacy, string, umap
pd.set_option('display.max_colwidth', 50)
nlp = spacy.load('en_core_web_sm')
!pip install spacy-langdetect
from spacy_langdetect import LanguageDetector
nlp.add_pipe(LanguageDetector(), name='language_detector',
last=True)
# Defining 'detect language' function
def detect_language(tweet):
```

```
    try:
    doc = nlp(t)
    language = doc._.language['language']
    score = doc._.language['score']
    except:
    language = ''
    print('sth goes wrong')
    return language, score
    languages = []
scores = []
for i, t in df['tweet'].iteritems():
    l, s = detect_language(t)
    if i % 500 ==0:
    print(i, t, l)
    languages.append(l)
    scores.append(s)

df['lng'] = languages
df['lng_score'] = scores
# Check the output
df[['tweet', 'lng','lng_score']].sample(10).round(2)
df.lng.value_counts()
# Keep only the 'English' language
df = df.loc[df.lng == 'en']
# Example sentences
it was such a beautiful sunset
l'anno prossimo verremo di nuovo
お勧めのレストランです。
```

Output:

ID	Tweet	lng	lng_score
1745	it was such a beautiful sunset	en	1.0
1648	l'anno prossimo verremo di nuovo	it	1.0
885	お勧めのレストランです。	jp	0.9

The next cleaning step is removing contractions. This step should be performed before word tokenisation as NLTK has a built-in method dealing with contractions. NLTK, however, only separates contractions without expanding them.

```
## Expand contractions

# creating an empty list
expanded_words = []
for word in text.split():
    # using contractions.fix to expand the shortened words
    expanded_words.append(contractions.fix(word))

expanded_text = ' '.join(expanded_words)
print('Original text: ' + text)
```

```
print('\n')
print('Expanded_text: ' + expanded_text)

# Example sentence
She'll be airport in 30 mins. We are supposed to catch the arrival,
aren't we?
I'd love to welcome her personally. It'll be an awesome vacation.

Output:
She will be airport in 30 mins. We are supposed to catch the arrival, are
not we? I would love to welcome her personally. it will be an awesome
vacation.
```

Next, we will move on to stop word removal.

```
## Stopword removal

df['tweet'] = df['tweet'].apply(lambda x: " ".join(x for x in x.split()
if x not in stop))
df['tweet'].sample(5)
```

It might be the case that the predefined stop words do not suffice for your task, in which case you might need to additionally define your own set of stop words.

```
## Adding your own stopwords

add_words = ["also","retweet","comment",]
stop_words = set(stopwords.words("english"))
stop_added = stop_words.union(add_words)
df['tweet'] = df['tweet'].apply(lambda x: " ".join(x for x in x.split()
if x not in stop_added))
df['tweet'].sample(3)
```

Texts that come from social media platforms in particular, like Twitter, Facebook, or Instagram, and contain URLs should be removed as they hinder the NLP algorithms' detection of the actual meaning of a sentence (Baldwin et al., 2013; Sarker & Gonzalez, 2016; Kumar & Babu, 2019). As they merely create noise in our data, we will delete all URLs in our corpus.

```
## Removing URLs

def remove_urls (vTEXT):
 vTEXT = re.sub(r'(https|http)?:\/\/(\w|\.|\/|\?|\=|\&|\%)*\b', '',
vTEXT, flags=re.MULTILINE)
 return(vTEXT)
df['tweet'] = df.tweet.apply(remove_urls)
df.tweet.head()
```

```
# Example sentence
sentence= 'NPR can provide useful information https://www.npr.org'
remove_urls(sentence)

Output:
'NPR can provide useful information'
```

We may also need to strip the texts of any HTML tags.

```
## Stripping HTML tags

def strip_html_tags(text):
 soup = BeautifulSoup(text, "html.parser")
 stripped_text = soup.get_text()
 return stripped_text
df['tweet'] = df['tweet'].apply(lambda x: strip_html_tags(x))
df['tweet'].head()

# Example sentence
strip_html_tags('People and neighbors are for the most part very
friendly,</a> even in stores</td>')

Output:
'People and neighbors are for the most part very friendly, even in stores'
```

Unless you want to analyse emojis, which could indeed be interesting as they convey information about feelings, they should also be removed from the data.

```
## Removing emojis

def remove_emoji(text):
  emoji_pattern = re.compile("["
                u"\U0001F600-\U0001F64F" # emoticons
                u"\U0001F300-\U0001F5FF" # symbols & pictographs
                u"\U0001F680-\U0001F6FF" # transport & map symbols
                u"\U0001F1E0-\U0001F1FF" # flags
                u"\U00002702-\U000027B0"
                u"\U000024C2-\U0001F251"
                "]+", flags=re.UNICODE)
  return emoji_pattern.sub(r'', text)

# Example
remove_emoji("Have fun with NLP!😀😀")
Output:
'Have fun with NLP! '

# Remove all emojis from tweets
df['tweet'] = df['tweet'].apply(lambda x: remove_emoji(x))
```

In the steps above, we only removed the emojis, but, in this case, it is also relevant to remove emoticons.

```
## Remove emoticons

!pip install emot
from emot.emo_unicode import UNICODE_EMO, EMOTICONS

# Function for removing emoticons
def remove_emoticons(text):
 emoticon_pattern = re.compile(u'(' + u'|'.join(k for k in EMOTICONS) +
u')')
 return emoticon_pattern.sub(r'', text)
df['tweet'] = df['tweet'].apply(lambda x: remove_emoticons(x))
df.tweet.sample(5)

# Example
remove_emoticons( "This hotel is just awesome :)" )

Output:
'This hotel is just awesome'
```

Next, we want to group together the inflected versions of a word, which allows us to analyse them as a single term. In this case, we will apply lemmatisation only.

```
## Lemmatisation

word_list = ["tourist", "booking", "rating","itinerary","recreation]
print("{0:20}{1:20}{2:20}".format("Original
Word","Stemmed","Lemmatized"))

for word in word_list:
 print("{0:20}{1:20}{2:20}".format(word,porter.stem(word),
wordnet_lemmatizer.lemmatize(word,pos="v")))

Output:
Original Word Stemmed Lemmatised
tourist tourist tourist
booking book book
rating rate rat
itinerary itinerari itinerary
recreation recreat recreation


## Lemmatisation & Tokenisation

def LemmaSentence(sentence):
 token_words=word_tokenize(sentence)
 token_words
 lemma_sentence=[]
 for word in token_words:
 lemma_sentence.append(wordnet_lemmatizer.lemmatize(word, pos='v'))
 lemma_sentence.append(" ")
 return "".join(lemma_sentence)
```

```
df['tweet'] = df['tweet'].apply(lambda x: LemmaSentence(x))
df.tweet.sample(3)

Output:
671 make friends new city # travelsomeday # armcha...
2098 enter # colombia emergency # passport # travel...
2143 free things phoenix, # arizona # travelsomeda...
```

N-Grams are usually done together with the tokenisation process (Thanaki, 2017). As we did tokenisation together with the lemmatisation above, here, an individual, typical n-Gram code is presented.

```
## n-grams

from nltk import ngrams
sentence = 'Information is the lifeblood of tourism'
n = 3 # defines the number of tokens. Here a tri-gram
ngramsres = ngrams(sentence.split(), n)
for grams in ngramsres:
 print(grams)

Output:
('Information', 'is', 'the')
('is', 'the', 'lifeblood')
('the', 'lifeblood', 'of')
('lifeblood', 'of', 'tourism')
```

Spelling correction helps to identify typos and correct them. This is needed as, otherwise, the number of features would increase, and, for example, “standing” and “standng” words would be regarded as two different words. As previously mentioned, there are many different Python packages that use different approaches for spelling corrections, yet their execution time and performance differ drastically.

```
## Spell correction

# Example sentences for spell correction
print(TextBlob('In Chicago, a lakeside is a goodf lcation to relax').
correct())
print(TextBlob('Minesota is briming with natual and cltural bauty').
correct())

Output:
‘In Chicago, a lakeside is a good location to relax’
‘Minnesota is brimming with natural and cultural beauty’

%time df['tweet'][:10].apply(lambda x: str(TextBlob(x).correct()))
```

As punctuation marks create noise in the data, they should also be excluded.

```
## Removing punctuation

df['tweet'] = df['tweet'].str.replace('[^\w\s]','')
```

The following NER and POS example is not based on the Twitter dataset; instead, a short paragraph has been used to showcase these tools. *spaCy* allows the importation of a huge number of models for many different languages and in different sizes. For this example, we will load the smallest pipeline "en_core_web_sm".

```
## Named Entity Recognition (NER)

import spacy
from spacy import displacy

nlp = spacy.load("en_core_web_sm")

doc = nlp(u"On September 29th, Amazon announced that it is going to launch virtual tours and activities. The launch will start in U.S soon and the service will be available in Europe next year. Jeff Bezos is happy about this 1 Billion $ business.")

for tokens in doc.ents:
 print(tokens)
```

Let us see which entities *spaCy* was able to extract.

```
Output:

September 29th
Amazon
U.S
Europe
next year
Jeff Bezos
this 1 Billion $
```

Next, we would like that all identified entities are highlighted by colour based on entity category.

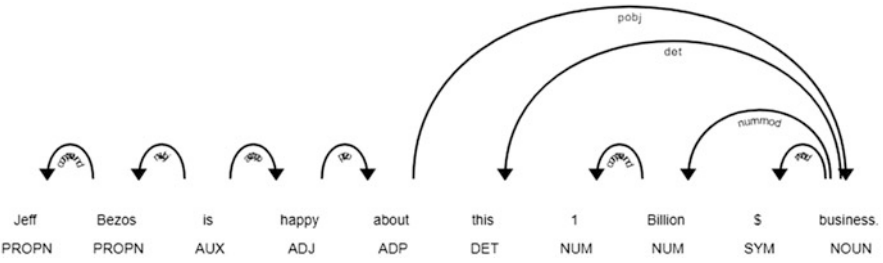
```
displacy.render(doc, style="ent", jupyter=True)
```

Finally, POS can be performed so as to see the different lemmas in this sentence.

```
## Part of Speech Tagging (POS)

def show_lemmas(doc):
 for token in doc:
 print(f'{token.text:{12}} {token.pos_:{6}} {token.lemma:<{22}} {token.
lemma_}')

show_lemmas(doc)

Output:
On ADP 5640369432778651323 on
September PROPN 4843961188194855601 September
29th NOUN 14143533789794879514 29th
, PUNCT 2593208677638477497 ,
Amazon PROPN 7854762596996286480 Amazon
announced VERB 6100327276114004729 announce
that SCONJ 4380130941430378203 that
it PRON 561228191312463089 -PRON-
is AUX 10382539506755952630 be
going VERB 8004577259940138793 go
to PART 3791531372978436496 to
launch VERB 1882817931903534611 launch
virtual ADJ 14425350317076788470 virtual
tours NOUN 6598908817114149421 tour
```

Using *spaCy*'s displaCy visualiser, we then get an overview of our example sentence along with its dependencies.

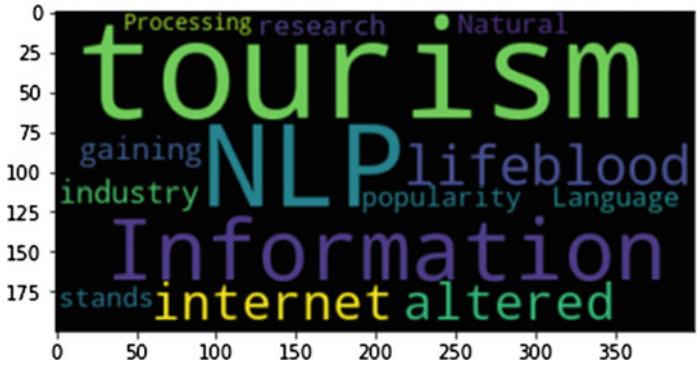
```
displacy.render(doc, style="dep", jupyter=True, options={"distance" : 90})
```



To obtain a first brief overview of the pre-processed text, simple wordclouds are most suitable as they show the frequency of the single words on the basis of the displayed size.

```
## Wordcloud

from wordcloud import WordCloud
import matplotlib.pyplot as plt

text = 'Information is the lifeblood of Tourism. The internet has altered
the tourism industry. NLP is gaining popularity in tourism research. NLP
stands for Natural Language Processing. '
# Generate a word cloud image
wordcloud = WordCloud().generate(text)

plt.imshow(wordcloud, interpolation="bilinear")
plt.show()
```

Output:



6.1 Tips for Using Python for an NLP Study

Despite the fact that we import packages and predefined functions many times, in many cases, it is important to check how the code is written since a code may be doing more than what it is originally designed for. For example, while using the contractions functions in Python, we need to pay attention to how it works as the "contractions" package requires apostrophes in order to identify them. In this situation, if you happen to have removed all punctuations before the contractions, then the contraction function will not be able to find apostrophes and, in turn, fail to perform its task.

Service Section
Main Application Fields: Natural Language Processing is a prominent field of Data Science, which is especially important in tourism due to the many different text data created by customers. Thus, opinions, sentiments, topics, and much more can be extracted from texts.

Limitations and Pitfalls: In order to be able to work analytically with texts, intensive pre-processing is required in most cases. This process step is very labour-intensive and error-prone.

Similar Methods and Methods to Combine with: If the text data gets vectorized (represented in numerical values), then any other ML task can be performed (classification, regression, clustering, etc.).

Code: The Python code is available at: https://github.com/DataScience-in-Tourism/Chapter-15-Introduction-Natural-Language-Processing

Further Readings and Other Sources

Quantum stat provides a list with more than 300 NLP Colab-Notebooks, providing an excellent overview by describing the notebook, the language-model used, and the NLP tasks it is designed for. https://notebooks.quantumstat.com/

Ivan Bilan, the author of chapter 19 (Entity Matching), has established "The NLP Pandect", an incredible comprehensive and helpful collection covering almost all topics on NLP. Among them are compendiums, conference papers, NLP datasets, links to podcasts, newsletters, meetups, YouTube channels, and much more. https://github.com/ivan-bilan/The-NLP-Pandect

The University of Michigan offers a complete NLP course on Youtube https://tinyurl.com/NLP-michigan, and we also recommend to check for free courses on coursea.org, like the ones from DeepLearning.AI https://tinyurl.com/deeplearningai-course or the HSE University https://tinyurl.com/NLP-HSE-course.

Finally, a great NLP course is also provided by Lena Voita, who is teaching at the Yandex School of Data Analysis. https://lena-voita.github.io/nlp_course.html

References

Aicher, J., Asiimwe, F., Batchuluun, B., Hauschild, M., Zöhrer, M., & Egger, R. (2016). Online hotel reviews: Rating symbols or text… text or rating symbols? That is the question! In A. Inversini & R. Schegg (Eds.), *Information and communication Technologies in Tourism 2016* (pp. 369–382). Springer International Publishing.

Alaei, A. R., Becken, S., & Stantic, B. (2017). Sentiment analysis in tourism: Capitalising on big data. *Journal of Travel Research, 1*(9), 175–191.

Albishre, K., Albathan, M., & Li, Y. (2015, December). Effective 20 newsgroups dataset cleaning. In *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)* (Vol. 3, pp. 98–101). IEEE.

Anandarajan, M., Hill, C., & Nolan, T. (2019). *Practical text analytics* (Vol. 2). Springer International Publishing.

Baldwin, T., Cook, P., Lui, M., MacKinlay, A., & Wang, L. (2013, October). How noisy social media text, how different social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing* (pp. 356–364).

Bird, S., Loper, E., & Klein, E. (2009). *Natural language processing with python*. O'Reilly Media.

Blondel, V. D., & Senellart, P. P. (2002). Automatic extraction of synonyms in a dictionary. vertex, 1, x1.

Boyarskaya, E. (2019). Ambiguity matters in linguistics and translation. *Слово.ру: балтийский акцент, 10*(3), 81–93. https://doi.org/10.5922/2225-5346-2019-3-6

Bussière, K. (2018). Chapter 4 – Text analysis (digital humanities - a primer). Available online at https://carletonu.pressbooks.pub/digh5000/chapter/chapter-4-text-analysis/.

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.

Chang, Y. C., Ku, C. H., & Chen, C. H. (2020). Using deep learning and visual analytics to explore hotel reviews and responses. *Tourism Management, 80*, 104129.

Chantrapornchai, C., & Tunsakul, A. (2019). Information extraction based on named entity for tourism corpus. In *2019 16th International Joint Conference on Computer Science and Software Engineering* (pp. 187–192). IEEE.

Conti, E., & Lexhagen, M. (2020). Instagramming nature-based tourism experiences: A netnographic study of online photography and value creation. *Tourism Management Perspectives, 34*, 2–3.

Cook, P., Evert, S., Schäfer, R., & Stemle, E. (Eds.). (2016). *Proceedings of the 10th Web as Corpus Workshop*. Association for Computational Linguistics.

Egger, R. (2010). Theorizing web 2.0 phenomena in tourism: A sociological signpost. *Information Technology & Tourism, 12*(2), 125–137. https://doi.org/10.3727/109830510X12887971002666

Fielding, N. G., Lee, R. M., & Blank, G. (2017). *The SAGE handbook of online research methods*. SAGE Publications Ltd.

García-Pablos, A., Cuadros, M., & Linaza, M. T. (2016). Automatic analysis of textual hotel reviews. *Information Technology & Tourism, 16*(1), 45–69.

Guerreiro, J., & Rita, P. (2020). How to predict explicit recommendations in online reviews using text mining and sentiment analysis. *Journal of Hospitality and Tourism Management, 43*, 269–272.

Han, H. J.; Mankad, S.; Gavirneni, N.; Verma, R. (2016). What guests really think of your hotel: Text analytics of online customer reviews. *Cornell Hospitality report, 16*(2), 3–17. Available online at https://scholarship.sha.cornell.edu/cgi/viewcontent.cgi?article=1003&context=chrreports, checked on 4/5/2019.

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

Hapke, H. M., Lane, H., & Howard, C. (2019). *Natural language processing in action*. Manning.

Hazem, A., & Daille, B. (2018, May). Word embedding approach for synonym extraction of multi-word terms. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

IDC (2018). *Time Crunch: Equalising time spent on data management vs analytics*. https://blogs.idc.com/2018/08/23/time-crunch-equalizing-time-spent-on-data-management-vs-analytics/

Ignatow, G., & Mihalcea, R. (2017). *Text mining: A guidebook for the social sciences*. SAGE Publications, Inc.

Kannan, S., & Gurusamy, V. (2014). Pre-processing techniques for text mining. *International Journal of Computer Science & Communication Networks, 5*(1), 7–16.

Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., & Nithya, M. (2014). Preprocessing techniques for text mining. *International Journal of Computer Science & Communication Networks, 5*(1), 7–16.

Kao, A., & Poteet, S. R. (2007). *Natural language processing and text mining*. Springer.

Keung, P., Lu, Y., Szarvas, G., & Smith, N. A. (2020). *The multilingual Amazon reviews corpus*.

Kumar, C. P., & Babu, L. D. (2019). Novel text pre-processing framework for sentiment analysis. In *Smart intelligent computing and applications* (pp. 309–317). Springer.

Li, S., Li, G., Law, R., & Paradies, Y. (2020). Racism in tourism reviews. *Tourism Management, 80*, 104100.

Li, Q., Li, S., Zhang, S., Hu, J., & Hu, J. (2019). A review of text corpus-based tourism big data mining. *Applied Sciences, 9*(16), 3300. https://doi.org/10.3390/app9163300

Ma, Y., Xiang, Z., Du, Q., & Fan, W. (2018). Effects of user-provided photos on hotel review helpfulness: An analytical approach with deep leaning. *International Journal of Hospitality Management, 71*, 120–131.

MacCartney, B. (2014). Understanding natural language understanding. ACM SIGAI Bay Area Chapter Inaugural Meeting, 2014. Available online at https://nlp.stanford.edu/~wcmac/papers/20140716-UNLU.pdf.

Manning, C. (2019, March 21). *Coreference Resolution [Video]*. Youtube. https://www.youtube.com/watch?v=i19m4GzBhfc&list=PLoROMvodv4rOhcuXMZkNm7j3fVwBBY42z&index=16&ab_channel=stanfordonline

Markopoulos, G., Mikros, G., Iliadi, A., & Liontos, M. (2015). Sentiment analysis of hotel reviews in Greek: A comparison of unigram features. In *Cultural tourism in a digital era* (pp. 373–383). Springer.

Mendez, J. R., Iglesias, E. L., Fdez-Riverola, F., Diaz, F., & Corchado, J. M. (2005, November). Tokenising, stemming and stopword removal on anti-spam filtering domain. In *Conference of the Spanish Association for Artificial Intelligence* (pp. 449–458). Springer.

Merriam-Webster. (2021). Contraction. In Merriam-Webster.com dictionary. Retrieved January 14, 2021, from. https://www.merriam-webster.com/dictionary/contraction

Munezero, M., Montero, C. S., Sutinen, E., & Pajunen, J. (2014). Are they different? Affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE Transactions on Affective Computing, 5*(2), 101–111.

Poon, A. (1993). *Tourism, technology and competitive strategies*. CAB International.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137.

Porter, M. F. (2001). *Snowball: A language for stemming algorithms*. Available online at http://snowball.tartarus.org/texts/introduction.html.

Qi, P., Dozat, T., Zhang, Y., Manning, C. D., 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies.*

Ricci, F. (2020). Recommender systems in Tourism. In Z. Xiang, M. Fuchs, U. Gretzel, & W. Höpken (Eds.), *Handbook of e-Tourism* (pp. 1–18). Springer International Publishing; Imprint Springer.

Rockwell, G. (2003). What is text analysis, really? *Literary and Linguistic Computing, 18*(2), 209–219.

Saralegi, X., & Leturia, I. (2007). Kimatu, a tool for cleaning non-content text parts from HTML docs. In *Proceedings of the 3rd Web as Corpus Workshop* (pp. 163–167).

Sarkar, D. (2019). *Text analytics with python*. Apress.

Sarker, A., & Gonzalez, G. (2016, December). Data, tools and resources for mining social media drug chatter. In *Proceedings of the fifth workshop on building and evaluating resources for biomedical text mining (BioTxtM2016)* (pp. 99–107).

Schütze, H., Manning, C. D., & Raghavan, P. (2008). *Introduction to information retrieval* (Vol. 39, pp. 1041–4347). Cambridge University Press.

Siemens, R. (1996). *Lemmatization and parsing with TACT pre-processing programs*. Digital Studies/Le champ numérique.

Thanaki, J. (2017). *Python natural language processing. Explore NLP with machine learning and deep learning techniques*. Packt.

Tsai, C.-F., Chen, K., Hu, Y.-H., & Chen, W.-K. (2020). Improving text summarization of online hotel reviews with review helpfulness and sentiment. *In Tourism Management, 80*, 104122. https://doi.org/10.1016/j.tourman.2020.104122

Vijayarani, S., Ilamathi, M. J., & Nithya, M. (2015). Pre-processing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks, 5*(1), 7–16.

Wennker, P. (2020). *Künstliche Intelligenz in der Praxis. Anwendung in Unternehmen und Branchen: KI wettbewerbs- und zukunftsorientiert Einsetzen*. Springer Gabler. Available online at https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=6326361

Xiang, Z. (2018). From digitisation to the age of acceleration: On information technology and tourism. *Tourism Management Perspectives, 25*, 147–150.

Xiang, Z., Du, Q., Ma, Y., & Fan, W. (2017). A comparative analysis of major online review platforms: Implications for social media analytics in hospitality and tourism. *Tourism Management, 58*, 51–65.

Yang, L., Cao, H., Hao, F., Zhang, W. Z., & Ahmad, M. (2020). Research on tourism question answering system based on xi'an tourism knowledge graph. *Journal of Physics: Conference Series, 1616*(1), 12090. https://doi.org/10.1088/1742-6596/1616/1/012090

Yu, J., & Egger, R. (2021). Tourist experiences at overcrowded attractions: A text analytics approach. In W. Wörndl, C. Koo, & J. L. Stienmetz (Eds.), *Information and Communication Technologies in Tourism 2021. Proceedings of the ENTER 2021 eTourism Conference, January 19–22, 2021* (pp. 231–243). Springer.