JOHANNES KEPLER
UNIVERSITY LINZ

Submitted by
Veronika Arefeva

Performed at
Institute of Business Informatics - Data & Knowledge Engineering and Institute of Computational Perception - Multimedia Mining and Search Group

Thesis Supervisors
o. Univ.-Prof. Dipl.-Ing. Dr. Michael Schrefl and
Univ.-Prof. Dipl.-Ing. Mag. Dr. Markus Schedl

Assistant Thesis Supervisors
FH.-Prof. Mag. Dr. Roman Egger and Ass.-Prof. Dr. Navid Rekabsaz

September 2022

Similarity-based ranking of European tourism destinations leveraging Airbnb experiences and custom pre-trained TourBERT embeddings

Master's Thesis
to obtain the academic degree of
Master of Science
in the Master's Program
Business Informatics

JOHANNES KEPLER UNIVERSITY LINZ

Table of Contents

List of Tables

List of Figures

JOHANNES KEPLER UNIVERSITY LINZ

List of Abbreviations

Abbreviation	Explanation
ALBERT	A Lite BERT
AUC	Area under ROC-curve
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bi-directional LSTM
BIO	Beginning-Inside-Outside
BPE	Byte-pair Encoding
CBOW	Continuous Bag Of Words
CF	Collaborative Filtering
CNN	Convolutional Neural Network
CRF	Conditional Random Fields
ELMo	Embeddings from Language Model
GAN	Generative Adversarial Network
GELU	Gaussian Error Linear Unit
GPT	Generative Pre-trained Transformer
GPU	Graphical Processing Unit
KG	Knowledge Graph
LDA	Latent Dirichlet Allocation
LSTM	Long Short-Term Memory
MLM	Masked Language Modeling
MTE	Memorable Tourism Experience
NER	Named Entity Recognition
NLP	Natural Language Processing

NLTK	Natural Language Processing Toolkit
NSP	Next Sentence Prediction
PCA	Principal Component Analysis
POF	Perceived Overall Fit
POI	Point Of Interest
QA	Question Answering
RE	Relation Extraction
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RQ	Research Question
SGD	Stochastic Gradient Descent
t-SNE	T-distributed Stochastic Neighbor Embedding
TDPS	Tourist-destination personality similarity
tf-idf	Term frequency–inverse document frequency
TKG	Tourism Knowledge Graph
TPU	Tensor Processing Unit
UMAP	Uniform Manifold Approximation and Projection
USP	Unique Selling Proposition
VQA	Visual Question Answering

1. Introduction

Tourism is one of the largest industries worldwide, and it continues to grow exponentially with the abundance of user content generated on a daily basis (Shaikh and Kulkarni, 2020). The expansion of social networking and recent advancements in Artificial Intelligence and Machine Learning have thus opened up a new era of data mining, providing novel opportunities for the analysis of large amounts of data. Specifically to the tourism industry, user-generated data such as social media posts, online reviews, and ratings, amongst others, are of greater importance as they provide a solid basis for personalized recommendations and targeted advertisement. Nowadays, it is impossible to create a personalized service without intelligent techniques for Big Data processing; therefore, in tourism, modern online services with personalized recommendations can be considered a product of a vast amount of user-generated content that has undergone the most advanced machine learning algorithms (Zhang et al., 2021).

One of the most widespread sources of tourism data are online reviews, which include user comments and feedback on their experiences, ratings for purchased services or products, and impressions about recently visited locations, popular tourist destinations, and so forth. As a result, the availability of textual data on the Internet has allowed for the creation of reliable datasets for various NLP-tasks including text classification, topic modeling, question answering (QA), and text summarization, to name but a few. Methods as such can cultivate large business value for the tourism industry by automating many processes such as targeted advertising, personalized recommendations, and more (Li et al., 2019). A general framework for Big Data mining in the tourism domain, provided by Li et al. (2019), can be seen in Figure 1 below:

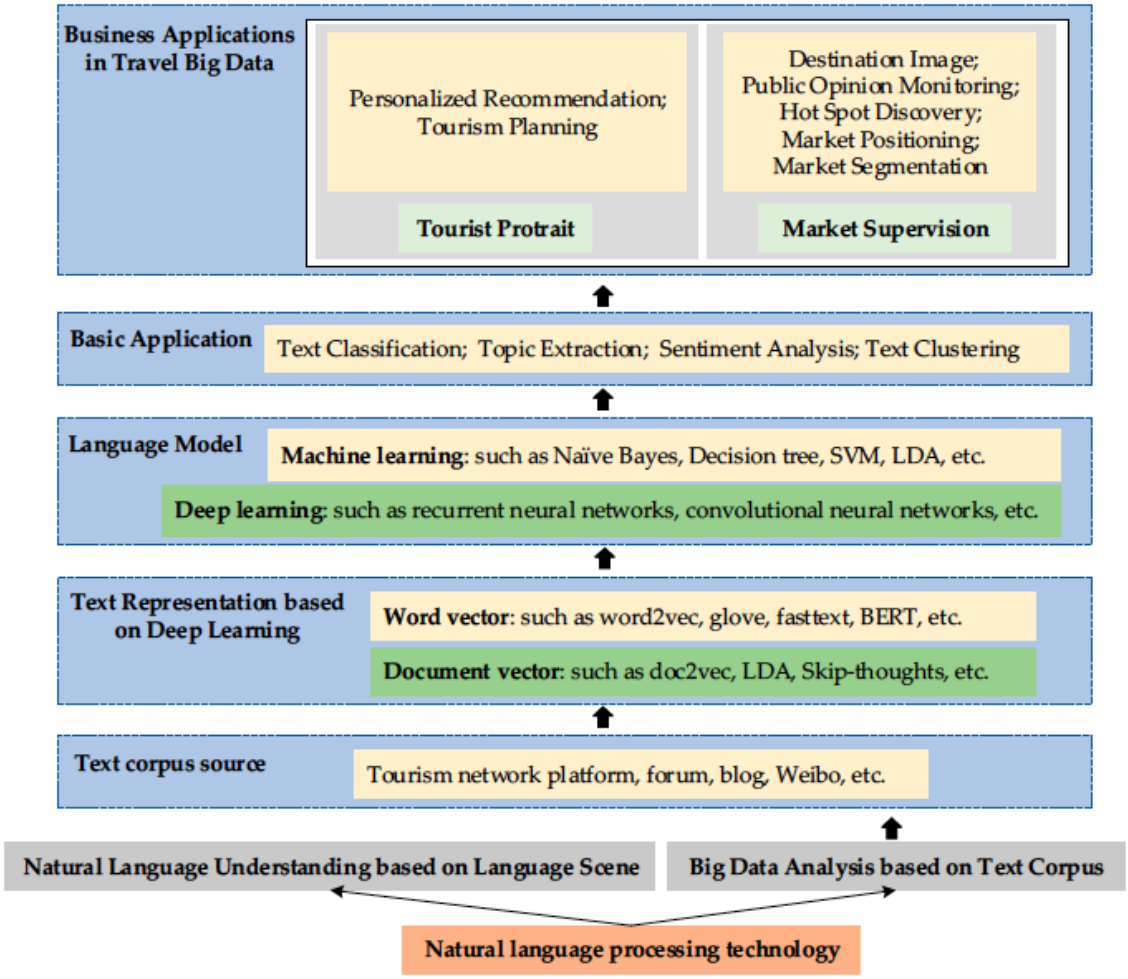


Figure 1: Text corpus-based tourism involving Big Data mining (Li et al., 2019).

Figure 1 shows a general approach towards Big Data processing applied to textual data, collected from tourism-related social media. The initial step at the bottom involves data collection, which is often performed using web-scraping technologies. After data collection, i.e., once textual data becomes available, it must be transformed into a numerical equivalent using NLP techniques, one of the main focuses of this work. Thereafter, depending on the final business goal one wishes to achieve, different machine learning algorithms can be applied. The chosen algorithm then exhausts the numerical vectors created in the previous step. By performing at least one, or a combination, of the tasks listed in the framework above, many problematic business aspects in the tourism industry, including personalized recommendation systems, destination image exploration, opinion monitoring, and market segmentation, can be addressed. For example, the Airbnb platform is one example of a personalized recommendation

system that provides many offerings in a variety of categories, also referred to as experiences. An experience at a certain location is a special offer covering activities like sightseeing, photo shoots, sports, and others, all provided by locals. The importance of experiences in the context of tourism offerings has been studied intensely by many researchers and will be explained in more detail in the subsequent paragraphs.

Society has entered a new stage in which what is known as the service-based economy has emerged into a version of experience economy, providing a fresh foundation for destination management organizations (Mehmetoglu and Engen, 2011). Within this concept, consumer behavior tends to place an emphasis on emotional and participative aspects over functional and rational ones (Morgan et al., 2009, Sthapit and Jimenez-Barreto, 2018, Chang, 2018). Furthermore, since the service-oriented economy, mostly centered around accommodation and transportation (e.g., flights), has seen decreases in attention in recent years due to much more affordable tourism services, tourists are now searching for more unique experiences (Morgan et al., 2009).

Chang (2018) as well as Sthapit and Jimenez-Barreto (2018) discuss experiences within the realm of memorable tourism experience (MTE), with experiences providing new economic value after services (Sthapit and Jimenez-Barreto, 2018). An experience differs from a service in that, unlike a product or a service, it is associated with individuals' needs to create their own identities and to shape their personalities in a life characterized by increased freedom and an improved economy. Hence, experiences are important for people's self-perception (Mehmetoglu and Engen, 2011). The uniqueness of an experience also contributes to a trip's memorability (Cheng, 2016) and is therefore an event that is more easily remembered (Sthapit and Jimenez-Barreto, 2018). Chang (2018) considers an experience from a postmodernist's point of view, where postmodernism is associated with diversified interests, motivations, and activities.

However, strategies of creating unique and memorable experiences have yet to be studied sufficiently and recognized by destination managers (Morgan et al., 2009, Chang, 2018). On this note, destination management organizations are familiar with offerings provided by other destinations or professional tourism agencies but are unaware of the services offered by locals. Being familiar with the experience economy and capable of enhancing marketing strategies with unique MTEs can bring about a significant competitive advantage (Morgan et al., 2009), raise

industry revenue through increased consumer experienced utility (Chang, 2018), and raise customer satisfaction (Mehmetoglu and Engen, 2011). Sthapit and Jimenez-Barreto (2016) attribute the importance of experiences for the tourism industry to the fact that tourists who have positive MTEs are more likely to visit a destination again as well as recommend it to others.

This thesis aims to fill the aforementioned gap by means of a destination comparison based on services provided by locals while, at the same time, also providing a basis for a destination recommender system. While the first contribution is important for destination management organizations, the latter is of greater importance for individual tourists. From a business perspective, the concept of destination similarity based on experiences can provide a valuable tool for destination management organizations since it helps them to better understand their competition and therefore determine their unique selling propositions (USP) more efficiently. From a tourist's point of view, recommendations based on the aspect of destination similarity can play a key role when choosing a destination.

As the main source for experience data, this thesis makes use of the Airbnb platform in order to analyze the similarities between 69 major European destinations based on their experiences. Airbnb is one of the key market leaders in the tourism experience economy, combining multiple USPs such as lower costs, interactions with the local community, and unique experiences, which altogether differentiate Airbnb from other tourism platforms (Cheng, 2016). Each experience on Airbnb contains a description ("What you will do"), rating, price, geographical position, and information about the host as well as some administrative aspects.

Thus, this thesis utilizes the textual descriptions of Airbnb experiences as the main component of a destination profile. In order to enable similarity comparison between different destination profiles, Natural Language Processing (NLP) will be applied to obtain a numerical representation for the text corpus. According to Li et al. (2019), this task is one of the most comprehensive in tourism data mining and is based on a concept of word embeddings. A word embedding encapsulates the meaning of a word, ultimately allowing models "to reason". By knowing a word's meaning, they start to learn contextual and grammatical dependencies and are able to give answers to questions, predict a missing piece of text, guess synonyms for a given word, etc. Moreover, converting a text into a vector allows for algebraic operations to be applied, which is one way of enabling the success of machine learning algorithms.

Currently, one of the most advanced natural language models is the Bidirectional Encoder Representations from Transformers (a.k.a., BERT), developed by Devlin et al. in 2018 by Google Research and used, for example, in Google Search engine. As it is publicly available, the model as well as its source code can be downloaded and adjusted to a researcher's specific needs. BERT was pre-trained on the entire English Wikipedia and Books Corpus, which are both based on a general vocabulary obtained from processing large amounts of historical data. Therefore, this model may lack an understanding of linguistic peculiarities and special terminology used in specific domains. As a result, researchers have started to develop their own domain-specific models so that BERTs for biological, clinical, cyber, software, and financial domains have already come to exist. There is, however, no publicly available BERT model for tourism. This work thus aims to create a BERT model for the tourism industry, which is expected to surpass the efficacy of the general BERT model when it comes to various machine learning tasks for tourism-specific contexts.

1.1 Research questions

Based on the existing research and business gap described above, the three following research questions (RQs) and goals have been formulated for this thesis:

RQ1: Does a natural language model for the tourism domain (TourBERT) capture tourism-specific contexts better than a general language model (BERT-Base)?

To answer this question, a customized language model for the tourism domain will be created and utilized to address the subsequent research questions. The BERT pre-training method will be leveraged to create TourBERT – the first ever language model for the tourism domain. BERT pre-training will be performed from scratch using a combined dataset consisting of three million reviews from TripAdvisor and 46,000 sightseeing descriptions from Expedia. TourBERT will then be benchmarked against the BERT-Base model using both quantitative and qualitative evaluation methods; these methods embody both supervised and unsupervised machine learning tasks as well as an extensive user study. After evaluation, the final model will be used to establish destination profiles. One of the main contributions of this work is TourBERT's release to the open-source community, which should encourage many experts throughout the tourism domain to take advantage of the new model for specific business tasks.

RQ2: Which European countries provide similar tourism offers by locals on Airbnb?

In this regard, the aspect of destination similarity ranking is explored and a framework for the comparison of the most popular tourist destinations is introduced based on publicly available destination descriptions from Airbnb (under the “Experiences” section). The results of similarity research should deliver deeper insights into tourism offerings provided by locals and thus help destination managers to optimize their advertisement campaigns. Moreover, the novelty of the destination similarity framework introduced in this thesis should aid in giving destination management organizations a competitive advantage.

RQ3: How can TourBERT help to improve the quality of personalized recommendations?

Whereas the second research question addresses the problem of destination similarity from a business stance, this research question is intended to demonstrate a way in which users or tourists could benefit from experience-based destination similarity. In order to adequately answer this research question, a destination recommendation web-service prototype will be constructed based on TourBERT, and a similarity framework will be introduced later on in this thesis.

1.2 Thesis structure

The present thesis is organized as follows: While chapter 2 provides an overview of existing approaches for destination similarity comparison, chapter 3 explains any existing embedding and language models, including BERT, in detail. Chapter 4 further demonstrates how BERT has been applied to specific tasks in the tourism domain, followed by chapter 5, which describes the technical solution of research questions defined in this work. Thereafter, chapter 6 gives a description of TourBERT and its training procedure, and chapter 7 contains a detailed description of the evaluation procedure and its results. In chapter 8, the results for the European destination comparison are provided, which is subsequently followed by chapter 9, demonstrating the developed destination recommendation service. Lastly, the discussion, conclusion, and limitations of this thesis are then provided in chapter 10.

2. Destination similarity

This chapter provides an overview of previous research regarding destination similarity. The extent of interest in destination similarity in the tourism domain has been rapidly growing alongside the increase of tourism offerings available on the Internet and through social media (Cao and Thomas, 2021). Owing to a digital transformation of society, tourists and travelers now have easier access to uncountable opportunities for their leisure, trips, and free-time activities. Since this vast amount of data, however, has introduced difficulties in gaining a quick overview when searching for one's next vacation destination, the exploration of destination similarity can provide a helpful grouping of offerings – one which a tourist might be keen of (Li et al., 2019). Moreover, not only are customers interested in obtaining quicker insights, but also destination managers are searching for rapid analysis tools that could aid in managing advertisement campaigns as well as recommendations more accurately, thus ultimately increasing overall profits and customer satisfaction (Flores-Muñoz, 2019).

Several studies investigating the destination similarity from both customers' and providers' point of views, for instance, where the comparison is based on different aspects like the geographical location of a city (Akdağ and Oter, 2011), trips and consumer goods prices (Alrasheed et al., 2020), or users' preferences in form of a user search history on different tourism and booking platforms (Ravi and Vairavasundaram, 2016), have been conducted. As an example, Bekk et al. (2016) investigated the level of impact of tourist-destination personality similarity (TDPS) and perceived overall fit (POF) on customer satisfaction and recommendation behavior by performing an extensive user study. TDPS refers to a concept in which personality profiles, for both destinations and tourists, are created in order to perform the act of matching amongst them and to find the best suitable destinations for a tourist as well as a potential customer for a given destination.

When creating a tourism-destination profile, a destination is normally characterized through images, reviews, descriptions, etc. Whereas tourists' expectations influence the choice of a destination, their opinions and behavior depend on their own personality and not on a destination personality. Therefore, it is important to note that a destination personality should match with the tourist's personality, i.e., with the tourist who would be visiting that destination. Moreover, TDPS is a multi-dimensional concept, involving different personality dimensions that

are usually studied separately for the tourist and the destination. Regarding a tourist profile, a personality can be defined based on specific categories, e.g., according to the Big Five model (McCrae and Costa, 1999), or using a broader range of types based on another categorization. Bekk et al's (2016) study evaluated tourist personality in the three following dimensions: sincerity, excitement, and sophistication, as proposed by Aaker et al. (2001). Yet, this may not have been exhaustive enough as various dimensions can have different predictive power and may be considered irrelevant for specific destination types. Regardless, they used personality ratings of the vacationer as well as the holiday destination to compute TDPS for their study.

On the other hand, POF is a different high-level concept, measuring the perceived fit between the person and the environment in an aggregate manner. In the user study by Bekk et al. (2016), POF was directly reported by their participants. One of the findings revealed that TDPS, as a more granular approach, drives POF to some extent; however, POF was an important predictor of tourist satisfaction and recommendation behavior. Furthermore, the results demonstrated that TDPS and POF are non-interchangeable concepts. As both approaches are attitude-based, it was shown that, together, they have a higher influence on satisfaction than on reported recommendation behavior (Bekk et al., 2016).

Alrasheed et al. (2020) created a hybrid recommender system for tourism, also using the concept of destination similarities. The recommender system has collaborative filtering (CF) (Schafer et al., 2007) as an underlying approach, and the system produces recommendations from a list of popular tourism destinations. The similarity between these destinations is computed based on user preferences. The system works as a two-step approach: First, users who are registered in the platform provide some basic preferences, in this case, their most preferred attractions as well as the weather conditions. Depending on those attributes, a user receives recommendations for destinations that are popular among similar users, where users' similarity is measured using the two types of attributes described above. In the second step, the list of destinations is filtered interactively so that the user is invited to provide additional information such as travel dates or a vacation budget as well as other basic attributes. After the user enters all the additional information, a new user profile is created, and an updated list of recommendations is displayed, ranked according to destination match scores. The full architecture of the system can be observed in Figure 2 below:

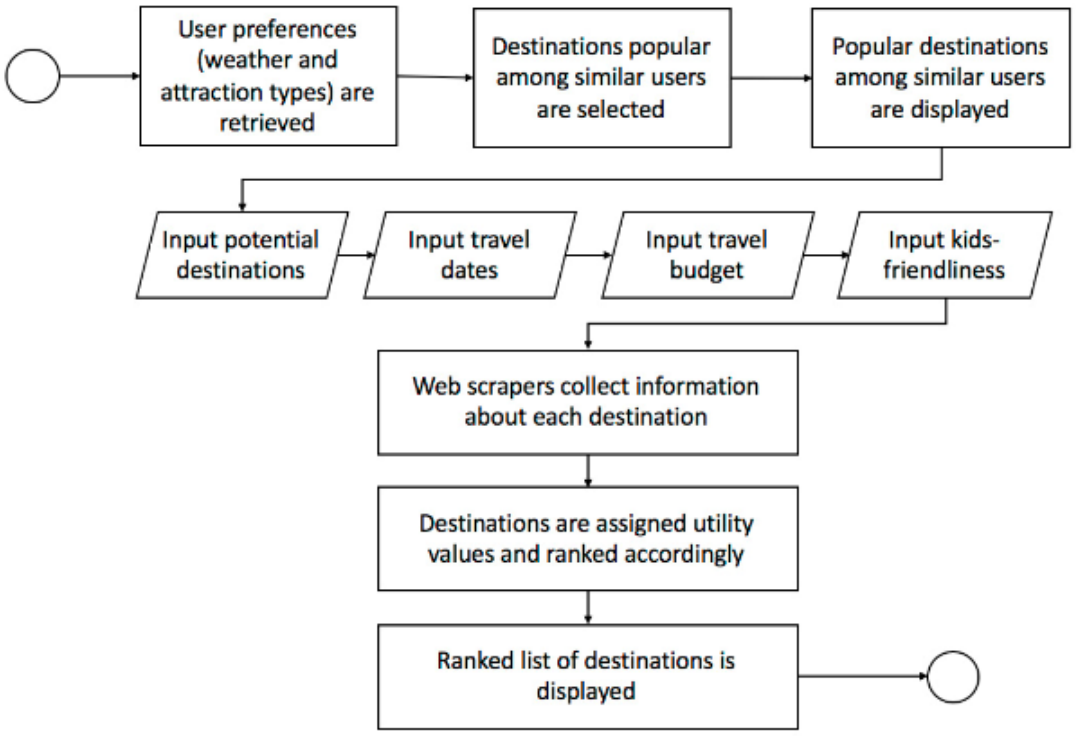


Figure 2: Flowchart of the destination recommendation process by Alrasheed et al. (2020).

In another study, Inversini et al. (2012) explored destinations' similarity based on user-generated picture tags, i.e., a geo-location tag. They relied on a method for predicting similar locations proposed by Clemens et al. (2010), which essentially relies on two hypotheses: 1) Due to specific user preferences, tourist visits' locations should be similar to some extent, and 2) if a user took a photo in a specific location, then it is most likely an indicator of this user liking and admiring this place. From this information, one can conclude that the destination similarity in this study was analyzed from the user's perspective. In turn, the researchers argued that the resulting similarities could be used to create a list of recommendations based on social media pictures shared by users. For their destination similarity experiment, destination-related tags from Flickr were thus crawled to build a description of a city. In the next step, a term frequency-inverse document frequency (tf-idf) feature matrix was created and similarities using the cosine distance measure were computed. At the end, a web-service was built in which a city can be entered as a query and a list consisting of the most similar cities is produced as a response. The most popular tags from Flickr were also shown below the similarity list. The results showed that, for example, Rome is quite similar to Venice, Florence, Milan, and other

Italian cities. The calculated similarities were evaluated by 113 students from the University of Milan, verifying that the model produces accurate recommendations.

In a more recent study conducted by Cao and Thomas (2021), destination similarity was investigated from the perspective of implicit user interest. One of the purposes of their research was to help circumnavigate COVID-19 travel restrictions so that a user could choose an alternative location based on destination similarity. Even tourism offices and online advertisement companies could profit from this similarity framework considering that their approach could improve sustainable tourism. In other words, by choosing destinations located closer to their homes, tourists could contribute to the reduction of over-tourism. To create a recommender system as such, the popular CF approach was adopted, which requires a similarity matrix calculation. In this case, a user's recent destination search history was used as input (Ravi and Vairavasundaram, 2016), and measures such as cluster consensus similarity and normed cluster consensus similarity were introduced in the framework. In addition to a user's search history, the user's origin was taken into consideration as well since user preferences can be influenced by origin or cultural differences, which also tends to impact travel behavior.

The approaches described above consider the concept of destination similarity in conjunction with recommender systems. Overall, conventional approaches, used to create user profiles based on previous user's search history or a restricted set of attributes, have been faced with much criticism over the years. Ricci (2002) summarizes existing approaches for tourism recommender systems as content-based approaches that use the same set of attributes to express both the user's needs as well as to describe a tourism "object" (e.g., a destination). To him, other approaches would provide the user with an opportunity to automatically classify himself/herself into one of the given tourists categories that already include all the basic user preferences. However, he ascribes the limitations of such systems to neither of them considering a concept of a "user-defined" trip in which a user could specify multiple preferences, not only including a desired destination but also a set of activity types or attractions one would like to visit at a certain location. While existing systems provide an opportunity for destination recommendations, they do not account for further details of trip planning and thus fail to provide any specific activities. Still, systems which partially introduced a higher level of granularity relied

on basic socioeconomic or psychological attributes describing the user's personality (Ricci, 2002).

Moreover, conventional CF approaches, despite their proven success in designed recommender systems, have still left much room for improvement in relation to travel object recommendations. This is because two users might have experienced the same trip in a completely different way, whereas CF is based on search or purchase history but does not take the user experience into consideration (Ricci, 2004). According to Ricci (2004), other limitations of existing approaches involve a general lack of user perspective incorporated into travel recommender systems as well as the "cold start" problem that recommender systems have to face in cases where no previous user's search history is available. Moreover, he emphasizes the importance of a recommender system's usability so that even an inexperienced user can easily interact with such a service. In another piece of work, Ricci et al. (2011) further point out the significance of using NLP techniques for the construction of semantic features, stating that this could represent a user profile in a more efficient way. In their opinion, the capability of generating new semantic features could potentially lead to unlocking unexpected and non-trivial relationships between items and user profiles.

To sum up, one can conclude that destination similarity has mostly been studied within a recommender systems concept. Different approaches have applied personality profiles, tourism typologies, or user searches on tourism platforms as well as social media data to establish user profiles. As for destination profiles, those have been created using attributes such as geographical location, price ranges, or statistical information, e.g., the number of overnights during a stay or a number of activities available for a certain category. In terms of Point Of Interest (POI) recommendation, only few approaches have been found involving textual features for destination representation. For example, Chang et al. (2018) used Instagram posts from different locations along with their texts to create a user-centered representation of a destination. Rahmani et al. (2019), on the other hand, applied location categories from online review platforms as the textual representation of a location in order to create a POI recommender system. Both approaches used Word2Vec to transform texts into numerical embeddings; nonetheless, neither of the described approaches leveraged recent NLP advancements or content-rich destination descriptions from extensive user reviews or tourism

destination offers for text analysis. Therefore, finding applications for the BERT model in terms of destination similarity or recommender systems for tourism was impossible.

What is more, the concept of tourism typologies has also been questioned by many researchers. Egger (2022), in response, discusses postmodern tourism as a consequence of remarkable social changes within the last decades and notes that tourism, as a social phenomenon, changes together with society. Compared to the last century, where the number of tourism offerings was limited and manageable, nowadays the diversity of leisure and holiday types raises abundant questions about existing tourism typologies, mainly because they are being rendered obsolete (Cecilia et al., 2011). Moreover, digitalization and social media platforms have provoked multiple social changes, also prompting new research methodologies (Ketter, 2021). Regardless, typologies that attempt to classify tourists into psychographic, demographic, or activity-based types tend to have a positive impact on creating travel recommendations (Coccossis and Constantoglou, 2008).

For the present work in particular, tourist typologies were not chosen to be included in the practical part due to the reasons described above and limitations of the data (scraped from Airbnb). Analyzing a tourist, or a user, in terms of typologies would have required collecting opinions or reviews of the activities on Airbnb in order to create TDPS. This work, alternatively, introduces a framework consisting of an aggregated approach for creating destination profiles. As will be explained in more detail later on, in the created system, tourists' preferences were transformed into aggregated representations as well, which simplifies the process of matching a destination and a tourist profile.

3. Embeddings and language models

Unlike humans, computers are only able to process information presented in a numerical format. Therefore, in order to be able to process any text, it must first be converted into numbers. There are many approaches that can be taken to accomplish this task to which numerous methods, split into count-based, non-context based, and context-based, exist. This thesis will solely cover non-context based and context-based models. Whereas some of the models are able to create word embeddings, other models are capable of creating representations for longer sequences, like sentences or entire documents. An embedding is a vector in a n-dimensional space, usually consisting of real-valued numbers that can capture the meaning of a word or a sentence (Egger, 2022). In this way, words which are semantically similar, e.g., “food” and “snacks”, or semantically related words, e.g., “sea” and “sailing”, are placed closer to each other in the n-dimensional space. In addition, more advanced models, such as ELMo or BERT, are able to capture the surrounding context of a word or a sentence so that the word’s meaning changes depending on the context it was mentioned in (hence, context-based). The ability of capturing context is especially important for tasks involving sentiment analysis, text summarization, and so on. Using certain distance metrics like Jaccard (Hancock, 2004), Levenshtein (Haldar and Mukhopadhyay, 2011), or cosine distance (Lahitani et al., 2016), one can determine how closely two different words, sentences, or documents are related to each other once their numerical representation has been obtained.

Non-context based embedding models, on the contrary, learn a “static” representation of a word in which the context of a word is not taken into consideration. However, a big advantage is that, compared to count-based models which ignore the word order in the original text, non-context based models are able to preserve the original word order (Mikilov and Chen, 2018). Some of the most popular non-context based embedding models include Word2vec, Doc2vec, FastText, and GloVe, which will be briefly explained in the next sections. Furthermore, context-based embedding models will be reviewed in section 3.4.

3.1 Word2vec and Doc2vec

Word2vec (Mikolov et al., 2013) had previously been considered the best-performing model, yielding state-of-the-art results in many NLP tasks, before transformers architecture appeared. Generally speaking, this predictive model can be trained using either of the two following algorithms: Continuous Bag Of Words (CBOW) or Skip-gram, which work in opposite ways. Whereas CBOW attempts to predict missing words in the middle of a sentence, Skip-gram predicts surrounding words based on a given word (Khattak et al., 2019). Concerning implementation, Word2vec learns distributed representations of words using a three-layer neural network, and one particular advantage is its robustness towards increases in the size of a dataset. Its training was optimized for increasing accuracy while simultaneously reducing computational complexity. Regardless of the underlying Word2vec architecture, all neural networks were trained using Stochastic Gradient Descent (SGD) and backpropagation algorithms (Zhong et al., 2018).

Both CBOW and Skip-gram are log-linear models that can better reduce computational complexity than other models that use non-linear hidden layers in their architectures. The architecture of CBOW is similar to a feed-forward neural network language model, and the projection layer is shared amongst all words so that the final output is produced through vector averaging. It is important to note that the order of the words does not lead to a different projection. To predict a current word, the previous and subsequent four words are used for prediction, which is done via a log-linear classifier (Mikolov et al., 2013). Compared to CBOW, the Skip-gram model attempts to predict a certain number of preceding and subsequent words for a given word within a specific context. Again, a log-linear classifier with continuous projection layer is used. As a visualization, the architectures of both the CBOW and Skip-gram models can be seen in Figure 3 below.

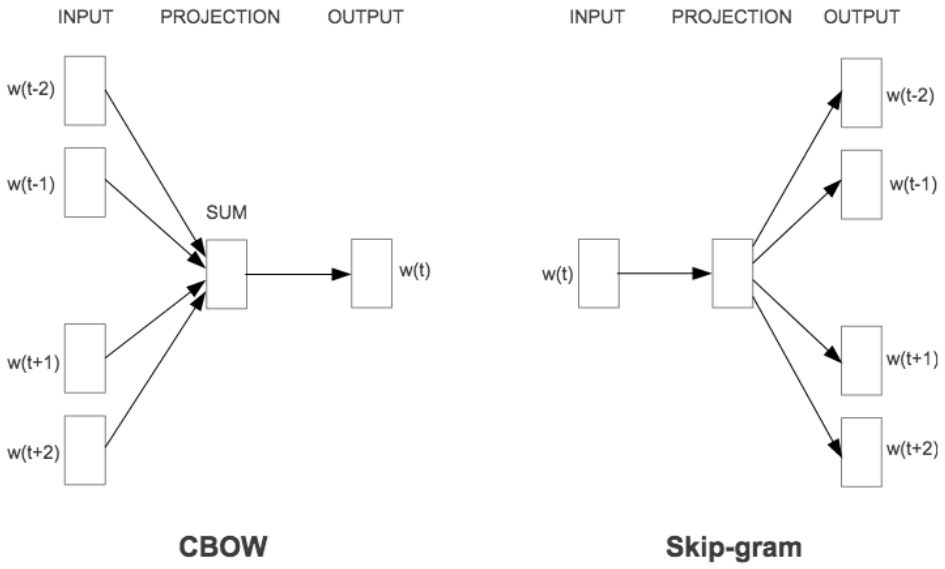


Figure 3: Architectures of CBOW and Skip-gram neural networks (Mikolov et al., 2013).

Word2vec is known to produce word vectors that might introduce difficulties when trying to create representations for longer sequences (e.g., sentences or documents). To combat this particular problem, the Doc2vec model was developed by Le and Mikolov (2014), which has a similar underlying approach as Word2vec and is trained in an unsupervised manner. Both Word2vec and Doc2vec can be trained on different corpora in order to create a domain-specific model. Although Li et al. (2019) claim that, due to the lack of tourism-specific datasets, no tourism-specific model has been developed, two models in particular can be mentioned here. On the one hand, the Tourism2vec was introduced by Hau et al. in 2019 for location and tourism type representation, while, on the other hand, a Doc2vec for tourism was built by Arefieva and Egger (2021) based on 3.6 million tourism-specific texts including reviews, sightseeing descriptions, etc.

3.2 GloVe

While Word2vec is a predictive approach, it fails to account for a word's frequency within a given context. This issue was solved thanks to the Global Vectors (GloVe) model, a log-bilinear regression model that combines both count-based and predictive approaches (Egger, 2022). Like Word2vec or Doc2vec, GloVe can also be trained from scratch; however, pre-trained GloVe

vectors can be downloaded using Python's gensim library. The model uses a word-word co-occurrence matrix, which contains, for each pair of words *ij,* the frequency of word *j* appearing in the context of word *i*. Using this matrix, co-occurrence conditional probabilities can be derived. On this note, it has been shown that some aspects of meaning are highly correlated with co-occurrence probabilities. Yet, despite the advantages of Word2vec and GloVe methods, they have a significant drawback: They cannot represent out-of-vocabulary words. The FastText model was successful in addressing this issue and will thus be briefly introduced in the next section.

3.3 FastText

The FastText model was developed in 2016 by Facebook AI Research (Bojanowski et al., 2017). Different from Word2vec or GloVe, which consider a word to be the smallest language unit, FastText uses character n-grams so that the word representation embodies a sum over all n-grams for that word. Using this approach, morphological information about words is incorporated into the final word representation. Since the model considers sub-words, it enables the creation of representations for previously unseen, i.e., out-of-vocabulary, words. Regarding implementation, FastText uses the abovementioned CBOW or Skip-gram algorithms, which are used in Word2vec as well.

3.4 BERT

A disadvantage of all the previously described methods is that they produce static vectors and do not account for the context of a word in which it was mentioned. This leads to decreased levels of accuracy and ignores the differentiation between multiple meanings of the same word depending on context.

Currently, one of the most advanced natural language representation models is BERT, which stands for Bidirectional Encoder Representation from Transformers (Devlin et al., 2018). This model was developed in 2018 by Google research and is used in Google Search and Google Translate, for example.

As was briefly mentioned in the introduction, BERT was trained on the entire English Wikipedia and Books Corpus, which consists of general vocabulary words and numerous historical events, to mention just a few. In the real world, however, knowing only general terms tends to be quite insufficient for communicating confidently with domain experts in areas like biology, finance, or healthcare. This statement can be projected onto embedding models as well, meaning that the knowledge of merely general vocabulary might negatively influence prediction quality when trying to solve domain-specific tasks from areas as those listed above (i.e., fields that require specific terminology).

Moreover, there are several words that have different meanings depending on the subject area in which they occur. For instance, words like “ticket”, “entrance”, “transfer”, “service”, or “experience” have different semantic meanings when perceiving them from a general or a tourism perspective. Therefore, in order to achieve better performance in domain-specific tasks, it is necessary for a language model to understand the peculiarities of domain-specific vocabulary or terminology. Take BERT variants as an example: They have been pre-trained for the financial sector (FinBERT) (Araci, 2019), the medical sector (Clinical BERT) (Alsentzer et al., 2019), for biomedical texts (BioBERT) (Lee et al., 2020), or for biomedical and computer science (SciBERT) (Beltagy and Cohan, 2019) (Arefieva and Egger, 2022). Also, because of model complexity, researchers tend to create architectural variants of the BERT model. Some of them are more lightweight but lack the original BERT accuracy, while others include alternative transformations that can be profitable depending on its intended use. Some architectural modifications of BERT will be briefly described after the original model architecture has been introduced.

3.4.1 Model architecture

The problem with previously used sequential models like Recurrent Neural Networks (RNNs) is that, because of their sequential architecture, they are only able to capture recent contexts, meaning that if a word is mentioned for the first time at the beginning of a sequence, they tend to “forget” this information when predicting a current word at the end of the sequence. In general, the weights of an RNN are updated in a procedure called backpropagation through time, which is used to calculate the gradient of the loss computed at a current point in time. The state of the network at the current point in time depends on its state at the previous point in time

as well as on its current input, i.e., the current element of the input sequence. The network state at the previous point in time, in turn, depends on its previous state so that this dependency is recursive until the initial point in time. Considering these connections, one of the steps when calculating the error gradient at a given point in time includes the recursive computation of multiple partial derivatives of the current state with respect to its previous state until the initial state. Those partial derivatives are then multiplied together via a chain rule. Increasing the length of the input sequence leads to an increase in the number of states and connections between them and thus also the partial derivatives. Multiplied together, they can result in a very small value that does not, however, contribute further to the error reduction; this is known as the Vanishing Gradient Problem (Ezen-Can, 2020). For example, in two consecutive sentences such as “A black car crossed the street during a red light. It then turned around”, “it” mentioned in the second sentence is not clear as to which object, “car” or “street”, it refers to.

This problem was partially addressed and eliminated to some extent through ELMo (Peters et al., 2018), which relies on the architecture of Long Short-Term Memory (LSTM). Embeddings from Language Model (ELMo) is a “shallow” bidirectional encoder based on two different LSTM blocks trained independently to capture context from left to right and from right to left. The final embedding is a concatenation of both outputs from two LSTM blocks. Regardless, the main issue with LSTM is its inability to focus on the most important words in a sentence. Whereas other recent deep learning language models, like the Open AI Generative Pre-trained Transformer (GPT) (Radford and Narasimhan, 2018), can capture context in one direction only, BERT, which contains a multi-layer transformer as its underlying architecture, can train in both directions simultaneously. A comparison of the ELMo, GPT, and BERT models is provided in Figure 4:

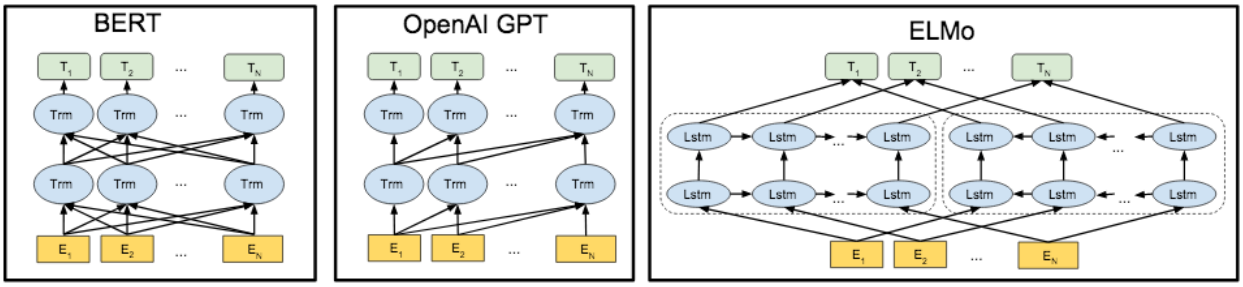


Figure 4: Pre-training architectures of BERT, OpenAI GPT, and ELMo (adapted from Devlin et al., 2019).

The aforementioned issues were solved in transformer models through usage of the attention mechanism. The transformer architecture was first proposed by Vaswani et al. (2017), who described both the transformer architecture (see Figure 5) and the attention mechanism.

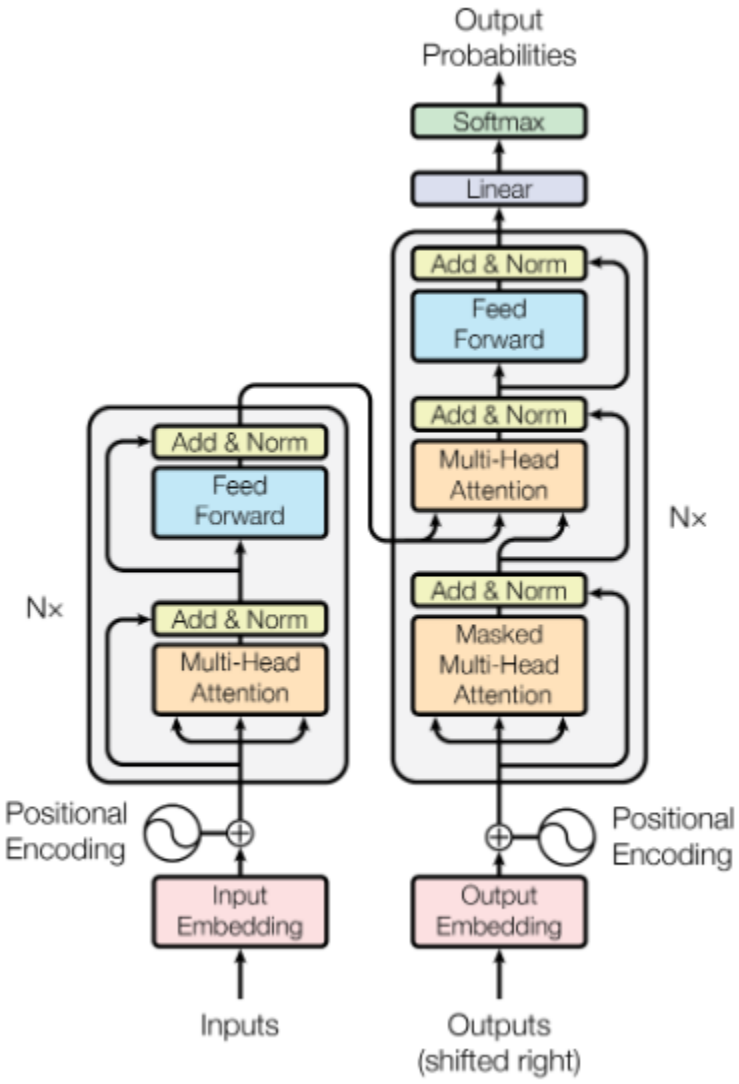


Figure 5: Transformer model architecture (Vaswani et al., 2017).

A transformer is a multi-layer neural network consisting of an encoder and a decoder. The encoder is used to create a numerical representation of the input, while the decoder is used to reconstruct the original input. Unlike an encoder, which uses self-attention to encode the input

text, a decoder uses previous words to decode the current word, i.e., processes input sequentially in one direction. Attention is a mechanism that allows for the focus to be placed on certain specific words more so than on others (e.g., think of people trying to distill information by keeping one's attention on the most relevant information).

Different from RNNs, transformers use multiple attention heads through which the information is passed linearly. This attention refers to scanning through different parts of sentences in order to try and discover some semantic or syntactic features. Whereas RNNs are unable to store an arbitrary amount of information, in transformers this issue can simply be resolved by adding more attention blocks. Furthermore, because a transformer can look at any word at the same time, word order becomes problematic; yet, this can be eliminated through storing input positions, ensuring that whenever the model processes multiple sequences in parallel, it also tracks the original order in which they were inserted into the model.

As already touched upon, BERT is a transformer-based model which was pre-trained on a large dataset consisting of unlabeled text, including the entire English Wikipedia (2,500M words) and Book Corpus (800M words). The large size of this dataset has enabled the model to gain a deeper understanding of language functioning. Moreover, the bidirectionality of BERT has allowed for the capturing of information in both directions, i.e., from left to right and vice-versa, so that the context of a given token has been captured from both sides during the training phase. To illustrate why bidirectionality plays an important role in language comprehension, an example of two sentences using the same word "key" have been provided:

- "I left my keys from the car at home";
- "The key to making healthy decisions is to respect your future self" (Jacobs, 2012).

From these two examples above, one can note that the identical word "key" is mentioned in different contexts, therefore containing a different meaning. Models like Word2vec or Fasttext are incapable of capturing such distinct word meanings depending on the context and always return the same vector for the same word. On the other hand, if one were to produce a BERT vector for the word "key" in both sentences, these would indeed come back as absolutely different. In doing so, the linguistic peculiarities of words and entire sentences can be collected, ultimately leading to better predictions and decision-making.

Compared to transformers, BERT uses only the encoder part of the architecture because the goal is merely to create a language model for feature extraction. The encoder in BERT-Base contains 12 layers, and the encoder in BERT-Large includes 24 layers, whereas the original transformer encoder block consists of six layers only. The encoders from BERT-Base and BERT-Large also have 12 and 16 attention heads. The encoder is followed by a feed-forward network with 768 and 1024 hidden units, and the total number of the parameters is 110 million and 340 million for BERT-Base and BERT-Large, respectively. Unlike transformers, which are limited on the number of NLP tasks they can solve (because decoders translate encoder output back into its original language), BERT, since it only uses an encoder, can add different architectures on top of it and solve multiple down-stream NLP tasks like text classification and summarization, QA, sentiment analysis, and more. In other words, different layers can be attached on top of the encoder. Since an encoder returns embeddings, many algebraic operations can be performed on them, e.g., two vectors can be compared for similarity using cosine distance or other metrics.

The architecture of BERT allows for more information about the context to be learned and to capture the meaning of a word. Still, questions about the model's interpretability remain: Though many researchers use these models to achieve state-of-the-art results, the exact functioning thereof is still unknown. As a result, BERT belongs to non-explainable AI.

3.4.2 Input format for BERT pre-training

The input data for BERT pre-training consists of three parts:

- Position embeddings: BERT learns and uses positional embeddings to express the position of words in a sentence. These are added to overcome the limitation of the underlying transformers model, which, unlike RNN, is its inability to capture "sequence" or "order" information;
- Segment embeddings: BERT can also take sentence pairs as input for tasks like QA. That is why it learns a unique embedding for the first and the second sentences: to help the model distinguish between them. In Figure 6 below, all the tokens marked as EA and EB belong, respectively, to sentence A and B;

- Token embeddings: These are the embeddings learned for a specific token from the WordPiece vocabulary.

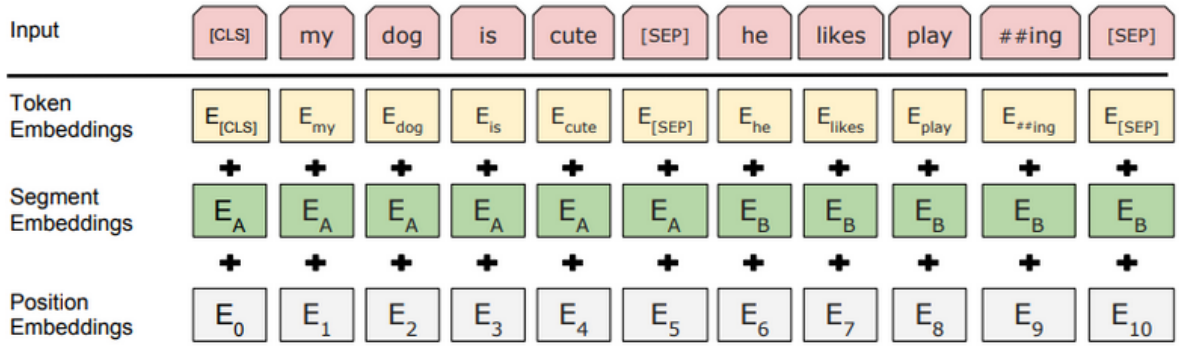


Figure 6: Structure of the input for BERT pre-training (Devlin et al., 2018).

For a given token, its input representation is constructed by summing up the corresponding token, segment, and position embeddings. In order to obtain tokens from the original text, the process of tokenization must be applied. Several techniques exist for tokenization, a text pre-processing method that splits an input text into words or sub-words. The resulting units are then converted to token ids and stored as a table for further searches by a transformer-based model. Despite there being multiple algorithms that perform tokenization using different rules, punctuation, whitespaces, and other features as well as probabilistic methods, the most widely used tokenization algorithms include Byte-Pair encoding (BPE), WordPiece, and SentencePiece, all of which will be covered later in this chapter. BERT, DistilBERT, and ELECTRA, for instance, operate with WordPiece, while SentencePiece is used in ALBERT, XLNet, Marian, and T5. BPE, contrarily, is adopted in models like GPT, RoBERTa, XLM, and byte-level BPE, which is a modification of BPE used in GPT-2.

Simple word tokenization could be implemented as splitting text into words by using whitespaces and punctuation. For example, using such an approach, the sentence “I didn’t know that BERT is the best model” would become [“I”, “didn”, “ ‘ ”, ”t”, “know”, “that”, “BERT”, “is”, “the”, “best”, “model”]. However, this has multiple disadvantages: Primarily, the number of representations to learn would be extensive if a vocabulary consists of unique words, and, furthermore, as could be observed from the example above, the word “didn’t” became [“didn”, “ ‘ ”, “t”], leading to a loss of meaning. In order to reduce the number of unique words, a

character-level tokenization can be performed, which would significantly decrease the size of the vocabulary. Nevertheless, such an approach could lead to performance decrease since character representations would then fail to carry any meaningful information, as compared to a word representation. Languages like Chinese, however, profit from this since one character in Chinese might correspond to an entire word or even a sentence. Regarding all other languages, a hybrid between word and character tokenization is used in order to reduce the size of the vocabulary while simultaneously preserving meaningful representations. A sub-word tokenization is utilized in almost all transformer-based models and relies on the following principle: Words that are frequently used in a corpus retain their original form, while words that are used less frequently are split into meaningful sub-words. For example, if the word “meaningful” were considered to be a rare word, it could be split into the sub-words “meaning” and “ful”, which would have a higher frequency as either standing alone or as sub-words in the dataset. The advantage of this approach is the smaller size of vocabulary as well as the preservation of meaningful context-independent representations. Moreover, words which had never been seen by the model beforehand can be split into known sub-words. The most popular tokenization techniques will now be described:

- **Byte-Pair Encoding**: Byte-Pair Encoding (BPE) was introduced by Sennrich et al. (2016). For the first step, the text is split into words using a simple space or rule-based tokenization in order to calculate the frequency of each word in the dataset. After the frequency of each word has been determined, a base vocabulary consisting of all the unique symbols in the words is created. The BPE algorithm then determines merge rules in an iterative process so that, during the next iteration, a pair of symbols that frequently appear together are merged as a new symbol of the vocabulary. The process repeats itself until the vocabulary reaches the predefined size. Thereafter, the resulting merge rules are applied to new input data so it can be split into sub-words present in the vocabulary. Any unrecognizable symbols or sub-words are replaced with an “<unk>” token, an abbreviation for “unknown”, by the tokenizer. GPT, as an example, has a vocabulary size of 40,478, consisting of 478 base symbols and 40,000 merged symbols.

- **Byte-level BPE**: In case of a base vocabulary, consisting of all unique symbols, being too large, e.g., to take all unicode symbols, byte encoding can be used instead so that the maximum size of the base vocabulary is always equal to 256. An advantage of such

encoding is that every new unknown symbol receives its byte-representation without the need for an “<unk>” token. For example, GPT-2 uses this approach and has a vocabulary size of 50,257: 256 bytes corresponding to base tokens, a special token determining the ending of the text, and 50,000 symbols resulting from the merge process.

- **WordPiece**: The WordPiece algorithm was found by Schuster et al. (2012) and is a modification of BPE. Likewise to BPE, WordPiece initializes the vocabulary with all unique symbols but learns merge rules in a different manner. Rather than calculating the frequencies of each symbol and merging together the most frequently occurring pairs, in WordPiece a pair with the maximum likelihood is chosen for merge. Among all symbol pairs “xy”, where “y” follows “x”, a pair is chosen where the probability of “xy” divided by the probabilities of “x” and “y” is maximal. In other words, the loss after merging two symbols into a new symbol should be minimized.

- **SentencePiece**: SentencePiece was found by Kudo et al. (2018). Whereas all the previously described algorithms have assumed that the whitespace tokenization is used as a pre-processing technique to split the text into words, this technique is inapplicable to languages like Chinese, for example. Therefore, SentencePiece adds the whitespace as a base symbol to the base vocabulary. After that, the vocabulary is constructed via the WordPiece or Unigram algorithms.

- **Unigram**: In contrast to WordPiece or BPE, which expand their vocabulary through merge rules, Unigram works in an opposite direction. This means that it includes, for instance, all the words obtained after pre-tokenization in the base vocabulary and removes symbols from it iteratively. At each step, the symbol that leads to the lowest increase of the total loss is removed. Again, pruning is performed until a desired vocabulary size has been reached. In addition to saving vocabulary during the training process, the probability of each token is saved as well. This is required since the algorithm does not rely on merge rules, and this leads to several possible tokenizations derived from the same input. However, usually Unigram uses tokenization that outputs the highest probability.

It is important to note that although SentencePiece is technically not used in BERT, the current work still implements one of the BERT models for tourism with SentencePiece tokenizer, opening up an opportunity to train a multi-language tourism-specific model in the future.

3.4.3 BERT pre-training

BERT was pre-trained on two tasks simultaneously, namely, Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). The novelty of such a pre-training task is that they do not require any manual labeling of the training data. Concerning MLM, this task predicts hidden words using the following strategy: 15% out of all of the words in the training dataset were masked; 80% thereof were replaced with the [MASK] token, 10% were replaced with a wrong token, and the remaining 10% were masked with a correct token. The NSP task, on the other hand, aims to predict whether, in a given pair of sentences, the second sentence is a continuation of the first one. In this task, BERT learns to predict not just a word but the entire sentence, given the current sentence. The advantage of BERT compared to RNNs, which could only examine previous words in order to predict the current one, is that BERT also knows the context after the masked words, rendering prediction easier for the model. Masking thus provides the advantage of not requiring any manual labeling. The 80/10/10 split was chosen for the MLM setting because if all words would be replaced with the [MASK] token, the model would not learn any information about other words, rather, only about those which it needs to predict. Replacing some of the tokens with random and correct words ensures that the model never knows whether the current word is correct or incorrect (non-masked words). Moreover, if replacing them with wrong words were ignored, then the model would always know that the current word is consistently correct and shift towards static vectors. On the other hand, if replacing words with correct words were disregarded, the model would know that the current word is always wrong.

BERT pre-training was done for one million steps, and the WordPiece tokenizer was used for input text pre-processing, resulting in a vocabulary size of 30,522 tokens. The input for each token is the sum of token, segment, and position embeddings. To obtain token embeddings, BERT outputs vectors for each input token from the sequence. For a pair of sequences, an embedding of a [SEP] token is added to every input token in order to indicate whether it belongs to sequence A or sequence B. To obtain a word embedding for a word that was split into

multiple tokens, one can run the model on a single word and use [CLS] token embeddings, which is an aggregated output.

The pre-training dataset contains 800 million words for BooksCorpus and 2,500 million words for English Wikipedia, which resulted in approximately 40 training epochs over a 3.3 billion word corpus. The following hyperparameters were used: batch size of 256 sequences with each sequence containing 512 tokens, Adam optimizer with the learning rate of 1e-4, L2 weight decay of 0.01, learning rate warmup over the first 10,000 steps, and a linear decay of learning rate. The dropout probability was kept at 0.1 for all layers, and the Gaussian Error Linear Unit (GELU) activation function (Hendrycs and Gimpel, 2016) was used instead of the standard Rectified Linear Unit (ReLU) function:

$$ReLU(x) = (x)^{+} = max(0, x),$$

$$GELU(x) = x * \Phi(x),$$

where $\Phi(x)$ is the cumulative distribution function for Gaussian distribution. Hendrycs and Gimpel (2016) attribute the advantages of GELU over RELU to the fact that GELU is non-monotonic and a non-linear function that introduces curvature in the positive domain, thus allowing for better approximation of complex functions. Moreover, it has been empirically proven that models with GELU activation function outperformed those with RELU activation functions (Hendrycs and Gimpel, 2016).

Overall, the train loss for BERT pre-training can be defined as the sum of the mean MLM likelihood and the mean NSP likelihood (Devlin et al., 2019). Pre-training of both BERT-Base and BERT-Large models took four days on four Cloud Tensor Processing Unit (TPU) clusters (16 TPU chips in total). TPU is a special processor optimized for fast matrix computations. The model was pre-trained with an input sequence length of 128 for the majority of the steps, and then only in the final steps with a length of 512 because of the complexity of the attention mechanism, which is quadratic to the sequence length (Devlin et al., 2019). For fine-tuning, all settings were left unchanged except for different batch sizes, learning rates, and number of training epochs.

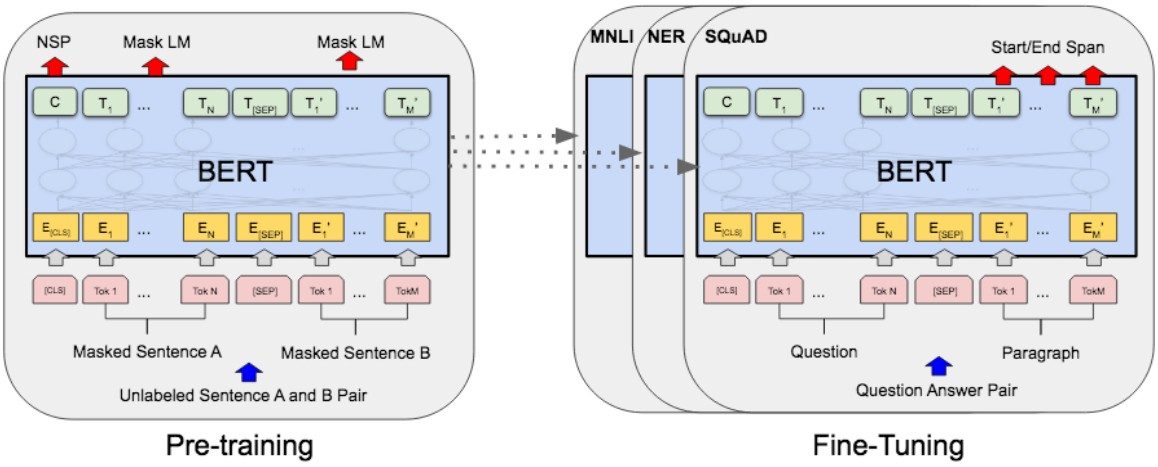


Figure 7: The pre-training and fine-tuning of BERT (Devlin et al., 2019).

Regarding a model's weights initialization, pre-training procedures can be classified into pre-training from scratch and further pre-training. Pre-training from scratch implies that the model weights are randomly initialized at the starting point, while a continuous pre-training procedure uses a different weights initialization strategy in which the model is initialized using some of the checkpoints that had already been pre-trained from scratch and possess the same model architecture.

For example, the previously discussed domain-specific BERT variants work with different pre-training techniques. Alsentzer et al. (2019) pre-trained two models starting from BERT-Base and BioBERT checkpoints, respectively, and also Araci (2019) did further pre-training of the BERT-Base checkpoint to create a language model for the financial domain. Additionally, Shin et al. (2020) pre-trained BioMegatron from scratch as well as from BERT-Base checkpoints using the vocabulary of the BERT-Base-uncased model in the first setting and the vocabulary learned from the PubMed dataset (Sayers et al., 2020), consisting of 30,000 and 50,000 words, in the second setting. Meanwhile, the authors of Clinical BioBERT (Alsentzer et al., 2019) released two different models for the clinical domain, pre-trained on different corpora. During the evaluation phase, it was shown that, depending on a specific vocabulary used for down-stream tasks, different models achieve better performance. While Clinical BioBERT and Clinical Discharge Summary BERT models find their application in both the clinical and medical

domains, the authors of BioBERT (Lee et al., 2019) applied the PubMed text corpus consisting of scientific paper abstracts from the biological domain and 4.5 billion words in total.

However, while solving real-world problems, scientific vocabulary may not cover the general language exerted in daily business and vice-versa. As for the tourism domain, the model created and used throughout this work was pre-trained from scratch, i.e., no initial checkpoints like BERT-Base were used to initialize the model. This can be explained through the fact that the training data was large enough to build a custom vocabulary and to cover features of both general and tourism-specific languages.

3.4.4 Fine-tuning and downstream tasks

Without significantly changing the architecture, it is possible to solve a variety of other tasks using the same model. The advantage of transfer learning is that many researchers can benefit from a model pre-trained on a large data corpus, a task that might be hard to obtain or collect individually. Since models can be built on top of the BERT model, knowledge that has already been gained can be transferred to a domain-specific model. Another advantage is that one can save and reduce the training time by selecting BERT layers that need to be fine-tuned. According to Stevens and Su (2021), though it still has yet to be proven, it seems to be the case that layers in the beginning of the model are more responsible for syntactic features, whereas top layers tend to capture semantics and context-specific information (Rogers et al., 2020). This implies that the last layers are most suitable for fine-tuning when performing a domain-specific task. Also, a much smaller amount of data is more sufficient for achieving greater accuracy. For example, while pre-training from scratch requires millions of sentences, a few hundreds or thousands might suffice to solve a given downstream task. At the end, one can specify all layers or a couple of last layers to be fine-tuned or to even “freeze” all layers except for the last one, which is directly responsible for making predictions.

A down-stream task in NLP is a machine learning task designed to solve a concrete problem, more often in a supervised manner. Some examples include, inter alia, Named Entity Recognition (NER), Relationship Extraction (RE), QA, text classification, and sentiment analysis. In contrast, examples of unsupervised down-stream tasks are clustering, topic modeling, synonyms search, and similarity comparison, amongst others. From an architectural point of

view, usually a single or a couple of layers need to be attached on top of BERT's architecture, which implement a specific prediction function calculated on BERT's output. For instance, softmax and feed-forward layers can be used for text classification. These architectures will be explained more in detail in the practical part of this thesis. All in all, downstream tasks use almost the same architecture, except for output layers, and are initialized with pre-trained weights.

3.4.5 Architectural variants of BERT

As described above, BERT's complexity is very high and thus requires abundant computational and financial resources to pre-train a model. Furthermore, the model is often expensive in terms of inference time and hosting resources when using BERT in real-world applications. For such reasons, the research community has developed numerous architectural variants of BERT, which can be classified into the two following categories:

- Architectures designed to reduce BERT's complexity and improve the training and inference time while maintaining as much accuracy from the original BERT as possible;
- Architectures designed to improve BERT's accuracy through modifications of either pre-training tasks or of the original architecture, which mostly lead to increases in the model's complexity.

The first group of BERT's architectural alternatives embody Distillation BERT (DistilBERT) (Sanh et al., 2020), A lite BERT (ALBERT), TinyBERT, and others. DistilBERT, for instance, is 60% smaller than the original BERT-Base model and uses a pre-training process built on knowledge distillation. Knowledge distillation is a process designed to compress a model and to build a smaller model, called a student, that learns from a bigger model, known as the teacher. The progress of knowledge transmission from a bigger model to a smaller one is measured through a distillation loss, which is calculated based on soft target probabilities predicted by the teacher model. Another "compressed" version of BERT is ALBERT (Lan et al., 2020), consisting of only 12 million parameters, i.e., nine times smaller than the number of parameters in BERT-Base (108 million parameters). Although ALBERT is pre-trained on the same data as BERT, it uses different techniques for the training process like embedding matrix factorization,

cross-layer parameter sharing, and inter-sentence coherence prediction as the loss function. Finally, TinyBERT (Jiao et al., 2020) is an extension of DistilBERT, with the difference being that while in DistilBERT the knowledge is shared between the output layers of the student and teacher networks, in TinyBERT knowledge distillation is performed for all intermediate layers of the model.

As for the second group, the most well-known models yielding state-of-the-art results that tend to surpass BERT in some down-stream tasks involve Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA) (Clark et al., 2020) developed together with Google Research, and a Robustly Optimized BERT Pretraining Approach (RoBERTa) (Liu et al., 2019) developed by Facebook AI. The main difference between BERT and ELECTRA is the pre-training task to which Replaced Token Detection is applied instead of MLM. Whereas in MLM the task is to predict a randomly masked token replaced with [MASK], in the Replaced Token Detection tokens are replaced with fake tokens using another model so that the pre-trained model must predict whether or not the current token is in its original form. Also, while only 15% of random tokens were masked in BERT, all tokens were replaced with fake tokens in ELECTRA, which allowed this model to surpass BERT's performance in GLUE tasks. The pre-training procedure for ELECTRA is similar to Generative Adversarial Networks (GAN) training, where generator and discriminator networks are competing against each other during the training procedure. Figure 8 below illustrates the training procedure of ELECTRA using a concrete example.

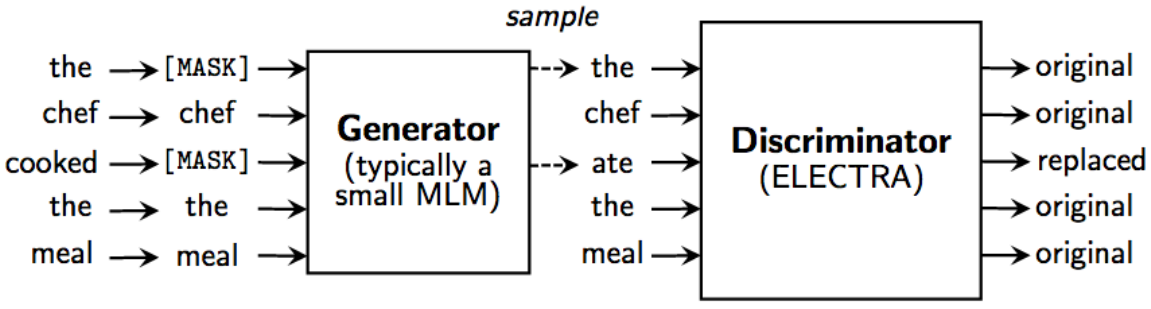


Figure 8: Pre-training procedure of ELECTRA (Clark et al., 2020).

Moving on to RoBERTa, this model has a larger number of parameters than BERT-Base but possesses almost the same architecture; the major difference between the two models lies in their pre-training procedure. Unlike BERT, RoBERTa uses dynamic masking instead of the static

one, which essentially means that different parts of the same sentence are masked depending on a training epoch. Moreover, the NSP task is discarded for pre-training because it was shown in various experiments that the removal of NSP loss from the training procedure leads to better results. Overall, RoBERTa has outperformed BERT-Base on SQuAD, MNLI-m, and a couple of other tasks. Nonetheless, it is important to bear in mind that RoBERTa was trained on a larger dataset, which, in addition to BERT's training data, uses Common Crawl News (CC-News) and Open WebText datasets. In total, this exceeds the size of BERT training data by ten times. The final difference to note is that along with a bigger dataset, a bigger training batch size must be put into place.

4. NLP and BERT applications in tourism

During an extensive literature review, several examples of BERT-based applications in the tourism domain were explored. A brief summary of examples reviewed in this chapter can be found in Table 1 below:

Name	*Title*	*Objective*	*Downstream Tasks*
Arreola, Garcia, Ramos-Zavaleta & Rodríguez (2021)	An Embeddings Based Recommendation System for Mexican Tourism. Submission to the REST-MEX Shared Task at IberLEF 2021	To generate a recommendation system for tourist sites in Mexico (REST-MEX) using Spanish data	Natural language inference (XNLI), paraphrasing (PAWS-X), NER, POS tagging, document classification (BERT+XgBoost)
Chantrapornchai & Tunsaku (2021)	Information Extraction on Tourism Domain using SpaCy and BERT	To compare two approaches (BERT & SpaCy) for constructing tourism ontology	NER, text classification
Leyi, Han, Fan & Yuehua (2019)	A Method of Chinese Tourism Named Entity Recognition Based on BBLC Model	To improve the efficiency of named entity identification in Chinese tourism	NER
Kim & Zhuang (2021)	A BERT-Based Multi-Criteria Recommender System for Hotel Promotion Management	To develop a multi-criteria recommender system for hotels based on predicted multi-aspect ratings (rating predicted by BERT)	Attribute prediction using multi-criteria CF
Li, Qiu & Jiang (2019)	Research on Tourism Destination Attraction Based on Deep Learning	To propose a model based on deep learning that can explore the attraction and uniqueness of tourism	Masked Language Modeling, Next Sentence Prediction, classification with softmax classifier
Yanuar & Shiramatsu (2020)	Aspect Extraction for Tourist Spot Review in Indonesian Language using BERT2	To perform aspect-based sentiment analysis of	BERT further pre-training and fine-tuning,

		Indonesian TripAdvisor reviews	aspect-based sentiment classification
Hu & Nuo (2019)	A Deep Learning Approach for Chinese Tourism Field Attribute Extraction	To construct a knowledge graph for tourism in a province/autonomous region of China	NER, Attribute extraction (BERT-ResCNNs-BiLSTM-CRF)
Siregar & Chahyati (2020)	Visual Question Answering for Monas Tourism Object using Deep Learning	To compile the first VQA dataset in Bahasa, Indonesia	Visual Question Answering, multi-label classification (ResNet + BERT)
Chantrapornchai & Tunsaku (2020)	Information Extraction based on Named Entity for Tourism Corpus	To present a methodology for extracting tourism data from unstructured information	NER, Relationship Extraction
Phan & Do (2020)	BERT+vnKG: Using Deep Learning and Knowledge Graph to Improve Vietnamese Question Answering System	To develop a knowledge graph in Vietnam tourism	Question answering (BERT+vnKG)
Xiao, Wang, Yu, Zhang & Wu (2020)	A Practice of Tourism Knowledge Graph Construction based on Heterogeneous Information	To construct a systematic framework for building a Tourism Knowledge Graph for Hainan	NER (BERT+BiLSTM-CRF), Relationship extraction
Zhang, Cao, Hao, Yang, Ahmad & Li (2019)	The Chinese Knowledge Graph on Domain-Tourism	To construct a Chinese knowledge graph on domain-tourism	Entity alignment
Phan & Do (2022)	Developing a BERT based triple classification model using knowledge graph embedding for question answering system	To build a large Vietnam Tourism KG	Text classification, triple classification

Table 1: Summary of BERT and NLP applications in tourism.

For example, Arreola et al. (2021) used BERT for the Spanish language in order to create a recommendation system based on reviews of the most popular and most visited sites in Mexico. The data was crawled from the Internet, and the task solved in their research was the prediction of a review rating. Additional characteristics like tourist groups, tourism activities, etc., were also incorporated into BERT's input features. The final model was built using the BERT and XgBoost algorithm as a classifier. Next, Chantrapornchai and Tunsaku (2021) conducted an experiment of which the goal was to benchmark BERT against SpaCy framework for NER. The authors failed to disclose which algorithm was used behind SpaCy for entity recognition, but SpaCy typically adopts the Conditional Random Fields (CRF) algorithm. Reviews on restaurants, hotels, shopping, and tourism activities in Thailand were crawled from TripAdvisor, and those were processed in a two-step fashion: First, the reviews were classified into categories in order to improve text summarization, and then NER was conducted in order to extract entities such as name, location, and facility type. The experiment revealed that the BERT model outperformed SpaCy in both tasks.

Leyi et al. (2019) also developed a BERT-based NER model, but for the Chinese language. Due to the peculiarities of Chinese, for instance, the fact that words are often not split through whitespaces and cause tagging difficulties, applying conventional NER algorithms to Chinese texts has been a problematic endeavor. Moreover, complex grammatical dependencies tend to lead to difficulties in identifying the start and end position of an entity. It therefore comes as no surprise that an existing model for tourism NER in Chinese language ceased to exist. To alter this, researchers crawled reviews from the Ctrip website and tagged all entities manually using BIO-tagging rules (BIO stands for Beginning-Inside-Outside). Thereafter, they applied the BERT model to produce word embeddings and created a BiLSTM-CRF model for entity recognition in order to recognize people name, location name, organization name, time, and things. Their findings proved that the BERT-BiLSTM-CRF model outperformed the popular CRF model for entity recognition.

Kim and Zhuang (2021), on the other hand, developed a multi-criteria hotel recommender system based on crawled reviews from TripAdvisor. This platform provides the possibility to rate a hotel in six different categories such as location, service, etc. Since users sometimes write reviews without leaving ratings or rate only a couple of the six categories, researchers were able to utilize the BERT model to predict all six ratings based on the textual reviews posted for a

certain hotel. An approach as such thus establishes a solution for the cold start problem, where it is difficult to provide a personalized hotel recommendation without knowing a user's preferences. The final goal was to either predict an overall review rating or all six aspect-based ratings and use them for a personalized recommendation in the next step. The recommendation system was then implemented using a CF approach. According to the authors, the novelty of their approach lies in the multi-step method, which uses CF on ratings predicted by the BERT model, since many recommendation systems often only rely on the first step in which the BERT model is supposed to predict the rating. The advantage of the developed approach is that not only can hotels be recommended to users, but also customers can be recommended to hotels, ultimately improving the experience for both sides.

Li et al. (2019) conducted research using a BERT-like model to explore tourism destination attractions in China and to obtain a unique label for each tourist destination. Using travel notes from the Chinese tourism platform Mafengwo, travel notes for 512 cities with 500 notes per city were crawled. Moreover, both the Chinese and English names for every destination in all 512 cities were obtained. The goal was to explore, i.e., to learn the attraction of the tourist destination for each travel note. A tourism attraction is considered to be a word or a phrase, and attractions are predicted based on the words with the highest probability noticed by the model. To create a model, the BERT pre-training approach was used, but for a different model that uses only some elements of BERT (transformers architecture, attention mechanism, and the same pre-training data input format). The model thus consists of an input layer, transformer block, and a softmax output layer for predictions, while the encoder from transformers architecture consists of multiple layers where each layer contains two sub-layers, a self-attention layer, and a feed-forward neural network.

When it comes to Yanuar and Shiramatsu’s (2020) research, they used BERT to extract aspects for tourist spot review in Indonesia, and TripAdvisor reviews were crawled for aspect-based sentiment analysis. The goal of the model was to simplify the extraction of the relevant information in such a way that users can distinguish reviewers’ opinions towards the different aspects of a review that they might be interested in, for example, food or hotel service. All reviews were crawled in the Indonesian language and went through a pre-processing technique that accounted for any peculiarities in the language. The multilingual BERT model was further pre-trained with 4,220 review sentences in Indonesian, and an additional 501 sentences were

used for model fine-tuning. Although the multilingual BERT model already includes the Indonesian language, there is no model for Indonesian only; therefore, due to the limitations of the dataset crawled in the study, researchers decided to use multilingual BERT, and it was shown that after further pre-training and fine-tuning the model outperformed the multilingual BERT.

On the other side of the spectrum, Hu and Nuo (2019) performed attribute extraction for the retrieval of missing information about an entity in order to build a knowledge graph (KG) for Chinese tourism. Attraction descriptions were crawled from different Chinese websites and the problem was defined as a sequence labeling task. The resulting model was a BERT-ResCNNs-BiLSTM-CRF hybrid. The BERT model was first fine-tuned to obtain character embeddings for Chinese characters before using the ResCNN model to capture local feature representations from BERT for each character embedding. The ResCNN was used instead of the conventional Convolutional Neural Network (CNN) because ResCNN uses residual learning and tackles the gradient vanishing problem. The outputs of BERT and ResCNN were concatenated into a single vector and fed into the Bi-directional LSTM (BiLSTM) model as input. The goal of BiLSTM was to capture the context of a given sentence bidirectionally, i.e., both preceding and consecutive sentences or longer sequences. In the final step, the CRF classifier was trained for entity recognition and attribute extraction, and two datasets were used for evaluation: the publicly available MSRA dataset (Yao et al., 2012) and an additional dataset built and annotated manually by the researchers for the NER task. To crawl the data, the authors worked with Chinese websites like Baidu. The BIO2-approach was used to tag entities' attributes, which were represented in the five following categories: area, construction time, internal attraction, location, and nickname. The described approach outperformed other state-of-the-art hybrids on the MSRA dataset.

Siregar and Chahyati (2020) developed a model for Visual Question Answering (VQA) in the Indonesian language. The goal of the study was to first produce a dataset for VQA and then to construct a model and apply it to this dataset. In a VQA task, images are shown to the model and questions in the natural language are asked about a given picture. To build such a model, image embeddings were extracted using the ResNet model and BERT was used for the actual QA task. Questions were asked for seven categories: binary (with a possible answer "yes" or "no"), object (e.g., name), color, attributes (like winter temperature at a certain location), spatial

(predict a location), time, and activity (e.g., list activities that can be undertaken at a certain destination or place). The questions were constructed as a single sentence in the Indonesian language and did not contain any information requiring prior knowledge about the image. Each image was annotated with seven questions, one from each category. The dataset was annotated manually and encompasses, in total, 600 images and 2,447 question sentences with 276 unique answers. The final task was defined as a multi-label classification problem. In order to ensure that the output of the current pipeline component matches with the input of the subsequent one, the following approach was applied: ResNet produced a vector with a dimensionality of 1x1024 and BERT encoded the question as a 768-dimensional vector. The vectors from BERT were reduced using Global Max Pooling to compress it to a dimensionality of 1x1536 after which a layer normalization was applied. Finally, it was transformed into a 1x1024 vector using a dense layer with the tanh activation, and the image and sentence vectors were aggregated as element-wise products. The resulting vector was then passed through a feed-forward network and became input for the softmax classifier. In the end, it could be demonstrated that BERT outperformed other embedding models like Fasttext, Word2Vec, and others.

Next up, Chantrapornchai and Tunsaku (2020) implemented an information extraction method for tourism in Thailand. The machine learning problem was defined as a NER task for recognition of entity types such as name, location, or facility. The training dataset was constructed using descriptions of various hotels in Thailand scraped from TripAdvisor, Traveloka, and Hotels.com. After data extraction, the TextRank algorithm for extractive summarization was applied so as to reduce the size of the original text and keep only the most relevant sentences. Once summarization was complete, the data was filtered using HTML tags, which formed the basis of the dataset's annotations. Fields extracted from the HTML tags included location, facility, nearby, hotel name, and others. The annotated dataset was used for training both NER and Relationship Extraction (RE) models. Regarding the former model, SpaCy was benchmarked against BERT, achieving state-of-the-art results, and in the next step, was used to extract relationships between entities detected by the first model. The potential practical application of both models, according to the authors, could be the implementation of an ontology or a KG, which could introduce new opportunities for web-search improvement.

With respect to KGs, Phan and Do (2020) used BERT to build a KG in order to improve the existing QA system for Vietnamese tourism. The model answers questions about tourism places in Vietnam. As an underlying model, BERT for multilingual QA was used as it already contains the Vietnamese language. In order to enhance the accuracy of the existing QA model, a KG was built in addition to multilingual BERT. The system implements the following use-case: After a question is received, a subject extraction component is used to extract entities from the question. The extracted entities are then sent as input to the KG, where a search for extraction of the corresponding sub-graph containing those entities is performed. If such a sub-graph exists, it is transformed into a natural language formulation; otherwise, it is added to the KG. To add new sentences to the KG, the VnCoreNLP library was used to create triples consisting of two entities and a predicate describing a relationship between those two. For example, in the sentence “Da Nang has Golden Bridge”, names “Da Nang” and “Golden Bridge” are entities and “has” is a predicate that establishes the relationship. In the final step, the natural language formulation and the question are passed to the BERT model for the final answer prediction. Using Vietnamese travel websites, 300 QA-pairs and 4,600 relationships between entities were built, and the dataset was used for both KG and BERT QA models. This research provided proof that the multilingual BERT for QA in combination with a KG outperformed both LSTM and the existing standalone multilingual BERT for QA.

Similarly, Xiao et al. (2020) constructed a framework for building a KG for tourism in Hainan, China with the goal of improving tourism search engines and recommender and QA systems. Multiple datasets were formed using data from Chinese tourism web-platforms in order to ultimately build different NER models as well as the Chinese BERT model. Entity types like Duty Free Shop, Golf Course, Specialty, Snacks, and others were used to construct the training dataset. However, the experiment established that BERT was surpassed by a BiLSTM-CRF model. In the next step, the relationship extraction task was solved using a BiLSTM-CNN pipeline to construct a KG, which is populated with triples consisting of extracted entities and relationships between them. The resulting system contained 34,079 entities for 13 entity types, 46 relation types, and 441,371 triples in total. The novelty of this framework lies in its reusability when building new KGs from data found on other websites.

Another KG for Chinese tourism was built by Chantrapornchai and Tunsaku (2020). The main purpose of their research was to enrich existing KGs involving Chinese tourism by incorporating

new triples built from unstructured data crawled from the Internet. First, zhishi.me and CN-DBpedia (Chinese DBpedia – a general-purpose knowledge base) websites were used to download existing knowledge sub-graphs containing travel-related knowledge. Subsequently, unstructured text data about travel, attractions, and scenic spots was crawled from the wikivoyage.org and ctrip.com websites to produce word vectors using the BERT model for the Chinese language. The obtained word vectors were applied to the entity alignment task, which is essentially the process of determining whether two different entities with the same meaning point to the same object in the real world. Entity alignment was performed using cosine distance between a pair of BERT word vectors. Those entities that had levels of similarity above a certain threshold, e.g., 85%, were considered to belong to the same object and added as a new expression for the same entity in the knowledge base (since one entity is allowed to have multiple expressions). The protege tool was used to build a KG stored as subject-predicate-object triples in RDF format, and the resulting ontology was then stored in the Neo4j graph database.

Phan and Do (2022) also built a KG, but for Vietnamese tourism. The purpose of their survey was to improve existing QA systems about tourism since existing QA models do not always output an expected answer or fail to capture all the information about the subject of a question in cases where a formulated question might be missing some information required for an accurate answer. A BERT-based triple classification model was developed in order to obtain KG embeddings and use those as input for the QA model. Content-based and link-based information was combined for KG representation learning, and triples were classified into three categories: base class, derived class, or non-existent class. Then, BERT was used to build two classifiers: For the first one, triples were converted into natural text and used for context classification, while the second classifier was built for triple classification for link information extraction.

This overview of existing approaches in terms of BERT and NLP in tourism has testified to the fact that most of the problems solved included more KGs and QA systems and less recommender systems. Moreover, the majority of approaches used BERT in conjunction with an ontology, being one of the components of a recommender system, and rarely utilized NLP for creating profiles of recommended tourism items (Ricci et al., 2011). It is also important to note that all the developed applications were applied to tourism within a single country, for instance,

Vietnam, China, Indonesia, or Mexico. Since these countries have complex languages, the majority of contributions were related to improving existing BERT models (especially for the Chinese or Indonesian languages) using manually constructed datasets and architectural modifications. However, the main limitation was that the described tasks were limited to tourism within a single country, without taking international tourism into consideration. More specifically, covering tourism in Europe has not been attempted at all, which is one of the main contributions of this thesis. Last but not least, the recommender systems described in the literature review above also failed to include international destination recommendation, which further attributes to the fact that all the systems were developed for internal tourism. As such, this thesis aims to bridge this gap by proposing a framework that can be used for European destination recommendation on an international level.

5. Methodology

To address the first research question, "*Does a natural language model for the tourism domain (TourBERT) capture tourism-specific contexts better than a general language model (BERT-Base)?"*, the TourBERT model was pre-trained from scratch and benchmarked against the BERT-Base-uncased model. To perform model evaluation, publicly available data from the Internet was resorted to. The overall goal of the model evaluation was to illustrate that TourBERT can outperform BERT-Base-uncased in a set of NLP tasks based on tourism-specific data. The model training procedure will first be described in detail in chapter 6, followed by a description of the evaluation procedure in chapter 7.

To address the second research question of *"Which European countries provide similar tourism offers by locals on Airbnb?"* and to determine the European cities that have the most similar tourism offerings, the process depicted in Figure 9 below was designed.

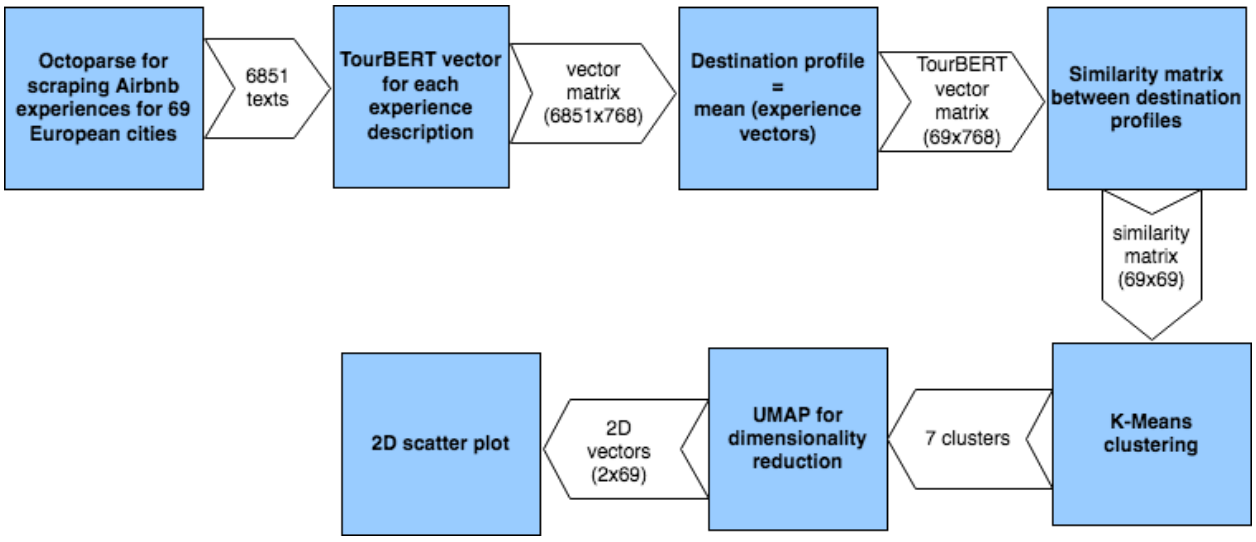


Figure 9: Proposed destination profile similarity framework.

First, the data from the Airbnb website was collected using Octoparse[1], a tool that allows researchers to set up a scraping workflow using a simple drag-and-drop user interface. Unlike other scraping libraries, Octoparse does not require any manual coding, which is oftentimes

[1] https://www.octoparse.com/

unsuccessful due to a strong anti-bot protection used on platforms like Airbnb. An embedded browser enables a user to navigate a webpage directly from Octoparse so that it saves the user's actions and builds the workflow automatically, and page contents can be extracted using corresponding HTML tags. An example of a screenshot from Octoparse can be seen in Figure 10 below:

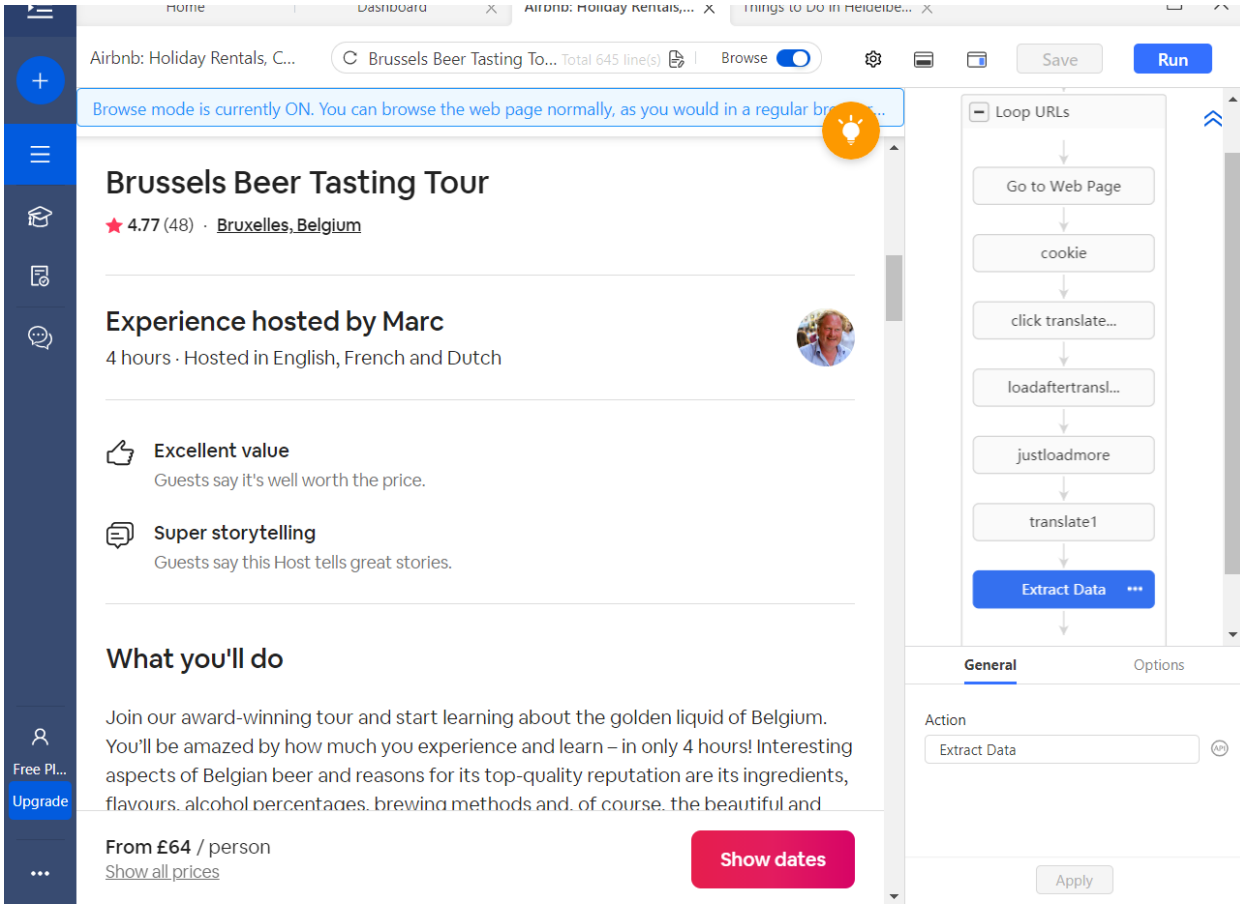


Figure 10: Octoparse UI with workflow for crawling Airbnb experiences.

For this study, destination profiles were created for 69 cities in total, which were taken from the ECM Benchmarking report according to the number of overnight stays reported in 2019. A smaller sample of a couple dozen, instead of hundreds, was chosen in order to provide a holistic overview and similarity comparison of the most relevant European cities in terms of tourism intensity. Originally, approximately 90 cities were selected for the analysis; however, the scraping of the experiences resulted in successful extraction for only 69 of them.

Using Octoparse, 6,851 experience descriptions for a total of 69 destinations were scraped. The subsequent step in the proposed process was to pass each experience description through the new TourBERT model, resulting in a vector matrix with the dimensionality of 6851x768. Afterwards, vectors representing experiences for the same location were averaged together to create a destination profile. In the next step, cosine similarity was utilized to calculate pairwise similarities between destination profiles. To achieve a higher level of granularity, the k-Means clustering algorithm was used for segmentation of the destination offerings. Destination profiles were then split into groups, where it was assumed that destinations belonging to the same group have similar tourism offers.

For results visualization, the Uniform Manifold Approximation and Projection for Dimensionality Reduction (UMAP) algorithm was used in order to transform each 768-dimensional vector into a two-dimensional point. The reason underlying the choice for UMAP is based on its several advantages when compared to other algorithms like T-distributed Stochastic Neighbor Embedding (t-SNE). For instance, the fact that UMAP has the ability to preserve global data structure in addition to its local structure. Moreover, it preserves connections between points as well as their similarity (McInnes, 2020) and allows for the visualization of the relative proximity of points. For this thesis, such visualizations are crucial as it is not only important to be able to obtain a set of well-separated clusters, but to also understand which cities have dissimilar offerings (which can be achieved by placing dissimilar points far away from each other). The final results visualization was completed using a scatter plot. In this way, the second research question could be addressed through the comparison of Airbnb offerings for the most popular European cities. The survey result was displayed on a map with geographical distances between objects being replaced with cosine similarity, and latitude and longitude being replaced with two-dimensional coordinates produced by UMAP.

When focusing on the third research question, *“How can TourBERT help to improve the quality of personalized recommendations?”*, the results and algorithms applied for the second research question were partially used and incorporated into a web-service that allows users to get an impression of destination similarity. Moreover, a user has the opportunity to enter a textual description of his/her specific preferences, which are then transformed into a two-dimensional user point displayed on the destination similarity map. Using the described approach, the user

can perceive such a result as a personalized recommendation. The implementation details are summarized in the architectural diagram in Figure 11:

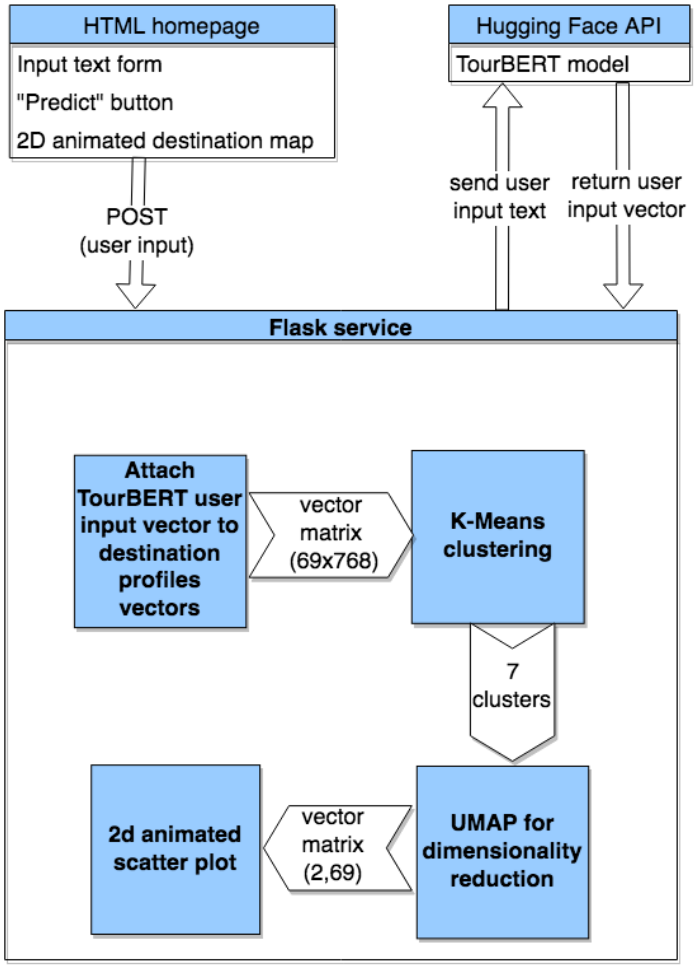


Figure 11: Proposed web-service architecture for the personalized recommendations prototype based on TourBERT embeddings and the destination profile similarity framework.

The web-service prototype consists of the three following components: an HTML user interface, an API for the TourBERT model, which is hosted by Hugging Face Model Hub, and the backend, i.e., the logic component implemented using Python's Flask framework. The HTML page includes an input text field and a button, "predict", bound to a REST "POST" method, which sends a user input request to the Flask service where it is processed through the TourBERT model. It will then be directly called from Flask using Hugging Face Hub API. The new user vector is appended to the list of previously computed destination profile vectors, and the new

resulting matrix undergoes a similar process as described in the proposal for the second research question. It first gets re-clustered using the k-Means algorithm to build seven clusters before the UMAP dimensionality reduction method is applied to plot the resulting vectors on a two-dimensional scatter plot. The resulting scatter plot is then recorded as a video file and is redisplayed on the HTML page to show the user his/her new potential location. The animation of the scatter plot should allow the user to track where his/her vector will move next on the map as soon as any changes in preferences has been made. This approach will be described in more detail in chapter 9.

As the basis for the destination profile, descriptions of cities and countries from the Airbnb platform were crawled, i.e., what are known as “Experiences”. These experiences provide numerous short descriptions about main tourism offerings, like shopping or restaurant tours, etc. It was noted that the number of experiences typically ranges from a few to a few hundred, which is considered to be sufficient for creating a valuable destination representation.

6. TourBERT - Model training

This chapter delves into the actual TourBERT pre-training procedure in which sub-chapter 6.1 provides an overview of the hardware used for the model's pre-training and evaluation, and sub-chapter 6.2 describes the preparation of the training data as well as some settings of the pre-training procedure.

6.1 Hardware setup

For the model training, this research made use of Google Colab Pro, which provides Graphical Processing Unit (GPU) and TPU resources for $10 per month. TPU is a special processor optimized for computationally expensive matrix operations. For all the evaluation tasks, a single compute instance with 25GB RAM and a single GPU provided by Google Colab Pro was used, while interactive visualizations were conducted on a MacBook Air with OS X El Capitan and 8GB RAM. For the development of the web-service, a Lenovo ideapad 510 with a Windows 10 operating system, 8GB RAM, and 4GB NVIDIA GPU was used.

6.2 Train settings

For pre-training, the original BERT implementation in Python TensorFlow was applied, and in all the experiments, the architecture of the BERT-Base-uncased model with 12 hidden layers and 12 attention heads with 110 million parameters was used. Due to the fact that the model was pre-trained from scratch, no initial checkpoints were provided for the weights' initialization.

The pre-training was done from scratch by following the official guidelines provided by BERT authors (Devlin et al., 2019). In the current setting, a combined dataset consisting of three million reviews from TripAdvisor and 46,000 sight-seeing descriptions from Expedia was created to which both datasets were collected for worldwide destinations. To create inputs for BERT, the dataset was split into sentences using Python's Natural Language Processing ToolKit (NLTK) library, which contains the SentenceTokenizer class for punctuation tokenization of a given text sequence into sentences. Splitting the corpus resulted in 22,601,333 sentences in total. In the

next step, SentencePiece and WordPiece tokenizers were trained using the resulting dataset to obtain a custom tourism-specific vocabulary.

To avoid out-of-memory issues, 5,120,000 sentences, which equates to about 22.65% of the dataset, were sampled for SentencePiece, and the final vocabulary included 32,000 tokens, inclusive of BERT's special tokens. WordPiece was trained in a standard fashion with a vocabulary size of 30,522. For pre-training, the scripts were adapted and modified from the official BERT Github repository[2], and BERT pre-training was done on a single TPU instance for one million steps, amounting to approximately 58 hours in total. It is significant to note that the model was not initialized with the BERT-Base checkpoint as has been previously done by other researchers who have created their own domain-specific BERT versions. The authors of BERT recommend using BERT-Base as an initial checkpoint if one does not have enough training data, which, in this study, was not the case.

This experiment follows the idea of trying several pre-training settings, including different options for vocabulary corpora. Some of the pre-training settings have been listed in Table 2 below:

Model architecture	Tokenizer	Initial Checkpoint	Vocabulary
BERT-Base-uncased	WordPiece	None	domain specific
BERT-Base-uncased	SentencePiece	None	domain specific

Table 2: TourBERT pre-training settings.

2 https://github.com/google-research/bert

7. TourBERT - Model evaluation

This chapter describes approaches used to benchmark TourBERT against the BERT-Base-uncased model. To define the best-performing model, both quantitative and qualitative evaluation methods were used, and the following supervised and unsupervised down-stream NLP-tasks were performed: multi-label classification, binary classification, topic modeling, dimensionality reduction, and nearest neighbor search. All tasks were performed using datasets that contain publicly available data from the Internet.

For qualitative evaluation, topic modeling, dimensionality reduction, and interactive visualization techniques were used in order to create insightful deliverables for human evaluation. A user study was also performed, the results of which were evaluated based on their statistical significance using hypothesis tests. For quantitative evaluation, supervised tasks like classification and different model metrics (e.g., accuracy, AUC-score, and others) are reported.

7.1 Unsupervised evaluation

This sub-chapter describes the qualitative evaluation procedure of the TourBERT model. Multiple methods from unsupervised machine learning and statistical modeling were selected in order to assess different aspects of the model. Section 7.1.1 begins by explaining the comparison of context-independent vector distribution for TourBERT versus other domain-specific BERT models, while section 7.1.2 presents results acquired from nearest neighbor, i.e., a synonym search performed with TourBERT and BERT-Base models. Sub-chapter 7.1.3 then describes the topic modeling approach applied to the Instagram posts using, once again, both TourBERT and BERT-Base models. Thereafter, section 7.1.4 provides a detailed overview of the user study conducted to assess how users perceive the quality of the topic modeling results from 7.1.3, and, lastly, sub-chapter 7.1.5 provides an overview of the vector down-projection using TourBERT and BERT-Base vectors.

7.1.1 Context-independent vector distribution

Prior to conducting both unsupervised and supervised model evaluation, the quality of the pre-trained token embeddings was assessed, and a vocabulary check was performed by

plotting the distribution of pairwise cosine similarities between words from the respective models' vocabulary. Figure 12 below depicts the results for four different domain-specific models including TourBERT. Each plot shows the average number of neighbors for a given word in the vocabulary, dependent on a certain cosine similarity threshold.

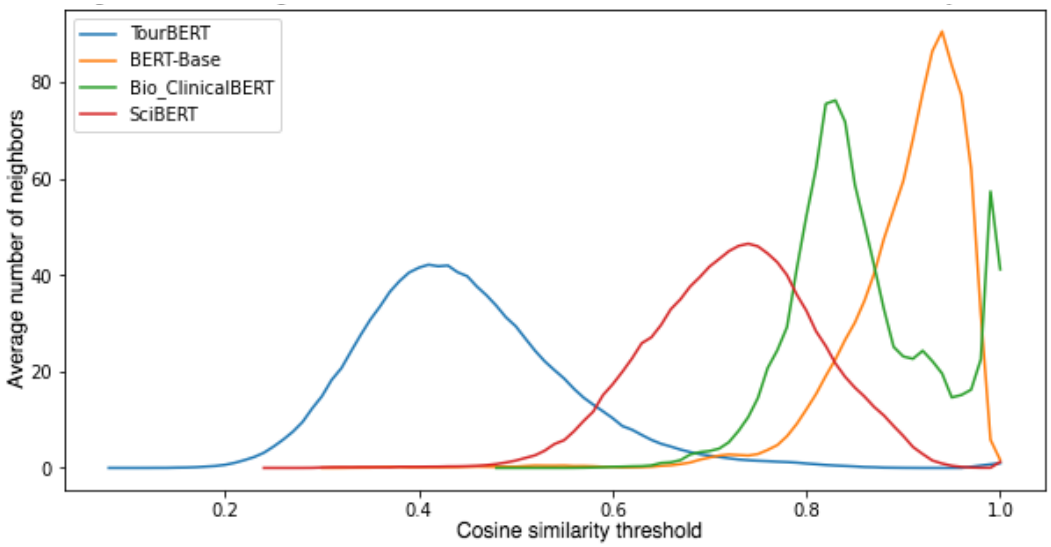


Figure 12: Comparison of word vectors' similarity distribution across four pre-trained models: BERT-Base, TourBERT, Bio_ClinicalBERT, and SciBERT.

For example, the distribution for TourBERT vectors shows that each word has about 20 neighbors with, on average, a similarity of 0.3. Also, one can observe that TourBERT seems to have the same neighbors for similarity values ranging from 0.8 to 1, which can be determined from the longer distribution tail on the right.

In order to produce such a plot for each model, 1,000 random words were chosen from its respective vocabulary and sampled. After, embeddings were produced for each word, and the cosine similarity matrix was computed using vector normalization and inner product operations. Subsequently, statistics showing the number of neighbors at a certain cosine similarity value for each word was computed. In order to reduce the number of distinct cosine similarity values, the similarity matrix was rounded to two decimals. Once the neighbor counts for each similarity value were computed for each word, neighbor counts across all the words for the same cosine similarity value were averaged. From the distribution plot, one can notice that the distributions for nearly all the models except TourBERT are highly skewed to the right. The fact that the distribution for the Bio_ClinicalBERT model is close to that of the BERT-Base model can be

explained by the nature of the vocabulary and the checkpoints used for the pre-training of both models. Since Bio_ClinicalBERT was initialized from the Bio_BERT's checkpoint, which, beforehand, was initialized from the BERT-Base model's checkpoint, both Bio_ClinicalBERT and BERT-Base seem to have quite similar distribution shapes. The SciBERT model, though it was also initialized from the BERT-Base checkpoint, has a similar distribution shape as TourBERT's, partially due to the fact that both SciBERT and TourBERT were pre-trained using a custom vocabulary. When looking at the distribution of the TourBERT model, one can see that it is slightly skewed to the left. The different shape of the distribution can be explicated by this model having been pre-trained from scratch using a custom vocabulary and without having been initialized from the BERT-Base checkpoint or using BERT-Base vocabulary, unlike other domain-specific models. Moreover, the long tail of the TourBERT similarity distribution on the right indicates that, with an increasing similarity threshold, the number of neighbors drops significantly. The fact that all distributions have long tails on the left side, however, imply a high number of orthogonal vectors, which could be an indicator of well-isolated words' neighborhoods.

To sum up, it is expected that the TourBERT model should achieve higher quality word neighborhoods and thus capture the tourism-specific context better than BERT-Base. To confirm or reject these assumptions, a nearest neighbor search for both TourBERT and BERT-Base models was performed, as is described in the next section below.

7.1.2 Nearest neighbor search

As part of the unsupervised model evaluation, a synonyms search using both BERT-Base and TourBERT models was performed. In NLP terms, a synonyms search task is defined as a nearest neighbor search problem. This task is solved by measuring similarity between a given pair of words, or any other smaller (e.g., tokens) or bigger (e.g., phrases, sentences, or even entire documents) units. To calculate words' similarity, cosine distance is typically used as the metric. In this section, the nearest neighbor search was conducted to find the closest words for a given set of words that were manually defined by a tourism domain expert. These words have multiple semantic meanings and, in particular, possess a special meaning for the tourism domain. Some examples include words such as “authenticity”, “experience”, “entrance”, ”transfer”, and others. A total of 10 words were selected for this experiment.

Prior to conducting this experiment, it was hypothesized that BERT-Base should produce a list of nearest neighbors unlocking the general meaning of the predefined words and ignoring its relation to the tourism domain. On the contrary, TourBERT is expected to output words that are very close to a given word within the tourism context. For example, the word “transfer” generally has multiple meanings and is usually associated with “transformation”, “transplantation”, and so on. However, from a tourists’ perspective, associations with “transfer” would likely involve words such as “taxi”, “pickup”, or “hotel transfer”.

From a technical point of view, the native implementation of BERT does not allow for querying “most similar words”. This is because, unlike Word2Vec or FastText models, BERT does not contain static vectors; rather, it produces them dynamically and can output two completely different vectors for the same word based on the context in which it was mentioned. Since the intention is still to compare words as standalone context-independent units, an algorithm was constructed that allows for any BERT-like model to query its vocabulary in order to find the words that are most similar. The algorithm works as follows: In the first step, pairwise similarities between all the words in BERT’s vocabulary are computed, resulting in a 30,522x30,522 matrix. Using KDTree algorithm from the Python’s Sklearn library, a search index is built upon that matrix, which allows for fast querying.

The results for BERT-Base and TourBERT models are provided in Tables 3 and 4, respectively. Each table contains the set of pre-selected words in the top row as well as the top eight most similar words for each one of them.

authenticity	*experience*	*entrance*	*attraction*	*ticket*	*destination*	*guide*	*transfer*	*sightseeing*	*service*
legitimacy	teach	shelter	attractions	tickets	dying	companion	recovery	trees	vessel
sincerity	heal	entrances	restaurant	fare	choice	entry	exchange	fireworks	authority
competence	communicate	archway	hotel	fares	lame	visit	imaging	shops	headquarters
authorship	consume	gate	exhibit	card	address	database	restoring	pacing	facility
flexibility	learn	roof	pavilion	trains	exit	forum	sale	comedy	workshop
integrity	eat	causeway	nightclub	bus	partner	workshop	comparison	prostitutes	circulation
conscience	consider	tenants	mall	metro	correction	access	recovering	sidewalk	companion
characterization	experiences	exit	ballroom	freight	priorities	google	screening	nights	operation

Table 3: Synonyms Search with BERT-Base.

authenticity	*experience*	*entrance*	*attraction*	*ticket*	*destination*	*guide*	*transfer*	*sightseeing*	*service*
uniqueness	adventure	entry	destination	entry	spot	tourguide	transport	exploring	staff
ambiance	enjoyment	admittance	feature	entrance	attraction	driver	pickup	sights	personnel
originality	opportunity	admission	landmark	wristband	place	interpreter	transportation	attractions	hospitality
intimacy	expere	ticket	place	admission	point	narrator	journey	exploration	frontdesk
charm	trip	fee	institution	fee	itinerary	host	limousine	nightlife	housekeeping
accuracy	excursion	carpark	museum	pass	hotspot	leader	shuttle	hiking	guiding
flare	activity	payment	spot	payment	venture	proprietor	taxi	outings	cleanliness
warmth	attraction	gate	site	card	hangout	organizer	minivan	excursions	roomservice

Table 4: Synonyms Search with TourBERT.

When comparing the neighbors generated by BERT-Base and TourBERT, one can see that TourBERT captures, almost perfectly, the tourism-specific meaning of a given word. On the contrary, BERT-Base captures a more generic meaning of the same word. For example, TourBERT associates the word “ticket” with “entrance” and “wristband”, whereas BERT-Base places the word within the scope of public transport and outputs neighbors such as “trains”, “bus”, and “metro”. To provide another example, the word “destination” is associated via the BERT-Base model with words like “dying”, “choice”, “lame”, and “address”, while TourBERT outputs “spot”, “attraction”, “place”, and other words that are closely related to a “destination” in tourism contexts.

7.1.3 Topic modeling with Instagram #wanderlust posts

To perform topic modeling, 5,000 Instagram posts, each containing the #wanderlust hashtag, were collected manually using Python’s ScraPy library. Instead of using posts’ texts, which are often incomplete or sometimes even missing, each photo from these posts was annotated automatically using the label detection service provided by Google Cloud Vision API. A more detailed description of this approach can be found in Arefieva et al. (2021).

By means of this dataset, topic modeling was performed using both TourBERT and BERT-Base embeddings as input features for the k-Means clustering algorithm. For both the TourBERT and BERT-Base results, k-Means models with 25 clusters were reported. To choose the best cluster number, models for five to 50 topics were trained with the step of five and compared using

silhouette score. The silhouette score plots for the TourBERT and BERT-Base models are visualized in Figure 13 below:

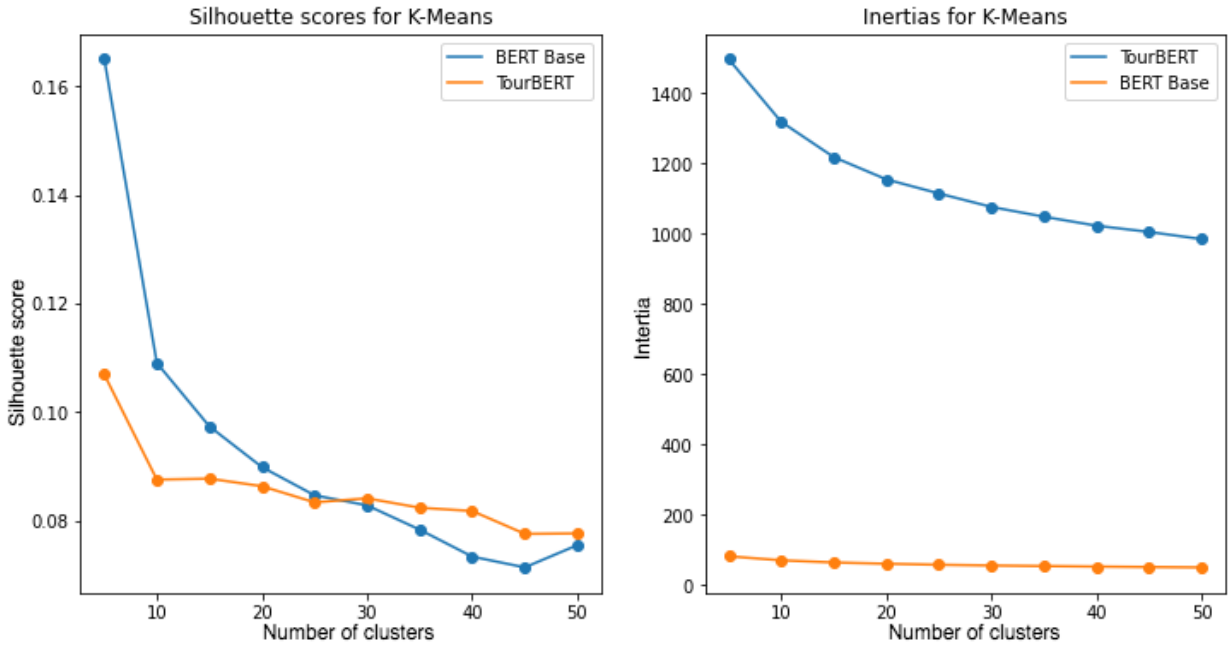


Figure 13: Silhouette scores and inertias for TourBERT and BERT-Base models for topic models from five to 50 clusters with the step of five.

Although the best cluster number should be chosen according to the highest silhouette score, the number of 25 clusters was selected because TourBERT and BERT-Base show comparable performance at that specific point. Moreover, a higher number of topics allows for model evaluation on a higher granularity level.

To visualize the topic modeling results, an interactive visualization tool was created, following the ideas from PyLDAvis (Sievert and Shirley, 2014). PyLDAvis is a library that enables the visualization of cluster centers on a two-dimensional plot while simultaneously displaying the most important words for each topic. In addition, the frequency of each word within a topic against its overall frequency over the entire dataset is shown. The reason PyLDAvis is unsuitable for visualizing k-Means clusters, although both Latent Dirichlet Allocation (LDA) and k-Means could be applied to solve the topic modeling problem, is because LDA is a probabilistic method, whereas k-Means is a non-probabilistic one. Thus, different methods are required for the interpretation of the results. Another significant difference between these two methods is

that LDA can assign the same document to multiple topics with different probabilities, but k-Means is a “hard” clustering algorithm that strictly assigns every sample to precisely one cluster without providing probability distribution over topics (i.e., the probability for each document belonging to a certain topic can be considered to be equal to one). To produce visualizations for a k-Means cluster model that would look similar to PyLDAvis dashboards, the following approach was used: Cluster features, i.e., the most representative words for each topic, were extracted using tf-idf vectorization (Lahitani et al., 2016) of all documents belonging to the same cluster. Using the resulting tf-idf matrix, the top 15 most important words for each topic were sampled. Next, for each word, its frequency within a topic as well as its frequency within the entire dataset were calculated for future visualization purposes.

The visualization dashboard consists of two blocks (see Figures 14 and 15). The first block on the left visualizes cluster centers as a two-dimensional plot, where the size of a cluster center is proportional to the cluster population size (cluster population size is defined as the number of samples belonging to the same cluster). The Principal Component Analysis (PCA) method was adopted to reduce the dimensionality of the vectors from 768 to only two dimensions. Each enumerated cluster center on the plot is clickable and changes the view dynamically in the second block on the right. When a user clicks on a particular topic (or enumerated circle on the plot), the top 15 most frequent words for a topic appear on the right half of the dashboard. For each word, the darker bars depict the word’s frequency within the entire dataset, while the lighter bars display the word’s frequency within the selected topic.

This visualization tool was created in 2020 around the same time when other similar publicly available tools for embedding-based topic modeling and visualization, e.g. Top2Vec (Angelov, 2020) or BERTopic (Grootendorst, 2020), were first published. The technical implementation of the visualization dashboard is based on Python’s Sklearn and Altair libraries. The resulting interactive dashboards are shown in Figures 14 and 15:

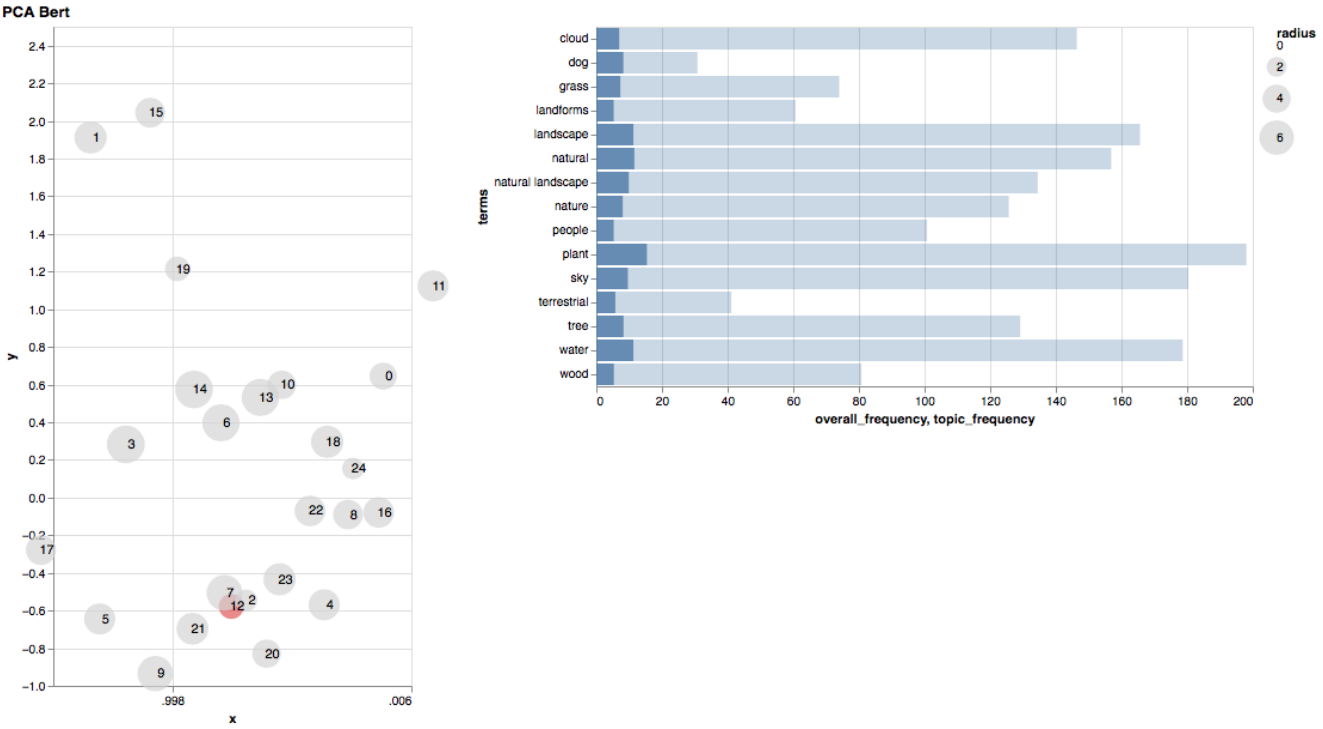


Figure 14: Topic model for 25 clusters created with BERT-Base.

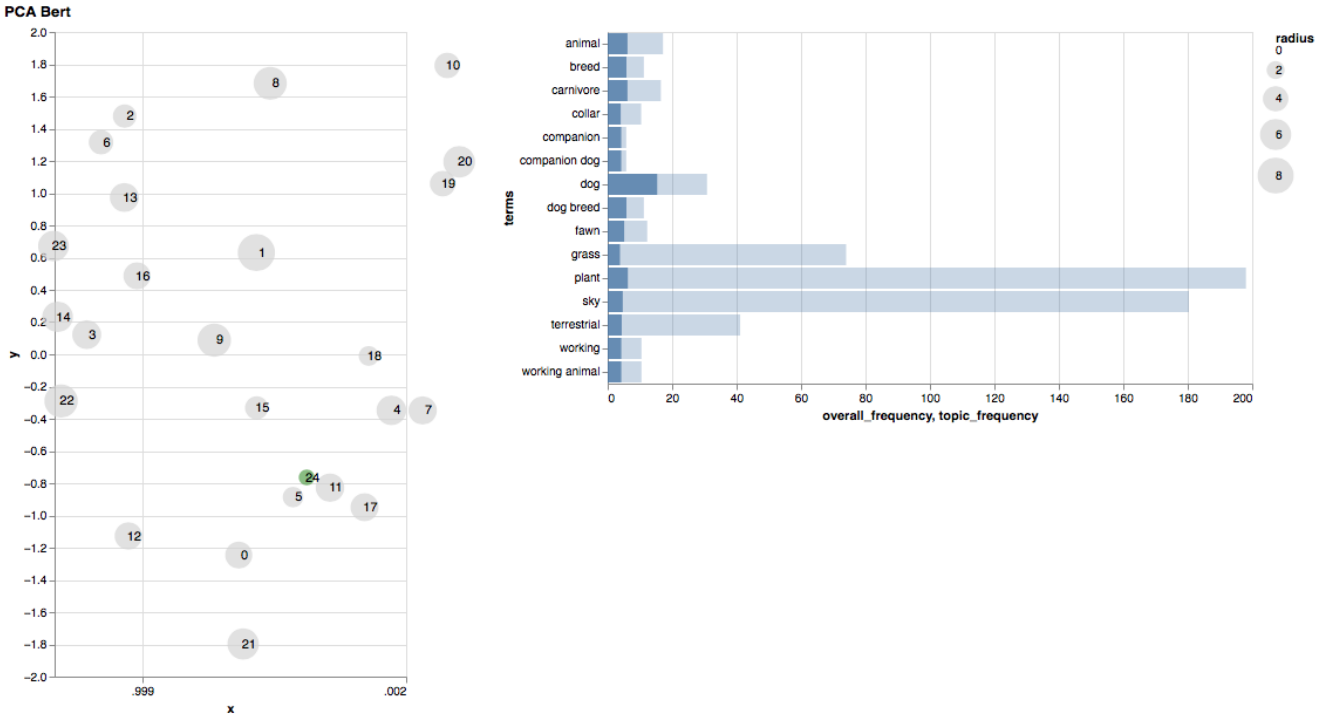


Figure 15: Topic model for 25 clusters created with TourBERT.

Based on the figures above, it can be said that clusters created with TourBERT vectors are better separated from each other than those produced with BERT-Base vectors. Furthermore, by looking at specific topics, one can see that TourBERT generates more consistent topics with less confusion. Some of the topic words are displayed in Tables 5 and 6 for both BERT-Base and TourBERT:

Topic #	Words
0	fashion, sleeve, shoulder, flash, flash photography, photography, street, street fashion, smile, hair, neck, eyewear, eyebrow, happy, sky
1	shades, tints, tints shades, plant, black, sky, shirt, bicycle, photography, white, font, sleeve, wood, building, automotive
2	sky, nature, water, landscape, plant, natural, cloud, people, tree, people nature, natural landscape, water sky, happy, cloud sky, azure
3	automotive, vehicle, sky, tire, font, plant, landscape, design, wood, art, building, rectangle, cloud, water, lighting
4	plant, natural, water, landscape, natural landscape, sky, ecoregion, cloud, tree, mountain, nature, cloud sky, highland, community, plant community
5	water, landforms, sky, coastal, coastal oceanic, oceanic, oceanic landforms, landscape, cloud, natural, beach, water sky, natural landscape, azure, plant
6	people, nature, sky, smile, people nature, sunglasses, flash, flash photography, photography, water, sleeve, care, vision, vision care, eyewear
7	landscape, sky, plant, cloud, natural, natural landscape, water, tree, building, nature, cloud sky, mountain, vehicle, people, blue
8	water, sky, cloud, landscape, plant, natural, natural landscape, resources, water resources, building, tree, mountain, cloud sky, water sky, nature
9	landscape, plant, natural, sky, water, natural landscape, nature, cloud, tree, grass, people, people nature, cloud sky, sky plant, wood
10	fashion, happy, sky, people, nature, photography, flash, flash photography, eyewear, smile, people nature, care, vision, vision care, plant
11	plant, sky, water, natural, landscape, ecoregion, tree, natural landscape, cloud, photography, fashion, flash, flash photography, smile, happy
12	plant, natural, landscape, water, natural landscape, sky, tree, dog, nature, grass, cloud, terrestrial, wood, people, landforms
13	building, sky, plant, window, vehicle, facade, tree, wood, design, house, automotive, tire, cloud, road, city
14	vehicle, automotive, sky, building, plant, tire, font, design, art, window, cloud, tree, wood, rectangle, lighting
15	plant, shades, tints, tints shades, sky, wood, black, fashion, bicycle, photography, rectangle, people, white, building, font
16	plant, water, natural, sky, landscape, natural landscape, cloud, ecoregion, mountain, tree, cloud sky, community, plant community, resources, water resources
17	landscape, plant, water, sky, natural, natural landscape, shades, tints, tints shades, tree, cloud, landforms, wood, coastal, coastal oceanic
18	fashion, sleeve, flash, flash photography, photography, street, street fashion, lip, shoulder, eyelash, eyebrow, smile, hairstyle, sky, neck
19	water, sky, equipment, cloud, equipment supplies, supplies, boating, boating equipment, boats, boats boating, landforms, boat, watercraft, coastal, coastal oceanic
20	water, landscape, natural, plant, sky, cloud, natural landscape, mountain, tree, nature, cloud sky, azure, highland, resources, water resources
21	plant, water, sky, nature, landscape, natural, cloud, tree, people, natural landscape, people nature, grass, cloud sky, mountain, building
22	sky, plant, cloud, water, landscape, building, natural, tree, natural landscape, mountain, cloud sky, window, nature, travel,

	road
23	plant, natural, sky, landscape, water, natural landscape, tree, cloud, nature, terrestrial, terrestrial plant, flower, grass, petal, wood
24	food, sky, cuisine, ingredient, recipe, tableware, dish, food tableware, ingredient recipe, water, tableware ingredient, staple, staple food, plate, produce

Table 5: Topic words for 25 topics produced with BERT-Base vectors.

Topic #	Words
0	plant, sky, tree, building, road, landscape, wood, cloud, road surface, surface, grass, window, sky plant, leisure, water
1	diving, underwater, water, fluid, marine, equipment, biology, marine biology, organism, fish, water underwater, liquid, diving equipment, underwater diving, blue
2	beach, people, water, sky, people beach, cloud, nature, people nature, water sky, azure, happy, travel, beach people, coastal, coastal oceanic
3	landscape, mountain, natural, sky, cloud, natural landscape, plant, slope, tree, cloud sky, highland, snow, sky mountain, terrain, sky plant
4	font, art, arts, event, rectangle, brand, design, pattern, graphics, photography, happy, painting, magenta, logo, visual
5	building, sky, window, facade, tower, design, urban, city, cloud, urban design, plant, sky building, road, house, building window
6	water, sky, afterglow, cloud, dusk, atmosphere, landscape, natural, natural landscape, sky atmosphere, cloud sky, sunlight, sunset, water sky, tree
7	tableware, drinkware, table, bottle, cup, dishware, food, glass, wood, plant, furniture, device, stemware, kitchen, wine
8	people, nature, sky, people nature, flash, flash photography, photography, happy, water, smile, plant, cloud, leg, gesture, tree
9	water, sky, equipment, boat, watercraft, cloud, vehicle, lake, supplies, boating, boating equipment, boats, boats boating, equipment supplies, water sky
10	care, vision, vision care, sunglasses, sleeve, eyewear, goggles, glasses, sky, dress, fashion, smile, shirt, flash, flash photography
11	automotive, vehicle, tire, bicycle, wheel, motor, motor vehicle, automotive tire, vehicle automotive, sky, lighting, automotive lighting, car, plant, tire wheel
12	plant, landscape, natural, natural landscape, sky, tree, nature, grass, community, plant community, cloud, people, people nature, water, sky plant
13	sky, water, cloud, landscape, natural, atmosphere, cloud sky, blue, natural landscape, azure, plant, nature, tree, horizon, sunlight
14	water, natural, landscape, sky, natural landscape, cloud, plant, nature, mountain, resources, water resources, ecoregion, tree, cloud sky, water sky
15	temple, sky, building, architecture, plant, facade, city, cloud, art, travel, tree, leisure, sculpture, world, monument
16	nature, plant, people nature, people, sky, happy, tree, landscape, cloud, natural, water, grass, natural landscape, travel, leisure
17	wood, design, building, rectangle, interior, interior design, window, shades, tints, tints shades, property, font, furniture, flooring, plant
18	food, cuisine, ingredient, tableware, recipe, dish, food tableware, ingredient recipe, produce, staple, staple food, cuisine dish, tableware ingredient, plate, cake
19	fashion, street, street fashion, sleeve, eyewear, flash, flash photography, photography, shirt, happy, waist, smile, dress, design, shoe
20	lip, eyebrow, eyelash, smile, hair, chin, shoulder, skin, nose, forehead, hairstyle, neck, eye, lip chin, facial
21	plant, flower, tree, terrestrial, twig, landscape, terrestrial plant, natural, petal, natural landscape, branch, grass, wood, sky, flowering

22	water, natural, plant, landscape, landforms, natural landscape, fluvial, fluvial landforms, landforms streams, streams, resources, water resources, sky, watercourse, water water
23	water, landscape, landforms, natural, sky, coastal, coastal oceanic, oceanic, oceanic landforms, cloud, natural landscape, water sky, azure, resources, water resources
24	dog, plant, animal, carnivore, breed, dog breed, fawn, sky, terrestrial, working, working animal, companion, companion dog, collar, grass

Table 6: Topic words for 25 topics produced with TourBERT vectors.

Although the hashtag "*#wanderlust*" may bring photos containing, to some extent or another, nature landscapes to mind, it seems that the topic model produced with TourBERT vectors was also able to identify distinct topics including "underwater world" (topic 1), "beach activities" (topic 2), "food and drink" (topic 7), "vehicle" (topic 11), and "animals" (topic 24). An attempt to find similarly grouped clusters within the results from the BERT-Base model did not bring about much success as nearly each topic involves landscape descriptions. While several distinct topics were indeed found by the model, the majority of them contain mixed concepts, each including terms describing nature and/or landscapes.

For better visibility, and to gain a better sense of the quality and distinction between topics, an additional visualization for each of the two topic models was produced, as demonstrated in Figures 16 and 17 for TourBERT and BERT-Base, respectively. Each figure contains a table with the first column showing words for a given topic and all the subsequent columns showing the top 10 most similar samples (photos) for that topic.

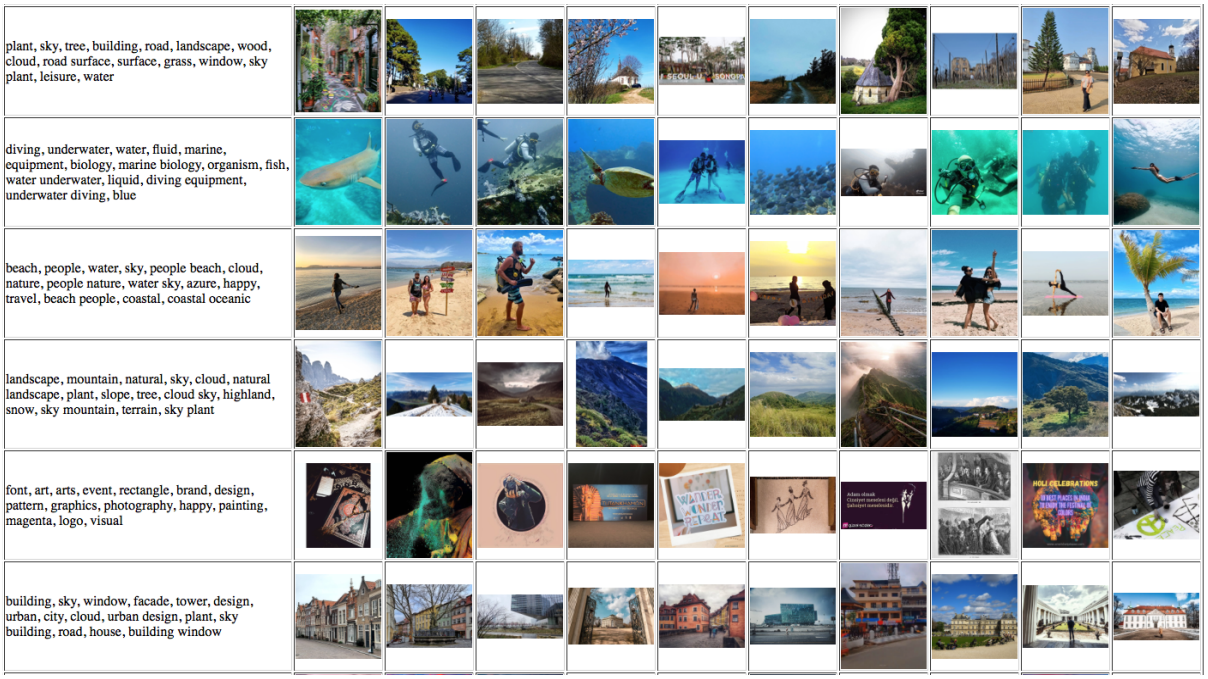


Figure 16: First six topics with cluster words and top 10 most similar images produced by the k-Means model using TourBERT vectors.

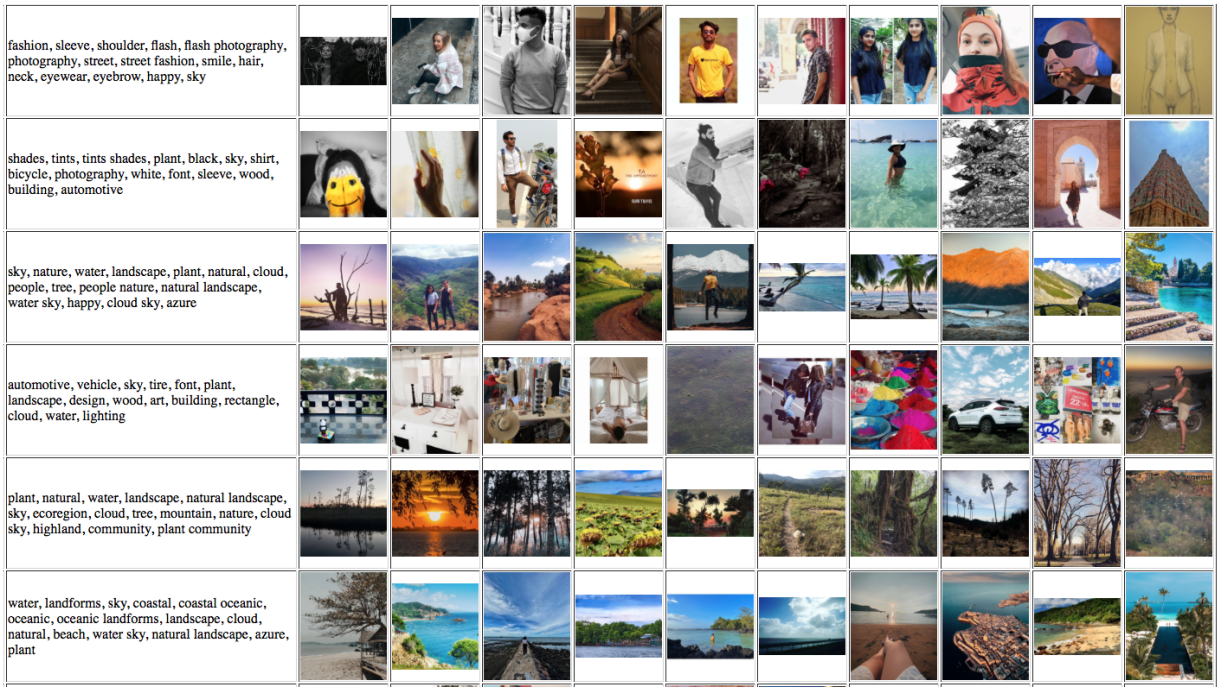


Figure 17: First six topics with cluster words and top 10 most similar images produced by the k-Means model using BERT-Base vectors.

Comparing the results from both models, one can see that the clusters produced with TourBERT vectors are better separated from each other when compared to those with BERT-Base, which sometimes include quite dissimilar photos within the same topic (e.g., in the third topic).

To further evaluate the homogeneity of the resulting topics as well as the overall quality of the model, an extensive user study, involving 82 independent participants, and statistical analysis of the results were performed. The user study and evaluation results will now be described in the subsequent section.

7.1.4 User study

A user study was designed for the same dataset of images and annotations. As so, a set of the 10 most similar photos for each of the 25 clusters from BERT-Base and TourBERT was created, and users were asked to evaluate how similar the photos within each of the 50 clusters are on a seven-point Likert scale (Likert, 1932). This evaluation approach is reliable in gaining an intersubjective perception of the quality of the clusters, similar to measuring the intercoder reliability in qualitative studies (Lavrakas, 2008). The image clusters were shown to participants in a rotating manner, i.e., alternating randomly. Every participant was invited to rate each of the 50 topics (25 for TourBERT and 25 for BERT-Base) using the available values ranging from 1 – “very similar” to 7 – “very different” (see Figure 18). In total, 82 users evaluated the results for both TourBERT and BERT-Base topic models. The purpose of this study was to evaluate whether participants perceive the quality of TourBERT to be better in general than the quality of the BERT-Base model. To derive quantitative measures, a paired t-test was selected to compare any differences between the ratings for TourBERT and BERT-Base. In the null hypothesis, it is assumed that the true mean difference between the paired samples is zero. On the contrary, the assumption under the alternative hypothesis is that the true mean difference between the paired samples is not equal to zero. Since the direction of difference, i.e., greater than zero, does not play any role in this experiment, a two-tailed test was chosen and conducted using SPSS software.

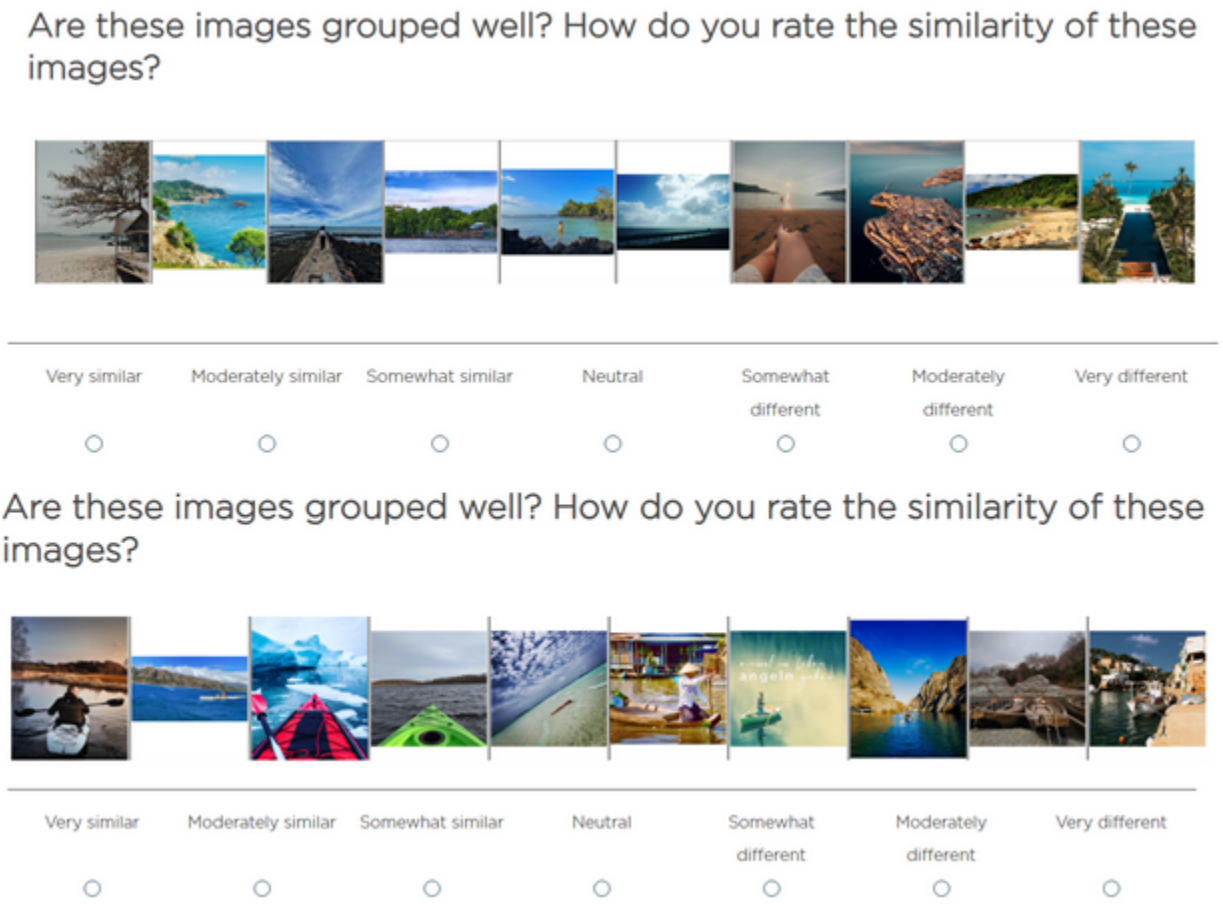


Figure 18: Two examples of image clusters.

Results are shown in Table 7 below, which consists of three separate tables. The first table displays the basic statistics for each sample group, like the mean value, the number of evaluations, the standard deviation, and the standard error mean. The second table, on the other hand, shows the output for the paired t-test. In the first column, the order of subtracting is shown, i.e., the values of TourBERT that were subtracted from those of BERT-Base. For paired differences, values such as the mean, standard deviation and standard error mean were calculated again as well as the lower and upper bounds of the 95%-confidence interval. The resulting *t*-value is equal to 21.898 and must be compared with the critical *t*-value for 81 degrees of freedom (degrees of freedom are calculated as *n-1* where *n* is the number of samples) in order to either reject or accept the null-hypothesis. By looking into the t-distribution table (two-sided), it was found that the critical *t*-value for 81 degrees of freedom and *p*-value of 0.05 equals 1.990. As the calculated *t*-value is greater than the critical *t*-value, the null

hypothesis could be rejected, and the conclusion was made that the differences between TourBERT and BERT-Base are not equal to zero, rendering the results as statistically significant.

In the third table, the effect sizes for paired samples were measured using Cohen's d and Hedges' correction coefficients. Cohen's d is typically used to measure the strength of a relationship between two population variables and is calculated as the difference of two means divided by the standard deviation of the data (Cohen, 1998). The resulting effect value is called magnitude and, like a t-distribution table, comes with a table that describes the effect volume depending on Cohen's d value ranging from 0.01 to 2. According to SPSS, the effect size is equal to 0.517, which is a medium-level effect. From the general results below, it can be concluded that perceived similarity of the annotated images was significantly better for TourBERT than BERT-Base.

Paired Samples Statistics

		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	BERT	3,7759	82	0,71655	0,07913
	TourBERT	2,5239	82	0,61724	0,06816

Paired Samples Test

		Paired Differences							
					95% Confidence Interval of the Difference				
		Mean	Std. Deviation	Std. Error Mean	Lower	Upper	t	df	Sig. (2-tailed)
Pair 1	BERT - TourBERT	1,252	0,51773	0,0571	1,13825	1,36577	21,898	81	0

Paired Samples Effect Sizes

			Standardizer	Point Estimate	95% Confidence Interval	
					Lower	Upper
Pair 1	BERT - TourBERT	Cohen's d	0,51773	2,418	1,986	2,846
		Hedges' correction	0,52015	2,407	1,977	2,833

Table 7: Results of the paired t-test for samples mean comparison for TourBERT and BERT-Base models.

7.1.5 Vector down-projection

In this experiment, another dataset containing images and manual textual annotations were used to produce BERT and TourBERT vectors for further down-projection and visualization, similar to what was executed in section 7.1.3 for topic modeling**.** The purpose of this experiment is to evaluate the cluster separation, where clusters result naturally from the down-projection method.

This dataset consists of 48 photos depicting different tourism activities like swimming, hiking, and snowboarding, amongst others. Using a simple web-service, 600 people were engaged to perform manual labeling on these photos; in other words, each person was asked to assign activity-related bi-gram tags to each photo. The compactness of this dataset allowed for usage of the TensorBoard projector service, which visualizes original pictures on a two- or three-dimensional plot.

For each photo annotation, TourBERT and BERT-Base vectors were produced, yet again, and visualized on a three-dimensional plot without applying clustering, but using UMAP with 9 neighbors instead. The results for both the BERT-Base and TourBERT models are provided in Figures 19 and 20 below:

Figure 19: Visualization of BERT-Base annotation vectors in Tensorboard Projector.

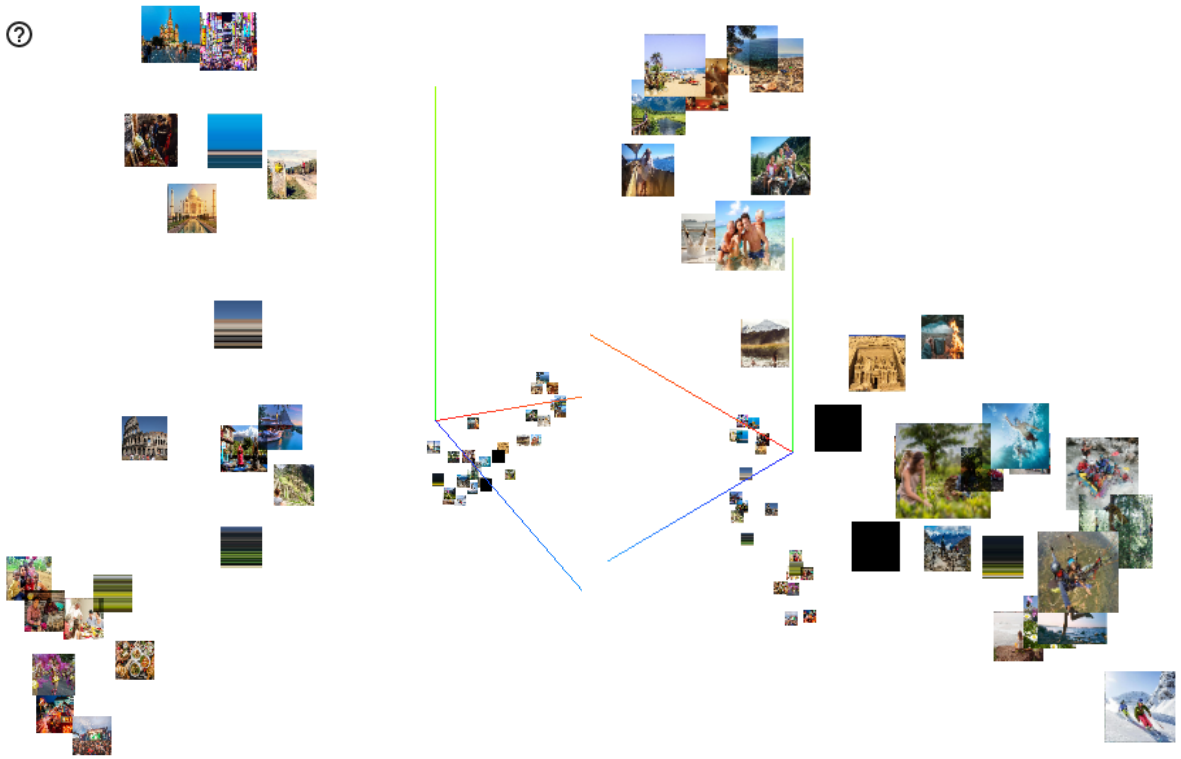


Figure 20: Visualization of TourBERT annotation vectors in Tensorboard Projector.

The figures above imply that TourBERT vectors result in better separated groups and that pictures within the same group comprise similar contents. Contrarily, looking at the results produced with BERT-Base vectors, it becomes evident that the photos are heavily mixed amongst each other and do not allow for the identification of well-separated/distinct groups.

7.2 Supervised evaluation - sentiment classification

In this sub-chapter, two different datasets for sentiment classification were applied for the quantitative evaluation of both BERT-Base and TourBERT models. The sub-chapter 7.2.1 describes the general methodology used for benchmarking TourBERT against BERT-Base in a supervised fashion, followed by sub-chapters 7.2.2 and 7.2.3, which describe multi-class and binary classification tasks, respectively, in more detail. Both sub-chapters also present the results for the TourBERT and BERT-Base models.

7.2.1 Methodology

There are several ways to perform classification with a BERT-like model, one of which (the most common approach yielding state-of-the-art results) is to attach a linear layer on top of the last hidden BERT output layer followed by a softmax layer. The linear layer transforms a multi-dimensional vector, 768 in case of BERT-Base, into a real-valued vector of length *N*, where *N* represents the number of classes in the dataset. Since the real values can be positive, negative, or zero, a function like softmax is required in order to transform it into a probability distribution over *N* classes, where values can be summed up to 1. The softmax function is defined as follows:

$$softmax(x_i) = \frac{exp(x_i)}{\sum_j exp(x_j)},$$

where x_i and x_j are values of the linear layer output vector x containing the dimensionality *1xN*, *j=0…N-1*, where *N* is the number of classes. The final output vector has values $softmax(x_i)$, which range from 0 to 1 and sum up to 1.

Another approach to constructing a BERT-based classifier is to use the BERT model only for the embedding inference, while using a different model, such as LSTM, as the classifier. In that case, BERT output vectors are fed into the LSTM neural network for further prediction. This approach is useful when input texts are longer than the BERT's maximum input length, allowing bigger texts to be split into a number of text chunks of equal length and to pass through BERT independently. The resulting vectors are then concatenated, together with preservation of the original text piece order, to create an input for LSTM that is capable of handling input sequences with varying lengths and managing the sequence order.

However, the choice of the final model architecture depends on the ultimate goal to be achieved via a specific classification task. If one seeks to achieve higher quality in a model and to use it in real-world applications, one should opt for a softmax classification layer or even more complex architectures like an LSTM. On the contrary, if the aim is to compare different models in order to uncover the best-performing model, one should choose a simpler classifier architecture, e.g., a feed-forward layer. The author is aware that this approach usually does not yield state-of-the-art

results, however, the goal of this evaluation is not to achieve the highest score on a given dataset but rather to show that the quality of TourBERT embeddings surpasses the BERT-Base model. Therefore, a single feed-forward layer on top of the BERT architecture was added.

In order to enable the architectural change described above, Pytorch's nn module was applied to create a sequential model consisting of three layers: a linear input layer with a dimensionality of (768, 50) and a hidden size of 50, a Rectifier Linear Unit (ReLU) layer, and a linear output layer with a dimensionality of (50, 2). ReLU is an activation function that transforms layer inputs to zero, if a value is negative, or otherwise keeps the original value. The final model output is the probability distribution over all labels, where the final prediction is the label that has the maximal probability.

Along with the model architecture design, the design of the training procedure can also vary depending on the final goal. For a higher model quality and for real-world applications with predefined data, continuous pre-training or BERT fine-tuning is a better option. Apart from that, evaluating the quality of the BERT embeddings themselves instead of the quality of the classifier is of main priority in this research. Therefore, all the layers of both the TourBERT and BERT-Base models were explicitly "frozen", and only the feed-forward layer on top of it was trained at the end. To "freeze" BERT, the dropout layer, which randomly drops model weights during training procedure, was deactivated. Second, the calculation of the gradient, which updates the model's weights through back-propagation, was disabled as well so that all the preceding layers of BERT remain unchanged.

Usually, to train such a deep neural network, the AdamW optimizer is chosen in place of the conventional gradient descent. In order to accelerate the training process and ensure convergence without skipping the local minima and to prevent overfitting in earlier stages, a linear scheduler with warm up was used. As an optimization metric, cross-entropy loss is typically used in classification problems and is computed with the following formula:

$$l_i(y, \hat{y}) = \sum_{c=0}^{C-1} y_c * log(\hat{y}_c),$$

where *C* is the number of classes, *i* is the number of the current input sample, y_c is 1 if the sample *i* belongs to class *c* or 0, and $\hat{y}_c$ is the predicted value obtained for $softmax(x_c)$. The loss for the entire dataset or for a batch with *N* samples is calculated as follows:

$$L(y, \hat{y}) = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{C=0}^{C-1} y_{nc} * log(\hat{y}_{nc}) y_{nc},$$

where *N* is the number of samples. A more detailed explanation of the loss and, in particular, of cross-entropy loss with softmax can be found in Barbiero et al.'s (2022) research.

For both sentiment classification tasks performed in this study, training was completed for two epochs using AdamW optimizer with default hyperparameters, cross-entropy loss as objective function, and the linear scheduler with warm up for the learning rate adjustment. The batch size was selected as 64 instead of 32, as recommended by BERT authors, because no BERT fine-tuning was performed and increasing the batch size accelerated the training process.

7.2.2 Multi-class classification

In a multi-class classification problem, each example or object can belong to only one class, meaning that object classes are mutually exclusive, and the model is required to predict a label given a set of more than two possible labels per choice. Given correct labels known in advance and labels that were predicted by the model, it is possible to evaluate the model's quality by calculating metrics such as validation loss, accuracy, precision, recall, and F1-score, which are computed using the following formulas:

$$acc(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N-1} 1(\hat{y}_i = y_i),$$

where *N* is the number of samples, y_i is the actual class label, $\hat{y}_i$ is the predicted class label for the input sample *i*, and $1(x)$ is the indicator function:

$$1(x) = 1 \; if \; \hat{y}_i = y_i, \; 0 \; otherwise.$$

$$recall = \frac{TP}{(TP+FP)},$$

$$precision = \frac{TP}{(TP+FN)},$$

$$F1 = \frac{2*(precision * recall)}{(precision + recall)},$$

where *TP* is the True Positive Rate, *FP* is the False Positive Rate, and *FN* is the False Negative Rate.

Validation loss is computed similarly to training loss, with the difference being that the validation dataset has a batch size of 1, which basically means that samples are processed sequentially. Therefore, the validation loss is computed as an average loss per sample. The model with the lowest loss and the highest accuracy, precision, recall, and F1-score is considered to be superior. Due to the fact that both BERT-Base- and TourBERT-based classifiers have identical classifier architectures, it is possible to conclude about the quality of the input embeddings as well. The goal of this experiment is to confirm that TourBERT embeddings lead to better classification results as compared to BERT-Base ones.

For this task, a dataset consisting of hotel reviews from Tripadvisor was used (Ray et al., 2021), which contains three labels, -1 = “negative”, 0 = ”neutral”, 1 = “positive”, and includes a total amount of 69,308 reviews. The dataset was first pre-processed and then split into training, validation, and testing sets according to the 80%/10%/10% proportion. The pre-processing procedure included lower-casing and the removal of punctuation and non-ASCII characters from the text.

The evaluation results for this task for all three models are presented in Table 8, while the plot in Figure 21 shows the average training batch loss for all three models. The loss was computed every 20 steps, i.e., after every 20 batches had been processed. The loss for all the batches was accumulated and then divided by the number of batches in order to obtain an average loss per batch.

	Validation set		Test set			
	Loss	Accuracy	Accuracy	Precision	Recall	F1
BERT-Base	0.4250	0.8190	0.81	0.66	0.4	0.42
TourBERT (WordPiece)	0.3146	0.8708	0.86	0.7	0.65	0.68
TourBERT (SentencePiece)	0.3166	0.8712	0.87	0.7	0.65	0.68

Table 8: Evaluation results for TourBERT and BERT-Base models for the Tripadvisor hotel review dataset.

Figure 21: Batch cross-entropy loss of TourBERT and BERT-Base on training data for the multi-label classification task.

7.2.3 Binary classification

A classification problem is defined as a binary classification if a model can choose between only two possible labels for a given example and each example can only belong to one class. For this task, a dataset of 515,000 reviews from European hotels was adopted, which is available online on the Kaggle platform[3]. This dataset contains attributes such as hotel name, the number of reviews, and its geographical position as well as negative and positive reviews from each reviewer. If a user has left only positive reviews, then the value for negative reviews is left blank and vice-versa. A subsequent pre-processing approach was used to extract only the positive and negative examples in order to prepare this dataset for a binary classification problem. This

[3] https://www.kaggle.com/jiashenliu/515k-hotel-reviews-data-in-europe

means that only reviews from users who left either solely negative or solely positive reviews were used. Using this approach, 35,000 positive and 35,000 negative reviews were sampled, resulting in 70,000 samples in total. Again, the dataset was split into training, validation, and testing sets according to the 80%/10%/10% proportion.

For binary classification, the same model architecture was used as for the multi-label classification, with the exception of the number of classes, which was changed to only two classes. The training procedure was conducted identically as in the previous task. In addition to accuracy, an Area Under ROC-curve (AUC-score) is also reported for binary classification**.** The Receiver Operating Characteristic (ROC) shows the True Positive Rate versus the False Positive Rate at varying acceptance probability thresholds. AUC is computed as the area under the ROC-curve. AUC scores for all three models are shown in Table 9 below:

	Validation set		Test set	
	Loss	Accuracy	Accuracy	AUC
BERT-Base	0.2296	0.9218	0.9279	0.97
TourBERT (WordPiece)	0.1371	0.9569	0.9633	0.99
TourBERT (SentencePiece)	0.1329	0.9586	0.9626	0.99

Table 9: Evaluation results for TourBERT and BERT-Base models for two sentiment classification datasets.

Figure 22: Batch cross-entropy loss of TourBERT and BERT-Base on training data for binary classification task.

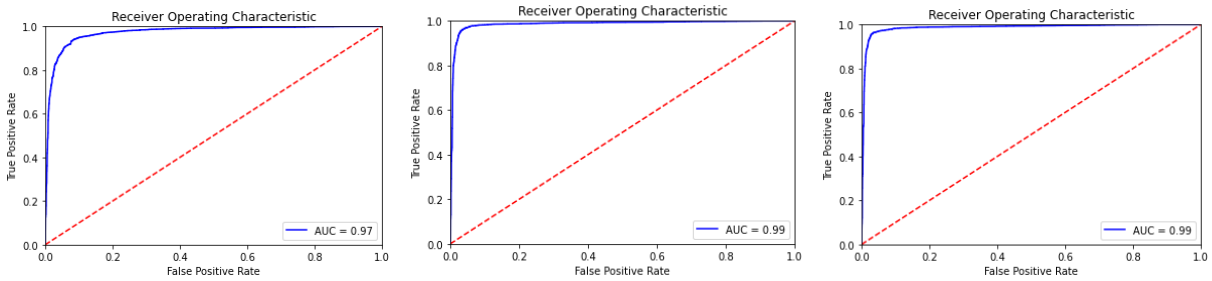


Figure 23: AUC score for binary classification for BERT-Base (left), TourBERT SentencePiece (middle), and TourBERT WordPiece (right) models.

According to the reported metrics, it is evident that TourBERT outperforms BERT-Base in both tasks and has lower loss and higher accuracy for both datasets. A consecutive conclusion is that TourBERT performs significantly better than BERT-Base on down-stream tasks involving datasets with tourism-specific contexts. To use TourBERT for future production purposes and real-world use-cases, one can fine-tune it for sufficient metrics improvement.

To conclude this section, it must be noted that the supervised evaluation setup was restricted to only the two tasks described above due to a lack of publicly available labeled datasets. However, this setting is considered to be sufficient enough for the evaluation of TourBERT because of the specificity of down-stream NLP tasks that can be performed in the tourism domain (Li et al., 2019).

8. Tourism destination similarities

This chapter will now provide the answers to the second research question, with the aim of evaluating the similarity of the main European destinations based on the descriptions of their tourism offerings on Airbnb Experiences.

As described in chapter 5, a TourBERT vector was produced for each Airbnb experience, and vectors belonging to the same city were averaged to obtain a single location vector. After that, vectors were down-projected for visualization using the UMAP algorithm. Following numerous experiments, it was shown that the default parameters of 15 neighbors and the minimum distance of 0.1 achieved better results. However, instead of the Euclidean distance, the cosine similarity metric was used, meaning that the number of components was left to two for visualization purposes. For better visual perception, k-Means clustering on location vectors was performed so that points on the scatter plot could be colored based on their cluster label. To select the best number of clusters, silhouette scores were computed for 15 different k-Means models, with cluster numbers ranging from five to 20. According to Figure 24 below, the maximum silhouette score was archived for a seven cluster solution.

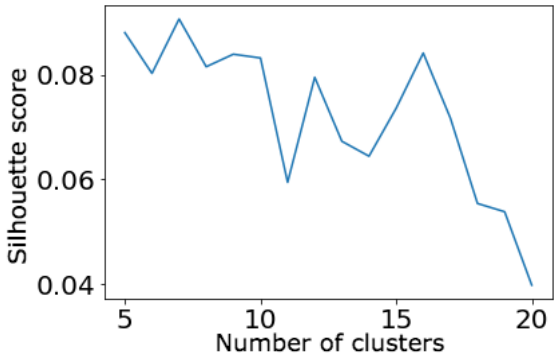


Figure 24: Silhouette scores of k-Means models with cluster numbers ranging from five to 20.

It should be mentioned that the original 768-dimensional, and not UMAP-reduced vectors, were clustered in order to preserve the original information encapsulated in the TourBERT vectors. Therefore, some of the visual clusters resulting naturally from UMAP may not correspond precisely to the clusters produced by the k-Means model. Figure 25 below shows the destination profiles obtained on a two-dimensional scatter plot.

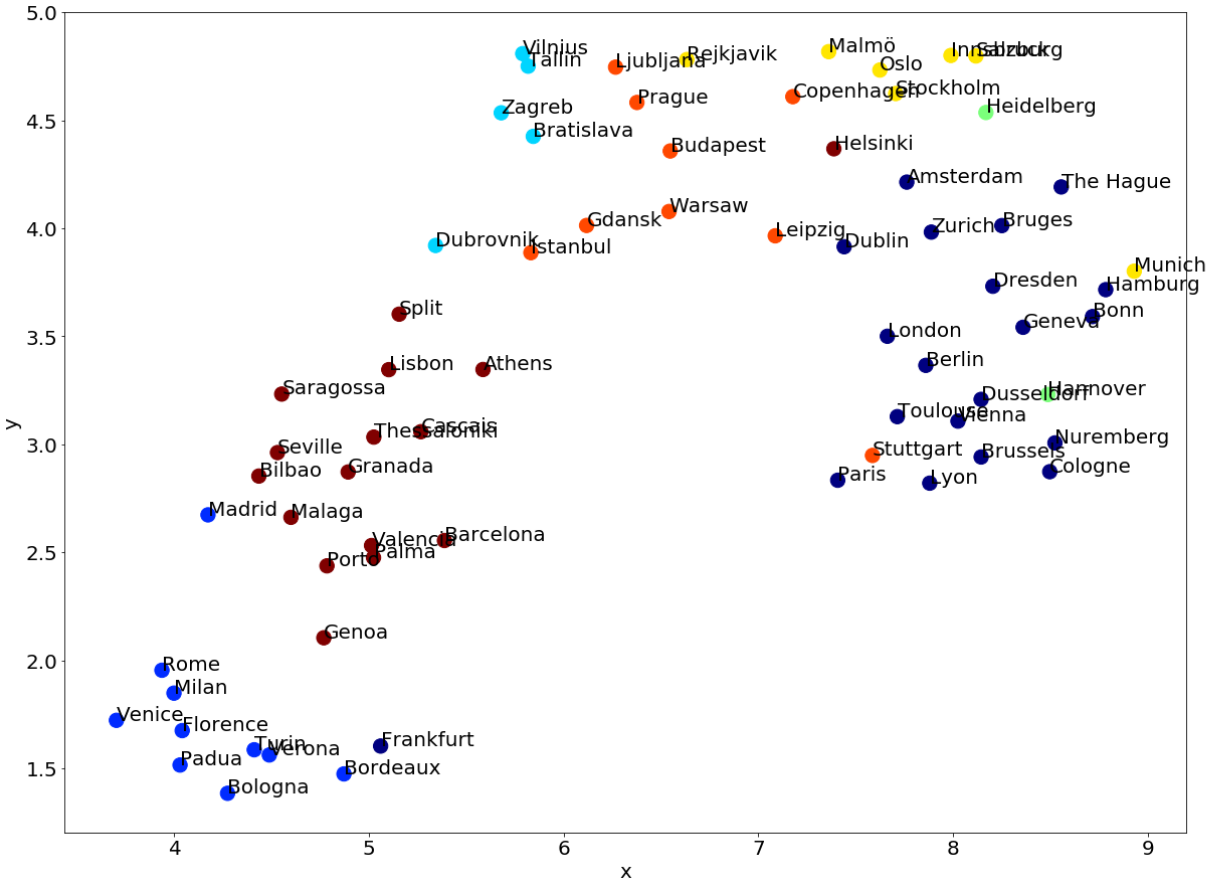


Figure 25: Similarity plot of the main European tourism destinations.

Based on the plot above, one can notice that some of the resulting groups reflect cities belonging to the same country on a map. For example, the left bottom corner contains mostly Italian cities, whereas the brown cluster above contains mainly Spanish cities. Nevertheless, there are also some mixed groups present, for instance, the dark blue cluster on the right contains not only German, Belgian, and French cities, but also other cities as well. A fascinating part of this research was to compare European capital cities, and one can see that main European destinations like London, Berlin, Vienna, Brussels, and Paris are actually placed quite close to each other. Interestingly, the biggest cities like London, Paris, or Berlin offer a wide range of experiences that cover multiple categories, e.g., sightseeing, restaurants, shopping, and many more. What they also have in common is that their offers concentrate mostly on exploring the center of the city and not focused on sports activities such as hiking or swimming, for example. Different from that group, capitals like Madrid, Rome, and Lisbon are placed further away from each other and are very distant from the other capitals, possessing almost an

opposite position than those included in the previously described cluster of central European capitals. In addition, clusters related to Spain, Italy, and Portugal rarely have any overlapping and are well-separated from other countries, which indicates that they provide quite special tourism offerings according to the model.

Another interesting fact is the grouping of cities based on southern and eastern European countries such as Croatia, Czech Republic, and Poland. Those clusters are located in the top area of the plot and are colored in light blue and orange. Vilnius, Tallinn, Zagreb, Bratislava, and Dubrovnik constitute the light blue cluster, while Ljubljana, Prague, Budapest, Warsaw, Gdansk, Istanbul, and Leipzig constitute the orange one. Interestingly, Stuttgart and Copenhagen both belong to the orange cluster as well but are located far from the cluster center. An explanation for this could be the fact that the points' coordinates are produced using UMAP and clusters are calculated based on the original 768-dimensional vectors, potentially resulting in slight inconsistencies due to the data loss through dimensionality reduction. Although Stuttgart, for example, fits better visually on the two-dimensional plot from the perspective of its original geographical location, UMAP might not have been able to fully capture the information about the offerings hidden in the TourBERT vector (i.e., this information got lost during visualization). Another cluster to pay attention to is the yellow cluster on the top of the plot, with the cities of Reykjavik, Malmö, Oslo, Innsbruck, and Salzburg. Without diving too much into detail, all of them have one main activity type in common that distinguishes them from all the other cities: skiing/snow activities in general. Again, one can see that Munich belongs to the same cluster according to k-Means but was placed far away from the cluster center because of UMAP.

In order to allow for better interpretation of the similarity analysis results, the cities were also visualized on a geographical land map using their original coordinates. Points were colored according to their cluster numbers obtained during the similarity search process. The comparison of the similarity plot to the original geographical map has multiple purposes. First, from a technical point of view, such a comparison aids in understanding whether a cluster membership is attributed to geographical aspects mentioned in experience descriptions or not. Second, it helps the user to comprehend the actual distance between his/her current location and the place that was recommended.

To compare the resulting projection with the original geographical map, all 69 cities used in this experiment were visualized on a real geographical map using Python and Kepler GL[4], an online software for geographical mapping. Using Python's Geopy library, the latitude and longitude coordinates of each city were obtained. After that, a .csv file with city names and their said coordinates was produced to upload to the Kepler GL service and allow for interactive geographical visualization. Clusters produced with k-Means were embedded into this map as well in order to simplify the comparison and see how the artificially created clusters compare to the original "geographical clusters". Figure 26 below shows the real geographical positions of the 69 cities used in this study.

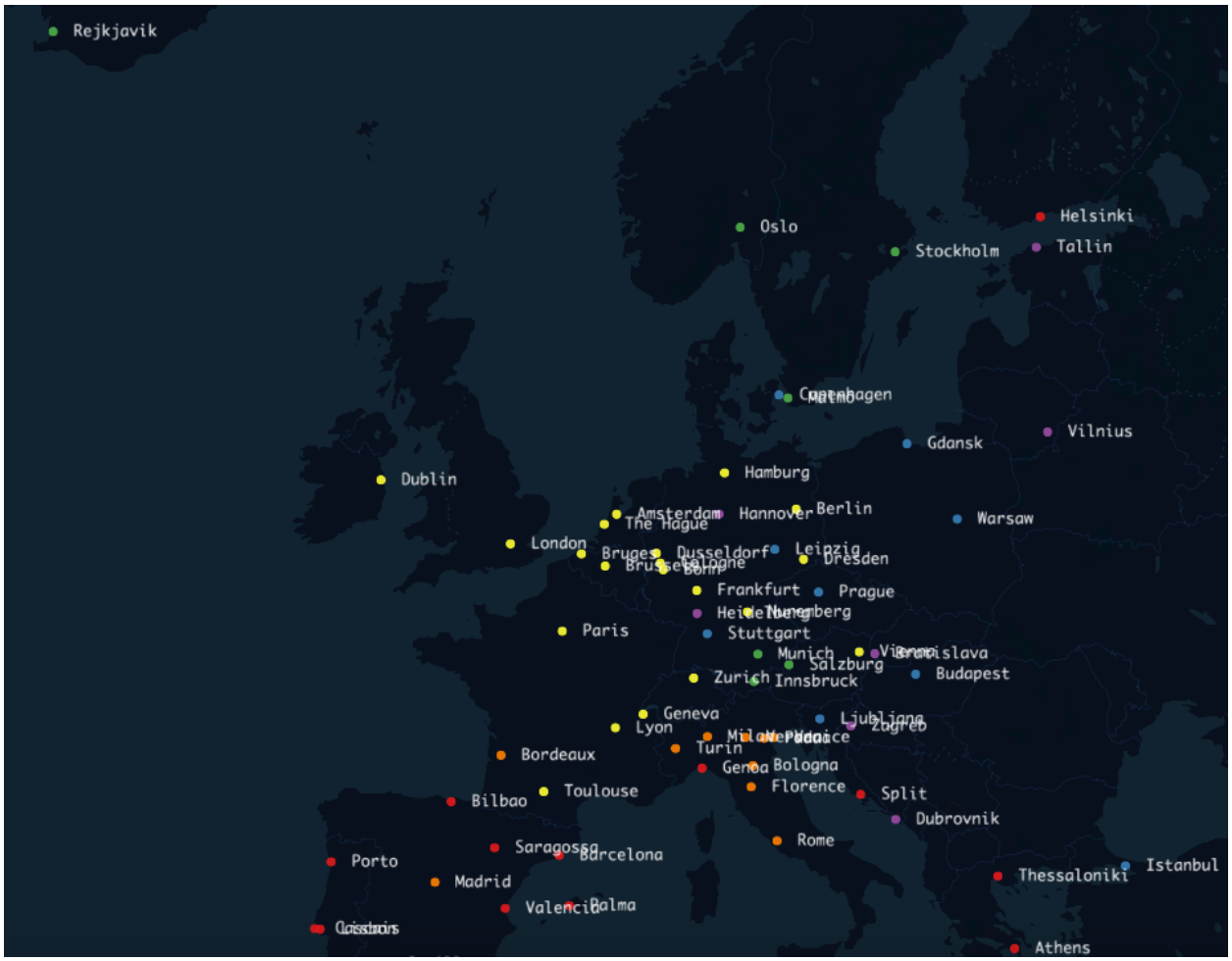


Figure 26: Map of 69 European cities used in the similarity comparison experiment.

[4] https://kepler.gl/demo

Country names were explicitly not displayed in order to avoid straining users with too much visualization content. One can see that the artificially created clusters overlap with original "geographical" clusters to some degree. For example, Spanish or Italian cities are well-grouped together, which provides evidence of different cities in Italy having similar tourism offerings in general, and the same applies to Spain. Interestingly, Reykjavik is located far away from all other cities but has similarities with Oslo and Stockholm, all considered northern cities. In order to better investigate the similarities within central and eastern Europe, the map was zoomed in on and can be seen in Figure 27 below:

Figure 27: Cities of central and eastern Europe.

From this map, one can notice two small yet distinct clusters: Heidelberg, Zagreb, Dubrovnik and Hannover in purple, and Munich, Salzburg, and Innsbruck in green. Whereas the second grouping could have been anticipated, the first cluster is of bigger interest. Despite the fact that the cities belonging to that group are placed far away from each other in the real world, they seem to have similar tourism offerings according to the model. Also, the red cluster in southern Europe might be of great interest as it is an indicator of a broader market existing mainly for summer tourism. All cities from that cluster have the fact that they are located near the

beach/bodies of water, and therefore have similar tourism offerings for the summer, in common. That includes many water activities, and it might also be a good complementary feature to include local food (e.g., fresh fish) in a summer tourism package. The similarities between those cities, despite being spread across the continent, might introduce consumers to a new opportunity when selecting their next tourism destination – by knowing that the general offerings are similar, they could pay more attention to the financial or geographical aspects of a trip, i.e., they could choose an offer that is cheaper or closer to their location, depending on their specific needs.

To gain deeper insights from the similarity analysis results and in order to understand the aspects of the offerings these similarities are based on, topic modeling was also performed. In order to extract the most important words for each group of cities that belong together (see Figure 26), the tf-idf approach was used. As a pre-processing step, a single text document per city was constructed by concatenating all the experience descriptions for that specific location together. The resulting texts were then lower-cased and cleared of punctuation and special characters. After, each city's group was analyzed separately; for example, the first group of cities contains 20 cities, therefore, 20 texts were passed to a tf-idf vectorizer, resulting in a 20xN matrix, where N denotes the number of unique bigrams. Finally, in order to acquire the most important bigrams for a given cluster of cities, rows of the tf-idf matrix were summed up and the top 30 bigrams with the maximum sum value were selected as output. The results for each cluster are shown in Table 10 below:

Cluster	Cities	Topic words
0	Amsterdam, Berlin, Dublin, London, Nuremberg, Paris, Vienna, Brussels, Cologne, Dresden, Düsseldorf, Frankfurt, Geneva, The Hague, Hamburg, Lyon, Bonn, Bruges, Toulouse, Zurich	old town, eiffel tower, walking tour, tai chi, street art, hidden gems, cathedral city, cologne cathedral, city tour, small group, cologne old, tour ends, notre dame, tour discover, buckingham palace, rally grounds, heart old, world famous, coffee reading, turkish coffee, old city, included price, city centre, step step, croix rousse, learn history, world heritage, learn make, fr dom
1	Bordeaux, Florence, Madrid, Milan, Rome, Venice, Bologna, Padua, Turin, Verona	lake como, euganean hills, olive oil, balsamic vinegar, wine tasting, saint emilion, walking tour, cooking class, vinegar modena, traditional balsamic, piazza navona, royal palace, trevi fountain, cinque terre, eternal city, fresh pasta, roman forum, glass wine, plaza mayor, san marco, parmigiano reggiano, learn make, piazza maggiore, abano terme, ice cream, palatine hill, sistine chapel, historic center, local products, new friends

2	Dubrovnik, Bratislava, Tallinn, Vilnius, Zagreb	old town, upper town, walking tour, city walls, food tour, price group, st mark, driver guide, beer tasting, danube hotel, inn radisson, park inn, radisson danube, old port, craft beer, street art, blue cave, old city, pile gate, rector palace, plitvice lakes, interesting stories, narrow streets, national park, history slovakia, meeting guide, vratna valley, zilina departure, free tour, main entrance
3	Heidelberg, Hannover	old bridge, ferry house, old town, body weight, burning process, house basse, old ferry, photo shooting, amazing photos, bridge castle, cafe course, cities world, couples families, course follows, cross neckar, cute cafe, dance dance, dance meditation, discuss expectations, expectations ideas, families want, follows safety, friends couples, hesitate discuss, high quality, ideas styling, kayak tour, kayak tours, meet cute, regarding covid
4	Munich, Innsbruck, Malmö, Oslo, Reykjavik, Salzburg, Stockholm	old town, northern lights, south coast, birds prey, eagles area, fallow deer, adolf hitler, icelandic nature, lava fields, bus stop, carriage ride, cinnamon rolls, gamla stan, kayak center, public transportation, king ludwig, hidden gems, tour starts, apple strudel, weather conditions, black sand, natural beauty, eagle nest, scenes filmed, time fits, trapp family, views old, von trapp, norwegian winter, og hesten
5	Istanbul, Prague, Budapest, Copenhagen, Gdansk, Leipzig, Ljubljana, Stuttgart, Warsaw	old town, charles bridge, grand bazaar, wine region, walking tour, salsa salsa, turkish coffee, ice cream, hagia sophia, blue mosque, galata tower, photo session, buda castle, matched cocktails, tour starts, fisherman bastion, town square, wine tasting, salsa basic, hidden gems, eagles area, jewish quarter, guide meet, vistula river, tivoli park, cream shops, specially matched, traditional turkish, street art, street food
6	Barcelona, Lisbon, Saragossa, Athens, Bilbao, Cascais, Genoa, Granada, Helsinki, Malaga, Palma, Porto, Seville, Split, Thessaloniki, Valencia	old town, port wine, national park, cinque terre, walking tour, sagrada familia, douro valley, diocletian palace, douro river, street art, wine tasting, natural park, blue cave, historic center, historical center, olive oil, basque country, plaza espa, crystal clear, tagus river, narrow streets, santa cruz, sierra nevada, world heritage, pena palace, food market, local products, local tapas, small group, hidden gems

Table 10: Topic modeling results for similarity analysis of the most popular European destinations based on Airbnb Experiences.

From the table above, it can be said that the first cluster (cluster 0) contains mostly central European capital cities and the offerings include, to a significant extent, sightseeing tours and photo shoots. The majority of the cities in clusters one and six are Italian and Spanish, respectively, with activities that are specific to their local culture, for example, food and cooking tours or wine/local food tastings. Compared to Italian cities, the Spanish cluster includes more offerings related to water activities, corresponding to cluster words “douro river” and “tagus

river". Cluster two is less specific and includes walking and driving activities as well as food and drinking tours. The third cluster, on the other hand, includes only two cities, Heidelberg and Hannover, which despite the usual sightseeing tours seem to provide more specific offerings like photo shoots, dancing courses, and kayak tours. Cluster four is different from all the other clusters in that it comprises cities from northern Europe that have different weather conditions affecting the range of offered activities. For example, topic words like "icelandic nature", "lava fields", "weather conditions", "black sand", "natural beauty", or "norwegian winter" describe the beauty of the natural landscapes of northern Europe as well as point to colder weather conditions.

Cluster five is interesting for this analysis because it includes Istanbul, a place containing few touristic and cultural commonalities with other cities present in that cluster. Although it might be difficult to initially guess the similarities amongst them, both geographical as well as economical aspects that bring Istanbul closer to Warsaw, Prague, and Budapest exist. First of all, Gdansk, Warsaw, Budapest, and Istanbul possess nearly the most eastern geographical coordinates (see Figure 26), and geographical position is known to influence the development of a given country and, in turn, its economic history. Tasan-Kok (2004) analyzed Budapest, Warsaw, and Istanbul based on their economical similarity and found that the expansion of neo-liberal capitalism in Hungary, Poland, and Turkey led to similar urban spatial changes in all three cities. The neo-liberal transformation of municipal governments in Budapest, Istanbul, and Warsaw thus showed similar urban development patterns (Tasan-Kok, 2004). Aspects as those described above are of significant importance for tourism managers, who could uncover unexpected touristic similarities attributed to political and economic changes that happened in previous decades.

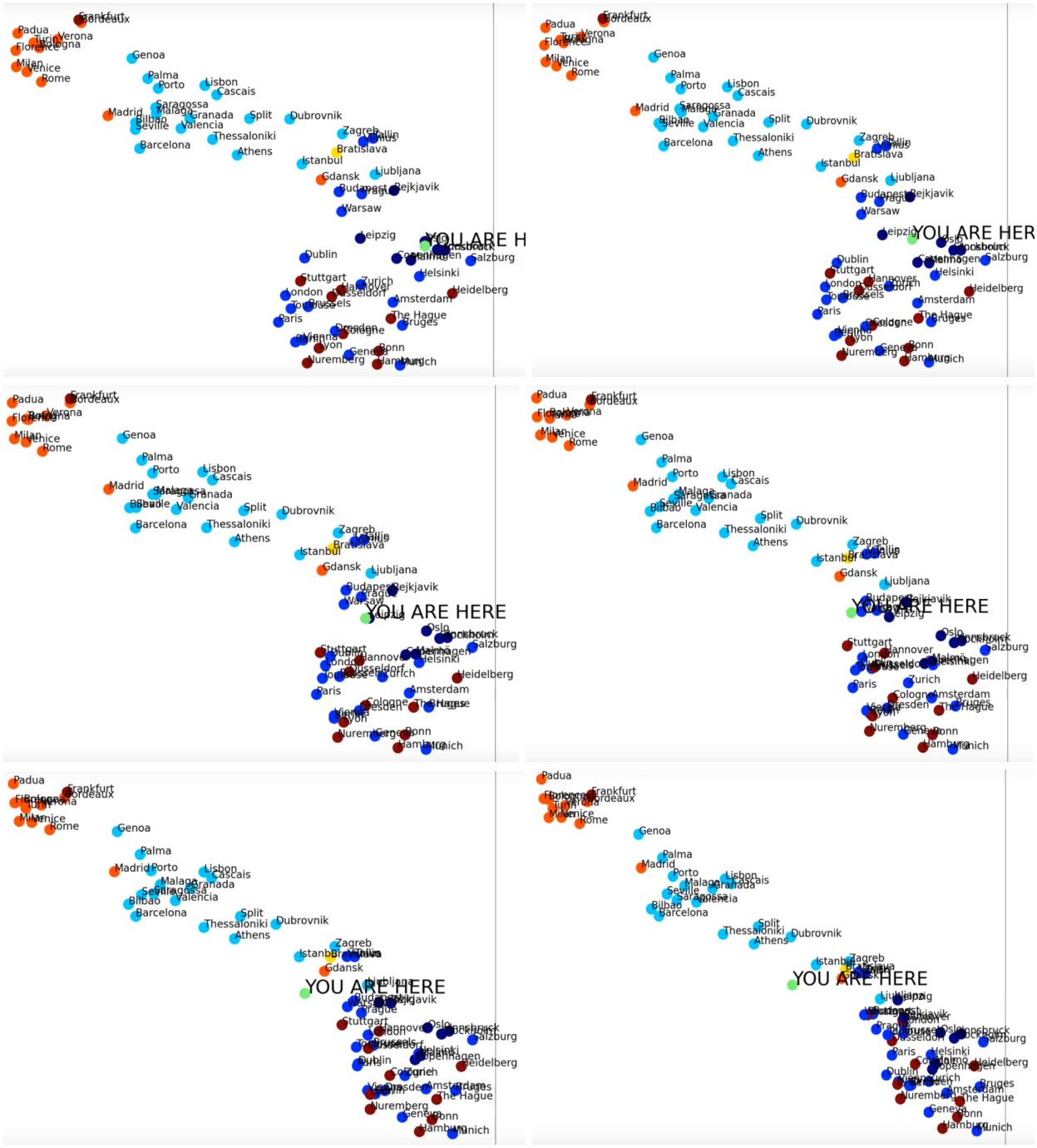
Since the set of cities for this research is limited to Europe only, the goal of a future study could be to embed more European cities as well as countries in order to provide an even better and more thorough overview of tourism offerings that might be useful for tourists all over the world. It is expected that bigger cultural differences will be reflected in the higher diversity of tourism offerings. For now, however, the next chapter will provide an overview of the web-service developed based on the aforementioned results in order to provide an opportunity for a user to select a destination based on his/her specific preferences described in a free-text form.

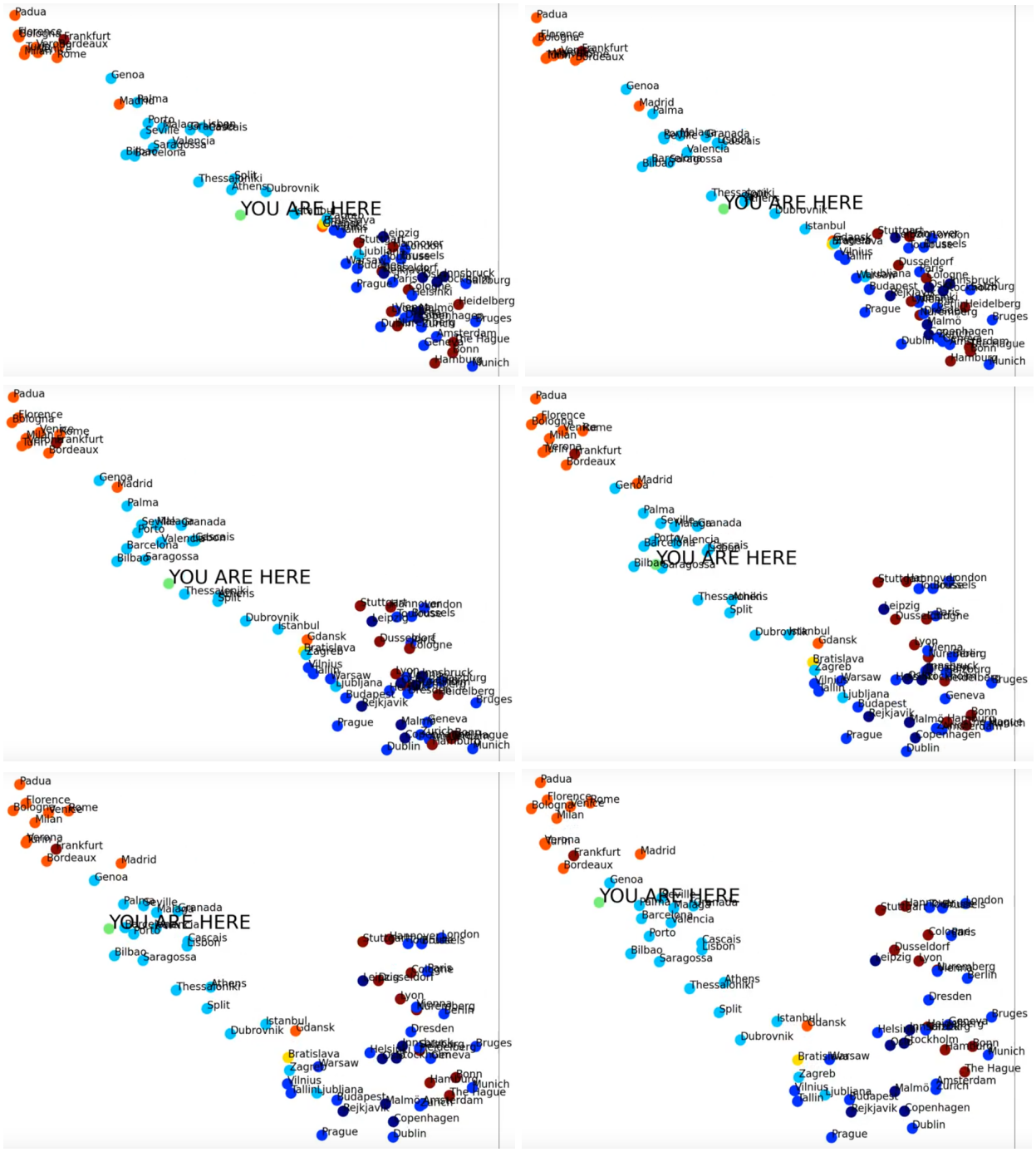
9. Application of TourBERT for a personalized destination recommendation service

In order to demonstrate how the real-world application could benefit from the results described in the previous chapter, a simple web-service was implemented that allows a user to be located on a similarity plot based on his/her specific preferences. This application currently implements a single use-case, where a user can enter a textual description of the activities he/she would like to undertake or experience, and by clicking on the “predict” button, the user’s input will be mapped onto a point on the similarity scatter plot. The resulting mapping can be interpreted as next destination recommendation based on available activities and offerings. As will be illustrated below, the advantage of such a service, and unlike other tourism web-platforms, is the absence of a need to enter a specific location if a user is unsure about his/her next tourism destination (Egger et al., 2007). Instead, the service is supposed to locate a user near the destination point, which is expected to have the most similar offerings and thus covers the user’s specific needs. Therefore, users might save a lot of time without having to look for a suitable offer in a specific location (Alrasheed et al., 2020). Also, by providing a wider range of destinations with similar offerings, a user might concentrate on other criteria for final decision-making, such as price or the distance from their current position to the recommended destination (Cao and Thomas, 2021).

The UI is displayed in Figure 29, comprising of a simple layout, some headers that help to understand what the app is about, an input text field where the user can enter his/her description of services, activities, or ideas of what the vacation should look like, and a “predict” button down below. After the user submits a request, it takes about 10 seconds to process the input on localhost and to rewrite the video file. After the video file is ready, the user is redirected to the homepage, which displays the video of the next recommendation of a place he/she could visit according to the inserted preferences. For example, if a user enters “to visit ski world cup and to walk in lots of snow and do some snow and mountain ski” into the description field, the model will bring the user closer to Innsbruck. When the user enters “to visit some southern country and try different pasta variations and old red wine” next, the model will recommend

visiting a place that is located near Bologna. Figure 28 shows all 15 frames of the video that would be produced if the descriptions were entered in the order described above:




YOU ARE HERE
YOU ARE HERE
YOU ARE HERE
YOU ARE HERE
YOU ARE HERE
YOU ARE HERE

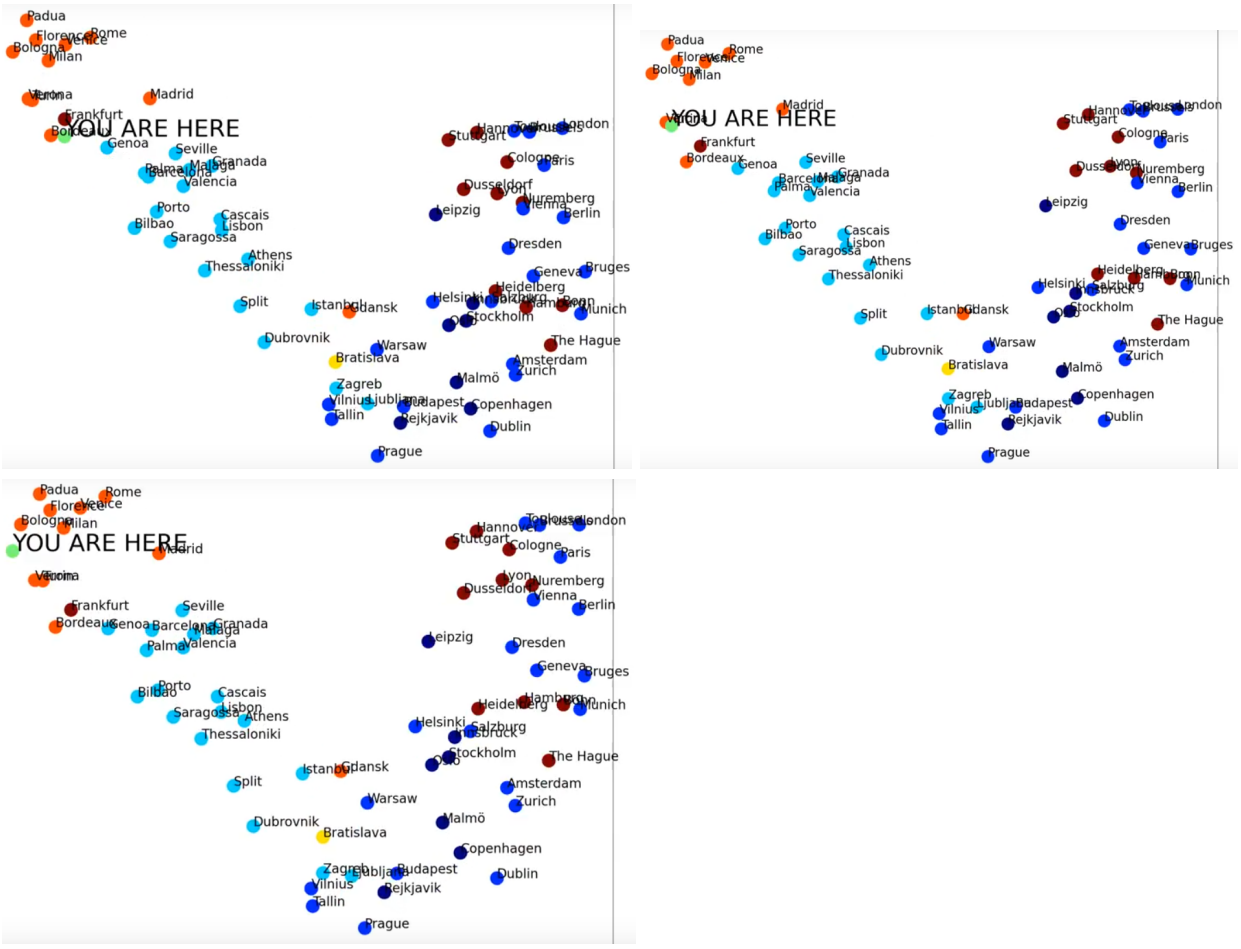


Figure 28: 15 frames of a video showing how points transition from state A to state B.

The pictures above represent an example video broken down into frames and showing how the user's point changes its position from Innsbruck to Bologna using the shortest path possible on the similarity plot. State A refers to a prediction as the user entered an input which was mapped to a location close to Innsbruck, whereas the final state B is located close to Bologna. Important to note is that points' cluster membership does not change after the map rearranges, which means that the visualization and predictions are consistent. The position of the user's point is circled in green on each picture for better perception. Some more videos in different resolutions are available on YouTube[5] [6]. Since, in the example above, the user has received Innsbruck as the first recommendation and Bologna as the second one (after having entered a different

[5] https://drive.google.com/file/d/1dQK-Lp9BRwVozdxPtgcGXR94nf85a4UP/view?usp=sharing
[6] https://drive.google.com/file/d/1m1tJiLfZl5lDtIuddIo_sUzNVnWJuVN9/view?usp=sharing

description for preferred activities), the service displays how the user's point would change position from the previous recommended location to the new one in an interactive manner. Also, because all city points change their locations after re-clustering as well, it makes it convenient for the user to track modified positions in live mode. As mentioned previously, the relative distance between the points rarely change from one recommendation to another; however, absolute positions can change, which is visible from the interactive plot in the video. As one can see in the example video, the movement of the points looks very similar to a rotation of the scatter plot in a three-dimensional space.

Though debiasing the model lies outside the scope of this work, it is still relevant to include initial thoughts on how the model could be debiased in the future since some biases were found during the usage of the model. For example, changing the input from "to visit ski world cup and to walk in lots of snow and do some snow hiking and mountain ski" to "to visit ski world cup places and to walk in the snow, to do some mountain skiing and to do some snow hiking" forces the model to change the position on the similarity plot from Innsbruck, Austria, to Malmö, Sweden. Although both cities offer some ski tours and contain mountainous regions, the points are still located somewhat far from each other on the similarity plot. This can be explained by the fact that during crawling of the experiences from Airbnb, the function of automatic translation was used in cases where the experience was not written in the English language. It is important to remember that the TourBERT model is able to process only English texts since the English BERT-Base model yields better results than the multilingual BERT model. Therefore, assuming that Airbnb uses Google Translate, which is technically supposed to use a BERT-like model, the assumption is that before TourBERT-model bias even comes into play, the model deals with the bias originated from the bias of the translated result. Plus, it is known that online translation services are biased in general (Bartl et al., 2020). From the example described above, it can be seen that only by changing the description a little bit, where the main change merely lies in switching the word order, the result is somewhat different. This might be potentially due to foreign languages often having a different word order, which leads the model to learn different language representations, regardless of the fact that in the real world two expressions from different languages might have almost the same exact meaning.

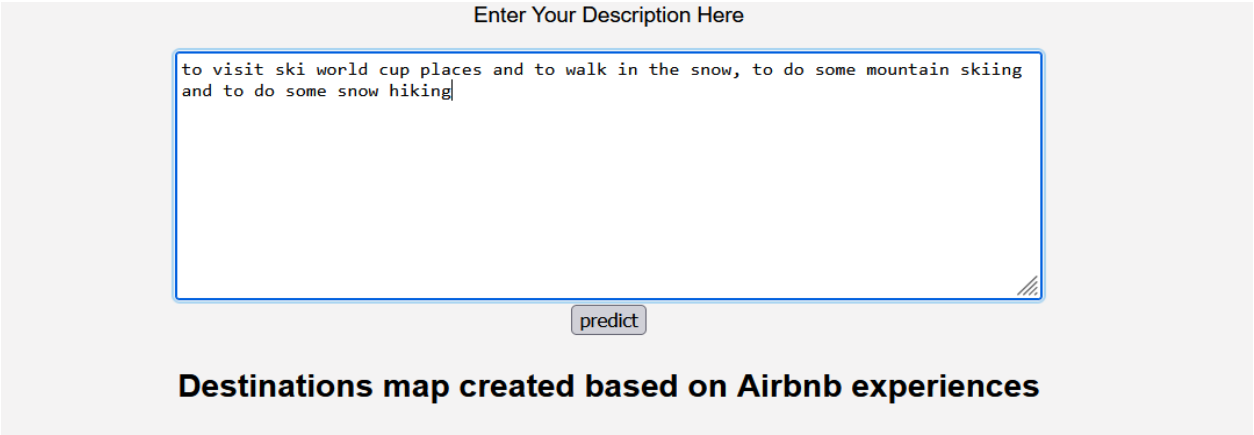


Figure 29: Flask UI for the destination recommendation web-service prototype.

All in all, the service was implemented using Python Flask framework, a RESTful web-service handling user requests as a POST-request. The request is initiated at the time when the user clicks on the “predict” button, sending the field input from the HTML form to the app through a POST-request. The request processing consists of several steps: 1) The input text is sent to the TourBERT model via Hugging Face API to obtain a user vector. 2) That vector is then attached

to a vector matrix, which already contains all the aggregated destination vectors that are clustered with k-Means and down-projected with UMAP. 3) As the final step, a new video is recorded showing how a user's point reaches his/her new recommended place from the previous point. If it is a first-time request, then the video will look like a static image; otherwise, if a user has previously received a prediction, then the old prediction will be the starting point for the next video produced at the time of the second prediction.

From an implementation perspective, the user's input is converted into a 768-dimensional TourBERT vector, which is then assigned to a cluster having the shortest distance between its cluster center and the user's vector in a 768-dimensional space. The point is then down-projected for visualization using the UMAP algorithm. It is important to note that each time the user provides a new piece of input, all of the points, including the new user vector, are down-projected instead of only the new point being down-projected to a low dimensional space. This is due to the fact that UMAP produces spherical clusters, and the projection of a new point exceeds the border of the similarity scatter plot, never reaching any position inside of it. Therefore, the decision was made to down-project all the points "from scratch" every time the user enters a new description. Nonetheless, although down-projected points change their absolute position with each new user input, the relationship between points remains almost unchanged, meaning that when the similarity plot becomes rotated after a new user input, the distance between the points is kept the same so that the visualization remains consistent.

To make the visualization interactive and allow a user to track how his/her position on a similarity map will change when entering a new description, the visualization is represented as a video showing how a user's point moves from the old position to the new one. Owing to the fact that all the destination points are down-projected each time the system receives a new user input, points corresponding to the cities will also change their absolute positions, as will be shown in the video. In order to demonstrate how points navigate during their state transition, a simple algorithm was created that allows for a point to reach destination B from destination A in a fixed number of steps using the shortest path possible from A to B. In this work, a denominator of 10 was used to determine the number of steps required for a single transition from each point. The video length equals three seconds so that the recording is shown, on average, at a speed of about 3-5 frames per second.

The source code for the app is available as a Github repository[7] since a trial to deploy the app with Heroku[8] for public access unfortunately resulted in memory errors due to the app having a size of 594MB and the maximum size limit for Heroku is 500MB. To start the app, one needs to install all the dependencies specified in the requirements.txt file, and run the app from the command line using the "python app.py" command. After the app starts up successfully, it should be available at localhost at http://127.0.0.1:5000/. Furthermore, it is important to modify the absolute path to the ffmpeg.exe binary file (ffmpeg library is used to save the interactive plot in .mp4-format) inside of the app.py file in case the app is run on Windows OS. The app works successfully with Python 3.7.

[7] https://github.com/VeronikaArefeva/tourbert_flaskapp
[8] https://dashboard.heroku.com/

10. Conclusion

This chapter summarizes the results of this thesis in connection with the three research questions described in chapter 1.1 and discusses the limitations of this research as well as gives a preface to any future work. In particular, sub-chapter 10.1 will provide a summary and implications of the results for all three research questions, while sub-chapter 10.2 will focus on the limitations of this thesis. Following that, sub-chapter 10.3 will describe the direction of future work that can be conducted in order to overcome the limitations described in 10.2 as well as to continue further research related to the topics discussed in this paper.

10.1 Summary and implications of the results

The main objective of this thesis is to answer the three research questions. In the following, the findings and results of this thesis in relation to the research questions will be discussed.

(RQ1) Does a natural language model for the tourism domain (TourBERT) capture tourism-specific context better than a general language model (BERT-Base)?

This research question addresses the lack of a language model for the tourism domain, as elaborated on in the introduction. The analysis of existing approaches has shown that, generally, unspecific domain language models were used to solve tourism-specific problems like sentiment analysis, NER, and to create recommender systems. In this work, TourBERT, a domain-specific model specialized for tourism, was pre-trained and benchmarked against the BERT-Base model. The results have demonstrated that TourBERT surpasses BERT-Base in all supervised tasks on tourism-specific datasets. In addition, a user study was conducted, which showed that, on average, users perceived the results obtained from TourBERT better than those from BERT-Base and that the results were statistically significant with a medium-level impact.

One of the main contributions of this thesis in relation to this research question is the release of the TourBERT model to the open-source community. The model is hosted on Hugging Face Model Hub and is accessible through this link: https://huggingface.co/veroman/TourBERT. In the past month, the model has been downloaded 3,481 times, indicating a significant interest from the research community as well as from the tourism industry.

This study will enable many researchers to utilize TourBERT in their studies, potentially leading to the following main implications: First, those who develop NLP-based systems for tourism can expect significant improvement in their results quality since TourBERT has been proven to capture tourism-specific context better than other language models. Moreover, TourBERT will help many researchers to gain deeper insights into their data, thus uncovering more unexpected, interesting patterns that can bring research in tourism to the next level.

From a business perspective, many data scientists working in the tourism industry will be able to benefit from the usage of TourBERT. Applications like recommender systems, intelligent assistants, chat bots, topic modeling, and sentiment analysis tools, amongst others, will improve through this model and continue to deliver state-of-the-art results. The improvement of applications will help managers to gain new USPs and increase their target audience as well as understand business needs in a more robust and comprehensive way.

(RQ2) Which European countries provide similar tourism offers by locals on Airbnb?

This research question addresses the limitations of existing methods on destination similarity analysis. The study of approaches used for destination similarity evaluation revealed that most researchers used this concept as one of the components of their recommender systems. The key factors for establishing a destination profile were either user preferences in the form of a user's previous search history or basic destination characteristics like prices, geographical locations, or available dates.

Despite the growing attention surrounding the destination similarity concept, this study is one of the first to adopt textual features for destination similarity analysis. Using experience descriptions from Airbnb, the most popular European cities could be investigated based on their offerings' similarity. The analysis of the resulting two-dimensional plot compared to the original geographical map of Europe indicated that some groups of cities are as closely related to each other as they are located on the real map. This can be explained through the cultural peculiarities of certain countries that have a strong cultural history, e.g., Italy or Germany.

Another interesting finding is that the resulting groupings of cities with similar activities can be partially attributed to their special landscapes, geographical position, and weather conditions. For example, cities located near the beach/water in southern Europe nearly all belong to the

same group. On the other hand, cities like Malmö or Innsbruck, located in mountainous regions, belong together as well. Apart from that, the biggest capital cities like Paris, Berlin, London, Amsterdam, and Dublin were grouped together, leading one to conclude that the offerings in big cities are similar to one extent or another since they offer general sightseeing, photo shoots, shopping, restaurants, and other tours that are not centered around the specialties of a local culture, unlike the smaller cities, for example.

The contribution of this research in conjunction with the second question has multiple implications for both academia and business communities alike. First and foremost, this study contributes to the growing literature on the experience economy by providing a novel framework for destination similarity analysis. Second, the reusability of the established method will allow many researchers from all over the world to extend this research of destination similarity to other continents. Finally, as shown in the third research question, the framework can be applied as a preliminary component of a destination recommender system. The TourBERT-enhanced features used in this thesis should encourage many researchers to apply this method in order to expand upon the capabilities of their tourism recommender systems.

Business wise, the contribution of this research is that many destination management and advertising companies can profit from these results since it can facilitate both information management and decision-making. By knowing that certain destinations are similar based on offerings provided by locals, many businesses can optimize their advertisement campaigns in such a way that it will help them to gain competitive USPs. Moreover, exploring experience offerings will help destination managers to familiarize themselves with non-professional offerings and identify the gaps in existing strategies for professional tourism offer development.

(RQ3) How can TourBERT help to improve the quality of personalized recommendations?

As was alluded to in the literature review, existing recommender systems in tourism suffer from a major lack of advanced NLP techniques. The majority of them were built with a user's search history data using conventional techniques, but in this work, a novel approach was proposed that is also profitable for future recommender systems in tourism. A web-service prototype was built using the results and artifacts obtained from research questions one and two. The service allows a user to enter his preferences in a free-text form, which will then be converted into a point on the destination similarity plot created for the second research question. Moreover, such

a system simplifies the search for users in that they can avoid having to enter many different attributes and filters, as is usually the case when searching for a destination on tourism platforms.

Additionally, this study is one of the first to establish both user and destination profiles based on textual descriptions in a free form. Whereas the destination profiles are based on Airbnb experiences, the user profiles are based on the textual descriptions of their preferences, which is a novel approach in itself as it is not based on tourist typologies or a predefined set of attributes. The proposed concept for both the destination and user profiles theoretically allows for the incorporation of an arbitrary number of categories and aspects of leisure activities, giving a user the opportunity to find the best suitable match available between their intentions and tourism market offerings.

New definitions of both a destination and user profile described in this work are of greater importance for destination managers as well. The use of the developed destination recommender system prototype can help many businesses to adopt their existing offers or create new tourism offerings based on collected free-text user searches. Expected outcomes of the integration of a TourBERT-based destination recommender framework include an increase in user satisfaction as well as business revenue.

10.2 Limitations

Regarding the first two research questions, this work utilized a set of countries that was limited to European countries and, so, similarity ranking was only provided for the most popular tourist destinations within that area. Moreover, all the datasets used throughout this work were either collected in or translated to the English language; therefore, the TourBERT model is only able to handle English texts. However, it was shown that the pre-training dataset based on the SentencePiece vocabulary allows TourBERT to achieve superior results. As a result, one of the future work directions could be to extend the existing model to multiple languages.

This master’s thesis has other limitations in terms of the third research question. Instead of an end-to-end personalized recommender system for travelers, a web-service prototype was created that allows users to enter their specific preferences in a free-text form and receive a

map where countries are rearranged according to their similarity. It is important to note that the web-service prototype's intended use is to show potential use-cases of the developed solution so that they can be extended to a broader set of functionalities and serve as a component of a personalized destination recommender system. Although the system was not evaluated extensively in the form of a user study, several meaningful examples were revealed, demonstrating the potential of the TourBERT application for destination recommender systems.

10.3 Future work

Considering the results and limitations of the current thesis, future work can be extended in three possible directions to corroborate the three research questions described prior. First, the TourBERT model was pre-trained on English texts only and therefore it is only able to understand the English language. However, social media platforms that act as pre-training data sources provide lots of information in multiple languages, which TourBERT could most certainly benefit from. One goal could therefore be to pre-train a multi-language TourBERT model in the future so that many researchers from different countries can use this model for the data collected in their native language. As previously discussed, the translation of any text from one language to another introduces certain biases that can lead to undesirable results. Hence, constructing a multi-language TourBERT model could instigate great improvements in debiasing language models.

Regarding the second research question, the research of tourist destination similarities was limited only to the most popular European cities. Thus, the expansion of this list would be another adequate extension of this research since more individuals as well as companies could benefit from the comparison of cities all around the globe. Such results could be integrated into a destination recommender system so that many tourists could explore the destination similarity framework and find their next most suitable location to visit. What is more, one other future direction could be to construct an application for the next destination recommendation to be hosted in a public cloud. For example, using Microsoft Azure App service, the destination recommender prototype could be made available to the research community as well as end consumers.

References

Akdag, G. (2011). Assessment of world tourism from a geographical perspective and a comparative view of leading destinations in the market. *Procedia Social and Behavioral Sciences*, 9.

Akdağ, G., & Oter, Z. (2011). Assessment of World Tourism from a Geographical Perspective and a Comparative View of Leading Destinations in the Market. *Procedia - Social and Behavioral Sciences*, *19*, 216–224. https://doi.org/10.1016/j.sbspro.2011.05.126

Alaei, A. R., Becken, S., & Stantic, B. (2019). Sentiment Analysis in Tourism: Capitalizing on Big Data. *Journal of Travel Research*, *58*(2), 175–191. https://doi.org/10.1177/0047287517747753

Alrasheed, H., Alzeer, A., Alhowimel, A., shameri, N., & Althyabi, A. (2020). A Multi-Level Tourism Destination Recommender System. *Procedia Computer Science*, *170*, 333–340. https://doi.org/10.1016/j.procs.2020.03.047

Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Jin, D., Naumann, T., & McDermott, M. B. A. (2019). Publicly Available Clinical BERT Embeddings. *ArXiv:1904.03323 [Cs]*. http://arxiv.org/abs/1904.03323

Araci, D. (2019). FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. *ArXiv:1908.10063 [Cs]*. http://arxiv.org/abs/1908.10063

Arefieva, V., & Egger, R. (2022). TourBERT: A pretrained language model for the tourism industry. *ArXiv:2201.07449 [Cs]*. https://doi.org/10.13140/RG.2.2.18322.99525

Arefieva, V., Egger, R., & Yu, J. (2021). A machine learning approach to cluster destination image on Instagram. *Tourism Management*, *85*, 104318. https://doi.org/10.1016/j.tourman.2021.104318

Arreola, J., Garcia, L., Ramos-Zavaleta, J., & Rodrıguez, A. (n.d.). *An Embeddings Based Recommendation System for Mexican Tourism. Submission to the REST-MEX Shared Task at IberLEF 202*. 8.

Barbiero, P., Ciravegna, G., Giannini, F., Lió, P., Gori, M., & Melacci, S. (2022). *Entropy-based Logic Explanations of Neural Networks* (arXiv:2106.06804). arXiv. http://arxiv.org/abs/2106.06804

Bartl, M., Nissim, M., & Gatt, A. (2020). Unmasking Contextual Stereotypes: Measuring and Mitigating BERT's Gender Bias. *ArXiv:2010.14534 [Cs]*. http://arxiv.org/abs/2010.14534

Bekk, M., Spörrle, M., & Kruse, J. (2016). The Benefits of Similarity between Tourist and Destination Personality. *Journal of Travel Research*, *55*, 1008–1021. https://doi.org/10.1177/0047287515606813

Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A Pretrained Language Model for Scientific Text. *ArXiv:1903.10676 [Cs]*. http://arxiv.org/abs/1903.10676

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *ArXiv:1607.04606 [Cs]*. http://arxiv.org/abs/1607.04606

Bommasani, R., Davis, K., & Cardie, C. (2020). Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4758–4781. https://doi.org/10.18653/v1/2020.acl-main.431

Boukkouri, H., Ferret, O., Lavergne, T., Noji, H., Zweigenbaum, P., & Tsujii, J. (2020). *CharacterBERT: Reconciling ELMo and BERT for Word-Level Open-Vocabulary Representations From Characters*. 6903–6915. https://doi.org/10.18653/v1/2020.coling-main.609

Cao, H., & Thomas, E. (2021). Destination similarity based on implicit user interest. *ArXiv:2102.06687 [Cs]*. http://arxiv.org/abs/2102.06687

Cecilia, S., Molnar, E., & Bunghez, M. (2011). TOURISM'S CHANGING FACE: NEW AGE TOURISM VERSUS OLD TOURISM. *Annals of Faculty of Economics*, *1*, 245–249.

Chang, B., Park, Y., Park, D., Kim, S., & Kang, J. (2018). Content-Aware Hierarchical Point-of-Interest Embedding Model for Successive POI Recommendation. *Proceedings of*

the Twenty-Seventh International Joint Conference on Artificial Intelligence, 3301–3307. https://doi.org/10.24963/ijcai.2018/458

Chang, S. (2018). Experience economy in the hospitality and tourism context. *Tourism Management Perspectives*, *27*, 83–90. https://doi.org/10.1016/j.tmp.2018.05.001

Chantrapornchai, C., & Tunsakul, A. (2019). Information Extraction based on Named Entity for Tourism Corpus. *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 187–192. https://doi.org/10.1109/JCSSE.2019.8864166

Cheng, M. (2016). Sharing economy: A review and agenda for future research. *International Journal of Hospitality Management*, *57*, 60–70. https://doi.org/10.1016/j.ijhm.2016.06.003

Coca-Stefaniak, J. A., Powell, R., Morrison, A. M., & Paulauskaite, D. (2017). Living Like a Local: Authentic Tourism Experiences and the Sharing Economy. *International Journal of Tourism Research*, *19*. https://doi.org/10.1002/jtr.2134

Coccossis, H., & Constantoglou, M. E. (2008). The Use Of Typologies In Tourism Planning: Problems And Conflicts. In H. Coccossis & Y. Psycharis (Eds.), *Regional Analysis and Policy* (pp. 273–295). Physica-Verlag HD. https://doi.org/10.1007/978-3-7908-2086-7_14

Cohen, E. (1979). A Phenomenology of Tourist Experience. *Sociology-the Journal of The British Sociological Association - SOCIOLOGY*, *13*, 179–201. https://doi.org/10.1177/003803857901300203

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.). Lawrence Erlbaum Associates.

Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, & Zhuowen Tu. (2012). Detecting texts of arbitrary orientations in natural images. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 1083–1090. https://doi.org/10.1109/CVPR.2012.6247787

Dalen, E. (1989). *Research into values and consumer trends in Norway*. https://doi.org/10.1016/0261-5177(89)90067-8

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv:1810.04805 [Cs]*. http://arxiv.org/abs/1810.04805

Dietz, L. W., Myftija, S., & Wörndl, W. (2019). *Designing a Conversational Travel Recommender System Based on Data-Driven Destination Characterization*. 5.

Dr. Egger, R., Hörl, J., Jellinek, B., & Jooss, M. (2007). *Virtual Tourism Content Network TANDEM - A Prototype for the Austrian Tourism Industry*. 175–184. https://doi.org/10.1007/978-3-211-69566-1_17

Egger, R. (2022a). Text Representations and Word Embeddings. In R. Egger (Ed.), *Applied Data Science in Tourism: Interdisciplinary Approaches, Methodologies, and Applications* (pp. 335–361). Springer International Publishing. https://doi.org/10.1007/978-3-030-88389-8_16

Egger, R. (2022b). Vectorize me. A machine learning approach to typologyze the postmodern tourist. (unpublished paper).

Ezen-Can, A. (2020). A Comparison of LSTM and BERT for Small Corpus. *ArXiv:2009.05451 [Cs]*. http://arxiv.org/abs/2009.05451

Farmaki, A., & Stergiou, D. P. (2019). Escaping loneliness Through Airbnb host-guest interactions. *Tourism Management*, *74*, 331–333. https://doi.org/10.1016/j.tourman.2019.04.006

Flores-Muñoz, F., Gutiérrez-Barroso, J., & Báez-García, A. J. (2019). Predictability and self-similarity in demand maturity of tourist destinations: The case of Tenerife. *Cuadernos de Economía - Spanish Journal of Economics and Finance*, *42*(118), 59–69.

Haldar, R., & Mukhopadhyay, D. (2011). *Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach* (arXiv:1101.1232). arXiv. https://doi.org/10.48550/arXiv.1101.1232

Han, Q., Zejnilovic, L., & Novais, M. A. (2019). Tourism2vec: An Adaptation of Word2vec to Investigate Tourism Spatio-Temporal Behaviour. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3350125

Hancock, J. (2004). *Jaccard Distance (Jaccard Index, Jaccard Similarity Coefficient)*. https://doi.org/10.1002/9780471650126.dob0956

Hu, Y., Nuo, M., & Tang, C. (2019). A Deep Learning Approach for Chinese Tourism Field Attribute Extraction. *2019 15th International Conference on Computational Intelligence and Security (CIS)*, 108–112. https://doi.org/10.1109/CIS.2019.00031

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling BERT for Natural Language Understanding. *ArXiv:1909.10351 [Cs]*. http://arxiv.org/abs/1909.10351

Karhula, A., Erola, J., Raab, M., & Fasang, A. (2019). Destination as a process: Sibling similarity in early socioeconomic trajectories. *Advances in Life Course Research*, *40*, 85–98. https://doi.org/10.1016/j.alcr.2019.04.015

Ketter, E. (2021). Millennial travel: Tourism micro-trends of European Generation Y. *Journal of Tourism Futures*, *7*(2), 192–196. https://doi.org/10.1108/JTF-10-2019-0106

Khattak, F. K., Jeblee, S., Pou-Prom, C., Abdalla, M., Meaney, C., & Rudzicz, F. (2019). A survey of word embeddings for clinical text. *Journal of Biomedical Informatics*, *100*, 100057. https://doi.org/10.1016/j.yjbinx.2019.100057

Kudo, T. (2018). Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. *ArXiv:1804.10959 [Cs]*. http://arxiv.org/abs/1804.10959

Kudo, T., & Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. *ArXiv:1808.06226 [Cs]*. http://arxiv.org/abs/1808.06226

Lahitani, A. R., Permanasari, A. E., & Setiawan, N. A. (2016). Cosine similarity to determine similarity measure: Study case in online essay assessment. *2016 4th International*

Conference on Cyber and IT Service Management, 1–6. https://doi.org/10.1109/CITSM.2016.7577578

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *ArXiv:1909.11942 [Cs]*. http://arxiv.org/abs/1909.11942

Lavrakas, P. (2008). *Encyclopedia of Survey Research Methods*. Sage Publications, Inc. https://doi.org/10.4135/9781412963947

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2019). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, btz682. https://doi.org/10.1093/bioinformatics/btz682

Li, Q., Li, S., Zhang, S., Hu, J., & Hu, J. (2019). A Review of Text Corpus-Based Tourism Big Data Mining. *Applied Sciences*, *9*, 3300. https://doi.org/10.3390/app9163300

Li, S., Qiu, C., & Jiang, M. (2019). Research on Tourism Destination Attraction Based on Deep Learning. *IOP Conference Series: Materials Science and Engineering*, *646*(1), 012026. https://doi.org/10.1088/1757-899X/646/1/012026

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of Psychology, 22 140,* 55.

Lutz, C., & Newlands, G. (2018). Consumer segmentation within the sharing economy: The case of Airbnb. *Journal of Business Research*, *88*, 187–196. https://doi.org/10.1016/j.jbusres.2018.03.019

McCrae, R. R., & Costa Jr., P. T. (1999). A Five-Factor theory of personality. In *Handbook of personality: Theory and research, 2nd ed* (pp. 139–153). Guilford Press.

McInnes, L., Healy, J., & Melville, J. (2020). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv:1802.03426 [Cs, Stat]*. http://arxiv.org/abs/1802.03426

Mehmetoglu, M., & Engen, M. (2011). Pine and Gilmore's Concept of Experience Economy and Its Dimensions: An Empirical Examination in Tourism. *Journal of Quality Assurance in Hospitality & Tourism*, *12*, 237–255. https://doi.org/10.1080/1528008X.2011.541847

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *ArXiv:1301.3781 [Cs]*. http://arxiv.org/abs/1301.3781

Morgan, M., Elbe, J., & de Esteban Curiel, J. (2009). Has the experience economy arrived? The views of destination managers in three visitor-dependent areas. *International Journal of Tourism Research*, *11*(2), 201–216. https://doi.org/10.1002/jtr.719

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. https://doi.org/10.3115/v1/D14-1162

Perreault, W. D., & And Others. (1977). A Psychographic Classification of Vacation Life Styles. *Journal of Leisure Research*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *ArXiv:1802.05365 [Cs]*. http://arxiv.org/abs/1802.05365

Phan, T., & Do, P. (2022). Developing a BERT based triple classification model using knowledge graph embedding for question answering system. *Applied Intelligence*, *52*. https://doi.org/10.1007/s10489-021-02460-w

Phan, T. H. V., & Do, P. (2020). BERT+vnKG: Using Deep Learning and Knowledge Graph to Improve Vietnamese Question Answering System. *International Journal of Advanced Computer Science and Applications*, *11*(7). https://doi.org/10.14569/IJACSA.2020.0110761

Radford, A., & Narasimhan, K. (2018). Improving Language Understanding by Generative Pre-Training. *Undefined*. https://www.semanticscholar.org/paper/Improving-Language-Understanding-by-Generative-Radford-Narasimhan/cd18800a0fe0b668a1cc19f2ec95b5003d0a5035

Rahmani, H. A., Aliannejadi, M., Mirzaei Zadeh, R., Baratchi, M., Afsharchi, M., & Crestani, F. (2019). *Category-Aware Location Embedding for Point-of-Interest Recommendation*.

Ravi, L., & Vairavasundaram, S. (2016). A Collaborative Location Based Travel Recommendation System through Enhanced Rating Prediction for the Group of Users. *Computational Intelligence and Neuroscience*, *2016*, e1291358. https://doi.org/10.1155/2016/1291358

Ricci, F. (2002). Travel recommender systems. *IEEE Intelligent Systems*. https://www.academia.edu/2888709/Travel_recommender_systems

Ricci, F., & Del Missier, F. (2004). Supporting Travel Decision Making Through Personalized Recommendation. In C.-M. Karat, J. O. Blom, & J. Karat (Eds.), *Designing Personalized User Experiences in eCommerce* (Vol. 5, pp. 231–251). Springer Netherlands. https://doi.org/10.1007/1-4020-2148-8_13

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to Recommender Systems Handbook. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender Systems Handbook* (pp. 1–35). Springer US. https://doi.org/10.1007/978-0-387-85820-3_1

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A Primer in BERTology: What we know about how BERT works. *ArXiv:2002.12327 [Cs]*. http://arxiv.org/abs/2002.12327

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *ArXiv:1910.01108 [Cs]*. http://arxiv.org/abs/1910.01108

Sayers, E. W., Beck, J., Brister, J. R., Bolton, E. E., Canese, K., Comeau, D. C., Funk, K., Ketter, A., Kim, S., Kimchi, A., Kitts, P. A., Kuznetsov, A., Lathrop, S., Lu, Z., McGarvey, K., Madden, T. L., Murphy, T. D., O'Leary, N., Phan, L., … Ostell, J. (2020). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*, *48*(D1), D9–D16. https://doi.org/10.1093/nar/gkz899

Schafer, B., J, B., Frankowski, D., Dan, Herlocker, Jon, Shilad, & Sen, S. (2007, January 1). *Collaborative Filtering Recommender Systems*.

Schuster, M., & Nakajima, K. (2012). *Japanese and Korean voice search*. 5149–5152. https://doi.org/10.1109/ICASSP.2012.6289079

Senel, L. K., Utlu, I., Yucesoy, V., Koc, A., & Cukur, T. (2018). Semantic Structure and Interpretability of Word Embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(10), 1769–1779. https://doi.org/10.1109/TASLP.2018.2837384

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. *ArXiv:1508.07909 [Cs]*. http://arxiv.org/abs/1508.07909

Shaikh, A., & Kulkarni, S. B. (2020). Natural Language Processing Applications for Tourism Sector. *IOSR Journal of Computer Engineering*, *22*(1), 27–35.

Shin, H.-C., Zhang, Y., Bakhturina, E., Puri, R., Patwary, M., Shoeybi, M., & Mani, R. (2020). BioMegatron: Larger Biomedical Domain Language Model. *ArXiv:2010.06060 [Cs]*. http://arxiv.org/abs/2010.06060

Sievert, C., & Shirley, K. (2014, June 26). *LDAvis: A method for visualizing and interpreting topics*. https://doi.org/10.13140/2.1.1394.3043

Siregar, A. H., & Chahyati, D. (2020). Visual Question Answering for Monas Tourism Object using Deep Learning. *2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 381–386. https://doi.org/10.1109/ICACSIS51025.2020.9263149

Smith, V. L. (1989). *Hosts and guests: The anthropology of tourism*. University of Pennsylvania Press.

Stevens, S., & Su, Y. (2021). An Investigation of Language Model Interpretability via Sentence Editing. *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 435–446. https://doi.org/10.18653/v1/2021.blackboxnlp-1.34

Sthapit, E., & Jiménez-Barreto, J. (2018). Exploring tourists' memorable hospitality experiences: An Airbnb perspective. *Tourism Management Perspectives*, *28*, 83–92. https://doi.org/10.1016/j.tmp.2018.08.006

Swarbrooke, J., & Horner, S. (2006). Consumer behaviour in tourism: Second edition. *Consumer Behaviour in Tourism: Second Edition*, 1–428. https://doi.org/10.4324/9780080466958

Tasan-Kok, T. (2004). *Budapest, Istanbul, and Warsaw: Institutional and spatial change*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *ArXiv:1706.03762 [Cs]*. http://arxiv.org/abs/1706.03762

Xiao, D., Wang, N., Yu, J., Zhang, C., & Wu, J. (2020). A Practice of Tourism Knowledge Graph Construction Based on Heterogeneous Information. In M. Sun, S. Li, Y. Zhang, Y. Liu, S. He, & G. Rao (Eds.), *Chinese Computational Linguistics* (Vol. 12522, pp. 159–173). Springer International Publishing. https://doi.org/10.1007/978-3-030-63031-7_12

Xue, L., Cao, H., Ye, F., & Qin, Y. (2019). A Method of Chinese Tourism Named Entity Recognition Based on BBLC Model. *2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 1722–1727. https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00307

Yanuar, M. R., & Shiramatsu, S. (2020). Aspect Extraction for Tourist Spot Review in Indonesian Language using BERT. *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 298–302. https://doi.org/10.1109/ICAIIC48513.2020.9065263

Zhang, Q., Lu, J., & Jin, Y. (2021). Artificial intelligence in recommender systems. *Complex & Intelligent Systems*, *7*(1), 439–457. https://doi.org/10.1007/s40747-020-00212-w

Zhang, W., Cao, H., Hao, F., Yang, L., Sherwani, M., & Li, Y. (2019). *The Chinese Knowledge Graph on Domain-Tourism* (pp. 20–27). https://doi.org/10.1007/978-981-32-9244-4_3

Zhong, J., Yi, X., Wang, J., Shao, Z., Wang, P., & Lin, S. (2018). *Deep Learning Based Data Governance for Chinese Electronic Health Record Analysis*. 91–103. https://doi.org/10.5121/csit.2018.80507

Zhuang, Y., & Kim, J. (2021). A BERT-Based Multi-Criteria Recommender System for Hotel Promotion Management. *Sustainability*, *13*(14), 8039. https://doi.org/10.3390/su13148039