Technische Universität München
TUM School of Engineering and Design

Fusion of Remote Sensing Images and Social Media Text Messages for Building Function Classification

Matthias Häberle

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz:
Prof. Dr. Martin Werner

Prüfer*Innen der Dissertation:

1. Prof. Dr.-Ing. habil. Xiaoxiang Zhu
2. Prof. Devis Tuia, Ph.D.
3. Guest Prof. Dr. Mrinalini Kochupillai

Die Dissertation wurde am 26.01.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 06.09.2022 angenommen.

Abstract

Detailed knowledge about urban land use, especially building functions, is beneficial to governments for planning resource demands like electricity consumption or infrastructure. Every building in a city has different requirements according to the infrastructure. Commercial buildings consume more energy and produce more waste. On the other hand, residential buildings consume less energy, but a residential settlement could require a hospital or schools. Since people are still migrating from rural areas to cities and perhaps to informal settlements, governments can face unclear situations over the urban land use and the resulting demands.

Remote sensing is the standard way of classifying urban land use. However, satellite imagery could suffer from coarse image resolution or clouds such that subtle changes on the ground might be undetected. Therefore, on-site information could be beneficial to gather more data about the situation on the ground. What if citizens themselves could deliver information as sensors from the ground using georeferenced Twitter data? Twitter data has been used before to classify buildings, however, mostly at a block level. In a fast and heterogeneously growing urban environment, building functions at a block level might be too coarse. Therefore, the main focus in this work is the classification of building functions at the level of individual buildings.

In this dissertation, georeferenced Twitter text messages obtained from 42 cities across the globe are utilized to develop a classification process that is not restricted to a specific region. The data is applied as citizen (in situ) sensors to classify buildings at an individual building level into "commercial", "residential", and "other". For text classification, state-of-the-art natural language processing methods such as fastText or BERT are used. Since modern (mega) cities are multicultural and therefore multilingual spaces, not only English tweets are considered to classify the building functions. To cover the actual language situation in the cities, a multilingual fastText word embedding has been trained on 14M multilingual tweets to have a baseline against the multilingual variant of BERT pretrained on more than 100 languages.

To combine the strength of the remote sensing birds-eye-view with the potential of in-situ sensors, a straightforward data decision fusion method is applied to further improve the results. For this, three computer vision models, VGG16, InceptionV3, and ResNet50, have been trained on high-resolution Google aerial images of the 42 cities.

The text classification results show that the monolingual BERT model can achieve accuracies up to 59%. The multilingual BERT model reaches a mean accuracy of 56% and can outperform the LSTM trained with word vectors obtained from the self-trained multilingual embedding achieving a mean accuracy of 55%. After fusing the decisions of the best monolingual text model (BERT) with the best vision model (VGG16, 71% accuracy), an accuracy of 75% can be reached. For the multilingual setting, the fusion of the best text model with the vision model (73% accuracy) yielded 75% accuracy. The remote sensing and text features seem complementary, and data fusion can improve the results. The approach proposed in this dissertation is straightforward, resource-saving, easy to replicate, and yields reasonable results that can be built upon.

Zusammenfassung

Detailliertes Wissen über die urbane Landnutzung, vor allem Gebäudefunktionen, für Regierungen, beispielsweise für die Ressourcenplanung nützlich. Jedes einzelne Gebäude einer Stadt stellt verschiedene Anforderungen an die Infrastruktur. Beispielsweise weisen kommerziell genutzte Gebäude einen höheren Energiebedarf auf und erzeugen mehr Abfall. Demgegenüber stehen Wohnsiedlungen, die weniger Energie verbrauchen, aber Krankenhäuser oder Schulen benötigen. Da Menschen noch immer von ländlichen Gegenden in Städte migrieren und dort vielleicht in eher informelle Siedlungen ziehen, sehen sich Regierungen mit einer unklaren urbanen Landnutzungslage und resultierendem Infrastrukturbedarf konfrontiert.

Fernerkundung ist der herkömmliche Weg, um urbane Landnutzung zu detektieren. Allerdings sind Satellitenbilder teilweise zu grob aufgelöst und können mit Wolken überdeckt sein, sodass detaillierte Änderungen am Boden nicht beobachtet werden können. Aufgrund dessen können Information, die direkt vor Ort erhoben werden, nützlich sein, um die Situation besser zu beleuchten. Was wäre, wenn die Bewohnerinnen und Bewohner der Städte selbst diese Informationen mit geo-referenzierten Twitter Daten liefern könnten? Zwar wurden Twitter Daten schon zuvor für die Gebäudeklassifizierung benutzt, allerdings fast ausnahmslos auf Häuserblockebene. In einem schnell und heterogen wachsenden urbanen Umfeld sind Informationen auf Häuserblockebene möglicherweise zu niedrig aufgelöst. Aus diesem Grund liegt das Hauptaugenmerk dieser Arbeit auf der Klassifizierung von Gebäudetypen auf einer individuellen Gebäudeebene.

In dieser Dissertation werden geo-referenzierte Tweets aus 42 globalen Städten gesammelt um einen Prozess zu entwickeln, der nicht auf eine bestimmte Region begrenzt ist. Die Daten werden als in situ-Bevölkerungssensor (citizen-sensor) verwendet, um einzelne Gebäude in Geschäftsgebäude, Wohngebäude, oder weiteres Gebäude zu klassifizieren. Für die Klassifizierung des Textes werden lege artis Methoden des Natural Language Processing, wie fastText oder BERT, eingesetzt. Da moderne (Mega-) Städte multikulturelle und mehrsprachige Orte sind, berücksichtigt die vorliegende Arbeit für die Gebäudeklassifizierung nicht nur englische Tweets. Um jene realen sprachlichen Gegebenheiten in den Städten zu berücksichtigen, wird eine mehrsprachige fastText Text Embedding mit 14 Millionen Tweets trainiert, um eine Ausgangsgrundlage für den Vergleich mit der auf mehr als 100 Sprachen vortrainierten BERT-Variante zu erstellen.

Damit die Stärke der Vogelperspektive der Fernerkundung und das Potenzial der in situ-Sensoren voll ausgeschöpft werden kann, wird eine unkomplizierte Fusion der Daten auf Entscheidungsebene angewandt, welche die Ergebnisse weiter verbessern soll. Dafür werden die drei Bildverarbeitungsmodelle VGG16, InceptionV3 und ResNet50 mit hochauflösenden Google Maps-Luftbildern der 42 Städte trainiert.

Die Ergebnisse der Textklassifizierung zeigen, dass das englischsprachige BERT-Modell mit einer Genauigkeit bis zu 59% am besten abschneidet. Das mehrsprachige BERT-Modell liefert mit einer durchschnittlichen Genauigkeit von 56% etwas bessere Ergebnisse als ein LSTM, das mit Wortvektoren der multilingualen fastText Embedding trainiert wird (durchschnittlich 55%). Nach der Entscheidungsfusion des besten einsprachigen Textmodells (BERT) und Bildmodells (VGG16, 71% Genauigkeit), kann eine Gesamtgenauigkeit von 75% erreicht werden. Für die mehrsprachige Umgebung erreichte die Fusionierung des besten Textmodells (LSTM und mehrsprachige fastText Embedding) mit dem besten Bildmodell (VGG16, 73% Genauigkeit) ebenfalls eine Gesamtgenauigkeit von 75%. Fernerkundungs- und Textmerkmale scheinen also komplementär und so kann die Fusionierung der Modalitäten das Gesamtergebnis der Gebäudeklassifikation verbessern. Das in dieser Arbeit angewandte Vorgehen ist unkompliziert, ressourcenschonend und einfach zu reproduzieren. Dennoch werden angemessene Ergebnisse erreicht, auf denen sich in zukünftiger Forschung aufbauen lässt.

Acknowledgments

The first person I would like to thank is Professor Xiaoxiang Zhu, who allowed me to dive into the world of urban remote sensing and to pursue my dream of achieving a Ph.D. Second, I am grateful for having Hannes Taubenböck as a mentor. He always had the right advice at the right time. I would like to thank Prof. Devis Tuia and Prof. Mrinalini Kochupillai, who agreed to examine my thesis. Additionally, I would like to thank Anna Kruspe and Karam Abdulahhad for their valuable advice regarding my Ph.D. and my research in general. Also, I am grateful to the Munich Aerospace team, who also provided great support during the dissertation phase.

Of course, I would like to thank Prof. Martin Werner. I had a lot of passionate and loud discussions about politics and the world with him, which made me think out of the box.

Also, I would like to thank Eike Jens Hoffmann. He had for almost every programming issue and computer problem a quick solution. He was also a patient teammate who helped wherever and whenever he could despite his packed schedule. I am deeply grateful for your help!

And I will not forget my other amazing DLR colleagues, Homa (best office mate ever), Sina, Eike (again), Erling, Gerald, and Lukas. We became friends and shared many great memories, jokes, movie quotes, football comments, discussions about politics. Traveling to Japan with you was unquestionably a “dear diary moment” I won’t forget. I would like to thank Erling, Homa, Sina, and Lukas, who were brave traveling buddies exploring Kyoto, Hiroshima, and Tokyo with me. A big thank you also goes to Gerald Baier, who showed us the most exciting places in Tokyo and let us crash in his apartment.

Furthermore, there are people in my life who are not directly attached to my work as a Ph.D. student but still had or have an impact (without specific order): Bady, Jochen, Yvonne, Nicole, Nadine, Stefan, Mady, Laura, Sophia, Julia, Yvonne M., Daniel Prince, Lisa, Kai, Sven, Jan, Inda, Tabea, Anja Weber, and Yasemin. I would like to thank the Sammet family with my Gotti, Onkel Dirk, Yvonne with Christian, Oliver with Marianne, Maximilian, and Julian. We had a lot of fun, discussions about football, the world, and the Riester Rente. I also would like to thank the Kroll and Schobel family with Tante Rosmarie, Onkel Waldemar, Sandra, and Janine, Tante Moni, Gette, Mark with Lisa, and Sven.

Without a doubt, I have to thank my wonderful girlfriend, Mara. She always had my back in this stressful period of my life. Thank you for your love, patience, support, and the joy you bring into my life. “Hallo”.

Finally, I would like to express my warmest gratitude to my parents, Ingrid and Rolf Häberle. You always supported me, and nothing of this would have been possible without you!

Contents

List of Figures

List of Tables

Acronyms

API Application Programming Interface

BERT Bidirectional Encoder Representations from Transformers

BoW Bag-of-Words

CNN Convolutional Neural Network

GIS Geographic Information System

IR information retrieval

LSTM Long short-term Memory

NER Named Entity Recognition

NLP Natural Language Processing

OOV Ouf-of-Vocabulary

OSM OpenStreetMap

RNN Recurrent Neural Network

SDG Sustainable Development Goal

SDGs Sustainable Development Goals

SVD Single Value Decomposition

TF-IDF Term Frequency-Inverse Document Frequency

TUM Technical University of Munich

U.N. United Nations

VGI Volunteered Geographic Information

1 Introduction

Thousands of years ago, humankind started to become sedentary and began living in small settlements. Even today, in the age of the Anthropocene, we still retain at least two facts in common with our ancestors. First, we still reside in buildings, and second, we still live in settlements that grow over decades, centuries, and millennia. The urge to migrate from rural areas into cities to find jobs and happiness persists. In 2050, the United Nations (U.N.) expects that 70% of the world's population will live in cities [5]. More and more people are seeking positive economic perspectives in cities. This reality brings new challenges to city planning and municipal governments, such as energy consumption [6], waste management [7], and sanitary facilities. As an example for the latter aspect, the lack of such amenities can result in dangerous situations, especially for female slum residents [8]. Therefore, the U.N. suggests in Sustainable Development Goal (SDG) number 11: "Make cities and human settlements inclusive, safe, resilient and sustainable." To achieve such goals, broad knowledge about a local urban configuration is desirable and inevitable. The correlation between well-being and the built-environment is eminent [9], hence not only demographic information is essential for local decision-makers but also detailed facts about constructions (e.g., a building). Some countries are maintaining registers, e.g., cadastre, for buildings or real estate, et cetera, and are including their functions or land use. In fast-growing city districts with a high migration rate, such registers could be outdated or not even present [10, 11]. However, knowing the function of an individual building could lead to more demand-orientated city planning and so working towards fulfilling the U.N. Sustainable Development Goals [12].

For the reasons mentioned above, the current dissertation follows the building function classification task *not* at a block or field level but at an individual building instance level. In this work, a *building function* is defined as the primary purpose of a building. For example, if a building accommodates a supermarket only, it is used commercially. On the other hand, a residential building is composed of apartments or is a one-family dwelling. In urban areas, a wide range of building functions are available. It turns out that for New York City, for example, commercial and residential buildings are the major categories with the highest occupation of space [13]. In this work, the functions have therefore been summarized to *commercial*, *residential*, and *other*. The *other* class summons additional building functions like religious or civic structures. The task itself, i.e., the *building function classification task*, describes the identification of the primary function of an *individiual* building located in an urban area.

As pointed out by [14], the traditional approach to collect information about urban configurations and land use is the exploitation of data generated by *ex situ* sources like spaceborne or airborne sensors and the use of machine learning algorithms to extract the critical information, e.g., [15]. However, classifying individual building functions from remote sensing imagery is complex. Spectral signatures of materials or shapes of the buildings are not always providing clear information about the usage of a particular building [16]. Therefore, additional information obtained from different sensors could support building function classification.

Over a decade ago, [17] proposed that citizens voluntarily could collect data as a *citizen sensor* and, in doing so, contribute to science. Goodchild's idea has been considered in many studies, and some of the internet users became *in situ* sensors. For example, OpenStreetMap (OSM) is a prominent example of the citizen sensor idea. OSM is a Geographic Information System (GIS) where users can collect and explicitly add geographical information about their neighborhoods. Researchers can query this citizen sensor map and utilize all of the information for their work. A popular data source used as a citizen sensor is Twitter. Twitter is a micro-blogging platform where users can publish a short, mostly informally written, text enriched with web URLs, mentions of other Twitter users, or emojis, which is then called a *tweet*. Some of the users add geographical information to a tweet to tag a place, landmark, or restaurant. Therefore, Twitter data was used in numerous spatial-related studies like traffic patterns [18] or demographic analysis [19] in the past. Hence, Twitter data has great potential to impact building function classification positively. Twitter data has been used before to classify the function of buildings. The linguistic features derived from classical information retrieval approaches like Term Frequency-Inverse Document Frequency (TF-IDF) have been utilized to categorize the building functions [20].

Since the pool of big space data seems to be vastly growing[1], deep learning [21, 22, 23] has prevailed as a ubiquitous tool in remote sensing [24, 25, 26]. An extensive research community is utilizing the vast amount of optical data acquired by various spaceborne sensors to explore urban configurations like urban land use [11, 27, 28, 14]. On the other hand, the utilization of more powerful Natural Language Processing (NLP) methods like (multilingual) word embeddings or large language models were not employed for building function classification even though they outperform the classic approaches in text classification with Twitter data [29].

Additionally, another non-build attribute of urban structures is the aspect of multilingualism [30, 31, 32, 33]. Migration not only brings people into cities but also imports a large variety of new dialects and languages. As might be expected, English is the *lingua franca* of Twitter [34]. However, a significant amount of tweets are written in several languages [35] and additionally, translingual constructs are embedded in tweets as well [36]. Considering all this, investigating approaches handling multilingual text input next to a monolingual one is natural. Studies from the past mainly looked into monolingual texts obtained from block-level resolution or a small area of interest with a limited amount of tweets, e.g., [20, 37, 38, 39].

For this reason, this dissertation investigates multidisciplinary approaches to fill this research gap for building function classification using linguistic features. Namely, taking a more extensive area covering 42 cities distributed over all continents and considering

[1] `https://www.dlr.de/eoc/en/desktopdefault.aspx/tabid-12632/22039_read-51751` [6.9.2021, 14:12]

the aspect of the diverse pool of languages in the cities. For this, NLP methods like word embedding algorithms and large neural language models are used to handle the vast amount of multilingual text accumulated from the 42 cities and generate meaningful feature vectors which are fed in neural networks for classification.

The results based on the text classification are then used as in situ sensors, i.e., citizen sensors, to support the remote sensing image classification. This approach could also tackle the issue brought up by the U.N. that urban configurations underlie rapid changes [40, 41]. The function of a building can change over time. Here, the nature of social media–topicality–is advantageous. In contrast, a satellite can only see the roof of a building–a disadvantage of remote sensing. A roof does not change when the function of a building changes. Therefore, the text messages from the ground could reveal more about the building's function and complement the information acquired by remote sensing. For the image classification, high-resolution Google aerial images of the 42 cities are used and classified with state-of-the-art computer vision deep learning architectures. The combination, i.e., fusion, of data obtained from different sensors is a widely accepted methodology in remote sensing [42, 2, 43].

1.1 Research Questions

The aim of this dissertation can be spelled out in the following research questions (R.Q.):

1. Can linguistic features derived from social media text messages such as Twitter contribute to building function classification at an individual building level?
2. How can the multilingual reality in urban areas be represented best so that they are beneficial for building function classification?
3. What is the effect of unbalanced and balanced datasets on building function classification?
4. Finally, are visual features derived from very high-resolution remote sensing images complementary to linguistic features?

1.2 Thesis Outline

The outline of the thesis is arranged as follows: the subsequent chapter 2 provides a more in-depth view on the techniques used in this work and is followed by chapter 3 where a more detailed discussion about state-of-the-art technology is given. In chapter 4 more insights about the area of interest and the datasets are shown. Furthermore, the status of geo-referenced tweets is discussed. Chapter 5 summarizes the results of the text classification which is then followed by chapter 6 where the employed modalities are fused. Finally, chapter 7 rounds off the current dissertation by discussing the results and looks into possible directions for future research.

2 Fundamentals

In this chapter, the techniques applied in this work are explained. First, the foundations of urban remote sensing and data fusion are discussed. An overview of deep learning methods follows, which briefly introduces architectures utilized in computer vision and Natural Language Processing (NLP). After that, the field of NLP, which might be rather unorthodox in the remote sensing community, is introduced. Fundamental technologies like the classic Term Frequency-Inverse Document Frequency approach, which is used as a method to produce a baseline for the text classification part, are explained. Since NLP is one of the core research fields in this dissertation, the introduction will be more extensive and detailed. Finally, the chapter will be concluded with a brief discussion.

2.1 Urban Remote Sensing

The dream to observe the earth from far above like a bird is as old as humankind. Today, that dream has not only come true but is in high-definition. We can not only watch the earth from a bird's-eye-view but from a spaceborne view as well. Since the first Sputnik satellite in 1957, remote sensing technology has evolved tremendously. Now, petabytes of earth-observation from spaceborne sensors like TanDEM-X[1], TerraSAR-X[2], the Sentinel missions[3], Landsat[4], Quickbird[5], or Planet[6] data are available[7]. The data is accessible in different spatial, spectral, and temporal resolutions.

In remote sensing, two kinds of sensors exist to capture the electromagnetic waves carrying information: active and passive sensors (cf. figure 2.1). Active sensors emit artificial light sources, like synthetic aperture radar (SAR), for registering an image [44, 45]. For further information about radar sensors, consider reading [46]. Passive sensors use natural light, e.g., sunlight, to acquire an image. They are able to detect visible, infrared, and thermal infrared bands of electromagnetic spectrum. The sensors, i.e., the radiometers, measure the sunlight reflected by the earth's surface [45]. Depending on the ground material, the signature of the reflected waves changes and differ in strength. Snow reflects the sunlight more than 90%, but water, on the other hand, only 6% [1].

[1] `https://tandemx-science.dlr.de/` [8.10.2021, 11:11]

[2] `https://www.dlr.de/content/en/missions/terrasar-x.html` [8.10.2021, 12:30]

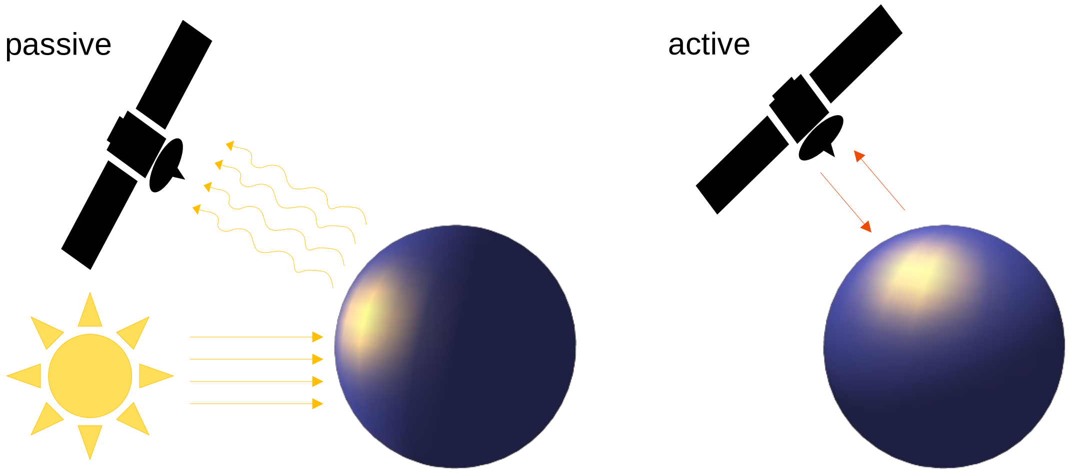
[3] `https://www.esa.int/Applications/Observing_the_Earth/Copernicus/The_Sentinel_missions` [8.10.2021, 11:12]

[4] `https://landsat.gsfc.nasa.gov/` [7.10.2021, 18:10]

[5] `https://earth.esa.int/eogateway/missions/quickbird-2` [7.10.2021, 18:13]

[6] `https://www.planet.com/` [7.10.2021, 18:14]

[7] `https://www.dlr.de/eoc/en/desktopdefault.aspx/tabid-12632/22039_read-51751` [6.9.2021, 14:12]



Satellite icons: Maxxl², CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0, via Wikimedia Commons (modified M. Häberle)

Figure 2.1: Difference of active and passive sensors. Image inspired by NASA Earthdata [1]

The differences between the reflected waves can be measured in different resolutions. Multispectral sensors, for example, can detect 3–10 bands, whereas hyperspectral sensors can measure even more bands (mostly airborne sensors) [1]. Of course, the spatial resolution can also vary. Optical images are composed of a pixel matrix where every pixel can store values up to 8 bits (radiometric resolution). One pixel can therefore store the energy levels of the light up to 256 digits (0–255) [1]. The spatial resolution also depends on how fine-grained a sensor is of capably storing the information of the recorded scene [1]. A pixel can represent a square of 100 × 100 meters, or, like Landsat 8, 30 × 30 meters [45]. More detailed imagery is possible. The GeoEye-1 satellite can register images in a panchromatic resolution of 0.41 meters and color images with 1.65 meters. Together with IKONOS and SPOT-5, Geo-Eye-1 delivers imagery for Google Earth [45].

Remote sensing data is used in a range of research areas and applications. For example, glacier monitoring [47], detection of invasive plants in rainforests [48], the impact of climate change on ocean productivity [49], disaster mapping [50], change detection [51], and crop identification [52].

Instead of observing the physical properties of the earth, urban remote sensing [41] focuses on settlement structures to monitor urban change [53], explore settlement morphology [54] and assess the usage of urban space, i.e., the land-use [44]. In great detail, urban remote sensing can help to discover and to analyze the human footprint on earth: where are settlements built [55]? How many people are living there [56]? What type of urban structure does the detected urban configuration represent [11]? Are the buildings in the area of interest more commercially used, or is it a residential area [27]?

All this information can help identify socio-economical patterns and augment the knowledge of the space we live in. Thus, data about urban agglomerations can contribute to solving the sustainability goals of the U.N. [12]. As stated in chapter 1, a key challenge today are the migration patterns into (expanding) cities. To quote the U.N. target 11.3 of SDG 11: "By 2030, enhance inclusive and sustainable urbanization and capacity for participatory, integrated and sustainable human settlement planning and management in all countries"[8].

But could it also be possible to tap new information to augment the pool of urban geospatial data? [57] elaborates that

> Geospatial information and EO, together with modern data processing and big data analytics, offer unprecedented opportunities to modernise national statistical systems and consequently to make a quantum leap in the capacities of countries to efficiently track all facets of sustainable development.
>
> EO (from satellite, airborne and in-situ sensors) provide accurate and reliable information on the state of the atmosphere, oceans, coasts, rivers, soil, crops, forests, ecosystems, natural resources, ice, snow and built infrastructure, as well as their change over time. These observations are directly or indirectly necessary for all functions of government, all economic sectors and many day-to-day activities of society. (p. 11)

Additionally, [12] point out that the usage of big data could be a way to close gaps in knowledge. More and more cities are implementing open data initiatives where geospatial data related to the cities are published and freely accessible. Los Angeles[9] and San Francisco[10], for example, provide cadastre data with parcel land-use labels. But what can researchers do if no official data is available [10]? As mentioned in the quote above, modern urban remote sensing should therefore integrate not only data from remote sensing sensors but also information gained from in situ sensors, i.e., sensors on-site. [12, p. 464] indicate, by citing the papers of [58] and [59], "that new geospatial information for sustainability (e.g. on the built environment, land use and management), could be derived from the integration of traditional EO approaches to data gathering with citizen science, crowd-sourcing, social sensing, big data analytics and the Internet of Things". [17] proposed the idea of the *citizen sensor*, i.e., social sensing, where citizens act as sensors to gather data about their direct environment.

Thus, the usage of in situ sensors such as (big) data gained from location-based social media could be analyzed and added to the toolbox of urban remote sensing [60]. Exploring this citizen sensor data about our immediate environment could help facilitate and achieve the goal of better citizen participation and more sustainable city planning.

Remote sensing images must be analyzed to discover the latter mentioned patterns or structures. Currently, classic machine learning algorithms like random forest models are used to analyze the images, e.g., [61]. However, as mentioned above, the amount of data is

[8] `https://sdgs.un.org/goals/goal11` [7.10.2021, 18:45]

[9] `https://data.lacity.org/` [7.10.2021, 16:32]

[10] `https://datasf.org/opendata/` [7.10.2021, 16:33]

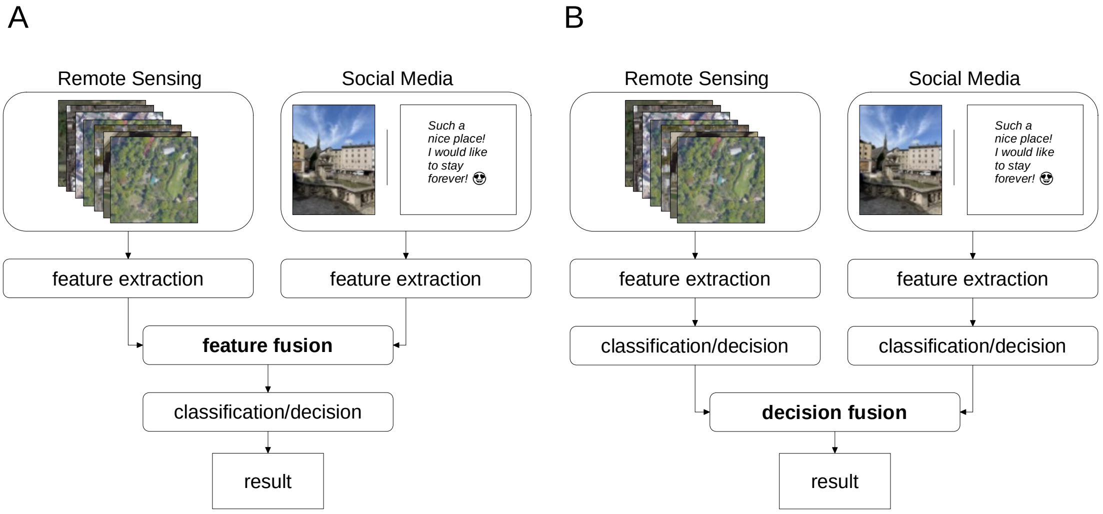


Figure 2.2: **A** feature fusion and **B** decision fusion depiction. The social media modalities can be images, text, metadata, or both. Image inspired by [2, p. 29, figure 14].Background images © TerraMetrics 2021, Google. Social media example image © private, the author.

rapidly increasing. In the present big data era, deep learning methods are becoming more and more accepted in the field of remote sensing. The superior performance in image classification makes deep learning a valuable tool for urban remote sensing [24, 11, 62, 63]. More details and insights into Deep learning are further discussed in chapter 2.3.

This section proposed the combination of data of two different sensors to gather more information for urban remote sensing. However, how the data of two (or more) sensors could be combined was left out. Therefore, the next section will discuss available methods to fuse data from various sensors and answer the latter question.

2.2 Fusion

As [64] states, human brains fuse sensory information to derive the best possible decision about a particular situation. In the engineering domain, sensor fusion has a long tradition. It can be described as the combination of observations done by different sensors in order to increase useful information for the studied phenomenon [64]. Of course, the sensors should be comprehensive enough to gain information. However, a mere duplication of the observations is not particularly helpful. Therefore, the fusion of sensor data can be done at different levels:

1. data fusion
2. feature fusion
3. decision fusion

Fusing information at the data level is mainly achieved by combining data from different sets to build a new set and using it for further feature extraction and decision making [64]. Feature level fusion utilizes already generated features, whereas decision level fusion takes already made decisions by different feature extractors and calculates the final decision [64]. The advantage of decision level fusion is computational efficiency and robustness against a sensor failure. When the information from one sensor is not available, the other sensor still provides a decision [64]. Whereas with feature fusion, a failure in the decision-making process is possible. However, a drawback of decision fusion is that possible information is lost because the features are used for classification independently [64].

The combination of specific sensor data is a traditional method in remote sensing to enhance the information quality [43, 2, 42]. As mentioned in section 2.1, the data pile collected by remote sensors increases rapidly. More information becomes available and awaits exploitation. However, the data collected by the many different sensors comes with trade-offs. E.g., a higher spatial resolution comes at the expense of a lower temporal or spectral resolution [2]. Therefore, data fusion can be used for pan-sharpening [65], spatiotemporal fusion [66], or multisensor image analysis [67].

In section 2.1, it was discussed that crowd-sourced (e.g., OSM) or location-based social media data (e.g, tweets), can be used to study urban areas. Whereas a coregistration via positioning systems of the data sources is rather easy [43], fusing two highly heterogeneous data sources is challenging [2]. Remote sensing and social media data cannot be fused directly together. [2] propose three distinct ways to combine the data (c.f. figure 2.2). First, features are extracted individually but are integrated into the same classification framework. The fusion of the two (or more) feature sets is done after the individual extraction paths are followed by the classification, e.g., [67, 68]. A second possibility is the decision level fusion. As explained above, two different classifiers first extract features of the two modalities, perform distinct classifications, and then afterward, the individual decisions are fused, e.g., [14]. A combination of both methods is also possible.

Further and more detailled information about data fusion can be found in [43, 2, 42, 69, 70].

Before data fusion can be performed, an accurate classifier or feature extractor should be selected. The following paragraph briefly introduces deep learning methods for computer vision and natural language processing tasks.

2.3 Deep Learning

The (human) brain is still a mysterious but fascinating organ. It automatically controls basal functions like breathing or the heartbeat [71]. While reading this dissertation, the brain simultaneously comprehends the text and creates consciousness. No wonder the (human) brain inspires scientists across the world. They are eager to learn from it and create digital micro-structures [72] which try to understand this organ. A goal of cognitive science is to "simulate" cognitive functions with algorithmic processes, e.g., the work of [73]. Computer scientists working on classification problems are trying to approximate the learning mechanism of the brain to increase the accuracy and flexibility of their models–which recalled *artificial neural networks*.

An axiom of neural networks has remained the same: learning from experience. Experience is gained by showing a model training samples x and minimizing the error to the target y. Often, the error is backpropagated [74], and a set of weights w is altered. The experience of the model is stored in weight matrices which are called *hidden units*. Over the last decades, the number of hidden units increased–the models became *deeper* and *larger* [21].

2.3.1 Convolutional Neural Networks

In the 1990s, the feature extractors of a classification task were often handcrafted and very task-specific. Therefore, the success of such a feature extractor was often dependent on the person's skill for creating it [75]. [76] propose the Convolutional Neural Network (CNN) architecture which has major benefits such as sparse interactions, parameter sharing, and equivalent representations [21]. A CNN can take size-normalized images instead of handcrafted features for classification. Via convolution and pooling layers, it can deal with data that may be distorted or irregular (e.g., handwritten numbers or letters). CNNs are efficient to train and are memory friendly [21]. The following sections discuss CNNs and other neural architectures in greater detail.

The basic CNN architecture has been constantly improved. For example, [77] increased the depth of a CNN by adding more convolutional layers. To achieve this, they decreased the filter size in all layers. This was done in order to tackle the demanding computational effort of the deep VGG models while at the same time making the model even deeper and wider, [78] which led to the invention of the Inception architecture: instead of only using one kernel size within a convolutional layer, it is possible to apply varying filter sizes at once to the input. In a so-called *Inception-module*, the convolutional layers with different filter sizes and max-pooling are concentrated. The output of a module to a subsequent layer is concatenated. In order to reduce computational complexity, the final inception module has a downsampling component to reduce the dimensionality of the input. By adding skip-connections before convolutional layers, [79, 80] the potential is there to build a very deep residual network and achieve state-of-the-art performance at the ImageNet classification task [81] with a smaller complexity than a VGG network.

In (urban) remote sensing, CNNs are nowadays a standard tool for image classification [82], building detection using high-resolution Google images [83], and locale climate zone classification using Sentinel-2 images [68]. For this reason, it also applied in this work to classify the obtained remote sensing images for building function classification (cf. sections 6.1).

2.3.2 Long Short-Term Memory, Transformers, and Attention

For NLP tasks, a different inventory of deep learning models is available. One of the crucial challenges in NLP is long-term dependencies [84, 21]. Specifically, it is an information unit that is needed for a decision, but this unit is located way back in time. The

Recurrent Neural Network (RNN) attempts to resolve this complication with an additional loop to feed information back to the input layer. The assistance from the "past" uses the network to memorize such long-term dependencies [73]. Basic RNNs, however, have difficulties learning longer sequences [84]. For this reason, [85] introduced the Long short-term Memory (LSTM) architecture. An LSTM can learn long-term dependencies better than regular recurrent networks by constantly modifying a cell state using gates. Such gates decide if the information is important for the current state. In sequence-to-sequence tasks, like translation tasks where long-term dependencies are natural, LSTMs can show their utility.

[86], for example, used two LSTM networks in an encoder-decoder setting [87] to translate English-French sentence pairs. The LSTM encoder encodes the sequence (of arbitrary length) into a fixed-length context vector c and the LSTM decoder decodes c into the translated text sequence.

The mapping of the information of a short sentence into a fixed vector representation works well. However, if the sentence length increase, the encoder might have trouble pressing all of the information into a fixed-length vector—the performance of encoders-decoders decrease when the sentence length increase [88]. [89] proposed a mechanism to overcome this issue. Instead of encoding the whole input sequence, it divides it into a sequence of context vectors for every word. The clue: the decoder focuses only on the vectors where helpful information is suspected to predict (or translate) the desired target word. The model knows where to focus by a weighting mechanism that scores how well the input and output agree. This method is widely known under the name *Attention*. By applying Attention, the performance in translation tasks with longer input sequences increased [89].

The development of Attention led to the advent of *Transformer* models [90]. A basic Transformer is also an encoder-decoder architecture. However, the recurrent loops are canceled, and the input sequence is presented at once instead of sequentially. For every input word, an embedding is created. Due to the omission of the sequential input, position information of single words is lost. Therefore, the words are embedded, and the position encoding is calculated to preserve information about the word order. During training, the decoder part is additionally fed with the entire target sequence that flows next to the received encoder input. Then, the output is transferred into word probabilities and the desired output sequence. Finally, the loss function evaluates the outcome.

The attention mechanism [90] used is called *self-attention* [91]. The main difference is that self-attention is not bridging the encoder and decoder. It is located within the encoder or decoder layer and relates all sequence words to each other. In [90], the self-attention is stacked in so-called *multi-head attention* layers. Furthermore, [90, p. 7] claims that "[n]ot only do individual attention heads learn to perform different tasks, many appear to exhibit behavior related to the syntactic and semantic structure of the sentences."

The Transformer showed superior performance, e.g., in machine translation tasks, over RNN-based encoder-decoder models, and it was faster to train [90]. For classification purposes or language modeling, the decoder part can be left out, e.g., like in the famous language model BERT [92] that will be introduced in section 2.5.4.

For a more exhaustive history of deep learning, [21, 23, 22] provide further reading material. The following section discusses the rather unusual field of natural language processing in the remote sensing community. A well-arranged overview of text preprocessing and representation is given.

2.4 Natural Language Processing in a Nutshell

The oldest human writing (discovered so far) has been found in Egypt and Mesopotamia dating back over 5,000 years [93]. Nowadays, humanity can type digital characters via a keyboard or even on a smartphone display. Writing in the digital age has become a common way of communication. We can chat with our friends or family, answer e-mails, or post comments on Twitter. In short, written texts have been a traditional way of information for millennia. In the digital age, many books, texts, invoices, and other documents are digitized or will be digitized. The vast amount of text data is interesting for commercial and scientific purposes who wish to extract information gaining insights and knowledge of the world.

Natural Language Processing investigates methods to transfer written text into a machine-readable format. Simultaneously, it should preserve as much information about syntactical and semantical features such that an interpretation of the data is possible [94]. The following section gives a brief overview of foundational techniques and algorithms applied in the field of natural language processing.

2.4.1 Tokens and n-grams

Some terms should be known before text preprocessing and algorithms can be understood properly. One term which appears frequently in NLP literature is *token*. A token is a discrete entity of a text corpus [21]. Normally, a token is a string of single letters that form a meaning-carrying processable unit, e.g., a word [95]. A sequence of tokens is called an n-gram. If a sequence contains one single token, it is referred to as *uni*-gram (*bi*-gram, *tri*-gram, and so on) [21]. The following example sentence *(a)* is a *tri*-gram:

(a) THE PENGUIN SHOUTS

A different category of n-grams are character n-grams. In contrast to the regular version, character n-grams are composed of tokens of a single character. Character *bi*-grams of the word *penguin*, for example, would be represented as tuples of letters (cf. *(b)*)

(b) PE EN NG GU UI IN

For more information, please consider [21, pp. 448-450] and [95, pp. 22-23]. Both token representations are used in the algorithms introduced in the upcoming paragraphs below.

2.4.2 Text Preprocessing

Imagine texts written by a professional author like a journalist who works for a high-ranked magazine, in comparison to random tweets, it is clear that sentences and writing styles can differ fundamentally. While most magazine articles have a clear structure, spell-checking, and editing, tweets are often drafted in a more unorthodox way. The authors of tweets mostly share web URLs leading to entertaining videos in the depths of the *www*, posting their latest holiday pictures, or uttering an opinion about whatsoever. Therefore, the unusual spelling of words or the creative way of using written language enriched with emojis is often utilized to emphasize the sentiment of the post or (maybe) the video's hilariousness.

While this utilization of text serves the user's expressiveness, it hinders the NLP scientist from analyzing the text straightforwardly. Of course, even formally written text would need to be preprocessed before a NLP task. The focus on preprocessing twitter text messages is temporarily redirected towards text preprocessing in general because there are several practices that are applicable to standard texts and informal texts.

One of the first steps in text preprocessing is often the lower-casing of all the words in a whitespace-separated text corpus [95]. The main reason for this step is that sometimes a word is written at the beginning of a sentence starting with a capital letter, while the same word in the middle of a sentence is spelled lower-cased.

(c) **W**hile the woman walked down the street, a man sneezed loudly.

(d) The woman walked down the street **w**hile a man sneezed loudly.

For a human reader the word *While* in sentence *(c)* and *while* in sentence *(d)* stay the same. However, a "machine reader" would detect two different words, and later, during the learning phase, it has to learn them: a word with a capital first letter and another word with a small first letter. In order to troubleshoot this, the first preprocessing would be the lower-casing of every single word in a text corpus. This step is even more essential using social media text. Consider the spelling of the following sentences (they could be from tweets):

(e) While the woman walked down the street, a man sneezed so **LOUD**.

Sentence *(e)* shows the informal spelling to stress the noisy act. The next example depicts even more irregular spelling:

(f) Tonight we will have the craziest **PaRtY** ever!

The quite unusual casing of the word presumably attempts to bring to attention that something crazy will happen. Lower-casing the corpus will help to limit the vocabulary and prevent storing or learning words with the same meaning. It should be mentioned at this point that lower-casing the corpus could cost some subtle semantic differences, for example, between *apple* and *Apple* (fruit vs. company). In this sense, lower-casing is used to concur with standard NLP procedures and to reduce the amount of the vocabulary of the highly irregular social media text, e.g., sentence *(f)*.

The second (language-independent) preprocessing step is the removal of punctuation. In most of NLP classification tasks, punctuation is removed since it rarely contributes to a classification task because it increases the complexity of a text sequence and further expands the vocabulary with useless tokens. In this dissertation, all punctuation is removed except for hyphens and apostrophes to preserve features like *wasn't* or *high-tech* [95].

The third standard preprocessing step is the removal of so-called *stop-words*. Stop-words are tokens like *I*, *and*, or *while* which have little or no meaning and occurring in a high frequency in a corpus [95]. The vast number of such words within a corpus could lead to a narrow focus of the classification algorithm on such words, resulting in a decreased classification performance. However, the choice of a researcher to remove stop-words or not depends highly on the task and algorithm. For example, [96] found out that the removal of high-frequency words was counterproductive while performing a statistical machine translation task. While traditional information retrieval methods like TF-IDF operating on word frequencies, the deletion of stop-words might be crucial. However, concerning geospatial natural language processing, the removal of stop-words could be disadvantageous. Phrases like *I'm at ...* could be helpful to identify targets located in an urban context. Furthermore, word embedding algorithms or language models take the context of a word, i.e., surrounding words, into account. That creates a deeper understanding of the language structure [97].

2.4.3 Tokenization

A crucial preprocessing step is called tokenization. It describes a procedure where tokens are selected and split by a separator, e.g., white space or a set of rules [95]. For example, punctuation should be separated from words and letters and treated as individual tokens (cf. example sentences *(g)* and *(h)*).

(g) The penguin, a lion, and a zebra walked out of the zoo.

(h) ["The", "penguin", "a", "lion", ",", "and", "a", "zebra", "walked", "out", "of", "the", "zoo", "."]

Of course, the examples above are relatively easy to process. However, sentences often include contractions *(don't)*, possesions *(it's)*, or hyphenations *(machine-readable)*.

(i) The penguin-style cup didn't belong to him.

(j) ["The", "penguin", "-", "style", "cup", "didn", "'", "t", "belong", "to", "him", "."]

Example sentences *(i)* and *(j)* show how difficult tokenization is. Dates, prices, or names like *Technische Universität München* are further complications. The tokenization via punctuation would lead to information loss. Hyphenations would be separated, and the hyphen itself would become a (meaningless) token. The meaning of contractions and possessions would also be dropped: the word, apostrophe, and the following abbreviated word like *not* would be completely torn apart (cf. *(j)*). Much more suitable tokenization would take additional information into account. For example, the popular Python package NLTK[11,12] [98] includes the word_tokenize[13] tokenizer. It is based on the algorithm of [99] that is able to split sentences. Furthermore, it utilizes information from the *Penn Treebank* [100][14] to split words (including apostrophes), punctuation, or even emojis into a list of tokens.

(k) ["The", "penguin-style", "cup", "did", "n't", "belong", "to", "him", "."]

The tokenized sentence *(k)* differs clearly from *(j)*. The hyphenated word is preserved as well as the contraction of *did*. The word is split into two parts: *did* and *n't*. The word *did* is preserved and also the negation as a separate token. This word description is beneficial if words are later supposed to be represented in a vector form (cf. paragraph 2.5.3). An additional aspect of word_tokenize is that it can be applied to text sequences written in several different languages (cf. footnote 13). Hence, a basic multilingual applicability is given.

Although the tokenizer mentioned above can deal with different languages, its application in a highly multilingual text environment is complex. For every (supported) language, the tokenizer must be initialized with the corresponding language. If a language is not supported, the tokenizer cannot be used, or the tokenization performance is poor. For Asian languages like Japanese or Chinese, it cannot be applied at all since only languages are supported where words are usually separated by whitespace. Chinese or Japanese words are not separated by whitespace. Therefore, a very different approach must be used.

Models like BERT (cf. [92] and paragraph 2.5.4), on the other hand, require special tokenization steps. For example, it expects special input tokens before training which are not provided by standard tokenization. Furthermore, the multilingual variant[15] supports 104 languages (including Chinese simplified, Chinese traditional, Japanese, and Korean). A multilingual setting calls for a solution that includes methods that can manage Asian languages as well. TensorFlow Hub[16] provides not only BERT models but also preprocessing models that prepare the text data to meet the BERT data representation and at the same time cope with multilingual texts. Such models are based on *WordPiece* models, which can break the text into single tokens with whitespace and/or for Asian characters split after Unicode.

Further (multilingual) tokenizers are, for example, *SentencePiece* that operates at a subword level and can handle every language after appropriate training [101] and *Moses*[17] which is a universal suite for multilingual natural language processing [102].

[11] Natural Language Processing Toolkit (NLTK): https://www.nltk.org [8.10.2021, 17:12]

[12] https://www.nltk.org/api/nltk.tokenize.html [8.10.2021, 17:12]

[13] https://www.nltk.org/api/nltk.tokenize.html#nltk.tokenize.word_tokenize [8.10.2021, 17:13]

[14] https://www.nltk.org/api/nltk.tokenize.html#nltk.tokenize.treebank.TreebankWordTokenizer [8.10.2021, 17:15]

[15] https://github.com/google-research/bert/blob/master/multilingual.md [3.11.2021, 11:32]

[16] https://tfhub.dev/google/collections/bert/1 [3.11.2021, 11:41]

[17] http://www.statmt.org/moses/ [3.11.2021, 11:54]

2.5 Text Sequence Representations

Since computers cannot read a text like humans, every text sequence must be transformed into a machine-readable representation. This procedure is referred to as vectorization[18,19]. During vectorization, characters or words are transferred into numerical feature vectors. Today, various algorithms with different characteristics exist to perform this operation. In the section below, a collection of basic approaches is introduced to give an overview of how this task can be solved.

2.5.1 One-hot-encoding

A simple machine-readable text sequence representation is one-hot-encoding. The vocabulary is represented in a $d \times w$ matrix where d_i is a word sequence, e.g., a tweet, and n is the total vocabulary size. The sequence is encoded as a n-sized vector with zeros and ones. Each word w that occurs in sequence d_i is set to one. All other terms remain zero. For example, the text sequence $d_1 =$ *the weather is nice* or $d_2 =$*the zoo is crowded* would be encoded as: However, one-hot-encoding cannot preserve information like the word order or the content of the document. Only the occurrence of a word can be stored. The following paragraph describes an algorithm that takes word frequencies into account to produce a term weighting score.

zoo	the	nice	train	weather	church	sky	is	crowded
0	1	1	0	1	0	0	1	0
1	1	0	0	1	0	0	1	1

Table 2.1: Simplified one-hot text sequence, e.g., tweet, representation.

2.5.2 Term Frequency-Inverse Document Frequency

In information retrieval (IR) or data mining tasks, the categorization or classification of documents into specific topics is often needed [103]. For example, if the task is to find out which document of a set of documents is about soccer, a person would read or at least skim the documents looking out for clues like regularly appearing terms such as *goalkeeper*, *foot*, *referee*, *goal*, et cetera. Luckily, scientists provided algorithms to solve this task more efficiently since it would be cumbersome for a person to perform for thousands of documents.

TF-IDF, for example, is a classic IR algorithm that performs the task above. It is based on the work of [104] and categorizes documents by occurring words within the

[18] `https://scikit-learn.org/stable/modules/feature_extraction.html#the-bag-of-words-representation` [3.11.2021, 8:32]

[19] `https://developers.google.com/machine-learning/guides/text-classification/step-3` [4.11.2021, 9:40]

documents. The algorithm takes the word count and the count of documents where the word appears into account. However, as [103] points out: it is not necessarily true that words that frequently appear automatically contribute to the document categorization. On the contrary, very high frequency words like *and* or the, usually do not tell us anything about the document itself and are usually filtered out before starting an IR task (for more details about stop-words please consider paragraph 2.4.2). On the other hand, are rare words inevitable hints for a document topic? [103] give a further example about word frequencies which takes aims at the opposite case. They draw attention to words like "albeit" and "notwithstanding" (p. 8). Those words are seldom, but they also do not help to categorize the document. Therefore, "[t]he difference between rare words that tell us something and those that do not have to do with the concentration of the useful words in just a few documents." [103, p. 8]. To take up the soccer example from above, the specific term *offside* would tell us more about a document containing football-related text sequences.

To measure the frequency of specific terms like *offside*, the first step of the TF-IDF algorithm is the computation of the *term frequency TF*. If D documents are available, then TF of word i in document j is calculated in the following way[20]:

$$TF_{ij} = \frac{f_{ij}}{max_k f_{kj}} \tag{2.1}$$

The frequency f_{ij} is then normalized by the maximum frequency of any term in the document. The word that exists most in j gets a score of 1 and all other words scores are fractions. After this step, the TF counts of the documents are stored in a $D \times$ vocabulary matrix. Ths form of vocabulary description is a so-called Bag-of-Words (BoW) representation.

The computation of the *inverse document frequency IDF* score is performed as follows:

$$IDF_i = log_2(\frac{D}{d_i}) \tag{2.2}$$

Word i can be found n times in all of the documents D. To prevent divsion through zero if an out-of-vocabulary word occurs, a 1 can be added the numerator and denominator[21].

Finally, TF and IDF can be multiplied to receive the final TF-IDF score of a word.

$$TF\text{-}IDF = TF_{ij} \times IDF_i \tag{2.3}$$

The higher the score of a word is, the more important is it to define a topic. Hence, TF-IDF is scoring, i.e., weighting words after their importance for each document [103, 95].If the latter steps are employed on a text corpus, the words can be vectorized and are machine-readable. Now, the desired NLP task, e.g., text classification, can be performed.

As mentioned earlier, TF-IDF only takes the word importance within a document into account, which might be good to get a general overview of a document's most important

[20] All listed TF-IDF equations were taken from [103, p. 8].

[21] `https://scikit-learn.org/stable/modules/feature_extraction.html#tfidf-term-weighting` [12.11.2021, 14:54]

terms or is helpful by removing stop-words. However, a word's context or semantic attributes are not considered, occurrences of Ouf-of-Vocabulary (OOV) words cannot be handled, and possible beneficial information is not used or is lost. 60 years ago, [105] comments:

> Even in the study of vocabulary [2] when ordered series of words are presented, such as kinship terms, parts of the body, terms of orientation in time and space, numerals, calendrical terms, names of social units, proper names of persons as well as of places,[3] it is essential that they be separately and severally attested in contexts of situation. It is, however, necessary to present them also in their commonest collocations. (p. 11)

He further argues that "[t]he *placing* of a *text* as a constituent in a context of situation contributes to the statement of meaning since situations are set up to recognize *use*. As Wittgenstein says, 'the meaning of words lies in their use.'[4]" [105, p. 11]. For this reason, algorithms have been developed which utilize the word's context, semantic, and syntactic information to generate a continuous feature vector of a word. In the subsequent paragraph, such techniques are introduced and discussed.

2.5.3 Word Vector Representations

With the quotation from above in mind, [105] continues with his famous sentence "You shall know a word by the company it keeps!" [105, p. 11]. What does this statement mean within the context of NLP tasks? By looking at the TF-IDF algorithm, it is noted that words are considered in isolation and are merely counted without taking context, i.e., neighboring words into account. As mentioned above, syntactical and semantical features of a word are lost.

In the early 1990s, [97, 106] pointed out that many NLP tasks, like word sense disambiguation[22], require semantics of words to be solved. He argues that "[...] the same content can be expressed with very different words, so that [...] two contexts could have a similarity measure of 0 although they are very close in meaning." [97, p. 787]. For this reason, he proposed a system with co-occurrence counts within a context window [97] and additionally of letter character $four$-grams [106] to create vector representations of words. A context of a word can be defined as words adjacent next to the word of interest. Therefore, a context window can be determined as an n-sized range of tokens before and after the token of interest. For example, a context window size of 2 of the word zoo in example sentence *(l)* would comprise the words {in, the} and {was, very}.

(l) The elephant we saw *in the* zoo *was very* large.

The key message of [97] is that the context of a word is taken into account to derive semantic meaning. If a co-occurrence matrix of words is available, the cosine distance for every word within the context window can be calculated. The normalized average of such word vectors could be described as the approximated semantic content of a

[22] E.g., *umfahren* vs. *umfahren* (German; avoid an obstacle vs. hit the obstacle)

word. [97, p. 787] further points out, "[i]f at least some of the words in the context are frequently used to describe what the current context is about then their vectors will pull the centroid toward the direction of that topic or content.". He claims that the proposed context vectors are more reliable than the BoW representation [97].

In his follow-up work, [106] proposed a method to calculate word vectors for single words. This differs from the approach above. A word vector was created by the cosine distance of the co-occurrence count of an adjacent word located in the context. To calculate a word vector of a single word, he used character $four$-grams. The word *football*, for instance, would be sliced into the following $four$-grams:

(m) FOOT OOTB OTBA TBAL BALL

Instead of utilizing a word co-occurrence matrix, he created a $four$-gram collocation matrix. This matrix holds the counts of occurrences of a $four$-gram i to the left of $four$-gram j. That count reflected the context of the two $four$-grams being found. Then, vector representations of the $four$-grams are created by Single Value Decomposition (SVD) [95, e.g., pp. 407-409] to reflect each $four$-gram count vector as a 97 dimensional real-valued vector. Composing a word vector by its context, context vectors must first be calculated for all positions the word appears in the text. All four-gram vectors are summed within a pre-defined window centered around the word of interest. Finally, the context vectors are summed and normalized. The result is the word vector of the word of interest [106]. Briefly, all contexts of word $four$-grams within a context window are used to create a vector representation of a word. With his work, [106] takes up the arguments of [105] and contributes significantly (amongst others) to word vector representations that preserve meaning.

2.5.4 Word Embeddings and Language Models

Word embeddings established a widely accepted technique to represent text in machine learning tasks [107, 94, 3, 108, 109]. Word embeddings provide a n-dimensional vector space representation of words that can preserve semantic and syntactic features. It is assumed that a word is described by the co-occurrence of adjacent context words. Often, large text corpora like Wikipedia or CommonCrawl are used to train such embeddings in an unsupervised setting. In a broad spectrum of application areas in natural language processing, e.g., word analogy [3] and similarity [110] task, or Named Entity Recognition (NER) [111], word embedding algorithms achieved good results. In the following paragraphs, important milestones of word embedding algorithms and language modeling are briefly introduced.

Word2vec was developed by [3] and is also a neural algorithm like [107]. It can be trained on much larger datasets and decrease training time because the hidden layer was omitted, leading to reduced computational complexity. The main innovations are the skip-gram and continuous bag of word (CBOW) models (cf. figure 2.3). While the skip-gram model predicts the context of a word within a context window, the CBOW model predicts the word by a given context. The algorithm is able to capture syntactic and semantic similarities of the training words. For example, the simple vector arithmetic

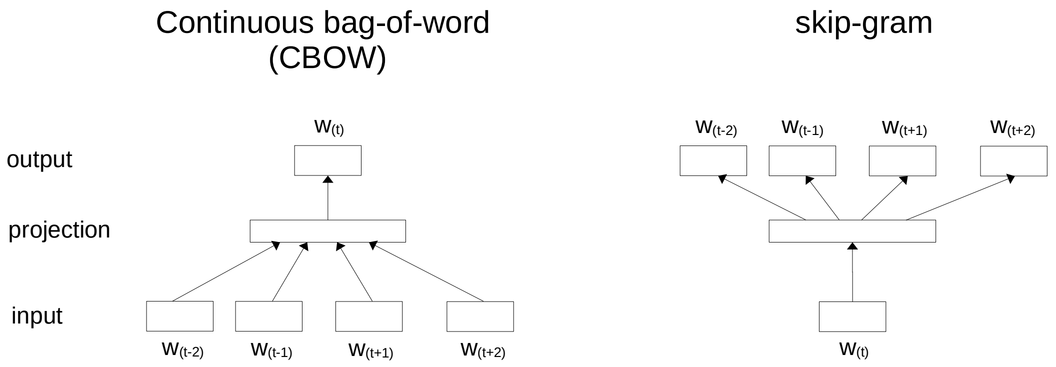


Figure 2.3: Continuous bag of word (CBOW) and skip-gram model. Figure inspired by figure 1 in [3, p. 5].

$paris - france + italy$ would result in an approximation of the word vector of *rome* [112]. Word2vec allowed the efficient training of large text corpora and created word vectors that preserve the syntactic and semantic meaning of words.

The embedding algorithm GloVe [108] used a different method the generate word vectors. GloVe is the acronym for *Global Vectors*, and in contrast to word2vec, GloVe utilizes a word co-occurrence matrix in addition to a local context window. GloVe outperformed word2vec various tasks, e.g., word similarity or named entity recognition (NER).

Both word2vec and GloVe cannot deal with out-vocabulary-words (OOV). The fastText algorithm [109], on the other hand, can generate word vectors by the sum of its n-gram-representations. By using subword information, it can approximate word vectors of words that were not in the training data and long word compositions[23] [109]. Not only the representations of compositions are better, but also the embedding of morphologically rich languages [113], e.g., German, Hebrew, or Arabic. Hence, fastText could be seen as an improved development of word2vec. Additionally, [114] trained word vectors for 157 languages using Wikipedia and CommonCrawl[24].

There are several ways to use word vectors in combination with a deep neural network. Before the vectors can be fed into the network, the text sequences must be vectorized (cf. paragraph 2.5). Instead of one-hot encoding or TF-IDF, every token (i.e., word, number, punctuation, etc.) of the complete text corpus is indexed by an integer and stored into a dictionary D. The dictionary can be seen as a vocabulary look-up table where every unique token is indexed by an integer. Text sequences like tweets can now be vectorized by encoding the tokens into the integer-indexed representation.

Currently, two *on-line* options are available, which can be employed directly to the classification pipeline. The first option comprises the creation of a completely new word embedding by training a randomly initialized embedding weight matrix[25]. The second option is the import of pretrained word vectors into the neural network environment as

[23] e.g., *Donaudampfschifffahrtsgesellschaft*

[24] `https://fasttext.cc/docs/en/crawl-vectors.html` [12.12.2021, 12:30]

[25] `https://www.tensorflow.org/api_docs/python/tf/keras/layers/Embedding` [2.11.2021, 10:54]

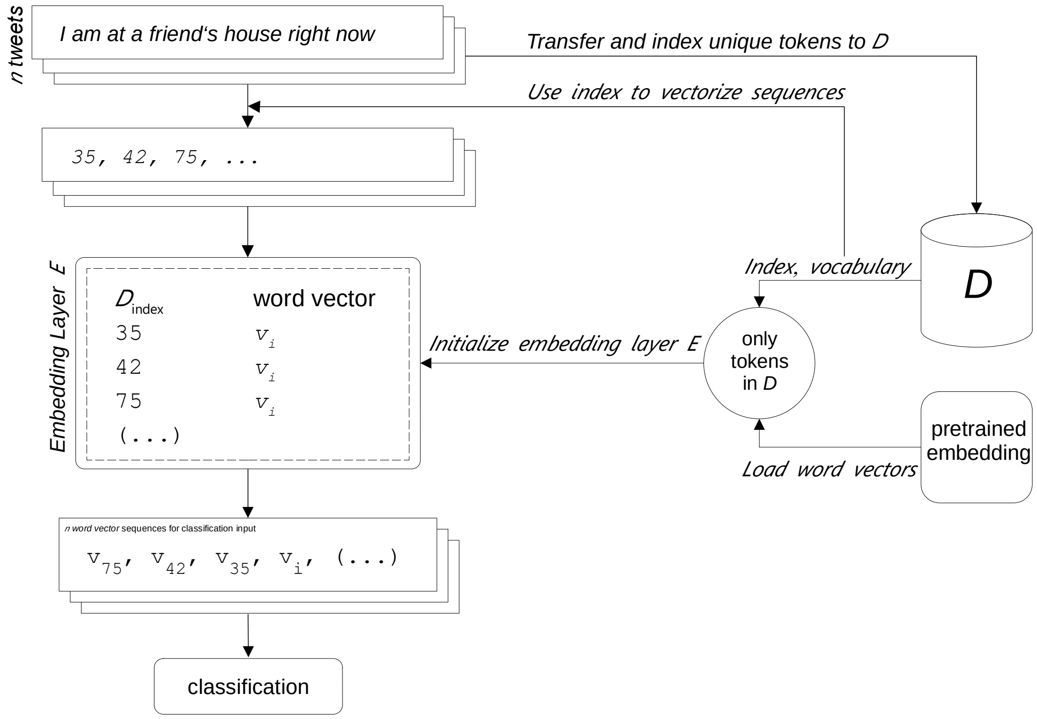

n tweets
I am at a friend's house right now
Transfer and index unique tokens to D
Use index to vectorize sequences
35, 42, 75, ...
Embedding Layer E
D index
word vector
35
42
75
(...)
v_i
v_i
v_i
Index, vocabulary
D
Initialize embedding layer E
only tokens in D
pretrained embedding
Load word vectors
n word vector sequences for classification input
v_{75}, v_{42}, v_{35}, v_i, (...)
classification

Figure 2.4: Vectorizing text using a word look-up table to integer-index sequences.

embedding weights. I.e., instead of training a completely new embedding, the already pre-trained word vectors are loaded and initialized. To reduce hardware resource consumption, only word vectors of tokens that are available in D are considered for the embedding matrix (cf. figure 2.4). The embedding layer is then initialized with the embedding matrix assembled with the token index and the word vectors of the tokens found in the pre-trained embedding. With the above integer-indexed sequences, the pre-trained word vectors can be loaded and word vector sequences compiled and used as input for classification. The third variant of using pre-trained embeddings is the creation of the word vector sequences *off-line*, i.e., independently of the classification pipeline. That means that the word vector sequences are generated and stored separately (e.g., in a text file). By using separately stored vector sequences, a look-up table is not necessary. However, it includes additional steps for generating, storing, and re-use.

The drawback of the word embedding models above is that only one word vector per word exists. However, the meaning of a word can change if a different context is given. In German, for example, the word *umfahren* has different meanings that cannot be represented by one single vector. Therefore, language models like ELMo [115], ULMFit [116] or BERT [92] can generate word vectors that are context-dependent. Specifically, the word vector of a word differs by its context.

Bidirectional Encoder Representations from Transformers (BERT) is based on the Transformer architecture [90] and was developed by Google researchers [92]. The encoder processes the entire text input sequence at once, allowing for the learning of the complete context of a word. The decoder part of predicts the desired text outcome. Another crucial difference between BERT and the above-discussed algorithms is how text sequences are presented to the model. Usually, a sequence with n words is fed into an algorithm, and the subsequent $n+1$ word must be predicted. That is identified as the bottleneck of context learning. In contrast, BERT presents an input sequence bidirectional, i.e., the words are not presented sequentially, but the whole sequence is presented at once. However, since this procedure would lead to the fact that the model would see the target words, the 15% of the words of sequences are masked (the authors refer to that as *masked LM*) [92]. With the generated contextualized word representations, BERT led to a state-of-the-art performance in tasks like question answering [117, SQuAD], or language understanding [118, GLUE]. Next to contextualized word vectors for each token, BERT models are also able to produce contextualized sentence representations. This is achieved by placing a particular token, the *CLS* token, as the first token of a sequence. [92, p. 4174] stressing that "The final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks.".

Google actually applying BERT in its own famous web search[26]. However, it was also finetuned for various NLP downstream tasks. These models are called *BERT experts*. For example, a version that is finetuned for bio-medical language processing [119] or German named entity recognition (NER) tasks [120]. For more technical details please consider the paper of [92], the Google blog entries in the foototes[27], or the review paper of [121].

However, since the original BERT models are very large, the computational resources and the resulting energy consumption to employ such models is high [122]. Therefore, some "smaller" variants of BERT are available. *MobileBERT* [123], for example, is BERT variant that is optimized for usage on mobile devices. It is smaller and faster but achieves fair results similar to the original BERT model. In this work, *DistilBERT* is used [122]. This BERT version is based on the student-teacher principle, where the student is trained to "mimic" the teacher's actions. DistilBERT achieves 97% of the original BERTs performance.

For the sake of completeness, several more large neural language models are available. For example, GPT-2 [124], XLNet [125], the parallelized training of very large language models–MegatronLM [126], or T5 [127].

2.6 Summary

In this chapter, the fundamentals of the research area and methods have been briefly introduced. Urban remote sensing seeks answers to questions *(inter alia)* concerning

[26] `https://www.blog.google/products/search/search-language-understanding-bert/` [6.12.2021, 12:32]

[27] `https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html` [6.12.2021, 13:37]

urban sustainability by using remote sensing methods. The information collected from the sensors is processed with deep learning computer vision models like VGG and data fusion to yield a maximum information output. Additionally, as sensors from the ground, social media data can be used to add more features distilled by NLP techniques such as word embeddings or large language models like BERT. In the next section, an overview of related work is given to contextualize the current work.

3 State-of-the-Art

In this section, trends and recent research related to this thesis are introduced. At first, urban remote sensing work is discussed, followed by discussing Twitter data within the context of geospatial research. Next, data fusion focussing on social media data is shown. The section is then concluded by introducing deep learning and natural language processing research.

3.1 Urban land-use and Deep Learning

Land-use is a classic task in the remote sensing community. The development in this area started by using handcrafted features and the application of decision trees [15]. Soon afterward, upcoming deep learning methods like convolutional neural networks started being used as well, and standard machine learning practices, like making use of ImageNet [81] weights have been adopted [128] too. It turned out that deep learning methods can yield higher land-cover prediction accuracies. The utilized a VGG16 network [77] were able to achieve good accuracies classifying agricultural areas, forests, and airports. The features of these classes seem to be general. [129] offers a morphological class scheme which includes next to vegetation-related classes like dense trees, also human-made classes such as high-rise or low-rise buildings. [130] and [131] prove that such classification schemes can be classified well with multi-temporal remote sensing information using deep learning and data fusion.

However, urban land-use classification is more complicated. In the past years whether deep learning-driven approaches can be applied to predict land-use by multispectral imagery. For example, the materials of the different building functions could be similar, which makes it hard to differentiate between functions [15, 16]. Additionally, classes could overlap, which makes it even more difficult to discriminate land-use classes [15]. However, similar difficulties are reported when separating high, mid, or low-density urban environments using high-resolution optical imagery [11]. It is reported that some land-use classes, like forests, are categorized well. Other high-level features such as "high density urban fabric" or "medium density urban fabric" suffer from lower accuracies. A possible explanation is owed the fact of highly subjective and complex to standardize land-use classes [11]. Classes like "high density urban fabric" or "medium density urban fabric" are blending into each other such that a clear border is hard to determine [11]. By using smaller patch sizes, e.g., 50m, lower classification results are documented. The authors suggest that smaller patch sizes might not induce a rich feature space [11]. On the other hand, information of smaller patch sizes could lead to enhanced classification results in urban land-cover classes, e.g., distinguishing building functions. Unfortunately, the authors are not providing detailed class-wise results on this matter even though they use high-resolution Google Maps imagery. [132] point out that for object-level classification, the resolution of many remote sensors is too coarse. A comparison of Quickbird and freely accessible Google Maps images reveal no significant differences in land-use classification accuracy. Google Maps imagery showed promising results in land-use/land-cover tasks [132]. It is pointed out in the literature that indeed high-resolution from Google Maps is a valuable resource for classifying land-use even at individual object level [133].

By taking the issues mentioned before into account, it might be beneficial to add additional data sources to increase the classification performance. However, the strength of remote sensing images is more located to identify physical layout on the ground [133]. It is noted that remote sensing images are might not enough to identify the detailed functionality of ground objects, and the augmentation of additional data is recommended [133] like different features like building height or floor area to classify land-use classes such as "office", "civic", "industrial", or "transportation" [15]. It turns out that the class "office" is often confused with "civic" and "industrial" [15]. Even though different features are used, it seems that classes with a similar urban context like "office" or "civic" are misclassified because they might be visual to similar [15].

The issues mentioned before are a sign of improving the land-use classification processes with additional data from the ground. Data acquired from in situ, i.e., on-site, sensors could deliver features related to the area of interest but possibly not detectable to spaceborne sensors. Therefore, the following section discusses the utilization of auxiliary data such as OSM or social media data within geospatial research and land-use classification.

3.2 Twitter and Geospatial Research

[17] proposed that citizens could act as an implicit sensor to detect urban phenomena. So to say, as an *in situ* sensor. OSM is a perfect example of the citizen sensor idea. It is created by a large mapping community to map the entire world since the spirit of OSM is that volunteers explore and map their neighborhoods. Even though OSM is criticized for its lack of completeness [134], all of this user-generated content is accessible for free, and research can be conducted in areas where no official data is available and at the same time mitigate research redundancy. Also, the in remote sensing rather unusual field of natural language processing can be utilized for solving geospatial research tasks. [135], for example, combined NLP and remote sensing images. The authors introduced a visual answering system based on a RNN (cf. section 2.3) and a ResNet50 (cf. section 2.3) trained with Sentinel-2 imagery. The features of the two modalities have been fused point-wise so that the model could reply to questions like "Is there a rural area?".

[136] found, for example, that georeferenced tweets can sketch administrative boundaries or blocks within dense urban areas visible or draw road networks. These are why georeferenced Twitter data is popular and widely used in urban research projects. The nature of Twitter data, however, is different than OSM. OSM data is maintained to create Volunteered Geographic Information (VGI). Twitter data, on the other hand, is not *a priori* created for geospatial research. Even so, new content in various forms is added

to the microblogging platform daily. Due to the easy-to-access data and the rich, and sometimes georeferenced, metadata, Twitter data offers a wide range of research topics. Therefore, Twitter data is not only interesting for NLP tasks like emotion detection and sarcasm prediction via emojis [137] but also for geospatial research as well. While this dissertation was being written, there was a worldwide pandemic caused by the SARS-CoV-2 virus (COVID-19, "Corona"). For a deeper understanding of the impact of such exceptional circumstances, [138] used a multilingual Twitter dataset to analyze the (digital) sentiment of European citizens during lockdown measures. Also, an exploration on the effects of anti-corona measures attempting to mitigate the virus was studied using *inter alia*, commuting data, and georeferenced tweets from Germany [139].

Besides the pandemic, linguistic features have been employed to conduct sentiment analysis and derive demographic characteristics of georeferenced U.S. tweets [140, 19], competitor analysis in the pizza industry [141], or event detection [142]. The combination with NLP techniques, e.g., word embeddings, and deep learning architectures, like CNNs, allows tackling complicated classification tasks like the identification of election-related tweets [143, 29].

Previous work shows that georeferenced Twitter data can also be utilized to investigate intra-urban characteristics. [144] used geo-located Twitter and Flickr[1] data to investigate how public parks in New York City are used and visited. [145] used tweets and derived TF-IDF scores from tagging OpenStreetMap objects in Great Britain, and [146] discussed the social relationship and mobility patterns by georeferenced Twitter data. Also, within the context of transportation, [147] mined Madrid-located tweets and monitored them in regard to complaints about the local subway system and mapped them to the spatial occurrence. The sentiment was detected by fine-tuning the multilingual variant of BERT with Spanish tweets.

Using linguistic features and patterns has also been applied in demographic research tasks. Also, for land-use classification, such features have been extracted. [148] employed tweets and Flickr images for an in-depth land-use classification in New York City and San Francisco. They made use of Latent Dirichlet Allocation (LDA) [149] to extract text features that were associated with Foursquare[2] venues. For the building function classification task, previous work utilized temporal-spatial analysis in combination with deep learning methods in order to extract settlement information [150]. Furthermore, [38] found different word clusters for commercial and residential tweets. For classification, vectorized word representations derived from word embedding algorithms, such as fastText [109], have been used to train a CNN. The deep learning model showed better performance than a Naïve Bayes and TF-IDF baseline. The study implies that deep learning and word embedding features are useful for building function classification. However, the study area was limited, and only one language was considered. In a subsequent study, [39] used tweet sentence vectors and neural network architecture to classify five classes of building functions at an individual building level in Berlin. Other than before, English was not the language of interest. Instead, a pretrained German fastText word

[1] `https://www.flickr.com/` [2.7.2021, 18:32]

[2] `https://foursquare.com/` [2.7.2021, 18:37]

embedding has been used to create the sentence vectors. The results showed that word embeddings could be used for individual building function classification. However, the study area was also limited to the Berlin urban area.

Recently, [20] introduced *abstaining* for building function classification. They focused on tackling the issue of class imbalance and how to exclude tweets from the classification that are not related to a building function. For text representation, they generated TF-IDF features from Los Angeles tweets posted in 2017 and 2018. The classification task covered the classes *commercial* and *residential*, which were derived from OSM. The tweets have been assigned to buildings via a nearest neighbor join, and the maximum distance was limited to 100m.

Their findings also support the claim that tweets can contribute to building function classification in urban areas. Using the abstaining method, they detected tweets where the content contained information about the building function. On the other hand, if content directly pointing towards a building function is necessary to classify a building, or implicit features also enable a successful classification. Furthermore, as pointed out by the authors, even though the challenge of unrelated tweets was tackled, the potential of multilingual text and the data fusion remains was not exploited. The study area and their amount of tweets were limited.

3.3 Multilingual Twitter Analysis and Text Representation

The latter sections extensively discussed the application of Twitter data on geospatial research tasks like land-use classification. Often, only sparse feature representations of the text are used, and additionally, text features obtained from a multilingual setting are rare even though Twitter is a highly multilingual space [35]. However, using tweets in several languages for classification tasks is not impossible. Some authors, for example, address multilanguage sentiment analysis based on Twitter text messages in a multilingual setting [151]. Before discussing the usage of multilingual tweets from a geospatial perspective, some explanation about multilingual text representation should be shown since working with multilingual data is challenging.

[152] also produced multilanguage word vectors in 59 languages by estimating projections of monolingual word vectors into the English vector space. [114] proposed various pre-trained word vectors in 157 languages trained on Wikipedia dumps and CommonCrawl. Word vectors for 100 different languages have been trained and preserved, for example, compositional semantic features of German multi-unit words [153]. The release of multilingual embedding models and word representation methods encourages to use of more multilingual text data. However, in a heavy multilingual space like Twitter, every language needs a separate embedding. In a downstream task like building function classification, a more holistic model covering several languages would be beneficial. Recently, [154] introduced a framework for multilingual sentence embeddings for 93 languages based on zero-shot transfer from English to other languages, even for those languages with underrepresented training data.

The latter examples focus on formally written language corpora. Indeed, on social media platforms such as Twitter, texts are often written in an informal spelling, mixed language, and (social media-specific) slang. For this reason, [155] developed RoVe (Robust Vectors), which tackles an issue of social media texts, namely typos. However, multilanguage texts are not taken into account by this approach. [156] used a linear translation approach [112] to map the word embeddings from one language into another to classify election-related tweets.

From a more application-oriented perspective, methods like multilingual universal sentence encoder [157] or the multilingual variant of BERT (cf. section 2.5.4) are more applicable. Because pre-trained models are open-sourced (e.g., via TensorFlow Hub[3]) and are easy to integrate into one's own models. For example, as mentioned in paragraph 3.2, sentiment analysis of multilingual tweets from Europe has been used to explore the tone and emotions of COVID-19 related tweets. [138] used (amongst other models) this analysis for a multilingual universal sentence encoder that supported 16 languages and a multilingual variant of BERT. The analysis reveals a measurable change of sentiment after announcing curfew measures.

Aside from technical aspects, multilingual Twitter data is rarely used in geospatial research. To the best of the author's knowledge, no work has been conducted on building function classification using multilingual Twitter text messages. Therefore, this dissertation attempts to close this gap.

3.4 Data Fusion with Social Media Data

The fusion of two or more data sources can improve classification results in various urban remote sensing-related tasks [43, 2, 42]. Fusing data from two remote sensors is a standard task in remote sensing. For example, [158] feature-fused the outcome of an RGB image classifier and a multispectral classifier. With this method, they were able to reach high accuracy scores by predicting urban land-use for Hong Kong and Shenzhen cities at a block level. However, the fusion with fundamentally different data sources is still challenging.

Nevertheless, research is exploring methods to utilize not only remotely sensed data but also geospatial data provided crowdsourcing platforms like in the OSM social mapping community. Here, the idea of the citizen sensor comes to life (cf. section 2.1). Scientists use the socially sensed data from OSM and fuse it with Landsat imagery. By the combination of the two modalities, land-use classification, e.g., for 24 classes, could be improved [159]. These results encourage the utilization of more user-generated, i.e., citizen sensor, data.

Not only data from OSM can be viewed as citizen sensor data. As discussed in section 3.2, Twitter data can also be used for geospatial research. The sheer volume and the possibility to tag local points of interest, or tagging the exact position (however, cf. section 4.2.1), is a valuable source of knowledge. For example, the combination of so-

[3] https://tfhub.dev/google/universal-sentence-encoder-multilingual-large/3 [9.11.2021, 18:02]

cial media data from Twitter and data from remote sensing can lead to improvements in flood detection [160], flood maps [161], and damage estimation [162]. Remote sensing and Twitter data can also be combined to understand socio-spatial contexts, e.g., if informal settlements have a different social media activity pattern in the megacity Mumbai [163]. Mumbai's urban structures types have been derived by high-resolution HR Quickbird data. The Twitter activity clusters in informal settlements (in proportion to the population density) were not so present, i.e., digital coldspots, as informal settlements. Hence, the combination of remote sensing and Twitter data deliver insightful information about the economic divide in megacities [163]. Also, in poverty mapping, Twitter data proved an attractive additional data source. Merging Twitter metadata and vectorized text with Nightlight data improved local poverty mapping for Africa [164]. These findings indicate that the combination of remote sensing data and social media data, i.e., citizen sensor data, is beneficial for geospatial tasks.

If Twitter data can increase the performance of applications mentioned above, the benefits of a step toward urban remote sensing tasks is evident. Since a large amount of social media data occurs in large urban areas [136], there might be information hidden in photos or text messages, which can improve classification results for land-cover or land-use tasks.

For land-use classification, [14] explored fusion methods of aerial images with street view images within the framework of a building function classification task. The results show that decision-level fusion by averaging prediction probabilities achieves the best classification performance. [62] showed the combination of several data sources for urban land-use mapping, and Google StreetView images for the ground perspective and Google Maps aerial images for the remote scene are used in that work. The proposed model can outperform models using only one data source.

[165], for example, also utilized metadata from georeferenced social media data and satellite imagery. They examined urban land-use in Haidian District (Beijing) using Weibo[4] and Gaofen-2 imagery. They divided the district into fields via OSM road data. For classification, they used textural and spectral features from the imagery. It also included the density and temporal patterns from georeferenced Weibo posts. [166] added to temporal and remote sensing data, linguistic features obtained from tweets, and improved land-use classification accuracy.

Also, in a land-cover task, the fusion of georeferenced Twitter data and remote sensing images can contribute positively to the classification performance. [167] proposed a data fusion framework to combine Twitter metadata and Sentinel-2 imagery. The target of the land-cover task was the classification of *Local Climate Zone* [129] classes. The definition of the classes ranges, for example, from bare soil, over scattered trees to an open high-rise building configuration. The area of interest was Washington, D.C., where the researchers collected a large georeferenced Twitter data set. As features, the tweet count, mean text length, or mean friends count (amongst others) were utilized per raster cell. They augmented the CNN architecture with the generated Twitter feature maps in one

[4] Chinese Twitter equivalent

experiment. This approach improved the classification result by increasing the F1 score concerning the baseline.

However, as pointed out in the previous sections, the researchers investigated small areas of interest. Therefore, the research gap of fusion remote sensing images and multilingual citizen sensor data from Twitter text messages has not been covered before.

3.5 Summary

A course sensor resolution and land-use classification at block or patch level might not be enough to provide detailed land-use or building function maps of settlements. Land-use classes flow into each other and impede land-use classification using remote sensing images at the patch level. Furthermore, almost all studies focused on a specific area of interest and limited data. However, tackling the U.N. Sustainable Development Goals demands a path towards a global and fine-grained building function classification at an individual building function level. Augmenting remote sensing data with in situ data like and social media data showed valuable results.

It is clear that adding data next to remote sensing imagery to classify land-use or building functions is not new—researchers before used linguistic features for building function classification. Like using remote sensing images alone, most of these studies focused on specific areas with minimal data. Furthermore, the capabilities of word embedding or (multilingual) language models have not been exploited so far. It is standard that various languages are spoken in highly heterogeneous urban areas. Therefore, utilizing multilingual language models like BERT or a self-trained multilingual Twitter word embedding seems to be a research gap regarding building function classification. To the best of the author's knowledge, a building function classification based on multilingual Twitter text features has not been done before. The decision level fusion of multilingual text features and remote sensing image features is also a novel application in the urban remote sensing field.

The following chapter concentrates on the Twitter data itself and the resulting datasets. It is shown, how the data can be obtained and processed. This includes the labeling and balancing of the Twitter data. Additionally, the Twitter georeference accuracy issue is discussed, and a straightforward approach is proposed to deal with this situation. The final compilation of the final multilingual text datasets is documented.

4 Datasets

First, an overview of the task itself is given, followed by an explanation of the Twitter dataset used. The approach to how tweets are assigned to buildings is explained. Furthermore, the challenge of geo-accuracy in relation to the Twitter data is discussed, and a solution for the building function classification task is offered.

4.1 Building Function Classification

The classification of building functions is a standard task in remote sensing. Often, some large areas or blocks are scrutinized to detect various land-cover and land-use classes such as vegetation or buildings [11] with high accuracies [158]. As pointed out in chapter 1, the urban structure can change over time. Large urban configurations are investigated at patch level or areas at block level with remote sensing methods only; detailed changes possibly remain undetected. This challenge is getting more vivid studying the functionality of an individual building. Figure 4.1 A depicts a residential area with green spaces and trees. No industrial plant or a large shopping mall is present in this scene. The building functions are evident (residential). Considering figure 4.1 B, a obvious functionality is not visible anymore. Even for a human, it is a hard task to identify a possible function. Is it a hotel or a university? Could it be a large multi-family house or a hospital? The answer can be found in section 6.4.2.

A conceivable way to tackle this challenge is shown in figure 4.1 C. The blue circles represent additional in situ sensor data, such as social media data that could be seen as citizen sensors. The supplementary information gained by these sensors could be helpful to associate a functionality to this building finally. In this dissertation, geo-referenced Twitter data is utilized as in situ sensors to generate additional linguistic features to support remote sensing to estimate a building function of an individual building.

Many land-use classification studies use various classes such as residential (scattered, dense), industrial, office, civic, et cetera for the task [11, 158, 15, 168]. It seems that urbanization influences commercial and residential areas [40]. When people migrate into cities, more residential areas are needed, impacting the infrastructure [169] and energy consumption [170]. Additionally, urbanization might cause a shift from agricultural toward industrial working places, or a service-oriented economy [40]. Therefore, studying commercial and residential buildings within the context of urbanization seems reasonable.

These classes mentioned above can be summarized to prototypical classes such as commercial or residential. This could be seen to reduce the complexity of the classification task at an individual building instance level. A third class, the *other* class, could solve as a container for buildings that cannot be categorized into commercial and residential. Therefore, the classes studied in this work are *commercial*, *residential*, and *other*. They

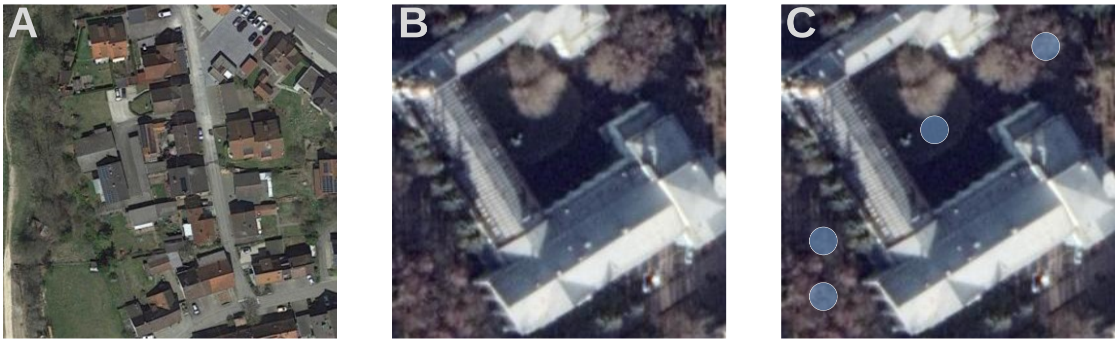


Figure 4.1: Building function classification using tweets as citizen sensors to support remote sensing. Blue circles in C represent in situ sensors. Background images © TerraMetrics 2021, Google.

are summarized from more fine-grained building functions derived from OSM building polygons (cf. sections 4.3 and 4.3.2). The following paragraphs describe the whole process of how a tweet becomes a citizen sensor to support urban building function classification.

4.2 Twitter Data

In this work, utilize social media text messages are obtained from the Twitter API v.1.1. The free access allows the download of approximately 1% of the live Twitter stream [171]. Additional filters can be set, for example, keywords or geographical filters like bounding boxes. For the current project, a bounding box covering the whole world was configured, i.e., only tweets with a geo-reference are received continuously. Twitter delivers a tweet in *JSON* file format, which allows a structured representation of the tweet's metadata. A tweet comprises many different metadata attributes such as a tweet-id, user-id, date, time, geographic information, and the text itself. All of the attributes are integrated into a so-called *data dictionary*. Normally, the coordinates of a geo-referenced tweet are stored in the data dictionary in the field `.coordinates.coordinates` which provides longitude and latitude floats. Furthermore, a tweet can also be geo-referenced when `.coordinates.coordinates` is empty. Namely, the `place` field also provides various information about the geo-reference. Here, a bounding box is stored which frames the present georeference in different granularities (cf. paragraph 4.2.1 for more detailed information). This means that a tweet can be geo-referenced without longitude and latitude information in the `.coordinates.coordinates` field. This attribute makes Twitter a widely used data resource in geospatial research (cf. chapter 3.2). The geo-referenced Twitter data was collected for approximately three and a half years for the whole world. In total, 589, 764, 252 geo-referenced tweets posted in 66 languages have been stored.

However, as mentioned in section 1, [172] has announced it will deactivate the precise geo-tagging in tweets. In the past, it was possible to embed a point coordinate of the current position in the tweet. This position was then at an individual device/person level (if the user permitted). How does this announcement impact geospatial research? The next section gives some insights into this matter by analyzing the results of a straightforward field experiment and comments on the possibilities for building function classification.

4.2.1 Geo-referencing precision using the Twitter and Instagram app

A Twitter user is still able to tag a tweet with a geo-location [173]. However, as locations, users can set the country or city, or they can choose among pre-selected locations such as a current neighborhood, a store, or a landmark (e.g., the Leaning Tower of Pisa) within a radius of approximately 200 meters of the mobile device. The locations are provided by Foursquare[1] and Yelp. Twitter notes in its help center that it is still possible to attach a precise geo-location to a tweet via the in-app camera of the iOS and Android Twitter apps. Furthermore, Twitter explicitly mentions that this information is available via the API [174]. [175] mentioned in their article that it is possible to get the point coordinates via an Instagram cross-posting on Twitter, i.e., a user can create an Instagram post and share it on his or her Twitter timeline. A brief analysis was conducted to evaluate the statement of Twitter and [175].

4.2.2 Brief field analysis of georeference precision

The aim of this short analysis was to tag a Tweet with point coordinates such that the exact location of the tweet could be identified and retrieved via the API as point coordinate. As a first step, the Twitter app (version 8.69.2) was installed on the author's mobile phone (Apple iPhone SE II, iOS 14.6), and all involved apps (iOS camera, Twitter, Instagram) were granted permission to access the phone's geo-information. Within the Twitter app privacy settings, the option *precise location* was opted-in. The Wi-Fi was deactivated by the opt-out switch in the *Settings*. To download the tweets from the author's Twitter timeline the Python package `tweepy` has been used[2].

For the analysis, Technical University of Munich (TUM) was chosen and the front lawn of the *Alte Pinakothek* was the exact tweet location (c.f. Figure 4.2) which is located in Maxvorstadt, a district of Munich, Germany. First, a tweet containing a short text sequence and a picture taken by the Twitter app's camera but without any geo-information has been posted. As expected, no polygons or points are present in the tweet JSON object's `coordinates` and `place` field. The next tweet was created the same way; however, this time, the suggested city (Munich) was included. The JSON reveals now a place field includes a string describing the place (Munich), a type (city), country (Germany), etc. and bounding box coordinates (c.f. Figure 4.2). Exact point coordinates were not present. In the third tweet, TUM was marked as a place. In the pulled JSON, no exact point coordinates of the mobile device were present; however, bounding box coordinates of the TUM main building were present. This bounding box has no surface area and cannot be seen on a plot. Therefore, the coordinates of the lower-left point were taken to visualize location (c.f. Figure 4.2). The place type was set to *poi* (point

[1] `https://foursquare.com/` [9.12.2021, 15:34]

[2] `https://docs.tweepy.org/en/stable/getting_started.html#hello-tweepy` [9.12.2021, 16:10]

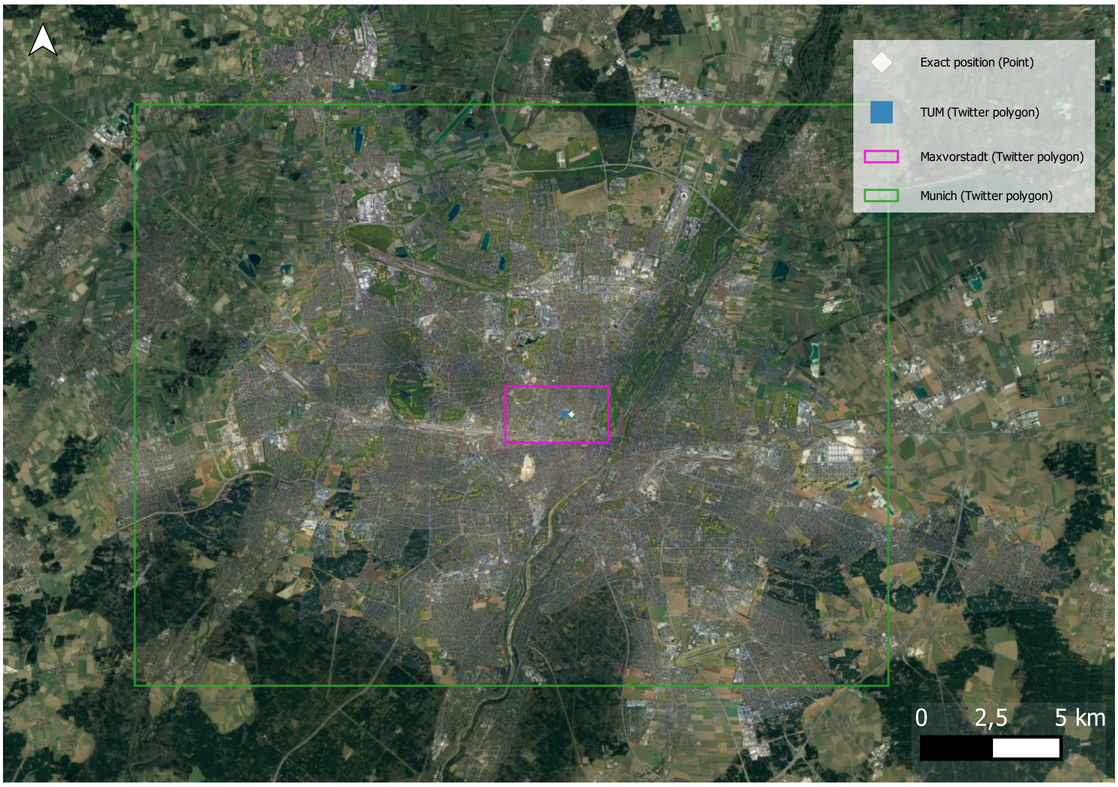


Figure 4.2: Polygons provided by Twitter. By creating the plot the TUM polygon had to be included as a point coordinate since the polygon had no surface area. Therefore, the box is a symbolic depiction of TUM's location. Background images © TerraMetrics 2021, Google.

of interest). The fourth tweet was again created in the same way, but the city district was tagged (Maxvorstadt). Again, no exact point coordinate was present but a polygon which comprises roughly Maxvorstadt was included (c.f. Figure 4.2). As a place type *neighborhood* was specified. It was not possible to tag the exact location of the mobile phone even though the Twitter documentation [174] proclaimed otherwise.

A cross-posting was created to check the statement of [175] that it is possible to receive exact point coordinates in the tweet JSON via an Instagram cross-post. Instagram suggests the same locations in the app if users want to tag an image with a place or a point of interest. Therefore, the reproduction of the tweets was straightforward. The results are shown in Figure 4.3. Now, if the tweets are queried, a point coordinate in the `coordinate` field is present. Despite the presence of a point coordinate, it is apparent in Figure 4.3 that even with a cross-post, it is not possible to receive the exact location of a tweet.

The results of this brief analysis include two significant findings and some minor findings. The first and the most critical finding is that it is not possible to receive the exact

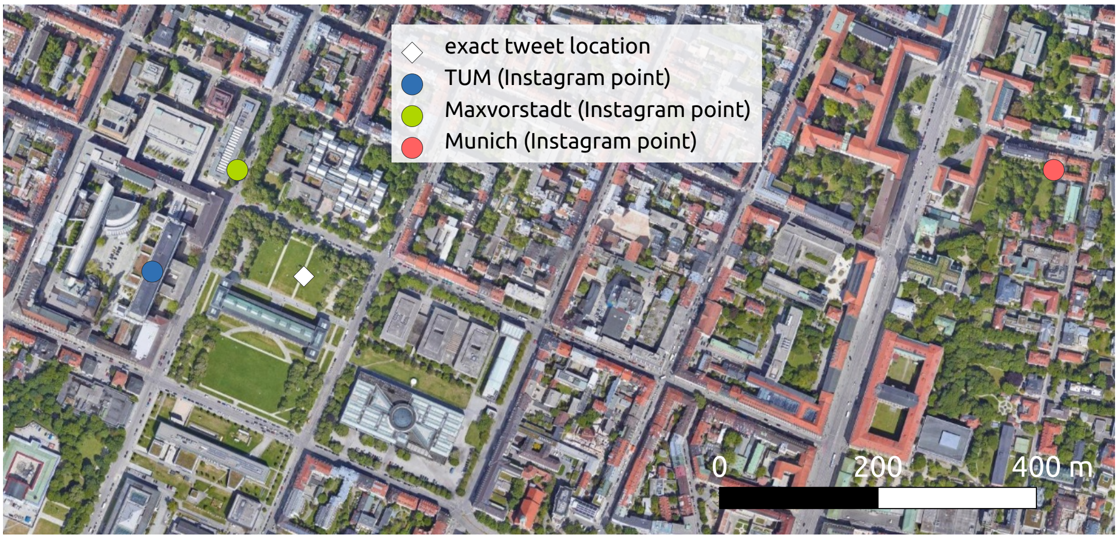


Figure 4.3: Points provided by the Twitter API. Other than merely tweeting via the Twitter iOS app, Instagram cross-posting reveals point coordinates in the retrieved tweet JSONs. However, the exact tweet position could not be retrieved via the API. Background images © TerraMetrics 2021, Google.

geo-location of a tweet regardless of the method applied (as of mid-2021). Secondly, places of interest such as TUM or city districts like Maxvorstadt can still be tagged in tweets. An Instagram cross-posting gives point coordinates of the place, district, or city, and Twitter provides a polygon without surface area at the exact coordinate of the location (e.g., TUM) and polygon coordinates for districts and cities. Furthermore, if a location is tagged via Twitter, a place type (e.g., city, point of interest, neighborhood) and an exact name of the location (e.g., Maxvorstadt) is given. A cross-post only includes the point coordinates, and as a place type, only the city is denoted no matter what type of location in the Instagram post was selected. Finally, in the Twitter app, it was not possible to select locations that were farther away than approximately 250 meters. Even when manually typing more distant locations like *Allianz Arena* or *Karlsplatz* (which is not very far from the exact tweet location), the app replied that location could not be found. On the other hand, Instagram permits the tagging of far more distant locations.

[4] investigated this issue at a data level and confirmed the findings above. They further provided detailed statistics informing about the distribution of geo-referenced attributes in Twitter data. From table 4.1 it is notable that a significant amount of tweets are posted via an official Twitter app. Those tweets are mostly tagged with a place tag. As mentioned earlier, they represent different granularity levels. The most coarse level is country level, followed by city and neighborhood level. The most exact place tag available is the *point of interest (poi)* tag. For example, landmarks, businesses, or restaurants can be flagged.

	number of tweets	total percentage
Total	25,756,667	100%
Twitter for Android	11,657,427	45%
Twitter for iPhone	9,410,305	37%
Twitter Web Client	2,982,670	12%
Twitter for iPad	506,638	2%
Twitter for Mac	12,040	0.05%
Instagram	22,151	0.09%
Foursquare (+Swarm)	2,034	≪0.01%
CareerArc 2.0	1,229	≪0.01%
Others	1,162,173	4.5%

Table 4.1: Distribution of tweet sources (e.g., apps). Mostly bots could be found in "Others". This table is obtained from [4, p. 214], Table 1.

In conclusion, even though the exact coordinates of a tweet (i.e., phone location) cannot be retrieved via the API anymore, the point or polygon coordinates of a point of interest are "exact" in the sense of an individual building level but *not* at the individual device/person level. Neighborhood and city locations are less exact [4]. Of course, the geo-privacy of the users increases with this practice. An analysis of the user's comments on the announcement of [172], reveals that the users welcomed the deactivation of this feature. For geospatial research, on the other hand, a valuable data source was decreased in geospatial accuracy. Therefore, additional measures needed to be executed to limit the impact of non-exact geo-referenced tweets like neighborhood, city, or country level.

4.2.3 Tweets used in the current Ph.D. thesis

The analysis reveals, it is clear that after mid-2019, no precise tweets at an individual device/person level are obtainable anymore. Therefore, the term *geo-referenced tweet* is related to a tweet with embedded information about an object in urban space. Since all tweets which have been collected for this work are "shipped" with a geo-reference, all tweets are considered. However, not "naïvly" anymore. Naïvely means in the context of building function classification, the hypothesis that a tweet with a point coordinate refers to the GPS location of the user. In tweets before mid-2019, it is the case but not after. That said, since it is feasible to tag landmarks or restaurants on Twitter and Instagram, it is possible that the tweet indeed carries (implicit) information about the object it is referring to. Therefore, the "real" location of a Twitter user is not needed to classify the building functions. Since a real-world location was tagged in the tweet, it could be that the tweet is referring to the tagged place even though the user is at the other side of town. In this study, it is hypothesized that a tweet text does not necessarily need explicit information about a certain building, e.g., "this is a company". Rather, a tweet including implicit information–like the presence or absence of specific vocabulary or the combination of words, e.g., "we sell high-quality apparel".

To summarize, for this study, all tweets with a point coordinate and tweets without point coordinates but with *poi* place type are considered. As pointed out earlier, such tweets incorporate also point coordinates in the form of a polygon without a surface. As also mentioned, the point coordinate tweets are not used "naïvly" anymore. Therefore, the impact of city and neighborhood-level tweets on building function classification must be limited. Section 4.5.4 provides further information.

4.3 Labeling of the Twitter data

The labeling process in this work is defined as the assignment of a tweet to one and only one building. The tweet then inherits the building function of the building. The building function is derived from OSM building polygons. To facilitate efficient geospatial queries like labeling or point-in-polygon checks as well as for attribute filtering, the streamed Twitter data have been stored in a PostgreSQL[3] database with PostGIS[4] extension [176].

4.3.1 OpenStreetMap Building Attributes

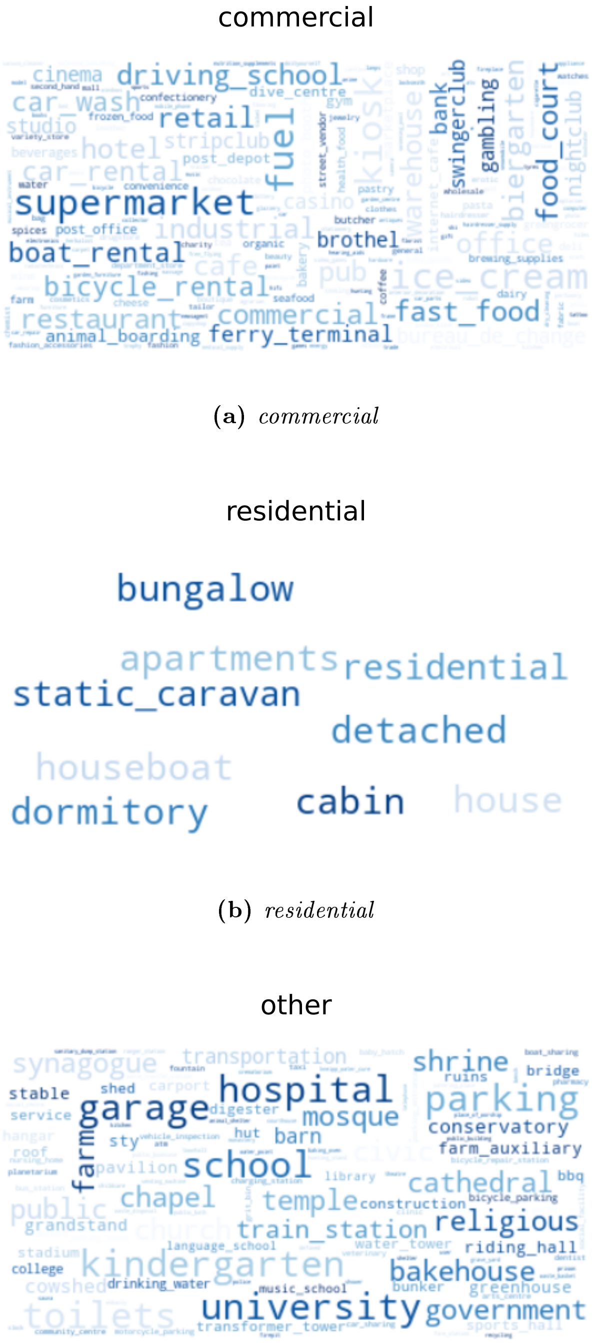
Although some cities, like Los Angeles, provide official data about parcels and their use, by far not every city publishes such data. For this reason, the tweets have been labeled with OSM building polygon attributes[5]. In more detail, building polygons include a *building tag* where several values can be associated with it. For example, `building:residential` implies that the polygon represents a residential building. To add a building polygon with more comprehensive information, users can add values to the *amenity tag*. Such amenity values determine additional functionality of a building which could be public bathrooms or Banks.

Since the building attributes provided by OSM are manifold, they were summarized to *commercial, residential, and other*. For example, the *residential* class includes tags like *bungalow* or *apartments*. The *commercial* class summarizes tags like *supermarket*, *retail*, or *fast food*. The *other* class comprises tags like *hospital*, *university*, or *religious*. In figure 4.4, wordclouds of the OSM tags are provided divided by class. Each word in the wordclouds represent a class mich was summerized.

4.3.2 Labeling process

As mentioned earlier, the labeling of the tweets is conducted via the PostgreSQL database with a PostGIS extension. PostGIS enables spatial queries and operations like distance measuring algorithms. The distance is measured via a PostGIS distance function[6] to

[3] `https://www.postgresql.org/` [9.9.2021, 13:39]
[4] `https://postgis.net/` [9.9.2021, 13:41]
[5] `https://wiki.openstreetmap.org/wiki/Key:building` [9.9.2021, 13:52]
[6] `https://postgis.net/docs/ST_DWithin.html` [10.9.2021, 10:45]



(a) *commercial*

(b) *residential*

(c) *other*

Figure 4.4: Wordclouds of summeraized individual OSM classes.

the next OSM building polygon with an OSM building attribute within a 50m limit to label the geo-referenced tweets. The distance limit of 50m has been chosen since a larger construction could have a plaza surrounding it and further include tweets in scattered residential areas. After the labeling process, some buildings exhibit more than one tweet. This state is from now referred to as $1:n$ relationship. A $1:n$ relationship means that one building n tweets can be assigned.

4.4 Areas of Interest

[177] proposed a benchmark dataset for locale climate zone [129] classification within the context of the *So2Sat*[7] project which is called *LCZ42*. In this dataset, 42 urban areas are considered which are distributed throughout all continents and cultural zones (c.f. figure 4.5). In addition to commonly investigated Western cities like New York, Paris, or Berlin cities from Southeast Asia like Jakarta or African metropolises like Nairobi have also been taken into account. Since geo-referenced Twitter data from the whole world is streamed, sub-sampling selection of the 42 urban areas is straightforward.

By using the city polygons proposed in [177], a subset from the large worldwide Twitter set was taken. From the overall dataset containing $589,764,252$ geo-referenced tweets, $113,059,673$ are located in the areas of interest (19.17%). This number can be further subdivided:

- 82,882,884 with point coordinates (longitude, latitude)
- 30,176,789 without coordinates but place tag point of interest (poi)

Since not every tweet can be assigned to a building that is within a distance of 50m, a significant data loss after the labeling process is expected (cf. section 4.3.2 for the labeling process). Therefore, the dataset size has reduced:

- 82,882,884 (points) to 22,956,700
- 30,176,789 (poi) to 3,729,446
- 66 languages to 65 (minus Dhivehi, dv)

In addition to the cultural and architectural diversity of the cities, the tweets received are written in various languages. Twitter officially supports 70 languages[8] and a token for unidentified languages *(unk)*. The dataset used in this dissertation is referred to as *LCZ42* to stick with the original designation used in [177].

4.5 Preparation of the Twitter Text datasets

Before the dataset for the building function classification task can be composed, some individual steps must be executed (cf. figure 4.6). The following paragraphs describe the

[7] `https://www.asg.ed.tum.de/sipeo/projects/so2sat/` [14.1.2022 12:27]

[8] `https://developer.twitter.com/en/docs/twitter-api/enterprise/powertrack-api/guides/operators`

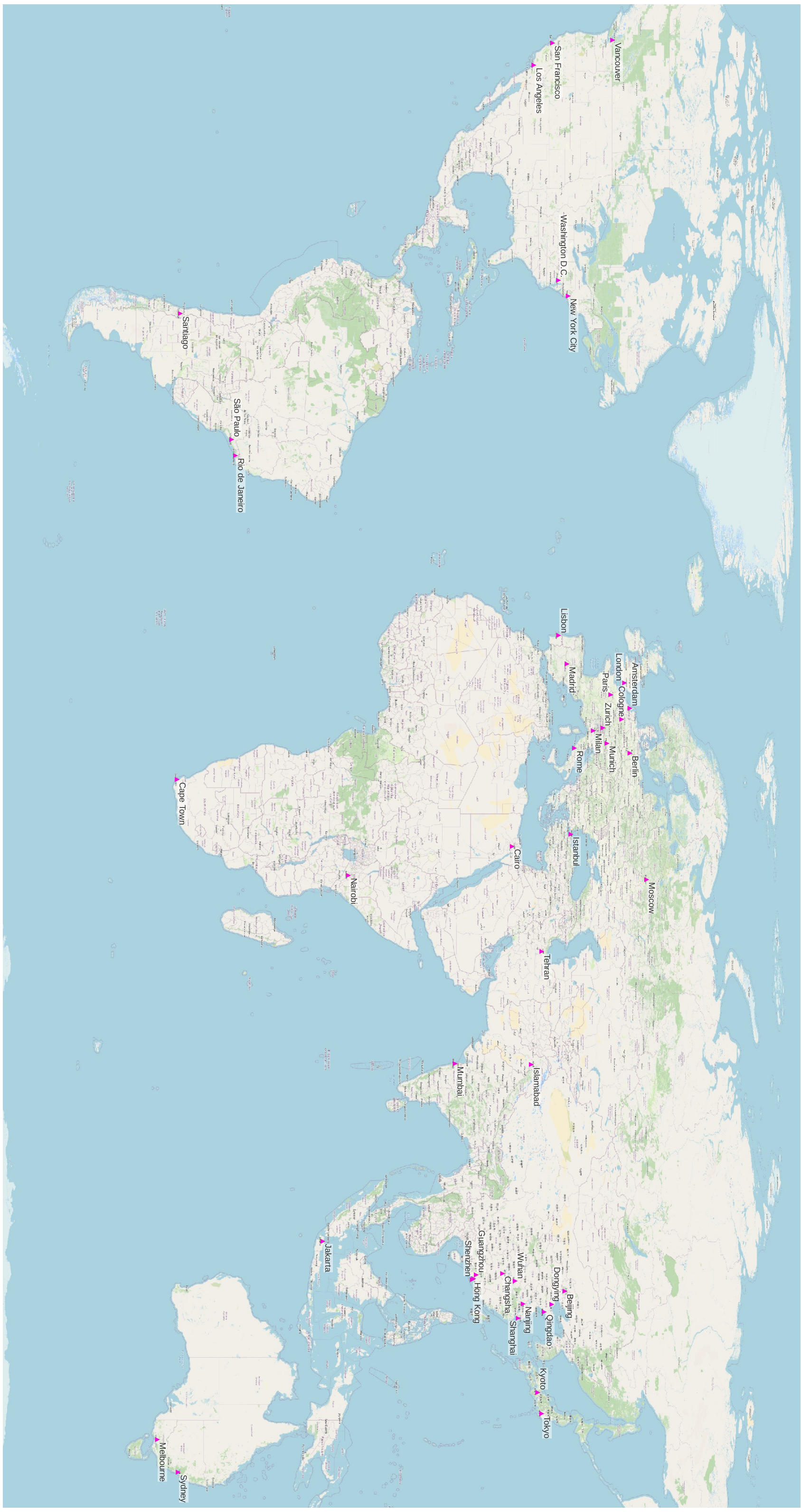


Figure 4.5: Distribution of the cities analysed. Background image © OpenStreeMap.org 2021

City	Region	Tweets	Buildings	Languages	Users
Amsterdam	Europe	562,642	42,039	48	60,278
Beijing	East Asia	24,889	1,596	42	8,889
Berlin	Europe	1,029,228	15,672	51	57,611
Cairo	transcontinental	74,817	631	42	10,880
Cape Town	Africa	128,660	2,201	43	19,220
Changsha	East Asia	5,358	243	27	1,846
Cologne	Europe	168,022	6,097	48	15,340
Dongying	East Asia	11	2	4	6
Guangzhou	East Asia	24,397	1,486	43	7,439
Hong Kong	East Asia	227,533	7,988	49	34,928
Islamabad	South Asia	8,288	472	39	3,406
Istanbul	transcontinental	3,491,443	17,482	51	214,292
Jakarta	Southeast Asia	615,430	10,964	43	126,499
Kyoto	East Asia	1,300,719	10,953	42	117,887
Lisbon	Europe	90,202	5,077	47	20,390
London	Europe	2,442,218	47,334	61	307,288
Los Angeles	North America	3,463,727	302,152	54	309,240
Madrid	Europe	4,48,108	16,683	46	86,772
Melbourne	Oceania	158,000	6,499	48	24,781
Milan	Europe	318,222	16,675	47	44,005
Moscow	transcontinental	655,484	24,549	51	49,176
Mumbai	South Asia	97,834	3,416	48	24,897
Munich	Europe	96,772	4,058	48	23,032
Nairobi	Africa	53,587	1,288	38	9,811
Nanjing	East Asia	30,516	424	38	2,895
New York City	North America	1,927,129	23,880	56	245,810
Paris	Europe	542,374	9,979	60	111,617
Qingdao	East Asia	4,130	334	28	1,585
Rio de Janeiro	South America	660,042	5,695	40	147,329
Rome	Europe	125,897	3,200	47	28,903
San Francisco	North America	945,607	6,206	50	99,603
Santiago	South America	1,019,367	14,475	40	51,168
São Paulo	South America	469,477	9,178	40	77,053
Shanghai	East Asia	41,377	1,852	41	12,235
Shenzhen	East Asia	4,145	209	24	1,761
Sydney	Oceania	212,931	6,301	50	36,048
Tehran	Middle East	52,078	1,484	46	7,801
Tokyo	East Asia	4,252,510	11,578	53	271,730
Vancouver	North America	332,699	4,673	46	32,507
Washington D.C.	North America	509,472	8,102	50	76,980
Zurich	Europe	64,638	2,764	46	10,501
Wuhan	East Asia	6,166	359	31	2,155

Table 4.2: Overview of the used cities.

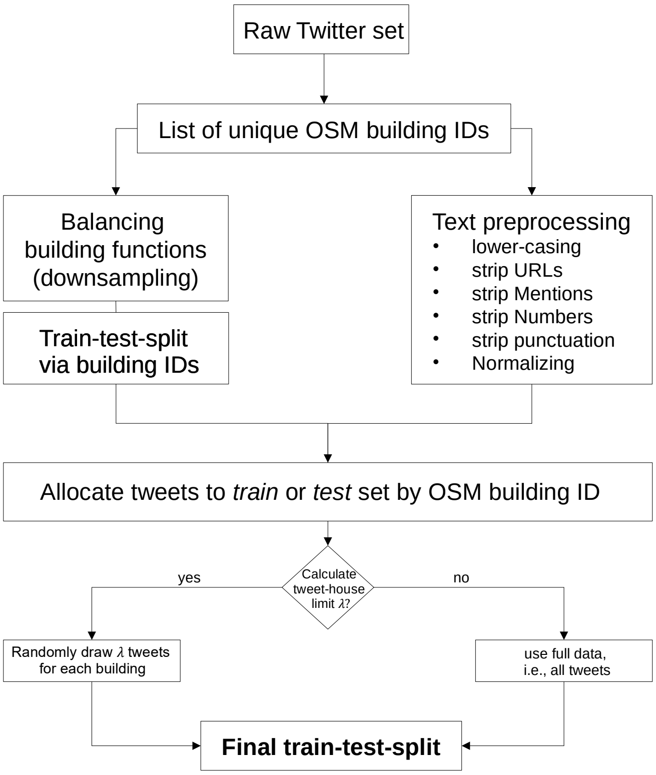

Raw Twitter set
List of unique OSM building IDs
Balancing building functions (downsampling)
Train-test-split via building IDs
Text preprocessing
• lower-casing
• strip URLs
• strip Mentions
• strip Numbers
• strip punctuation
• Normalizing
Allocate tweets to *train* or *test* set by OSM building ID
Calculate tweet-house limit λ?
yes
no
Randomly draw λ tweets for each building
use full data, i.e., all tweets
Final train-test-split

Figure 4.6: Composition process of the Twitter dataset.

train and test split via OSM building IDs, the subsequent data balancing process, and the text preprocessing. Finally, the preparation of the remote sensing images is explained.

4.5.1 Text preprocessing

After querying the data from the database, some initial text preprocessing steps are executed (cf. section 2.4.2). First, a building-wise text de-duplication is performed to make sure that a tweet text cannot appear twice. Every single tweet text of a building is concatenated with the corresponding building OSM ID to conduct the de-duplication. Additionally, the URL was stripped since many tweets only differ by URL. This string is stored in a Python *set*[9]. In Python, a set is an unordered container that includes hashable objects like text sequences. If a tweet text combined with the OSM id is already in the set, the data point containing this tweet will be rejected.

The monolingual text preprocessing steps include:

1. lower-casing
2. removing of numbers, mentions, URLs

[9] `https://docs.python.org/3.7/library/stdtypes.html#set-types-set-frozenset` [7.9.2021, 17:59]

3. removeing punctuation except apostrophes and hyphens
4. tokenization of the monolingual text (cf. section 2.4.3)
5. deletion of empty tweets

For more details about the preprocessing steps, see section 2.4.2.

A special preprocessing is required for (multilingual) DistilBERT models. Ready-to-use preprocessing models can be obtained which arrange the text in the special representation BERT requires internally (cf. section 2.4.3). For further information, please consider the BERT paper [92], paragraph 2.4.3, and the *huggingface* `transformer` library[10].

4.5.2 Balancing of the Dataset

Balancing the desired dataset before most machine learning tasks is an essential step. An over-representation of a class could lead to over-fitting, whereas an under-represented class could result in non-sufficient training, which results in unsatisfactory classification performance [21]. Hence, a balancing process aims to create a well-adjusted dataset to prevent over- and under-fitting of the classification algorithm. Various techniques can be used to perform dataset balancing, for example, by the application of oversampling, [178]. In contrast, under-sampling reduces available classes to the quantity of the minority class. Even though under-sampling is straightforward, good performance can achieve [178]. Hence, this dissertation utilizes under-sampling for balancing.

However, since several cities with different class distributions are used, the balancing process should also consider the distribution of classes per city. Therefore, a balancing algorithm was applied that also takes the city-wise class count into account. First, the class with the lowest support, i.e., buildings, is identified. This value is denoted as l. Next, the global target value β is calculated, which represents the optimal number of samples per class and city. l is divided by the total number of cities n to determine this number.

$$\beta = \frac{l}{n} \tag{4.1}$$

The calculated number β is the optimal downsampling threshold per class and city. However, some cities overshoot β easily. Therefore, the total number of buildings per class is calculated. After that, two possibilities are available: first, if the number samples s of a class is undershooting or equal to β, all samples are kept (cf. equation (4.2)).

$$\{s|s \leq \beta\} \tag{4.2}$$

If $s < \beta$, the difference of s and β is stored. Second, if

$$\{\beta|s > \beta\} \tag{4.3}$$

is true, then random downsampling takes place by drawing β samples (city-wise). The new samples, i.e., buildings, are remembered. After executing all these steps for all cities,

[10] `https://huggingface.co/docs/transformers/v4.14.1/en/model_doc/bert#transformers.BertTokenizerFast` [16.12.2021 10:04]

		tweets train	test	buildings train	test
mono.	*commercial*	1,223,785	298,986	40,388	10,012
	other	1,941,312	61,8235	40,404	9,996
	residential	361,775	75,284	40,168	10,232
	total tweets	3,526,872	992,505	120,960	30,240
multi.	*commercial*	3,762,683	1,153,615	69,662	17,278
	other	6,912,596	138,4165	69,424	17,516
	residential	785,770	202,612	69,570	17,370
	total tweets	11,461,049	2,740,392	208,656	52,164

Table 4.3: Final class distribution and train-test split.

the unused buildings are brought back. How many buildings are missing to approximate β independent of the city is now calculated. Lastly, the buildings are class-wise randomly drawn to "fill" the class count.

The classes are more evenly distributed amongst cities. The balancing process now yields a list composed of OSM id, city name, and building function. This list is used to create a train-test split. The next paragraph provides information about how the train-test split is produced.

4.5.3 Train and Test Split

The goal of a machine learning classification task is usually the categorization of a datapoint x to a target y. [21] express this as "The ability to perform well on previously unobserved inputs is called generalization" (p. 107). Before an acceptable generalization performance is achieved, a collection of data points is required to train the model. A set of unseen data is used to evaluate the trained model's performance. Therefore, dividing data into a train and test set is common in machine learning. Furthermore, the strict separation of training and testing data prevents the "leakage" of training data into the evaluation process, which affects the classification process negatively by biasing the model towards already seen data [179].

In this work, the train-test split is created via the OSM IDs obtained in the downsampling process described in paragraph 4.5.2 above. 80% of the data is assigned to the train set and 20% for the test set. For the validation set, 10% of the training data are subsampled. After the balancing process (cf. section 4.5.2), an overview of the created train-test split can be found in table 4.3. It can be seen that the number of tweets belonging to the *residential* class is smaller compared to the other two classes. The classification may be biased in favor of the *commercial* and *other* classes.

4.5.4 Dealing with imprecise Twitter geo-references during classification

Landmarks or locations like the White House, the Eiffel Tower, or the Brandenburger Tor usually have many tweets clustered around them. As discussed in section 4.2.2, tweets can have different levels of geo precision (e.g., GPS position vs. city level coordinate). For example, tweets at a city level could be located in the center of the city place polygon. During the labeling process, a building that happens to be next to the center might be assigned to the city level tweet. After the labeling process, a $1 : n$ relationship between buildings and tweets is generated. For reiteration, one building can be assigned to n tweets, which are closer than 50m. Here, the question arises of how the influence of landmark buildings and the Twitter geo-inaccuracy on building function classification can be limited.

To resolve this, a method is proposed to reduce the impact of buildings assigned to a high number of tweets on classification. The intuition behind the idea is to determine the tweet-house limit λ. The tweet-house limit expresses the maximum allowed number of tweets per building. It is calculated by determining the mean number of tweets per building. Different lower and upper bounds are defined to prevent landmark buildings with a very high tweet count biasing the computation of the mean. A lower bound can be described as the minimum possible number of tweets per building and an upper bound as the maximum of allowed tweets. The bounds are represented by elements p_l and p_u which are calculated. The bounds are calculated of a set of percentiles P_l and P_u.

$$\begin{aligned} \mathrm{P}_l &= \{1, 2, 5, 10, 15, 20, 25\} \\ \mathrm{P}_u &= \{99, 98, 95, 90, 85, 80, 75\} \end{aligned} \tag{4.4}$$

Then, the count of tweets per building is determined and every building which cannot fulfill condition (4.5) are not considered in calculating the mean number of tweets per building.

$$\{x | p_l \leq x \leq p_u\} \tag{4.5}$$

Where x is the total number of tweets of a specific building. If a building does not fulfill condition 4.5, it is not considered for calculating λ. Following this approach, nine monolingual and nine multilingual datasets are generated with different minimum and maximum tweet counts per building—also, the λ, i.e., the tweet-house-limit changes. Table 4.4 documents the calculated tweet-house limits, i.e., λ values using different percentile limits. The values decrease the stricter the thresholds are set. In the end, the smallest λ value is 3, while the largest is 30. For example, the dataset 9-95 has a λ value of 10. This follows that a building can only "have" a maximum of 10 tweets. If a building has fewer tweets as specified maximum, it can "keep" all tweets. However, if the total number of tweets is higher than the λ upper bound, the total number of tweets x is randomly downsampled until $x \leq \lambda$. In contrast, the full dataset has no limitations. All tweets are used in this dataset.

By using this method, 18 (9 monolingual and 9 multilingual) different datasets are created for classification. The results are investigated to find out if such a limitation of data can be useful for the building function classification task. For text classification

	Monolingual		Multilingual	
percentiles	mean tweets/building	λ	mean tweets/building	λ
full	29.89	∞	54.45	∞
100	29.89	30	54.45	54
1-99	19.27	19	25.04	25
2-98	15.11	15	19.50	20
5-95	9.95	10	12.64	13
10-90	6.51	7	8.18	8
15-85	4.73	5	6.07	6
20-80	3.84	4	4.77	5
25-75	3.18	3	3.97	4

Table 4.4: The calculated λ values to controll the maximum tweets per building. The λ values correspond to the mean number of tweets per building.

results, see tables 5.2 and 5.3 in section 5.6. To identify the datasets through out the thesis, they are referred to by the percentile limit, e.g., *5-95* or *10-90*.

4.5.5 Final collocation of the text dataset

Now, the approaches discussed in sections 4.5.2, 4.5.3, and 4.5.4 are applied to the data. As pointed out in section 4.3.2, every tweet have been assigned to a building by tagging it with the OSM ID and the correspondent building function label. All steps are the same for the mono and multilingual dataset.

The class-wise balancing introduced in 4.5.2 is performed, and with the resulting balanced class list, the train-test-split is generated. To create the train-test-split, every unique OSM ID in the balanced class list is collected and split into a 80 : 20 ratio using the method mentioned in paragraph 4.5.3. 80% of the OSM IDs are allocated to the training dataset and 20% to the test set, respectively. 10% of the data is sampled from the training dataset for validation during the training phase.

An additional aspect of building function classification with Twitter data is the unequal distribution of tweets and buildings. Remember, only tweets that are within a distance of 50m to its closest building are used for this work (cf. paragraph 4.3). Depending on the configuration of the urban area (e.g., dense or scattered) or the nature of a building (e.g., home or landmark), the number of tweets associated with a building within the selected 50m distance can vary. For instance, in London, a building with 584, 296 tweets has been found (OSM ID: 1101888552). It is a gift shop next to Trafalgar Square; therefore, as discussed in section 4.5.4, the total amount of tweets per building is limited to lower a possible bias of such buildings on classification.

All the steps depicted in figure 4.6 have now been executed, and a train-test split ready for classification has been created. The final distribution of tweets per (multilingual)

		Monolingual		Multilingual	
		tweets train	test	tweets train	test
commercial	*full*	1,223,785	298,986	3,762,683	1,153,615
	100	320,759	78,381	870,130	213,548
	1-99	257,863	63,464	598,415	147,600
	2-98	228,769	56,420	531,435	131,308
	5-95	184,168	45,577	417,995	103,377
	10-90	150,553	37,293	312,848	77,302
	15-85	123,176	30,588	260,808	64,456
	20-80	107,198	26,654	231,417	57,194
	25-75	89,005	22,155	198,967	49,189
other	*full*	1,941,312	618,235	6,912,596	1,384,165
	100	329,020	81,528	852,574	214,167
	1-99	262,052	65,129	584,126	146,974
	2-98	231,665	57,605	518,881	130,667
	5-95	185,637	46,128	408,196	102,889
	10-90	151,216	37,509	305,688	77,079
	15-85	123,302	30,569	255,014	64,345
	20-80	107,134	26,552	226,458	57,169
	25-75	88,841	22,030	195,081	49,243
residential	*full*	361,775	75,284	785,770	202,612
	100	140,962	35,411	377,299	94,003
	1-99	123,034	30,996	297,278	74,119
	2-98	114,172	28,867	275,114	68,724
	5-95	99,995	25,454	234,839	58,649
	10-90	88,621	22,646	193,715	48,361
	15-85	78,805	20,149	171,604	42,793
	20-80	72,696	18,595	158,467	39,480
	25-75	65,303	16,682	143,295	35,742

Table 4.5: Final class distribution and train-test split.

dataset created under consideration of the calculated tweet-house limitation using the λ value can be seen in table 4.5. It is important to note that between the buildings and the tweets in the dataset is a $1 : n$ relationship. I.e., that n tweets can be assigned to a specific building. For example, if TUM is in the test set, there could be 12 tweets assigned to the building id of TUM.

4.6 Training Data for the Multilingual fastText Embedding

As pointed out in section 1.1, how a diverse language pool in urban areas can be represented best for building function classification. To challenge this reserch question, a multilingual Twitter fastText word embedding is trained for the building function classification task. For this, a geo-referenced Twitter data subset from September 2021 was used. Specifically, data from calendar weeks 38 and 39 (Thursday to Thursday, seven days) have been considered. This period is not included in the LCZ42 Twitter datasets created above. The languages considered for the embedding have at least a share of 1% of the data. Therefore, only English, Turkish, Japanese, Portuguese, Indonesian, German, French, unidentified, Russian, Dutch, and Italian. These are the top eleven languages of the LCZ42 cities and together they are accounting for 91,97% of the tweets (cf. table 4.6).

The focus of creating this embedding is not on a transfer of semantics from one language to another. This means, for example, that the Spanish word *universidad* and the similar, but not identical, German word *Universität* are not necessarily adjacent in the vector space. However, it is quite possible with languages more similar to each other, like German and Dutch. The aim is more directed to generate a multilingual text representation where languages are clustered to generate their own feature space for each language. Such a representation could increase the quality feature representations of words that are misspelled like *Univerität* vs. *Universität*.

For languages like English or German, the same preprocessing steps are applied for the text classification dataset as mentioned above. However, no specific tokenization was performed. All words are then separated by whitespace. For the Japanese tweets, however, a slightly different approach was used. Since Asian languages like Japanese, Chinese, or Korean do not separate words with whitespace, the words must be split in a different way. In this work, the Python library *Janome*[11] v.0.4 was used to tokenize Japanese strings. It utilizes the MeCab[12] dictionary `mecab-ipadic-2.7.0-20070801`[13] dictionary including the Japanese new era (Reiwa) dictionary.

After the preprocessing step, $14,044,361$ multilingual tweets are available for training. For the final word embedding training, the data was roughly sorted into language branches[14]. First, Proto-Indo-European languages [180] are filled into the dataset. Starting with English, the biggest portion is the first language, followed by German and Dutch

[11] `https://mocobeta.github.io/janome/en/` [8.12.2021, 17:01]

[12] `https://taku910.github.io/mecab/` [visited 8.12.2021, 17:08]

[13] `http://jaist.dl.sourceforge.net/project/mecab/mecab-ipadic/2.7.0-20070801/mecab-ipadic-2.7.0-20070801.tar.gz` [8.12.2021, 17:06]

[14] `https://www.mustgo.com/worldlanguages/language-families/` [8.12.2021, 22:11]

language	code	count	%	language	code	count	%
English	en	8,522,616	45.06	Lithuanian	lt	4,745	0.03
Turkish	tr	2,795,470	14.78	Slovenian	sl	4,231	0.02
Japanese	ja	2,583,070	13.66	Vietnamese	vi	3,577	0.02
Portuguese	pt	846,849	4.48	Latvian	lv	3,158	0.02
Indonesian	in	459,482	2.43	Bulgarian	bg	2,985	0.02
German	de	457,710	2.42	Icelandic	is	2,685	0.01
French	fr	427,598	2.26	Marathi	mr	1,529	0.01
Undefined	und	360,958	1.91	Urdu	ur	1,431	0.01
Russian	ru	345,038	1.82	Hebrew	iw	800	0.004
Dutch	nl	316,202	1.67	Greek, M.	el	680	0.004
Italian	it	279,184	1.48	Serbian	sr	559	0.003
Chinese	zh	99,631	0.53	Bengali	bn	256	0.002
Finnish	fi	64,707	0.34	Nepali	ne	248	0.001
Estonian	et	61,419	0.32	Pushto	ps	95	0.0005
Tagalog	tl	56,221	0.30	Tamil	ta	84	0.0004
Arabic	ar	47,323	0.25	Panjabi	pa	76	0.0004
Danish	da	43,824	0.23	C. Kurdish	ckb	76	0.0003
Catalan	ca	34,394	0.18	Armenian	hy	63	0.0002
Persian	fa	29,177	0.15	Gujarati	gu	35	0.0002
Haitian	ht	26,349	0.14	Sindhi	sd	34	0.0002
Romanian	ro	21,966	0.12	Sinhala	si	22	0.0001
Welsh	cy	15,481	0.08	Lao	lo	22	0.0001
Polish	pl	14,357	0.08	Burmese	my	21	0.0001
Korean	ko	13,038	0.07	Malayalam	ml	21	0.0001
Swedish	sv	11,855	0.06	Georgian	ka	21	0.0001
Czech	cs	11,505	0.06	Amharic	am	16	0.0001
Norwegian	no	11,127	0.06	Telugu	te	14	0.0001
Thai	th	10,728	0.06	C. Khmer	km	9	0.0001
Hindi	hi	10,448	0.06	Kannada	kn	8	0.00004
Basque	eu	7,883	0.04	Tibetan	bo	5	0.00003
Hungarian	hu	7,671	0.04	Oriya	or	2	0.00001
Ukrainian	uk	6,547	0.03	Uighur	ug	1	0.00001

Table 4.6: Language distribution. The language abbreviations in the *code* column are encoded in ISO 639-1.

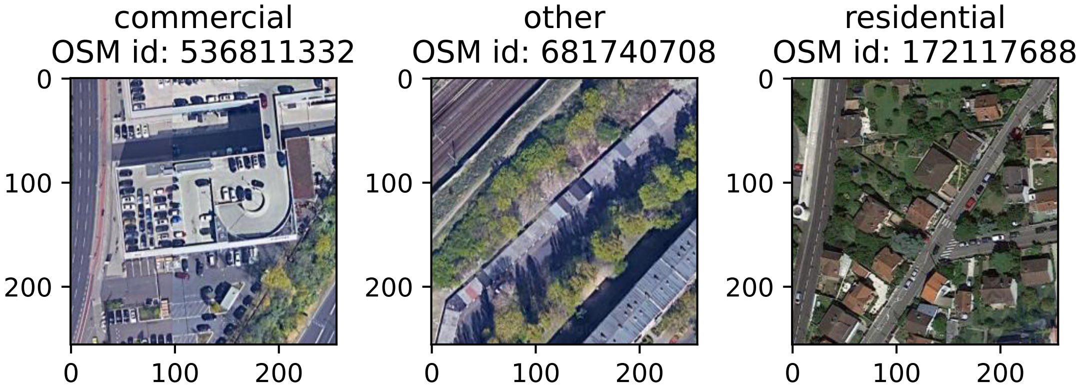


Figure 4.7: Examples of Google Maps very high resolution remote sensing aerial images. Background images © TerraMetrics 2021, Google.

(Germanic branch). Then, for the Romance branch, Portuguese, French, and Italian. After that, Russian (East Slavic), Turkish (Altaic languages), Indonesian (Austronesian), and Japanese (Altaic languages, disputed) are placed in the dataset. Finally, tweets "unknown" languages are placed into the dataset (mostly emojis). In total, the dataset has approximately 270 million words and 713, 099 unique words. The details of the training and the used hyperparameters can be found in section 5.4. The results are documented in section 5.6.

4.7 Preparation of the Aerial Image Dataset

Very high-resolution aerial images (VHR) provide a detailed depiction of an area of interest. However, such images are expensive to obtain. For this reason, the approach of [14] is followed. They used Google VHR images for *ex situ* building function classification obtained from the Google Maps satellite layer. The images are provided in WGS84 coordinate system obtainable in different zoom levels (up to 22)[15] and a tile size of 256×256. In this work, zoom level 18 is preferred to a spatial resolution of $0.48m$ in the area of interest in this work. The *ground sample distance gsd* on the desired zoom level z and a latitude lat would be computed with the following equation.

$$gsd(z, lat) = \frac{2\pi r_E \cos(lat)}{2^{(z+8)}} \tag{4.6}$$

Where r_E defines the equatorial radius of $6,378,137m$ [16]. Based on the OSM IDs of the Twitter text dataset, the building polygons of the buildings are obtained and the centroid calculated. Based on the centroid, Google images are downloaded. Table 4.7 shows the

[15] https://developers.google.com/maps/documentation/javascript/coordinates [9.12.2021, 15:32]
[16] https://wiki.openstreetmap.org/wiki/Zoom_levels [9.12.2021, 16:07]

	Monolingual		Multilingual	
	train	test	train	test
commercial	40,388	10,012	69,662	17,278
other	40,404	9,996	69,424	17,516
residential	40,168	10,232	69,570	17,370

Table 4.7: Distribution of Google VHR images.

distribution of the images. For every building, one image could have been downloaded. The single tiles have been compounded around the centroid to images with an extent of 256 × 256 (cf. figure 4.7).

4.8 Summary

This chapter discussed the Twitter data itself and the geo-reference accuracy issue (cf. section 4.2.1) and showed a possible approach on how to deal with it for the building function classification task (cf. section 4.5.4). The areas of interest, initially proposed by [177] that reflect a variety of cultural zones have been introduced (cf. section 4.4). Furthermore, the data labeling process of the tweets via OSM building function tags was showed and explained (cf. section 4.3). Also, the composition of the test dataset was discussed (cf. section 4.5). It included steps like text preprocessing, a train-test-split, and a data balancing process. Additionally, information about the fastText multilingual word embedding training dataset was given. Finally, the remote sensing image dataset composed of Google Maps satellite images was explained (cf. section 4.7).

The tweets have been prepared for their function as citizen sensors to support building function classification. The data is now ready for analysis. The upcoming chapter documents the results of the building function text classification task using the monolingual and multilingual datasets introduced above.

5 Building Function Classification with (multilingual) Linguistic Features

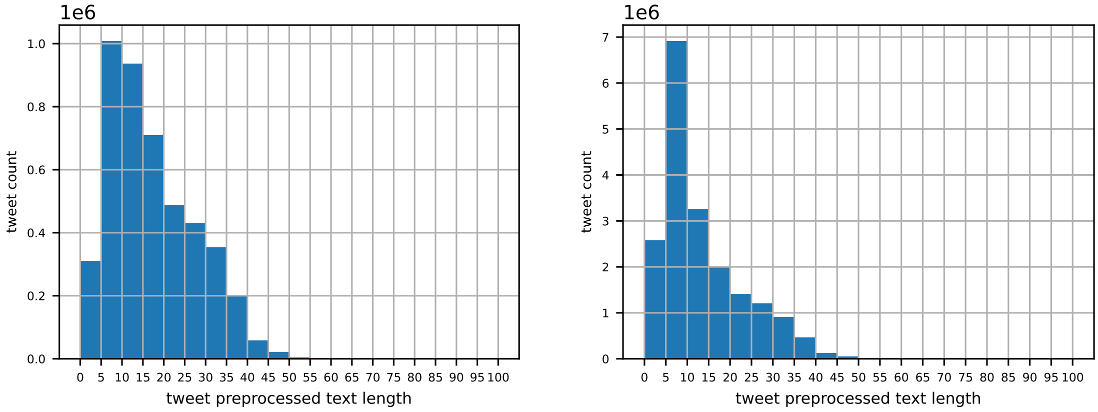
In this section, the text classification procedure is explained. All the results are discussed in more detail. First, the baseline results created with TF-IDF and a multinomial Naïve Bayes are discussed. After that, the procedure of creating word representations for the deep learning model configurations is explained and followed by the description of the training process. Also, the different approaches for monolingual and multilingual are shown. Afterward, the building function classification task results for both language modalities are discussed in detail. Subsequently, an in-depth feature analysis is given. Finally, the findings are discussed and summarized.

This section describes the building function classification using linguistic features from Twitter text messages. The main experiment is divided into five sub-experiments according to the input features and the classifiers that are used, namely, building function classification with:

1. a Naïve Bayes baseline with TF-IDF features,
2. an LSTM network using *virgin* word embedding which is trained on-the-fly,
3. an LSTM network fed with word vector sequences from a pre-trainend English fastText embedding,
4. an LSTM network fitted with word vector sequences from a self-trained multilingual Twitter fastText embedding,
5. and contextualized sentence embedding vectors from (multilingual) BERT.

It will be investigated which data representation will lead to the best classification performance. As pointed out in section 4.5.5, 9 monolingual and 9 multilingual datasets have been compiled. Each of them includes a different λ value, i.e., tweet-house limit, computed by different percentile thresholds (cf. section 4.5). The target is to find out what impact the limitation of tweets per house, i.e., a more balanced dataset has. Therefore, for every dataset, an individual model is trained. The text preprocessing and described in section 4.5.1.

The tweets of a building are randomly fed to the classifiers one by one to avoid exploding sequence lengths. This process is based on the fact of the $1 : n$ relationship of tweets to buildings. In theory, thousands of tweets can belong to a single (landmark) building in an unlimited dataset. If all tweets of that building are pooled, e.g., [181], the text sequence for the neural network would be enormous. As it was explained in section 2.3,



(a) Monolingual full dataset

(b) Multilingual full dataset

Figure 5.1: Histogram of the tweet text lengths of the mono- and multilingual datasets.

very long sequences hinder classifiers from performing well. Therefore, the tweets of a building are shown individually during training and inference. The maximum sequence length of a tweet was set to 40. The majority of the preprocessed tweets are between 1 and 40 tokens long. In the monolingual full dataset 98.60% and in the multilingual 99.29% of the tweets having this length (cf. figures 5.1a and 5.1b). All sequences are truncated with more than 40 tokens and padded with zeros to guarantee equal sequence lengths if a sequence length $<$ 40 tokens. BERT models, however, receive unlimited sequence lengths.

For the text classification, a Long Short-Term Memory (LSTM, cf. section 2.3.2) model is used. Because the classification itself is for all word embedding approaches the same, identical training settings are used if not explicitly mentioned otherwise. As output, a softmax fully-connected layer was added. Adam [182] was used as the optimizer, and as loss-function, categorical cross-entropy was employed. Furthermore, 10% of the training set were separated for validation.

5.1 Baseline with TF-IDF and Naïve Bayes

The first analysis serves as a baseline for the subsequent text classification task. It concentrates on traditional IR and machine learning algorithms. In more detail, the classic term weighting method TF-IDF is used (cf. section 2.5.2) and for classification a Naïve Bayes classifier [183, 184, 20]. Both algorithms are provided by the widely used Python machine learning library `scikit-learn` [185].

First, the text is transformed into a sparse representation using the TF-IDF algorithm. A lower word-count limit of 100 occurrences and an upper document frequency limit of 95% were set to limit the vocabulary. This ensures that typos and rare words are filtered out to a certain extent. Additionally, words that occur in more than 95% of the tweets

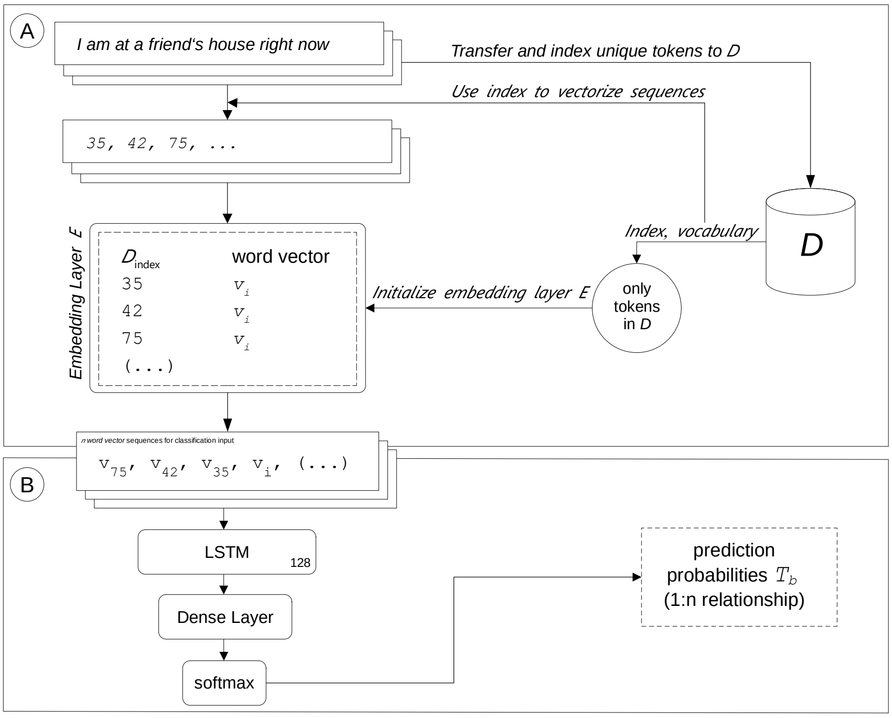

A
I am at a friend's house right now
Transfer and index unique tokens to D
Use index to vectorize sequences
35, 42, 75, ...
Embedding Layer E
D_{index}
word vector
35
v_i
42
v_i
75
v_i
(...)
Index, vocabulary
D
Initialize embedding layer E
only tokens in D
n word vector sequences for classification input
v_{75}, v_{42}, v_{35}, v_i, (...)
B
LSTM
128
Dense Layer
softmax
prediction probabilities T_b
(1:n relationship)

Figure 5.2: Text classification workflow of the virgin embedding. *A* shows the process of creating a look-up table and the transfer of word sequences into word vector sequences. *B* depicts the actual text classification part.

are considered as stopwords. To classify the text features, a multinomial Naïve Bayes was implemented using scikit-learn as well.

5.2 Word Embedding from scratch: the Virgin Word Embedding

The first approach includes a virgin, i.e., an "empty" embedding layer. It is initialized and trained with vocabulary derived from the tweet corpus. In other words: a new embedding is trained from scratch to generate 128-dimensional word vectors. The vocabulary size is limited to $35,000$ top words of the monolingual corpus. Instead of generating TF-IDF weights, all words are indexed with an integer (cf. section 2.5.4). The embedding generated by a Keras layer which is placed right before the network (cf. figure 5.2). The embedding layer is used to produce word vector sequences of the tweets and is "live" trained during the classification process. The word vector sequences are fed into a classic LSTM network (cf. section 2.3). The methods to vectorize the text and the implementation of the LSTM network have been taken from the machine learning and neural network framework *TensorFlow* [186]. In the following, this monolingual model is now referred to as *virgin embedding or LSTM-V*.

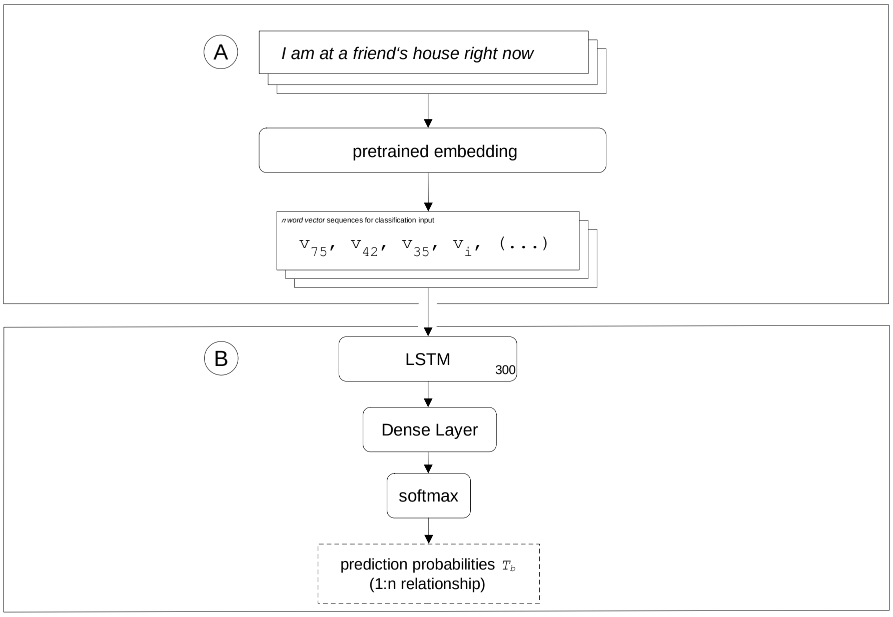

A
I am at a friend's house right now
pretrained embedding
n word vector sequences for classification input
v75, v42, v35, vi, (...)
B
LSTM
300
Dense Layer
softmax
prediction probabilities T_b
(1:n relationship)

Figure 5.3: Text classification workflow of the fastText embedding (mono- and multilingual). *A* shows the word vector querying from the pretrained fastText model. *B* depicts the actual text classification part.

5.2.1 Training

The embedding layer E of the virgin embedding is the first layer in the neural network construction with an embedding dimension of 128. As pointed out above, it accommodates the embedding matrix. For the whole experiment, its weights, i.e., the word vectors, are trainable. After E, an LSTM layer is arranged with 128 units (size of the word embedding) and 0.25 dropout. The virgin embedding model was trained for a maximum of 15 epochs. Early-stopping was applied to approximate an optimal training duration. Furthermore, if the learning reaches a plateau, the learning rate is reduced by a factor of 0.2. The batch size is set to 128 and the initial learning rate to $2.5e - 5$.

5.3 Pretrained Word Embedding: the fastText Vectors

The second variant of text representation includes a pre-trained English fastText embedding[1]. The pre-trained embedding is loaded and for every token a word vector is returned (cf. figure 5.3). For out-of-vocabulary (OOV) words, an approximated vector is given back based on the character n-grams of the OOV word (cf. section 2.5.4). This approach unleashes the full potential of fastText and so plays out the advantages it has against GloVe or word2vec, which are not able to deal well with OOV words. A further benefit of this approach: there is no word limit necessary since the classification can be done within

[1] `https://fasttext.cc/docs/en/crawl-vectors.html` [15.12.2021 18:40]

a reasonable timeframe. However, loading word vectors in this way increases the memory size considerably. A Python generator approach is recommended here. The vectors are 300-dimensional and are the outcome of training the fastText algorithm on Wikipedia dumps where additional CommonCrawl[2] data was added to increase the homogeneity of the training data [114]. An additional embedding layer is not needed. To load the fastText embedding and query the word vectors, the fastText Python library was used[3]. In the following, to this model is now referred to as *fastText model or LSTM-F*.

5.3.1 Training

The fastText classification model does not have an embedding layer. The inputs of the generated word vector sequences are directly fed into an LSTM network followed by a fully-connected classification layer with softmax activation. Each generated word vector is 300 dimensional. However, the fastText model was trained for 30 epochs with activated early-stopping. Also, the same approach of learning rate reduction was applied, like the virgin embedding. The learning rate was $5e-4$ and the batch size 128. Furthermore, a dropout of 0.25 was used.

5.4 Self-trained multilingual Twitter fastText Embedding

As mentioned in section 1, cities are multilingual urban structures. For this reason, the text classification task is extended from a monolingual to a multilingual text corpus. However, representing a multilingual text sequence with word vectors, e.g., obtained by a single fastText embedding, would return a very limited word vector sequence. For example, if a Spanish tweet should be converted with an English word embedding, the result would be very poor. Spanish tokens might be sporadically included in an English embedding; however, a meaningful vector-represented sequence cannot deviate. The tokens are simply not included. Even though fastText provides embeddings in 157 languages (cf. section 2.5.4 and footnote 1 on p. 58), the usage would be cumbersome. For every individual language, the embedding must be exchanged for an optimal feature yield. As an additional challenge, an individual preprocessing and tokenization must be executed for every language present in the corpus (cf. section 2.4.3). Therefore, a special multilingual training dataset to train the fastText embedding with unseen data was created (cf. section 4.6).

5.4.1 Training

The multilingual fastText CBOW and skip-gram embedding training settings are adopted from the fastText papers [109, 114] and standard settings. The embedding was trained using fastText v.0.9.2 for five epochs starting with a learning rate of 0.5. The n-gram size was set between 3 and 6. [109] reported this as a good choice across languages. The

[2] https://commoncrawl.org/ [15.12.2021 18:50]
[3] https://fasttext.cc/docs/en/python-module.html [15.12.2021 18:54]

		mLSTM-F CBOW		mLSTM-F skip-gram	
dataset	λ (m)	OA	κ	OA	κ
full	∞ (∞)	0.57	**0.228**	0.57	0.217
100	30 (54)	0.57	0.284	0.57	**0.293**
1-99	19 (25)	0.56	0.288	0.56	**0.295**
2-98	15 (20)	0.56	0.287	0.56	**0.294**
5-95	10 (13)	0.55	0.287	0.55	**0.294**
10-90	7 (8)	0.54	0.284	**0.55**	**0.293**
15-85	5 (6)	0.54	0.283	0.54	**0.292**
20-80	4 (5)	0.53	0.281	**0.54**	**0.291**
25-75	3 (4)	0.53	0.279	0.53	**0.284**

Table 5.1: CBOW vs. skip-gram building function classification results. The maximum number of tweets per building, i.e. the λ value, is noted in column λ (m), m stands for multilingual and is only valid for models with the prefix *m*.

output dimensionality was set to 300. The embedding yielding the highest scores in text classification is reported in the main results section 5.6.

For text classification, the same LSTM configuration and hyperparameters are used as the previous experiments are used. The model was trained for a maximum of 30 epochs with applied early-stopping. Also, the learning rate is reduced by a factor 0.2 if the learning reaches a plateau. Like above, the learning rate was set to $5e - 4$ and with a batch size of 128. A dropout of 0.25 was applied to prevent overfitting.

5.4.2 CBOW vs. skip-gram: preliminary results

As pointed out above two fastText embedding have been trained. In this paragraph it is analyzed which of the both models delivers the best building function classification results. Table 5.1 gives an overview of the classification results. Please note that all results refer to multilingual datasets. The classification was performed with an LSTM network. The training settings have been noted above. The LSTM trained with CBOW word vectors is refered to as *CBOW model* and the skip-gram trained LSTM is called *skip-gram model* in this section.

The CBOW accuracy ranges from 0.53 to 0.57, and the Kappa scores from 0.228 to 0.288. The skip-gram models also show accuracies from 0.53 to 0.57. In contrast to the CBOW model, the skip-gram models yield higher Kappa scores. This outcome could be pointing to a slightly better multilingual text representation. The skip-gram models can deal better with the irregular spelling of words which creates a high amount of irregular words and the short text sequences in general. Here, the differences between CBOW and skip-gram models might play a role (cf. section 2.5.4). In a blog post, the word2vec authors pointed out that skip-gram might show better results for infrequent words[4].

The classification results show almost equal results. However, the skip-gram models have a slight lead regarding the Kappa scores. It seems that LSTMs trained with multilingual skip-gram word vectors achieve slightly higher results. For this reason, the skip-gram model is used to compare the results against the other models. In the following, the winning skip-gram model is named as *selt-trained multilingual fastText model, multilingual fastText model, or mLSTM-F.*

5.5 Contextualized Sentence Embeddings obtained from (multilingual) BERT

The fifth option to generate the features for the text classification is using contextualized sentence embeddings yielded by a monolingual or multilingual BERT model (cf. figure 5.4). For the contextualized sentence embedding, the `CLS` token is used which is placed as the first token into every sequence during the special BERT preprocessing (cf. section 2.5.4). After BERT, a linear classification layer is following.

The classification with (multilingual) DistilBERT is implemented using the *huggingface* `transformers` Python library[5]. The classification architecture kept standard as described in huggingface documentation[6]. In the following, to this models is now referred to as *BERT* or *mBERT* for the multilingual variant.

5.5.1 Mono- and multilingual BERT models

For the mono- and multilingual BERT models, Adam was used as an optimizer, and the models have been trained for 3 epochs. The learning rate was set to $5e-6$. The small learning rate and the small number of training epochs were set because preliminary studies revealed quick overfitting to the Twitter data when using larger learning rates or more training epochs. All other parameters remained standard according to the huggingface transformer library.

5.6 Results

In this section, the results of the building function text classification is shown and discussed. A complete overview over the text classification results can be found in table 5.2 for overall scores, table 5.3 for class-wise peformance, figure 5.6, and figure 5.8 for confusion matrices.

Before the results are discussed in detail, it should be noted how the testing was performed. As pointed out in section 4.3.2, after the labeling process, buildings and

[4] `https://code.google.com/archive/p/word2vec/` [20.1.2022 18:38]

[5] `https://huggingface.co/` [1.12.2021, 12:05]

[6] `https://huggingface.co/transformers/model_doc/bert.html#bertforsequenceclassification` [1.12.2021, 13:24]

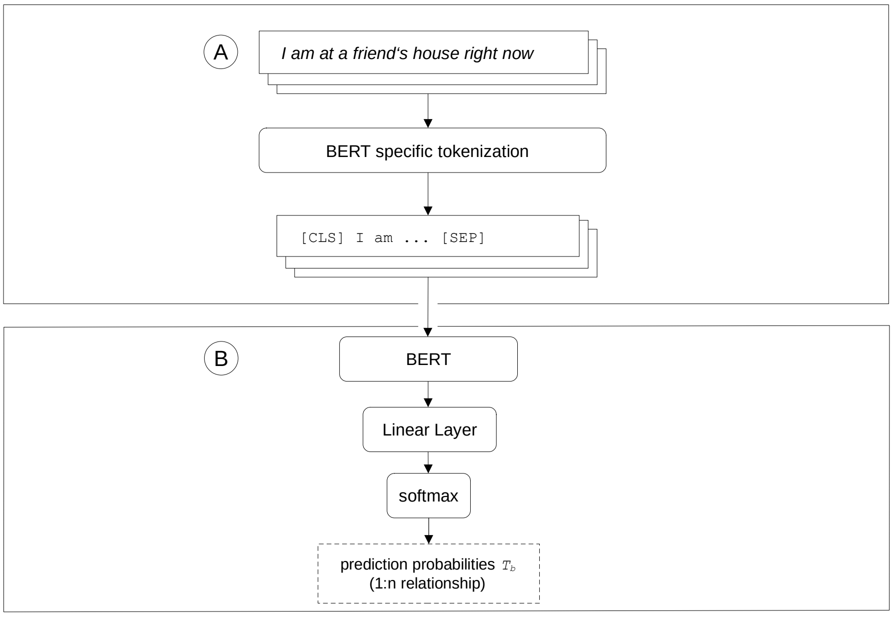

A
I am at a friend's house right now
BERT specific tokenization
[CLS] I am ... [SEP]
B
BERT
Linear Layer
softmax
prediction probabilities T_b
(1:n relationship)

Figure 5.4: Text classification workflow of the mono- and multilingual BERT models. *A* shows the special text preprocessing required for BERT. *B* depicts the actual text classification part.

corresponding tweets are in a $1:n$ relationship. Therefore, the testing was also performed tweet-wise. In section 6, however, the testing process will be at a building level.

5.6.1 Baseline

Table 5.2 shows the overall accuracy and the Kappa score. The overall accuracy achieved by the Naïve Bayes baseline never drops below 0.51. Interestingly, the overall accuracy of the model trained with the largest dataset (no tweet-house-limitation by a λ value) shows the highest accuracy of 0.59. On the other hand, the Kapp score of 0.128 is the third-lowest which points to overfitting. The highest Kappa score shows the model trained on the second-largest dataset of 0.273 and a 0.56 accuracy. It is also clear that the classification performance decreases when the dataset size also decreases. However, the performance drop of the datasets with a more restrictive λ value is small. For example from 100 to $2-98$ accuracy decreased by 0.02 and Kappa by 0.01. This moderate performance loss could signal that many tweets per building are not necessarily leading to significantly better performance.

		Monolingual								Multilingual			
		TFIDF-NB		LSTM-V		LSTM-F		BERT		mLSTM-F		mBERT	
dataset	λ (m)	OA	κ	OA	κ	OA	κ	OA	κ	OA	κ	OA	κ
full	∞ (∞)	0.59	0.128	0.52	0.073	0.53	0.146	0.59	0.248	0.57	0.217	0.63	0.336
100	30 (54)	0.56	0.273	0.57	0.293	0.58	0.305	0.59	0.332	0.57	0.293	0.57	0.305
1-99	19 (25)	0.55	0.267	0.56	0.292	0.57	0.303	0.58	0.332	0.56	0.295	0.57	0.307
2-98	15 (20)	0.54	0.263	0.56	0.284	0.57	0.301	0.58	0.329	0.56	0.294	0.56	0.305
5-95	10 (13)	0.54	0.259	0.55	0.284	0.55	0.297	0.57	0.331	0.55	0.294	0.56	0.308
10-90	7 (8)	0.53	0.256	0.54	0.274	0.55	0.295	0.57	0.327	0.55	0.293	0.55	0.302
15-85	5 (6)	0.52	0.252	0.53	0.270	0.54	0.288	0.56	0.321	0.54	0.292	0.55	0.303
20-80	4 (5)	0.52	0.251	0.51	0.239	0.54	0.291	0.56	0.321	0.54	0.291	0.55	0.301
25-75	3 (4)	0.51	0.248	0.50	0.236	0.53	0.289	0.55	0.320	0.54	0.284	0.54	0.298

Table 5.2: Text classification results. The scores presented in this table are overall accuracy (OA), and Cohen's Kappa (κ). The maximum number of tweets per building, i.e. the λ value, is noted in column λ (m), m stands for multilingual and is only valid for models with the prefix *m*.

For the class-wise performance, it can be noted, that the *commercial* and *other* class showing the best results (cf. table 5.3, column *TFIDF-MNB*). Both classes have the highest amount of individual tweets (c.f. table 4.5). It is natural the classifier adapts more to the dominant classes. The class-wise performance reflects this circumstance. The *residential* class has not less buildings in the dataset (cf. table 4.3). However, the total amount of tweets is lower (cf. table 4.5). Therefore, the performance is weaker as observed at *commercial* and *other.* On the other hand, the smaller the dataset gets, the performance of the *residential* class rises in precision, recall, and F1. The increase is owed to the fact of more balanced tweet amounts amongst the classes, which leads to a more "balanced" classification result.

The full dataset without tweet-house limitation produces a poorer prediction quality for the *commercial* and *residential* classes. As the overall results in table 5.2 indicated. For example, the datasets with λ values from 100 to $5 - 95$ produce almost identical results. This finding supports the claim that more data does not lead in all cases to better predictions.

5.6.2 LSTM and Virgin Embedding

The self trained virgin embedding shows overall accuracies from 0.50 to 0.57 and Kappa scores from 0.073 to 0.293 (cf. table 5.2, column *LSTM-V*). The strongest result is achieved with the model trained on the 100 dataset with an accuracy of 0.57 and a Kappa of 0.293. Comparing the overall accuracies to the Naïve Bayes accuracies, partially better results are detectable. Except for the dataset with no tweet-house limitation, the two smallest datasets showing lower accuracy and Kappa values as the Naïve Bayes result. The same pattern of slight to no decrease in performance can be seen in the results produced by the datasets 100 to $5 - 95$.

For the *commercial* class precision scores, the virgin embedding shows for all datasets almost the identical results as the Naïve Bayers models (cf. table 5.3, column *LSTM-V*). They are slightly lower. The same can be observed in the *residential* class results. Only the values for others are higher as the Naïve Bayes' numbers. If studying the recall values, a different picture can be drawn. For *commercial* and *residential*, the virgin embedding can outperform the baseline. Additionally, the F1 score is higher. Only for the *other* class is the baseline better.

Also, the trend of "less data, fair performance" mentioned above is visible in the virgin embedding results as well. Except for the *other* class, performance is not dropping when the amount of data is reduced. For the *residential* class, the precision, recall, and F1 score are climbing. The more balanced tweet distribution amongst classes might be the case for this behavior.

However, even though the virgin embedding can reach slightly higher recall and F1 scores, the gain of performance using deep learning methods with self-trained word representations, i.e., word vectors, is moderate but observable (F1 score). The training of a word embedding from scratch is a challenging task. The limited amount of words

			TFIDF-MNB			LSTM-V			LSTM-F		
	dataset	λ (m)	P	R	F1	P	R	F1	P	R	F1
commercial	*full*	∞ (∞)	0.42	0.31	0.36	0.36	0.42	0.39	0.46	0.43	0.45
	100	30 (54)	0.56	0.58	0.57	0.55	0.66	0.60	0.56	0.65	0.60
	1-99	19 (25)	0.55	0.57	0.56	0.54	0.65	0.59	0.56	0.62	0.59
	2-98	15 (20)	0.55	0.56	0.55	0.54	0.62	0.58	0.55	0.65	0.60
	5-95	10 (13)	0.54	0.54	0.54	0.53	0.62	0.57	0.54	0.63	0.58
	10-90	7 (8)	0.53	0.52	0.53	0.52	0.57	0.55	0.53	0.62	0.57
	15-85	5 (6)	0.52	0.50	0.51	0.51	0.56	0.54	0.53	0.59	0.55
	20-80	4 (5)	0.52	0.49	0.50	0.46	0.61	0.53	0.53	0.59	0.55
	25-75	3 (4)	0.50	0.48	0.49	0.45	0.62	0.51	0.51	0.57	0.54
other	*full*	∞ (∞)	0.66	0.79	0.72	0.65	0.62	0.63	0.68	0.63	0.65
	100	30 (54)	0.57	0.71	0.63	0.62	0.63	0.62	0.62	0.65	0.63
	1-99	19 (25)	0.56	0.70	0.62	0.62	0.60	0.61	0.60	0.67	0.63
	2-98	15 (20)	0.55	0.70	0.62	0.59	0.65	0.62	0.61	0.63	0.62
	5-95	10 (13)	0.54	0.70	0.61	0.60	0.60	0.60	0.60	0.62	0.61
	10-90	7 (8)	0.53	0.70	0.60	0.56	0.64	0.60	0.58	0.62	0.60
	15-85	5 (6)	0.52	0.70	0.59	0.56	0.61	0.58	0.57	0.61	0.59
	20-80	4 (5)	0.51	0.70	0.59	0.56	0.60	0.58	0.57	0.61	0.59
	25-75	3 (4)	0.50	0.69	0.58	0.56	0.56	0.56	0.56	0.58	0.57
residential	*full*	∞ (∞)	0.24	0.09	0.13	0.16	0.11	0.13	0.11	0.19	0.14
	100	30 (54)	0.47	0.18	0.26	0.45	0.23	0.30	0.45	0.23	0.30
	1-99	19 (25)	0.48	0.20	0.28	0.44	0.28	0.34	0.47	0.25	0.33
	2-98	15 (20)	0.49	0.21	0.29	0.47	0.24	0.32	0.49	0.25	0.33
	5-95	10 (13)	0.51	0.23	0.32	0.47	0.32	0.38	0.48	0.31	0.38
	10-90	7 (8)	0.53	0.26	0.34	0.50	0.31	0.38	0.51	0.32	0.39
	15-85	5 (6)	0.54	0.28	0.37	0.51	0.35	0.42	0.52	0.36	0.42
	20-80	4 (5)	0.56	0.30	0.39	0.57	0.24	0.33	0.53	0.37	0.44
	25-75	3 (4)	0.57	0.32	0.41	0.58	0.27	0.37	0.53	0.41	0.47
			BERT			mLSTM-F			mBERT		
commercial	*full*	∞ (∞)	0.50	0.55	0.53	0.55	0.51	0.53	0.61	0.67	0.64
	100	30 (54)	0.58	0.65	0.61	0.55	0.71	0.62	0.57	0.67	0.61
	1-99	19 (25)	0.57	0.67	0.61	0.56	0.65	0.60	0.56	0.66	0.61
	2-98	15 (20)	0.58	0.62	0.60	0.55	0.66	0.60	0.56	0.65	0.60
	5-95	10 (13)	0.56	0.64	0.60	0.55	0.64	0.59	0.56	0.64	0.59
	10-90	7 (8)	0.55	0.63	0.59	0.54	0.62	0.58	0.54	0.66	0.59
	15-85	5 (6)	0.53	0.65	0.58	0.54	0.61	0.57	0.53	0.64	0.58
	20-80	4 (5)	0.55	0.57	0.56	0.54	0.60	0.57	0.53	0.63	0.58
	25-75	3 (4)	0.54	0.54	0.54	0.54	0.54	0.54	0.54	0.58	0.56
other	*full*	∞ (∞)	0.74	0.66	0.70	0.60	0.67	0.63	0.69	0.64	0.66
	100	30 (54)	0.63	0.66	0.65	0.61	0.61	0.61	0.62	0.63	0.62
	1-99	19 (25)	0.64	0.62	0.63	0.59	0.63	0.61	0.60	0.63	0.62
	2-98	15 (20)	0.61	0.67	0.64	0.58	0.64	0.61	0.61	0.59	0.60
	5-95	10 (13)	0.64	0.60	0.62	0.58	0.63	0.60	0.60	0.62	0.61
	10-90	7 (8)	0.62	0.37	0.43	0.57	0.62	0.60	0.60	0.59	0.59
	15-85	5 (6)	0.62	0.59	0.60	0.57	0.62	0.59	0.61	0.56	0.58
	20-80	4 (5)	0.58	0.65	0.61	0.56	0.62	0.59	0.58	0.60	0.59
	25-75	3 (4)	0.58	0.62	0.60	0.53	0.68	0.60	0.57	0.61	0.59
residential	*full*	∞ (∞)	0.12	0.18	0.15	0.39	0.24	0.29	0.31	0.27	0.29
	100	30 (54)	0.45	0.28	0.35	0.48	0.18	0.26	0.42	0.24	0.31
	1-99	19 (25)	0.46	0.32	0.38	0.47	0.25	0.33	0.46	0.26	0.34
	2-98	15 (20)	0.48	0.31	0.38	0.49	0.23	0.32	0.43	0.32	0.37
	5-95	10 (13)	0.47	0.40	0.43	0.49	0.27	0.35	0.47	0.32	0.38
	10-90	7 (8)	0.50	0.37	0.43	0.50	0.31	0.38	0.49	0.33	0.39
	15-85	5 (6)	0.52	0.39	0.45	0.50	0.33	0.40	0.47	0.38	0.42
	20-80	4 (5)	0.53	0.41	0.46	0.51	0.33	0.40	0.50	0.35	0.42
	25-75	3 (4)	0.53	0.48	0.50	0.53	0.33	0.41	0.49	0.38	0.43

Table 5.3: Monolingual classification results. The scores presented in this table are precision (P), recall (R), and F1 (F1). The maximum number of tweets per building, i.e. the λ value, is noted in column λ (m), m stands for multilingual and is only valid for models with the prefix *m*.

$(35,000)^7$, the short sequences and the informal language are possibly the primary performance factors. For this reason, a pre-trained word embedding, such as fastText (cf. 2.5.4) could provide more sophisticated feature vectors and approximations of out-of-vocabulary words, which is expected due to informally written tweets, leading possibly to higher classification scores.

5.6.3 LSTM and fastText Embedding

The LSTM models training with fastText word vector sequences show overall accuracies from 0.53 to 0.58 and Kappa scores from 0.146 to 0.305 (cf. table 5.2, column *LSTM-F*). Compared to Naïve Bayes and the virgin embedding, the fastText models outperform both of them (moderately). The model trained on the 100 dataset shows the highest kappa score. The results of the fastText embedding models are stable throughout the nine datasets—the largest dataset achieves the highest accuracy but the lowest Kappa score.

The class-wise performance is close to the virgin embedding. Slightly higher F1 scores compared to the Naïve Bayes and the virgin embedding are observable. Also, recall in some cases is also higher. By differentiating the F1 scores, the model best predicts the *other* class followed by the *commercial* class. "Residential" demonstrate the poorest performance. However, the fastText models reach slightly higher precision values compared to the virgin embedding. For *residential*, higher recall values than the baseline can be achieved. In comparison to the virgin embedding, elevated recall scores for the smaller datasets are also observable.

The fastText models also display the same effect as the latter models. A lower number of tweets per building does not lead to a large performance drop. The dataset without a tweet-house limit constantly shows weaker results (except for the *other* class). As stated in the paragraphs above, the *residential* benefits the most from a stricter tweet-house limit, i.e., a lower λ value. The stricter this value is set, the more balanced the number of tweets per class, leading to increased performance of the *residential* class.

In more than 50% of the cases, the fastText model shows higher recall values and higher F1 scores compared to the virgin embedding model. In 19 of 27 cases, the fastText embedding models achieve higher F1 scores than the baseline models. This finding indicates that a pre-trained English word embedding is able to deliver better features of the informal Twitter language and therefore leads to increased classification results for building function classification. However, this poses the question of whether the classification can be boosted by using more contextualized embeddings from BERT's state-of-the-art neural language model.

5.6.4 BERT

The BERT models achieve the highest accuracy values and Kappa ratings throughout the different datasets (cf. table 5.2). Except for the largest dataset *full*, all models

[7] Subsequent experiments with a much larger vocabulary, however, revealed no performance boosts.

trained are reaching almost the same Kappa scores ranging from 0.320 to 0.332, which outperform all other models and classification techniques.

The precision values reached by the BERT models are higher for all classes in comparison to every other model–with the exception that the Naïve Bayes outperforms BERT's *residential* precision values. Although, BERT shows higher recall values at the *residential* class, which indicates a lower miss-classification rate. The recall values of the BERT models are also higher almost everywhere. In 19 of 27 cases BERT's recall values are higher than the fastText models. However, it seems the baseline's recall values of the *other* class are unbeatable. Finally, the BERT models can achieve higher F1 scores than all other models in 22 cases. For the weak *residential* class, BERT reaches the F1 high score for all datasets. In 8 of 9 cases, they reach the best recall score for the *residential* class. Even though the amount of data is lower than *commercial* and *other*, BERT can produce better results.

The results of BERT demonstrate that contextualized linguistic features can generate better performance for building function classification. In particular for the underperforming *residential* class where BERT can achieve higher recall and F1 scores. This indicates that context also matters for building function classification with text. The following section scrutinizes the classification results in more detail.

5.6.5 LSTM and multilingual Twitter fastText Embedding

To investigate the language capabilities of the self-trained embedding, some examples are given by executing neares neighbor queries using the fastText tool. The cosine similiarity scores of the query words can be found in the boxes in table 5.4. For in the intra-language-space, fair nearest neighbors can be achieved. The results of the nearest neighbor queries are not as perfect as for a monolingual embedding trained on very large Wikipedia corpora with structered text. However, what can be shown is that for example spelling errors can be covered to certain degree to yield vectors which are still in the neighorhood of the misspelled word (cf. boxes 5.4b and 5.4c). Addiontally, plural forms of words can also be found in each others neighborhoods (cf. boxes 5.4a, 5.4b, 5.4c, or 5.4d). The trained embedding is also capable to some degree to grasp concepts. Box 5.4d shows the nearest neighbors of the word *car*. Synonyms, plural, and car-lelated words like *garage* appear. This can also be shown for the word *flat* in box 5.4e. When querying the German political party *spd*, related words like *cdu* (another party), or *scholz* who is a member of the party and new German chancellor. This particular result is owed to the fact that during the time period the Twitter data was received, the *bundestagwahl* took place.

It can be seen that a real semantic transfer between languages is not present but was not intended from the onset (cf. section 4.6). Still, some word representations are in the neighboorhood of a different language. For example, box 5.4a in table 5.4 the German word *Universität* and the Dutch translation *Universiteit* or Spanish *universidad* and Portuguese (Brasilian) *universitária* (although admittedly pretty easy). Dispite the language transfer being rather modest, this illustrates the nearest neighbor examples showing the potential of self-trained embeddings for a specific down-stream task like building function classification.

universidade	0.861558
universidades	0.847378
diversidad	0.795884
marchadeladiversidad	0.720463
universitario	0.715051
universitárias	0.712705
universitária	0.704549
universitaria	0.704191
universitário	0.693096
universitários	0.689477

(a) universidad

universität	0.924975
diversität	0.8945
universitäten	0.887677
tät	0.788678
arbeitskollege	0.773706
parität	0.771903
nervosität	0.762107
kapazität	0.756521
naivität	0.756388
flächendeckend	0.752941

(b) Universität

universität	0.852836
tät	0.835155
universitäten	0.823349
naivität	0.811663
nervosität	0.810769
loyalität	0.804597
luftqualität	0.80388
parität	0.801605
autorität	0.793169
kapazität	0.790221

(c) univesität (sic!)

cars	0.759282
car	0.662185
tires	0.643771
dealership	0.639101
vehicles	0.621912
garage	0.613661
motorcars	0.611477
escooter	0.605716
chev	0.60509
parkings	0.604357

(d) car

apartment	0.74131
flats	0.737592
bhk	0.721817
bhkflats	0.70644
forrent	0.68528
apartments	0.652898
flat"	0.651492
apartmen	0.642933
furnishing	0.638957
forsale	0.637791

(e) flat

cdu	0.793992
csu	0.718582
jamaikakoalition	0.718441
scholz	0.714119
bundestag	0.711409
koalition	0.69919
ampelkoalition	0.689255
bundestagwahl	0.688271
bundestags	0.682935
bundestagswahlkampf	0.680833

(f) spd

Table 5.4: Cosine similarity scores of the skipgram model when performing a nearest neighbor query.

The results of the text classification of the LSTM using word vectors received from the self-trained fastText skip-gram (cf. CBOW results in table 5.1) show stable performance throughout the datasets. The overall accuracy ranges from 0.54 to 0.57 and Kappa from 0.217 to 0.295.

The models can achieve similar precision scores throughout the different datasets with minimal distances (0.54 to 0.56). In contrast, the recall values dropping from the datasets 1-99 to 25-75 are quite clear. The impact on the F1 scores is modest but visible.

Similar performance patterns can be found by looking at the *other* class. While the "mid-range" datasets achieved relatively similar results, the minor datasets exhibited a noticeable drop in precision. On the other hand, the recall of 0.68 is a indicator of the 25-75 dataset and seems to be an outlier. The recall values of the other datasets are smaller, and the F1 scores are almost identical.

The precision scores of the *residential* class rise the smaller the dataset gets. This trend was also observable by checking the numbers of the other models. The underrepresented *residential* class benefits from a more balanced dataset. Same for recall: the values increase the smaller the dataset gets.

The self-trained multilingual fastText embedding achieves balanced and reasonable results given the naïve training approach and the relatively small dataset. It yields similar patterns as other models—for example, the increasing scores for the *residential* when the dataset size decreases.

However, what if the features are generated with a multilingual variant of DistilBERT, trained on vast text corpora? Can such features further increase the classification results? The following section discusses the performance of the text classification using BERT.

5.6.6 Multilingual BERT embedding results

The results of the multilingual text classification with the multilingual variant of DistilBERT showed slightly higher or equal results. The overall accuracies range from 0.54 to 0.63. By comparing the Kappa results with the other models, it can be seen that the values are moderately higher. They range from 0.298 to 0.336. Multilingual DistilBERT trained with the largest dataset achieves the highest overall scores. This circumstance is different from the other results where the models with the most tweets always performed worst.

The precision scores of the *commercial* class decrease marginally, notable for the slimmer the dataset gets. Also, the precision scores are almost equal compared to the model using the self-trained multilingual fastText vectors. The recall values are slightly higher than the values achieved by the multilingual fastText model. However, the multilingual BERT model can outperform the multilingual fastText model in 6 of 9 datasets at F1 scores. In three cases, the multilingual fastText model achieves the same or better results.

By looking at the results of the *other* class, it can be seen that the smaller the dataset gets, the lower the precision. The precision drops slightly but is still higher compared to the multilingual fastText model. However, the recall scores yielded by the multilingual fastText model were better in 7 of 9 cases and for the F1 score, the multilingual fastText embedding achieves higher scores in 4 of 9 cases. The multilingual BERT models have only in 4 cases (slightly) higher F1 scores compared to the fastText model.

For the *residential* class, the findings are similar. In 9 of 9 precision scores, the multilingual BERT model is outperformed by the multilingual fastText embedding model. The dataset 2-98 or 5-95 for example, are partially and relatively transparent. It is interesting that the recall values in both models are climbing the smaller the datasets get. Here, the BERT model reach higher scores. This finding stressed the 8 of 9 times higher F1 scores in comparison to the multilingual fastText embedding.

It can be stated, that the massive multilingual BERT model can *partially* outperform the model trained with self-trained multilingual Twitter word vectors in the building function classification. However, the multilingual BERT model can outperform the fastText model, it can be seen that the differences between the models are moderate. The outcome is evidence that particular tasks like the building function classification task seem to require more specific linguistic features in a multilinguistic setting. Even though BERT is slightly better, almost the same result can be achieved with smaller datasets and "lighter" model architectures.

5.7 Analysis of the Text Classification Results

The last section showed the classification results yielded by the different classifiers. A significant finding is definitely the increased performance reached by models exploiting more sophisticated word representations. The Kappa and F1 scores could be increased with (multilingual) fastText and BERT models. However, 0.63 seems to be a ceiling for accuracy for pure text classification. The weakest class here is *residential*, with F1 scores from 0.13 (baseline) to 0.50 (BERT). One reason is most likely the lower number of tweets per building, but could it also be the text responsible for the underperformance? Why have the classifiers had difficulty distinguishing the tweets, especially for the *residential* class? This section studies the results and gives possible answers to these questions.

The examined results are based on the results achieved with the dataset encoded $5-95$ because it leads to a fair classification result (concerning the number of tweets) amongst classes.

5.7.1 Overview

The collected tweets can exhibit variable distances to the assigned building. Figure 5.5 shows the log-scaled distribution of tweets between the distances of 0 to 50m to the assigned building. Most of the tweets are within a distance of $0 - 10$m to their assigned buildings—the distribution of the *true* predictions showing a slight decrease for more distant tweets. The *residential* class shows here a noticeable trend. The *false* predictions increase the further tweets are away from the buildings. This tendency is visible throughout the classes. The finding suggests that the 50m parameter chosen in section 4.3.2 seems to be reasonable. Further research could utilize this finding and filter even stricter tweets via distance to increase classification performance. Figure 5.6 shows the overall classification results of the individual cities. It can be seen that cities like New York, Los Angeles, and Washington show the best results. It can be noted that no city exhibits strong performance variances. However, it can also be seen that some cities have lower Kappa scores. A possible explanation for the poor results of the models trained on the datasets without tweet-house-limitations is overfitting. By looking at the subfigures in figure 5.7 it can be seen that is fact the case. The models trained on the large datasets are overfitting very fast. By considering the loss plots belonging to models of the smaller datasets, it can be seen that the overfitting is either not present (cf. subfigure 5.7b) or is not onsetting immediately (cf. subfigure 5.7d)

Continuing on, the model predictions are analyzed in more detail. For this analysis, the models with the most balanced results are chosen. Therefore, the models with the dataset encoding 5-95 are used. The results with these datasets achieved slightly lower accuracy. However, the Kappa scores are almost on par, and the *residential* class shows better results. The confusion matrices of the individual models showing similar classification distributions (cf. figure 5.8). Figure 5.8a depicts the classification distribution of the best performing Naïve Bayes model. It can be seen that *commercial* and *other* are classified relatively well. The *residential* class, on the other hand, is classified poorly. "Residential" tweets are mistaken with *other* and *commercial* tweets. However, the bias towards the *other* class is slightly higher. Almost the same trend can be observed when looking at the confusion matrix in figure 5.8b of the virgin embedding. The dominant classes, *commercial* and *other*, are also inferred satisfyingly. Unfortunately, the *residential* class is classified poorly as well. The *commercial* class is here also the most frequent

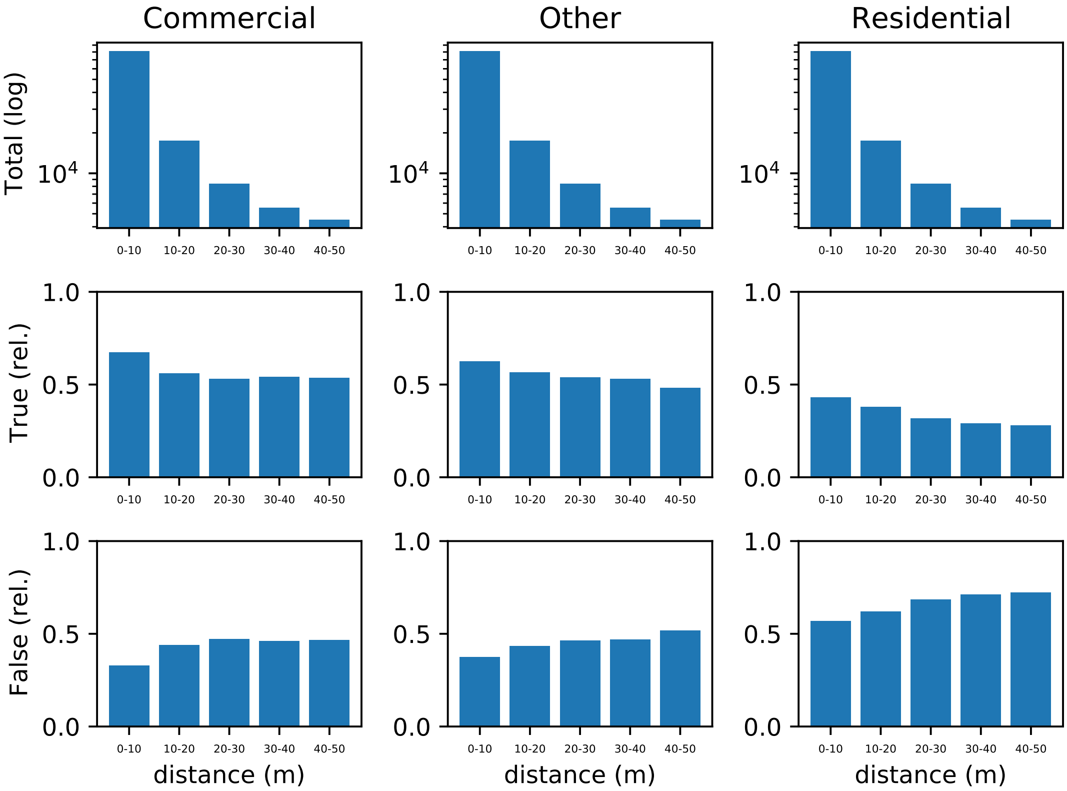


Figure 5.5: Distance plot of the $5-95$ BERT model. The upper row shows the tweet-distance distriburion within 50m. The rows below display the *true* and *false* distribution of tweets according to the distance. The distances are normalized.

miss-classification. The third model, i.e., the fastText model drawing the same picture (cf. figure 5.8c). Commercial and *other* are categorized fine, while the *residential* class underperforms visibly. According to the findings above, *residential* tweets are mainly confused with the *commercial* class. BERT also shows the same classification pattern as the three models above. The *commercial* and *other* class are categorized well. The BERT results for the *residential* class are slightly better. However, the *residential* class is more confused with the *commercial* class (cf. figure 5.8d). By examining the confusion matrices of the multilingual models, the same pattern can be seen (cf. figures 5.8e and 5.8f). Both models have achieved acceptable recall values for *commercial* and *other*, but for *residential*, both models confuse *commercial* and *residential* more than *residential* and *other*.

Therefore, the following paragraph analyzes the text classification in more detail. Some prototype tweets are selected, and their texts scrutinized. The goal is to find a possible explanation for the misclassifications besides class imbalance and if a text pattern can be detected.

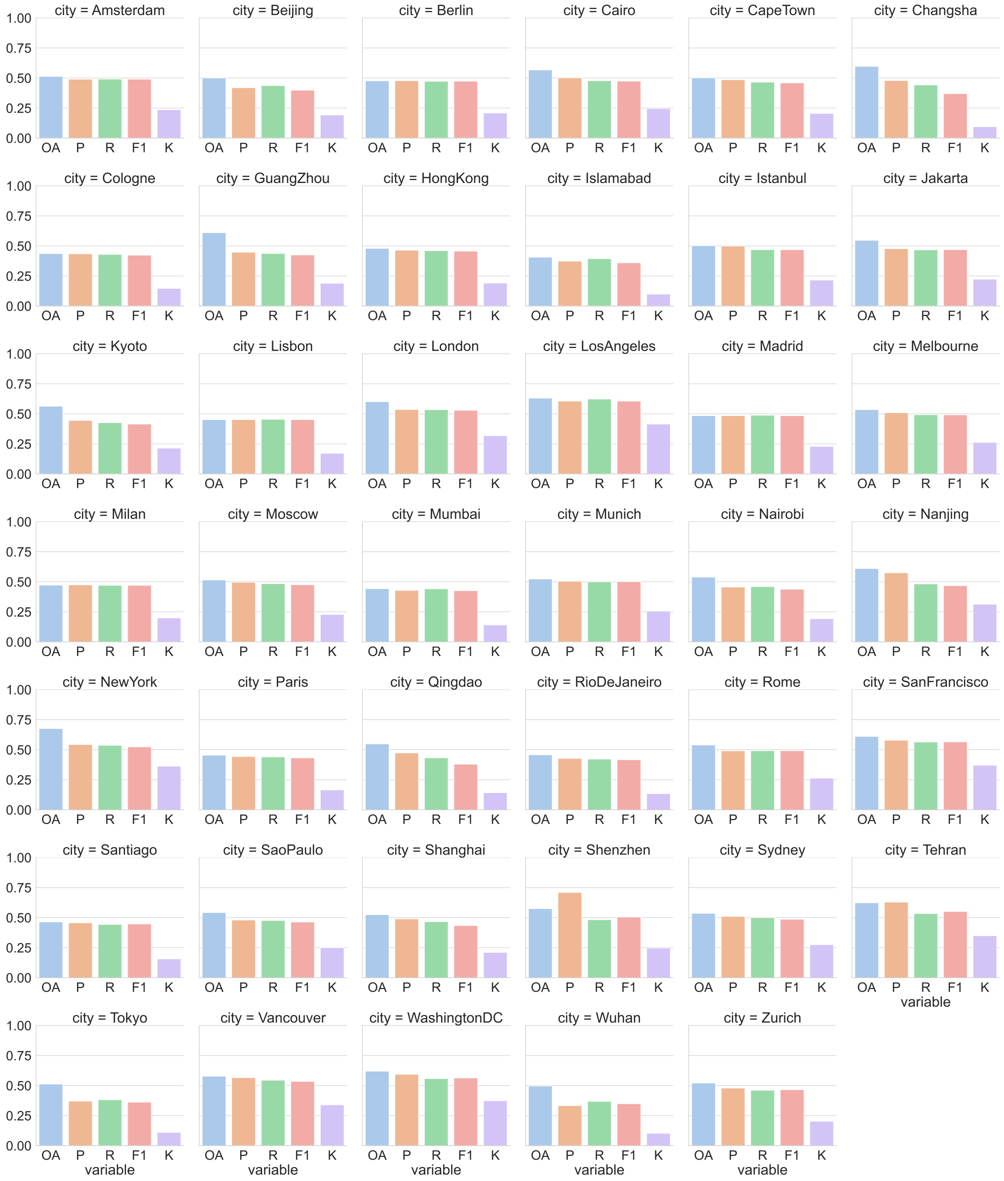
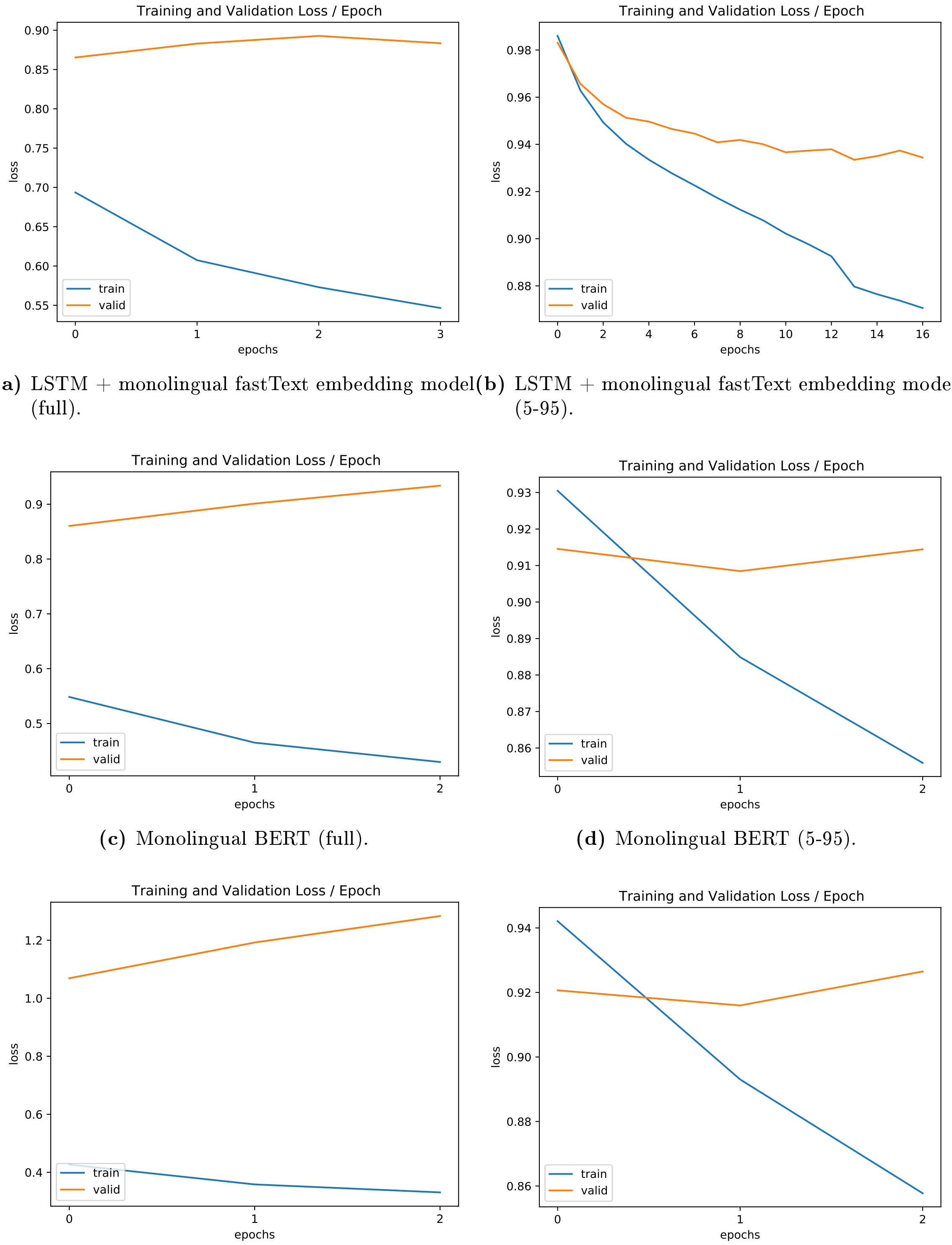


Figure 5.6: City-wise results of the 5 − 95 BERT model. The numbers are showing the macro scores of the individual cities. The scores presented in this figure are overall accuracy (OA), precision (P), recall (R), F1 (F1), and Kappa (K).



(a) LSTM + monolingual fastText embedding model (full).

(b) LSTM + monolingual fastText embedding model (5-95).

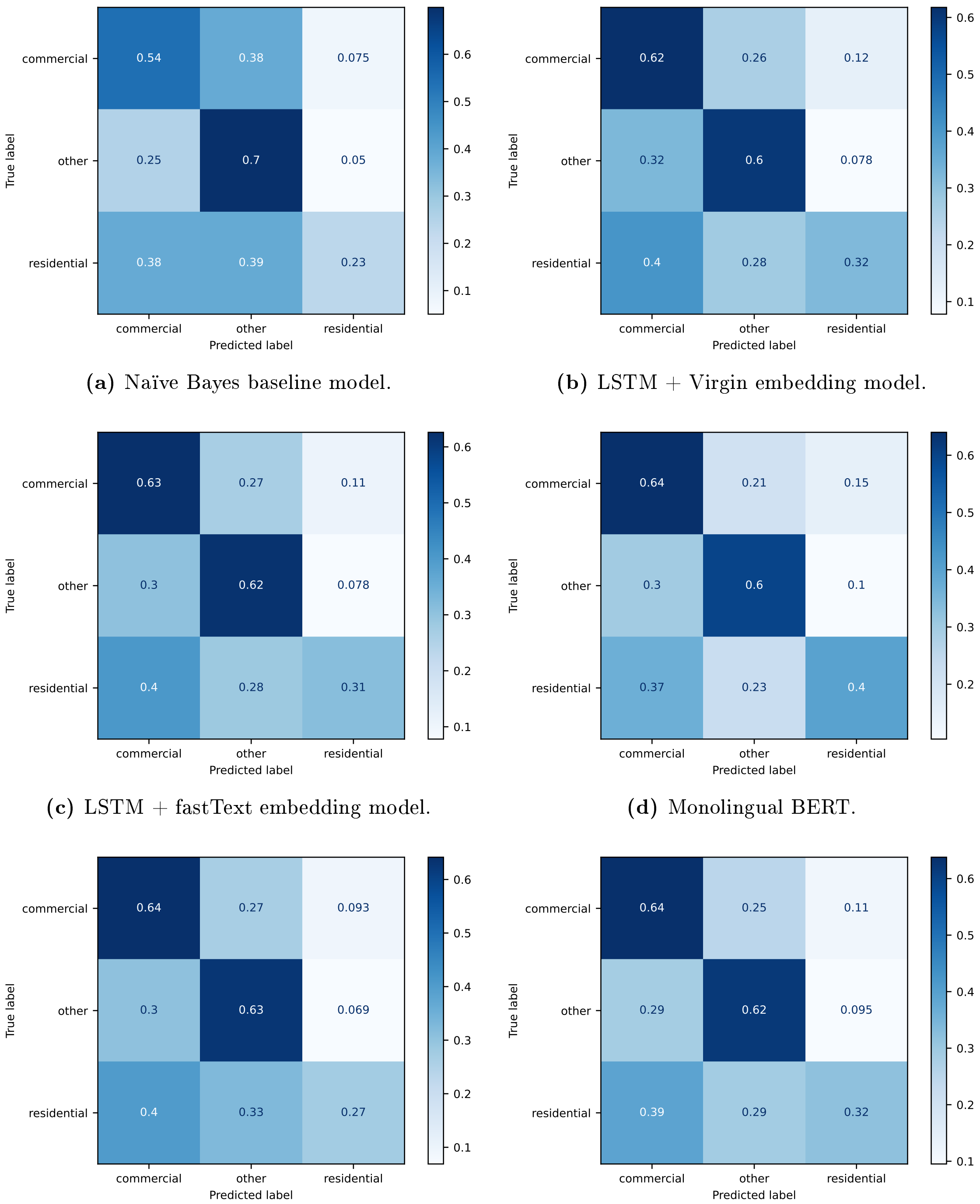
(c) Monolingual BERT (full).

(d) Monolingual BERT (5-95).

(e) Multilingual BERT (full).

(f) Multilingual BERT (5-95).

Figure 5.7: Training and validation loss histories.



(a) Naïve Bayes baseline model.

(b) LSTM + Virgin embedding model.

(c) LSTM + fastText embedding model.

(d) Monolingual BERT.

(e) LSTM + Multilingual Twitter fastText embedding.

(f) Multilingual BERT.

Figure 5.8: Confusion matrices of the 5-95 dataset model results. Values normalized after *true* predictions, i.e., recall (row-wise normalization).

5.7.2 Feature Analysis

Before the misclassification is investigated, a brief overview of correctly classified tweets is given. It can be seen in table 5.5 that the tweets showing a term and topic pattern. Commercial tweets (examples 1–4) are often food-related. The example tweets given here mostly talk about lunch break, having a drink, restaurants, or specific foods. Tweet 4 is an example of code-switching, i.e., the changed language within context. The text of the tweet is mainly English, but the user refers to a sort of beer and a location type spelled out in German. Examples 5–7 are positive examples of the *other* class. The tweets exhibit terms which are related to education like *classroom*, *college*, or religious content. Example 8 refers to getting out at the next stop (most likely public transportation in Paris). The examples of the *residential* class include watching the rain, offering kitchen utensils, a home-coming scenario in Portuguese, and mowing the lawn at home in Italian. Considering the confusion matrices 5.8a, 5.8b, 5.8c, 5.8d again, it can also be seen that misclassifications are happening. In particular, the *residential* class is underperforming. Why can *commercial* and *other* tweets be classified relatively well while *residential* tweets' predictions are mostly poor? One reason is the imbalanced number of tweets per class. However, as [20] point out, challenges should be expected when tweets are posted referring–referring to mix-used buildings. They hypothesized that a main source of error could be the mixed usage of some buildings. For example, an apartment building can also accommodate *commercial* entities like small stores, doctors' offices, a bakery, or a restaurant. If a tweet is tagged with a point of interest of this particular place, it inherits the OSM label of the building. When the building is labeled as *residential*, the tweet is tagged with *residential* regardless if the tweet refers to the small restaurant on the ground floor (cf. figure 5.9). Nevertheless, is the mixed-use building reflected in the tweet text? Multilingual tweets of all classes are selected to scrutinize the content to answer these questions. The goal is to detect text features that might explain the classification pattern discovered above. Table 5.6 gives an overview over selected examples to corroborate the statement from above. Here, tweets have been selected which the BERT model trained with dataset 9-95 predicted wrong. Examples 13–15 in table 5.6 show misclassifications of residential buildings. The content of the selected tweets is about drinking, cafeteria, restaurants, menus, and a club. It is clear that the tweets refer not to a residential home but to gastronomy venues that are tagged as *commercial* (cf. appendix 4.4a). Example 15 mentions a nightclub at a different location in the city.

One explanation might be that a person is tweeting from home about the venue. By looking at examples 16–18, the tweets are falsely categorized as *other* buildings. The text includes words like *Universidad*, medical terms, and *college*. Education and medical facilities can be related to the *other* class (cf. figure 4.4c). Example 16 is also an exemplary case of mixed language on Twitter. The same picture draws examples 19–20. Here, the *other* texts are mistaken as *commercial* tweets. Terms like office, work, or job opportunities might be related to business and *commercial* topics. Examples 20 and 21 are additionally difficult due to a mixed usage of possible topic words. The tweets include both job-related vocabulary like *job* or *hiring*. On the other hand, terms that could be used within a medical context like *nursing* or *care* could confuse the classifier.

#	OSM id	true	predicted	text
1	517469104	commercial	commercial	*tom yum seafood only at thai street grand metropolitan mall lv cobakoba wearekobagroup thaistreet grand metropolitan*
2	1569299520	commercial	commercial	*lmao how do you forget the back door lol lucky went around that side to see it trinity restaurant & bar*
3	763397640	commercial	commercial	*milano agosto pausa pranzo mezzogiornodifuoco milan august summer summertime canicule agosto pranzo lunch stazionecentralemilano break lunchbreak via vittor pisani* (it)
4	433047806	commercial	commercial	*drinking jever pilsener by friesisches brauhaus zu jever cronn dark side office* (mixed)
5	13880363	other	other	*voila the zombies zombie makeup studentvoice undead schooldistricts fxmakeup classroom studentsuccess studentengagement*
6	900261706	other	other	*serving up breakfastsandwiches & lunchsandwiches woodbridge berkeley college middlesex*
7	574421030	other	other	*bom diaa hj domingo dia do senhor fe gratidao amominhaigreja amominhareligiao em paróquia santa rita de cássia mirandópolis* (es)
8	290986426	other	other	*prochain arrêt étréchy* (fr)
9	1582147372	residential	residential	*mumbai rains and the beautiful views one can see mumbairain rainyday*
10	1547616508	residential	residential	*various kitchen stuff in london unitedkingdom zerowaste free*
11	1396624836	residential	residential	*sempre bom estar com meus irmãos em condomínio green village* (pt)
12	1043827438	residential	residential	*primo taglio dell'anno erba taglio giardino garden sole sun home casa pernate casa"* (it)

Table 5.5: Correctly classified examples predicted by DistilBERT and multilingual DistilBERT. To maintain the privacy level as high as possible, names or other markers which could identify a specific user are removed and replaced with [NAME]. Examples 3, 4, 7, 8, 11, and 12 are selected from predictions made by the model trained on the multilingual fastText embedding (9-95 dataset).

#	OSM id	true	predicted	text
13	422941234	residential	commercial	*drinking an original by cafeteria monti photo*
14	1492303944	residential	commercial	*liked it lot thanks morris for taking me there first time see french menu translation in hk and perfect one far from the approximate ones you find in paris chinese restaurants onedimsum*
15	372558734	residential	commercial	*and present playgroundlondonuk at circa theclub with dj [NAME] on the decks and me on the door come and join us doors open at pm [NAME]*
16	403913772	commercial	other	*i'm at universidad mayor in santiago de chile*
17	1302733506	commercial	other	*before and after nomorebraces bracesoff [NAME] orthodontics*
18	1022385648	commercial	other	*halloween at tmg college australia tmg college australia rto id*
19	52377106	other	commercial	*captured from the office nofilter industries sunset travelplaces workplace lovephotography amazingworld nature naturalphotography photography*
20	296540782	other	commercial	*want to work at sunrise senior living we're hiring in vancouver bc click for details nursing*
21	714822352	other	commercial	*see our latest vancouver bc nursing job opportunity and click the link in our bio to apply personal care giver at sunrise senior living*
22	780511446	residential	other	*ao partir pão os nossos olhos se abrem reconhecemos quem tu és sabe enquanto adorávamos com esse louvor eu perguntei deus que seria pra mim partir pão daí ele me* (pt)
23	1222452938	other	commercial	*una pausa para seguir trabajando* (es)
24	648693538	residential	commercial	*con este vídeo de nuestro escaparate más carnavalero os deseamos feliz carnaval en kidspace* (es)
25	796720174	commercial	other	*i'm at librerías cuesta de moyano in madrid comunidad de madrid* (es)

Table 5.6: Misclassification examples. To maintain the privacy level as high as possible, names or other markers which could identify a specific user are removed and replaced with [NAME]. Examples 22–25 are selected from predictions made by the model trained on the multilingual fastText embedding (9-95 dataset).

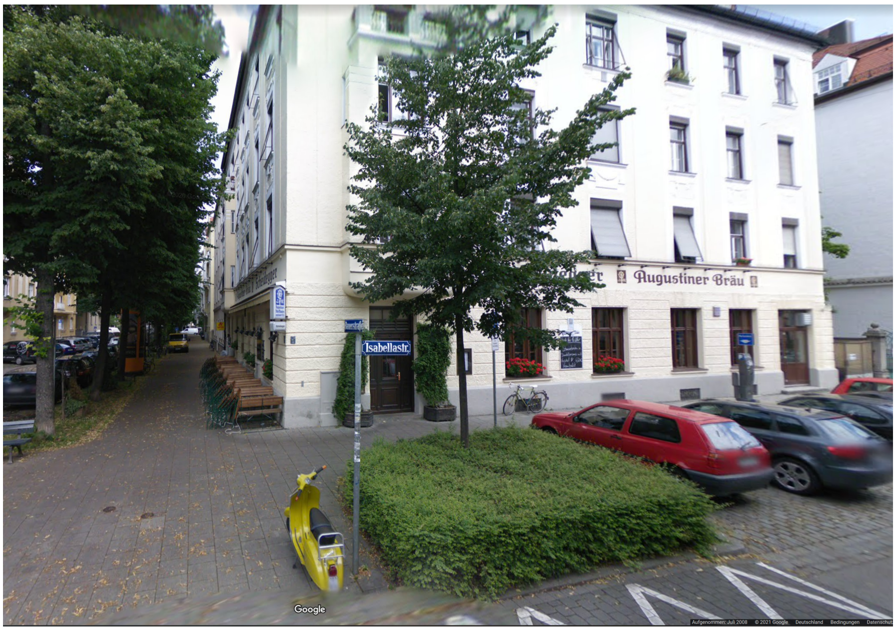


Figure 5.9: Prototypical mixed-used building located in Munich (screenshot taken from Google StreetView). © Google 2021.

The analysis of the tweets indicates two findings. First, by reading the example tweets in table 5.6 the issue of mixed-used buildings is visible. The tweets clearly referring to a *commercial* entities but are labeled as *residential* (cf. figure 5.9). The same scenario is present in all of the documented examples above. The challenge of unclear labels hinders the text classification from reaching higher accuracies [20].

However, the second finding is that the text classification is principally working. Tweets with words that could be assigned to a certain building function, e.g., food-related terms (commercial), are classified "wrong" only because of the problematic labeling situation. In reality, the tweet *is* a tweet that refers to a commercially used entity in urban space, but the label is *other*. The text in example 16 indicates that the user is at a university. The classifier recognizes a possible class-related term and therefore classifies the tweets as *other*. The true label, however, is *commercial*. Example 20 in table 5.6 is showing this circumstance vividly. The tweet contains work-related terms and is classified as *commercial*. The assumption that the text classification is working could also be derived by looking at examples 22–25. Example 22 is written in Portuguese and has spiritual and religious content. Even though the true label is *residential*, the classifier identified the text as *other*. This category covers churches, temples, or other spiritual facilities. The Spanish tweet depicted in example 23 is about taking a break from work which the classifier could interpret as a commercial context. Therefore, the tweet was classified as *commercial*. Example 24 could be a mixed-use building. The text is about a carnival video presumably from a place named *kidspace* which could be a daycare center or an indoor playground. The last classification example, 25, is about a street of small book stores in Madrid *Librerías de la Cuesta de Moyano*. The word *librerías* could be used in tweets posted from libraries or book stores. Both words could refer to a library or a book store in Spanish. In this case, the classifier might decide the tweet came from a library assigned to the *other* class. For this case, more local knowledge for the classifier would be beneficial.

5.8 Discussion

In general, deep learning methods can boost the performance of the building function classification at an individual building level. The Naïve Bayes trained with TF-IDF features states with a solid baseline. The three deep learning methods achieved partially higher F1 scores. DistilBERT reaches the highest monolingual classification scores, followed by the LSTM trained with pre-trained English fastText embeddings. For the multilingual part, the self-trained fastText embedding and an LSTM classifier can achieve on par results, however, is slightly outperformed by the multilingual variant of DistilBERT by F1 in 5 of the 9 different datasets.

Even though the differences between the models are moderate, what could be learned is that models using subword information like fastText or BERT reach higher scores than models using features at a token level. Despite using sentence embeddings composed of the tweets' subword information, the monolingual BERT model sets the benchmark. It achieves higher accuracies and Kappa scores for most of the datasets. The multilingual models also show reasonable results by taking into account the language diversity [35], informal character of the tweets (cf. tables 5.6 and 5.5), or translingual constructs [36], e.g., example 4 in table 5.5. Particular interesting are the classification results yielded by the LSTM trained on the self-trained multilingual embeddings. As discussed in section 5.6.5, the embedding seems to deliver almost the same useful features as multilingual BERT. Highly specialized embeddings could be pivotal for the task investigated in this dissertation. In smaller datasets, they could be able to cover the informal Twitter language better at a (sub) word level and perhaps embed topics discussed on Twitter in a more meaningful way (cf. table 5.4, box 5.4f). It seems that multilingual BERT yields only significantly higher accuracies when the dataset is very large. Here, BERT can show its full potential. However, if the datasets get smaller, the differences in performance are not so explicit anymore. This finding states that almost the same results can be achieved by methods with less hunger for energy, economical reasonable hardware, and, along going, a decreased amount of training hours.

The impact of the different dataset sizes (cf. table 4.5), i.e., distinct tweet-house-limits (λ, cf. table 4.4), is also detectable. The large datesets, i.e., no tweet-house-limit yielding high accuracies (cf. table 5.2) but poorer label agreement, is reflected by the Kappa score. This could be evidence of overfitting toward the dominant classes. The plausibility of this direction could be shown in figure 5.7. If the tweet-house-limit rises, i.e., a more restrictive λ value, the overall accuracies drop, however, not strikingly. On the other hand, an increase of F1 values of the underrepresented *residential* class can be seen. In general this class has much less tweets (cf. tables 4.3 and 4.5).

The recommendation which can be given after reviewing the results is to establish a balancing of tweets per building. However, due to the different nature of the buildings (landmarks vs. residential houses), establishing a holistic rule is tricky. Strict downsampling would reduce the dataset to the bare number of buildings, and neglecting to balance leads to overfitting. Therefore, the combination of reducing tweets via a λ value and class-weights might be a way to impact the classification.

However, the analysis of the results of the text classification task confirms that the building function classification task at an individual building level is challenging. Particularly demanding is the situation of mixed building functions (cf. section 5.7.2). For example, a residential building accommodates a restaurant, a small supermarket, bakery, or other businesses [20]. As future research, a method for sample weights based on the distance of tweets to the building could be researched. Tweets closer to a building could be more related to a building. The farther a tweet is away, the higher the likelihood of a falsely assigned tweet to a building. Therefore, the sample weight of a tweet farther away could be reduced and lower its impact on classification. Figure 5.5 perhaps suggests that a sample weight based on the tweet's distance could be an option. Data fusion with social media images or street view-like images could reduce uncertainties regarding mixed-used images. For example, images like depicted in figure 5.9, could be collected and used as additional data source to support the classification.

In this chapter, the primary research question *1* has been answered if multilingual linguistic features can contribute to building function classification at an individual building level was explored. It can be answered in general with *yes*. Research question *2* has been answered as well. Models using subword information are a good approach for multilingual building function classification. The contextualized embeddings returned by BERT yielding strong classification results of multilingual sequences. Additionally, the linguistic reality of cities can be covered by features obtained from a self-trained multilingual embedding as well. LSTMs trained with these features achieve competitive results compared to BERT. The examples above show that they can handle multilingual social media text sequences well and lead to good classification results.

Nevertheless, the embedding can be further optimized by using more training data and sophisticated tokenizing. Additionally, more formal corpora, e.g., (multilingual) Wikipedia dumps, could be used as additional training data next to tweets. Finally, research question *3* regarding the impact of a balanced dataset could also be answered. It could have been shown, that the overall scores slightly decreased; however, the classification scores in underrepresented classes could be increased in some cases. They benefit from more balanced training data.

5.9 Summary

In this chapter, the training of the text classification models was introduced. The different settings and configurations of the classifiers have been explained. Also, the classification results of the different dataset sizes have been documented and discussed. Models taught on larger datasets, i.e., more tweets per building, reach higher scores but seem slightly to overfit the dominant classes *commercial* and *other*. On the other hand, models trained on smaller datasets achieve lower overall scores but more balanced results regarding the minor *residential* class.

Furthermore, a multilingual text analysis has been given. To sum up, deep learning methods, especially subword-based methods like fastText or BERT can reach higher classification results. It was also discovered, that an LSTM trained with word vectors from a self-trained multilingual Twitter fastText skip-gram model achieved partially on par classification scores as multilingual BERT or was slightly outperformed. Subword information and specialized (multilingual) linguistic features seem to be a recipe for multilingual building function classification at an individual building instance level across different urban areas.

The subsequent chapter introduces the remote sensing image classification results and discusses research question *4* asking about the usefulness of decision fusion. It is asked if the combination of text and vision is able to realize better classification outcomes for building function classification.

6 Fusion of Remote Sensing Images and Social Media Text Messages

This dissertation's first major research question was the building function classification on a building level using multilingual Twitter text messages. Section 5.6 shows promising results. The linguistic features proved as valuable data sources. The second core research objective is to combine, i.e., the fusion of remote sensing features and linguistic features to further improve the classification. This research question is discussed in the following sections.

6.1 Computer Vision Models

In this section, the selection of the computer vision model is explained. Since the main focus in this thesis is natural language processing, the computer vision models are only briefly discussed. All models are based on the work of [14] and are initialized with ImageNet [81] weights. [128] found that the practice of applying ImageNet weights can also be helpful in remote sensing image classification with relatively sparse datasets. The models used in this work are VGG16 [77], InceptionV3 [187], and ResNet50 [79]. The models have been fine-tuned with the Google aerial very high-resolution image dataset introduced in section 4.7 and table 4.7.

6.1.1 Training

Three state-of-the-art computer vision models have been fine-tuned on Google aerial images to predict the building functions of the selected 42 cities. The approach and fine-tuning process are based on [14] and for optimization, Adam [182] has been used and categorical cross-entropy as loss function is applied. For fine-tuning, two consecutive training steps are applied. First, all layers of the models are frozen except one dense layer. This layer is trained for 16 epochs. After this initial learning step, all layers are set to "trainable" one after another, beginning with the last layer. Meanwhile, the learning rate is decreased. The exact fine-tuning steps can be found in table 6.1.

6.1.2 Results of the Aerial Image Classification

First, an initial overview over the remote sensing image classification is given. Table 6.2 illustrate the overall classification results of the models. The InceptionV3 is performing well amongst classes. It achieves a good precision score predicting the *commercial* class. The *other* classes' results are more balanced. Recalling the text classification results (cf.

model	step	learning rate	epochs	# trained layers
InceptionV3	1	1e-4	16	1
	2	1e-5	16	311
ResNet50	1	1e-4	16	1
	2	1e-5	16	175
VGG16	1	1e-4	16	1
	2	1e-5	16	21

Table 6.1: Fine-tuning protocol on aerial imagery applied to selected computer vision models.

	Monolingual			Multilingual		
Model	OA	F1 (macro)	κ	OA	F1 (macro)	κ
InceptionV3	0.70	0.70	0.552	0.72	0.72	0.580
ResNet50	0.67	0.67	0.507	0.68	0.68	0.514
VGG16	0.71	0.71	0.570	0.73	0.73	0.588

Table 6.2: Aerial image classification results. The scores presented in this table are overall accuracy (OA), and Cohen's Kappa (κ), and F1 (macro).

All models perform stronger than the text models (cf. table 5.2). The veteran model VGG16 can achieve the best result measured in accuracy, F1, and Kappa scores. The weakest model denotes the ResNet50 network.

Table 6.3 shows the class-wise performance of the computer vision models.

table 5.3), where the *residential* class always performed below the two *other* classes, it can be seen here that the *residential* class yielded the highest precision, recall, and F1 score.

By comparing the class-wise performance of the ResNet50, it can be seen that the performance is not as good as the InceptionV3 numbers. Throughout all classes, the scores achieved are lower. However, the *residential* class performs best.

The VGG16 attains the highest precision scores except for the *residential* class. On the other hand, the recall score is the lowest for the *commercial* class. The *residential* class yields a strong recall of 0.91, which indicates a high true-positive rate.

Interestingly, the best performing class of all vision models is the *residential* class that underperforms in the text classification. A possible explanation is classifying residential homes with text are difficult opposed to remote sensing imagery. Residential places could have very distinct shapes and maybe more greenspaces, e.g., gardens. In contrast, the text classification struggles to identify homes because of a much smaller number of tweets and possibly much higher heterogeneity of the texts. From home, one can discuss everything.

		commercial			other			residential		
		P	R	F1	P	R	F1	P	R	F1
mono.	InceptionV3	0.70	0.62	0.66	0.66	0.66	0.66	0.74	0.83	0.78
	ResNet50V2	0.66	0.59	0.63	0.63	0.63	0.63	0.72	0.79	0.75
	VGG16	0.75	0.57	0.65	0.71	0.66	0.68	0.69	0.91	0.78
multi.	InceptionV3	0.69	0.71	0.70	0.69	0.67	0.68	0.79	0.78	0.78
	ResNet50V2	0.65	0.65	0.65	0.62	0.66	0.64	0.76	0.72	0.74
	VGG16	0.75	0.63	0.68	0.64	0.78	0.70	0.81	0.77	0.79

Table 6.3: Class-wise remote sensing image classification. The scores presented in this table are precision (P), recall (R), and F1 (F1).

The finding that a modality is better classifying a particular class, in this case *residential*, is a strong argument for data fusion. The two modalities could complement each other to improve the overall classification result for individual building function classification. The upcoming section investigates the second central research question raised in this dissertation: Does data fusion enhance building function classification at an individual building level.

6.2 Decision level Fusion of Social Media Data and Remote Sensing Aerial Images

In the following paragraphs, the results of the decision-level fusion of the aerial remote sensing images and the Twitter text messages are shown. First, the overall results are presented and followed by an overview of the class-wise performance. For the sake of clarity, only the F1 score is used as a metric in table 6.4. The decision level fusion is executed after the classification of the individual modalities (cf. section 2.2). Hence, the fusion process in this work is triggered after the text and remote sensing image classification (cf. figure 6.1, *1,2,3*). The softmax probabilities of the classifiers are avergaged (cf. figure 6.1, *4*).

As pointed out in section 4.5.5, after the dataset prapration procedure (cf. section 4.5) several tweets can be assigned to one building. That results in a $1 : n$ relationship between buildings and tweets. Since there is only one remote sensing image, a $1 : 1$ relationship must be established. Therefore, all the predictions of a specific building are averaged (cf. equation 6.1). In other words: a mini-fusion before the actual fusion process. After this step, the predictions are now ready for the building-wise fusion process.

The fusion process per building $b \in B$, where B describes a non-empty set of OSM building IDs can be formally defined by:

$$f_b = \text{argmax}\left[\frac{1}{2}\left(\left(\frac{1}{|T_b|}\sum_{\mathbf{t}\in T_b}\mathbf{t}\right) + \mathbf{i_b}\right)\right] \tag{6.1}$$

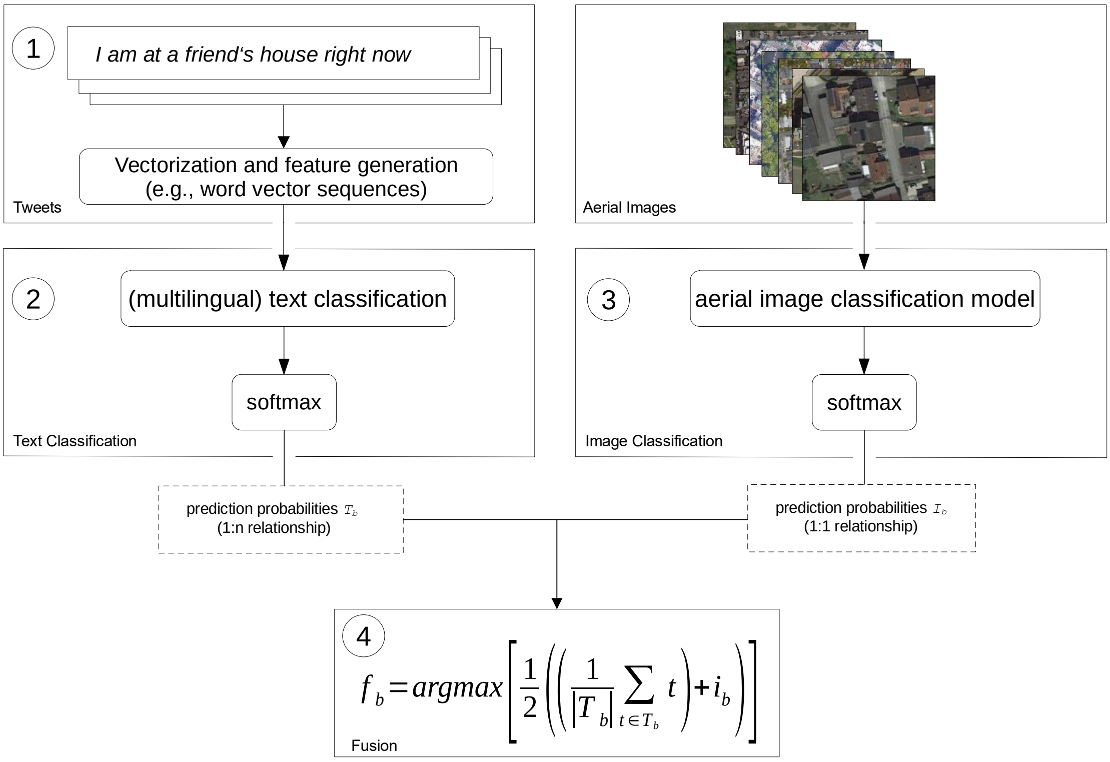

1
I am at a friend's house right now
Vectorization and feature generation (e.g., word vector sequences)
Tweets
Aerial Images
2
(multilingual) text classification
softmax
Text Classification
3
aerial image classification model
softmax
Image Classification
prediction probabilities T_b (1:n relationship)
prediction probabilities I_b (1:1 relationship)
4
$f_b = argmax\left[\frac{1}{2}\left(\left(\frac{1}{|T_b|}\sum_{t \in T_b} t\right) + i_b\right)\right]$
Fusion

Figure 6.1: Decision fusion framework. *1-2* shows the text classification module. *1* depicts vectorization and linguistic feature generation module. *2* depicts the actual text classification module. *3* depicts the aerial image classification module. *4* displays the fusion formula. Background images © TerraMetrics 2021, Google.

The predictions of both models are stored as probability vectors of the size $n_{classes}$. The text predictions are represented as a set of probability vectors T. A prediction made from a Tweet for a building b is noted as $\mathbf{t} \in T$. Analog to the text predictions, predictions for a building based on remote sensing features are specified as a set of probability vectors I where $\mathbf{i} \in I$.

The fusion methodology that is used in this work is straightforward, computationally cheap, and flexible. E.g., if more recent classification results are available, the fusion methods can be quickly applied without training a new model.

6.3 Results of the Decision Level Fusion

For the decision level fusion, three computer vision models yielding the largest Cohen's Kappa score are selected. The fusion was then conducted with the text classification results showing the most balanced results. In this case, the text classification results of the 5-95-encoded datasets are used. The fusion results are documented in table 6.4. Only the model combination yielding the best fusion result is presented.

Remember, that after labeling, buildings and tweets are in a 1:n relationship. This means that several tweets can be assigned to one building. During text classification, the analysis of the results have been conducted tweet-wise. Here in the fusion chapter, on the other hand, the analysis is performed building-wise. It can be seen in equation (6.1) that not only the propabilities of the text and image results are fused but also the text results. As mentioned in section 6.2, this enusures a 1:1 relationship between tweets and buildings. This process is also improving the text classification results. For the tweet-wise text classification see tables 5.2 and 5.3 in section 5.6.

Table 6.4 documenting the overall fusion results which denote the results of the key research question of data fusion. Fusing the Naïve Bayes model (OA 0.54, κ0.314) with the VGG16 model (OA 0.71, κ0.570) can achieve a better overall accuracy and a higher Kappa score (OA 0.73, κ0.602). Also, the class-wise performance can be elevated. The F1 score of the *commercial* class can be increased from 0.52 and 0.65 to 0.68. Also, the scores of the *other* category can be improved from 0.60 and 0.68 to 0.72. The *residential* class was classified pretty well by the VGG16 beforehand. Therefore, the increase is moderate but visible: from 0.48 and 0.78 to 0.79. The text results also improved slightly from a Kappa score of 0.259 to 0.314.

The fusion of the virgin model *LSTM-V + VGG16* also yields higher scores. The combination of the text model (OA 0.57, κ0.361) and the VGG (OA 71, κ0.570) produces better results (OA 0.75, κ0.621). The F1 score of the *commercial* class can be improved from 0.57 and 0.65 to 0.70. Here, the deep learning model can contribute to a higher result compared to the baseline model. The *other* class can also benefit and now has F1 scores from 0.61 and 0.68 to 0.73. Here, the die baseline was also beaten. The *residential* class also had improved results which can be reported with a strong F1 score of 0.80 after fusion.

	Monolingual												Multilingual					
	TFIDF-NB + VGG16			LSTM-V + VGG16			LSTM-F + VGG16			BERT + VGG16			mLSTM-F + VGG16			mBERT + VGG16		
	T	V	F	T	V	F	T	V	F	T	V	F	T	V	F	T	V	F
C	0.52	0.65	0.68	0.57	0.65	0.70	0.58	0.65	0.71	0.60	0.65	0.71	0.59	0.68	0.72	0.60	0.68	0.72
O	0.60	0.68	0.72	0.61	0.68	0.73	0.62	0.68	0.73	0.63	0.68	0.73	0.63	0.70	0.73	0.63	0.70	0.74
R	0.48	0.78	0.79	0.53	0.78	0.80	0.52	0.78	0.81	0.58	0.78	0.81	0.47	0.79	0.79	0.50	0.79	0.79
OA	0.54	0.71	0.73	0.57	0.71	0.75	0.58	0.71	0.75	0.60	0.71	0.75	0.58	0.73	0.75	0.58	0.73	0.75
κ	0.314	0.570	0.602	0.361	0.570	0.621	0.365	0.570	0.623	0.403	0.570	0.629	0.362	0.588	0.619	0.377	0.588	0.625

Table 6.4: Fusion results. The text results (T) refer to the results achieved with the dataset 5-95. C=Commercial O=Other$R$=Residential. The selected vision model (V) is the model which yields the highest fusion result (F). The scores presented in this table are overall accuracy (OA), Cohen's Kappa (κ), and class-wise F1 (macro).

The text classification results could be further improved from a Kappa score of 0.284 to 0.361 which outperforms the baseline.

By fusing the model trained with monolingual fastText embeddings, *LSTM-F + VGG16*, the overall outcomes could also be improved. The overall scores improve by fusing the two modalities from (OA 0.58, κ0.365) and (OA 0.71, κ0.570) to (OA 0.75, κ0.623). By comparing the results with the virgin model, it can be seen that the increase is subtle. For the *commercial*, the F1 scores improved from 0.58 and 0.65 to 0.71. Here, an elevation of 0.01 can be reached compared to the virgin embedding model. For the *other* class, the F1 raised from 0.62 and 0.68 to 0.73. No improvement can be found by comparing the differences between the virgin and the fastText model. The *residential* class F1 score can also slightly improve from 0.52 and 0.78 to 0.81. Here, the fusion F1 is slightly higher than the virgin embedding model. The text classification can be improved from 0.297 to 0.365, slightly higher by comparing the score to the virgin embedding result.

The best fusion result of all the monolingual models yields the combination of BERT and the VGG16, *BERT + VGG16*. The scores can be elevated from (OA 0.60, κ0.403) of the text model and (OA 71, κ0.570) of the vision model to fused (OA 0.75, κ0.629). However, by looking at the class-wise outcomes, it can be seen that the F1 scores are identical to the scores reached by the fastText model. For *commercial*, the F1 enhanced from 0.58 and 0.65 to 0.71. The scores of the *other* class improved from 0.62 and 0.68 to 0.73. For *residential*, F1 raised from 0.52 and 0.78 to 0.81. The text classification results benefit more from the fusion. They climbed from a Kappa of 0.331 to 0.403.

For the model trained with the multilingual fastText model, *mLSTM-F + VGG16* also positive results can be reported. The scores can be improved from (OA 0.58, κ0.362) and (OA 0.73, κ0.588) to (OA 0.75, κ0.619). For the *commercial* class, the scores can be improved from 0.59 and 0.68 to 0.72. The *other* class exhibits an improvement from 0.63 and 0.70 to 0.73. The F1 scores of the *residential* is 0.79 which means no improvement compared to the remote sensing results. The text classification result can also be enhanced. The cape score increased from 0.294 to 0.362.

Finally, the fusion of the multilingual BERT model with the VGG16 can also improve the outcome of the building function *(mBERT + VGG16)*. The scores climb from (OA 0.58, κ0.377) and (OA 0.73, κ0.588) to (OA 0.75, κ0.625). By comparing the achieved numbers of the model trained with multilingual fastText embedding with the BERT scores, it can be seen that the fusion could not improve further. For the *commercial* class, an improvement from 0.60 and 0.68 to 0.72 can be reached, quasi the same as the fastText model's outcome. The text classification results cannot be outperformed by the fastText model. The same situation can be found by looking at the F1 score of the *other* class: an improvement can be detected. The numbers can be elevated from 0.63 and 0.70 to 0.74 (fastText model fusion 0.73). For *other*, the text classification results are identical (0.63 and 0.63). The *residential* class' scores can be improved via decision fusion from 0.50 and 0.79 to 0.79. By looking at the F1 of the text result, it can be seen that the BERT model slightly outperformed the fastText model. The pure text classification results can also be improved by the data fusion from a Kappa score of 0.308 to 0.377.

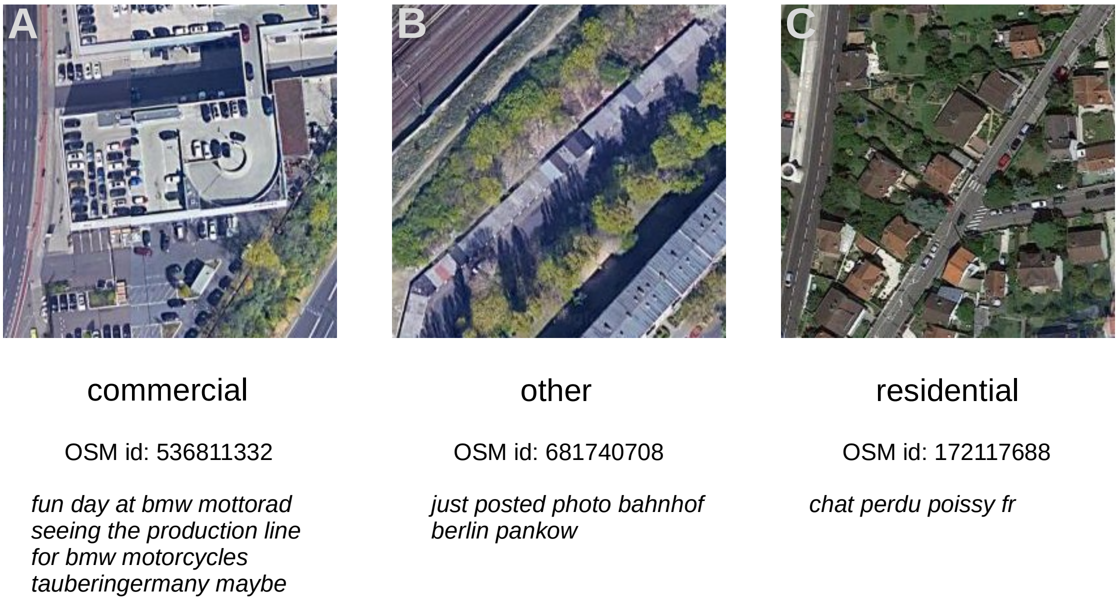

A
B
C
commercial
other
residential
OSM id: 536811332
OSM id: 681740708
OSM id: 172117688
fun day at bmw mottorad seeing the production line for bmw motorcycles tauberingermany maybe
just posted photo bahnhof berlin pankow
chat perdu poissy fr

Figure 6.2: Agreement between text and image predictions. Background images © TerraMetrics 2021, Google.

6.4 Fusion Analysis

The preceding section showed the classification results and that the data fusion with the decision fusion method is beneficial for building function classification. This section will show where the fusion of remote sensing images and Twitter text messages is helpful. The examples displayed in this section are taken from fusing the LSTM trained on the multilingual fastText embeddings (9-95) and the VGG16.

6.4.1 Agreement

Figure 6.2 depicts some examples of an agreement between the text and image classification. For example, figure 6.2 A, shows a large complex with (rooftop) parking spaces. The text refers to a German car company visiting a production line for motorcycles. Both classifiers agree on the building label *commercial*. Figure 6.2 B, showing tracks, a covered structure and greenspaces. The text is written in a mixed-language style. It is about posting a photo at Pankow station in Berlin. Public transportation sites are labeled as *other*. Also, both classifiers concur. The last agreement example, C, depicts scattered low-rise buildings with various vegetation patches in them. A tight road is separating some of the buildings. The French tweet text is about a missing cat *chat perdu*. The examples shown in figure 6.2 are quite harmonic, demonstrating that an agreement between remote sensing images and tweet text is possible. However, the main target of data fusion in this work is enhancing the building function classification. Is it possible that the two modalities "backing each other up" if one sensor is wrong or uncertain?

6.4.2 Can modalities support each other?

It is intended to illustrate how the fusion of social media text messages and remote sensing images can improve building function classification. Figure 6.3 lists some examples where one modality supported the other during classification and so identified the correct building label.

Below are each example's text and, a string encodes the classification decisions of the classifiers. For example, string *T:c V:o F:c* means that text (T) classified *commercial* (c), vision (V) *other* (o) and fusion (F) results in *commercial* (c). The spelled-out label below each remote sensing image is the true label.

Example A shows a large grayish rooftop with a small portion of vegetation. The tweet text is about that the person had orders to try a new camera. Words like *order* or *new* may correspond to a more commercially used vocabulary. The word *camera* could also be associated with a professional photographer. In this example, the text helped to classify the building correctly.

The following example, B, depicts a residential area with a relatively large open green space and scattered trees. The text is about how the author would like to be a bee, drinking nectar, being a child again. It seems that this text is confusing for the classifier. As a consequence, the text model miscategorizes the text as *other*. On the other hand, the remote sensing image model can correctly identify the building as *residential*. After fusing, the building is classified accurately into *residential*.

Example C demonstrates the issue of mixed-used buildings. The remote sensing image shows a residential building block in Berlin. The buildings encompass a courtyard with some trees. Additionally, a parking lot is visible, and a road is dissecting blocks from each other. The tweet, however, refers to a pharmacy (*apotheke*) (sic). A pharmacy is categorized as *other* building. As also pointed out in section 5.7.2, the classification of the text itself is "correct". However, the mixed-use of the building confuses the text classifier. The image classifier's prediction is true, and the fusion results yield the correct building label *residential*.

The image in example D displays a compound with paved roads and gray and a large roof. No vegetation what so ever is visible. These visual features could indicate a commercial place, e.g., a mall or an industrial plant. The text is about participating in workshops, e.g., painting or making kombucha[1]. Sometimes, text features could be confusing. The text classifier ranks these tweets as *residential*, and the image classifier can label the building correctly.

In the next example, E, however, shows that the text can also help to correct the vision classification. The image depicts crossroads and broad streets, vegetation with trees and bushes. The vision classifier tagged this image as *residential*, but the true label is *commercial*. The tweet helped to identify the correct label. The hungry Twitter user

[1] `https://en.wikipedia.org/wiki/Kombucha` [14.12.2021, 16:11]

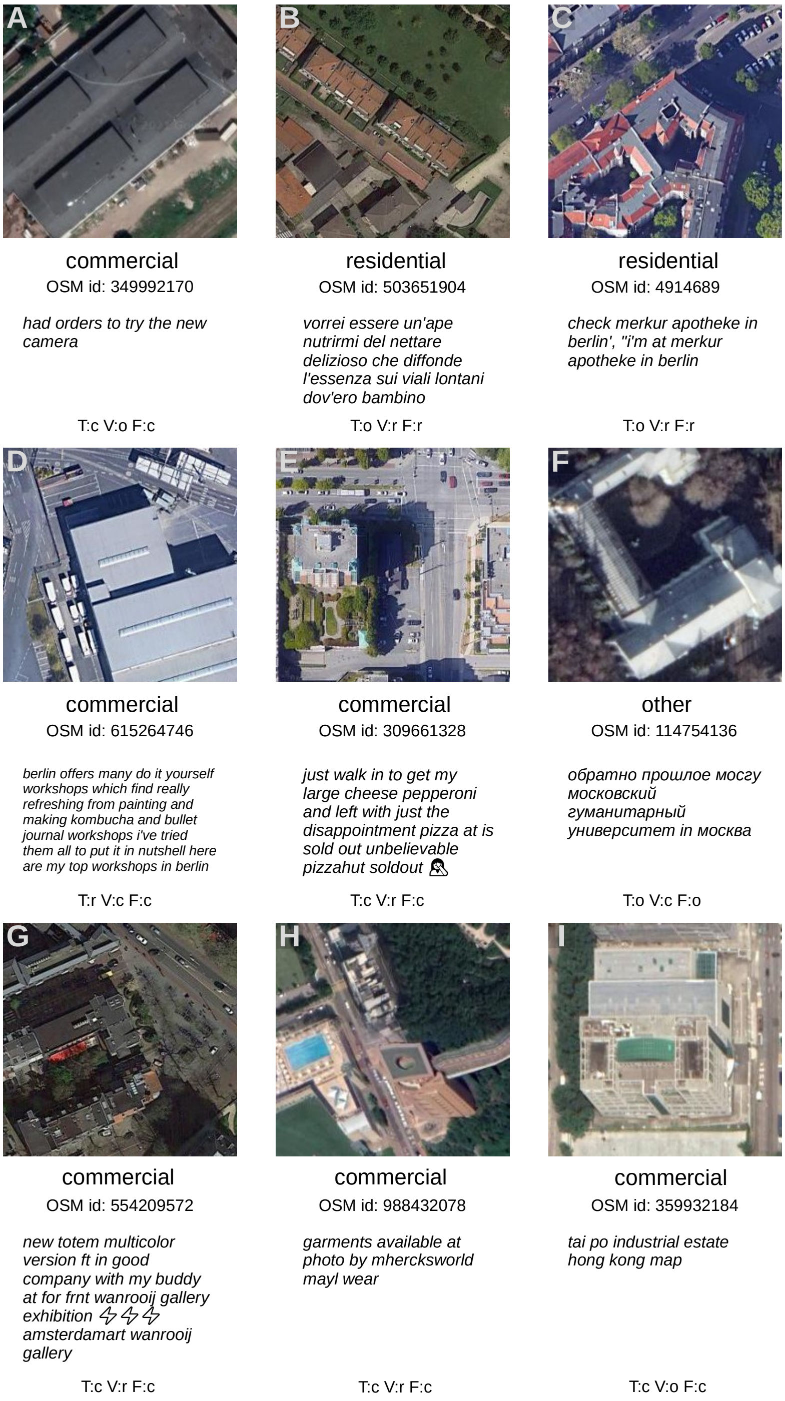

A
B
C
commercial
OSM id: 349992170
had orders to try the new camera
T:c V:o F:c
residential
OSM id: 503651904
vorrei essere un'ape nutrirmi del nettare delizioso che diffonde l'essenza sui viali lontani dov'ero bambino
T:o V:r F:r
residential
OSM id: 4914689
check merkur apotheke in berlin', "i'm at merkur apotheke in berlin
T:o V:r F:r
D
E
F
commercial
OSM id: 615264746
berlin offers many do it yourself workshops which find really refreshing from painting and making kombucha and bullet journal workshops i've tried them all to put it in nutshell here are my top workshops in berlin
T:r V:c F:c
commercial
OSM id: 309661328
just walk in to get my large cheese pepperoni and left with just the disappointment pizza at is sold out unbelievable pizzahut soldout 🤦‍♀️
T:c V:r F:c
other
OSM id: 114754136
обратно прошлое мосгу московский гуманитарный университет in москва
T:o V:c F:o
G
H
I
commercial
OSM id: 554209572
new totem multicolor version ft in good company with my buddy at for frnt wanrooij gallery exhibition ⚡⚡⚡ amsterdamart wanrooij gallery
T:c V:r F:c
commercial
OSM id: 988432078
garments available at photo by mhercksworld mayl wear
T:c V:r F:c
commercial
OSM id: 359932184
tai po industrial estate hong kong map
T:c V:o F:c

Figure 6.3: Data fusion examples. This figure shows examples were data fusion improved classification. The encodings below the images denote the classification pattern of the modalities. *T=Text, V=Vision, F=Fusion, c=commecial, o=other, r=residential.* All text predictions come from the LSTM trained with multilingual fastText vectors (9-95) and vision predictions from multilingual VGG16. Background images © TerraMetrics 2021, Google.

wanted a pizza, but the pizzeria was *soldout* (sic). As previously discussed in 5.7.2, it seems that food-related terms are classified as *commercial*. For this reason, the building could be correctly classified as *commercial*.

The following example, F, shows a larger building surrounded by trees. The vision model classifies the image into *commercial*. The text contrasts this result. The text is about Moscow University. The text can provide additional information regarding this building. After fusion, the building is correctly classified as *other*.

Example G displays a possible winter scene (trees without leaves), a block of buildings with a courtyard, and a road. The vision classifier marks the building as *commercial*, and the text refers to colors, good company, a buddy, a gallery, and an exhibition. Galleries are commercial places, and the word *company* (despite the semantics) could lead the text classifier into the direction of a commercial building. The fusion result yields correctly *commercial*.

The second last example, H, shows an ample green space with dense trees and grass-covered areas. Furthermore, a swimming pool is visible. The building is visibly hard to identify due to the comparatively coarse image resolution. The vision model classifies the building as *residential*. Here, the text can help: it is about an apparel brand and seems to be advertising a product. In the end, the building was correctly classified as *commercial*.

The final example, I, shows a blockish building with a flat roof, trees, and an adjacent road. The building is classified as *other* by the computer vision model. The text backs the classification up. It is about industrial real estate—the fusion result *commercial*, which is the proper label.

6.5 Discussion

In general, the above-documented data fusion results suggest that the straightforward fusion approach is working and improves the building function classification task results. The fusion results in table 6.4 demonstrate this fact.

By analyzing the fusion results, it can be seen that text and images achieve agreement of text and building function (cf. figure 6.2). Possible prototypical structures in the images and (multilingual) linguistic features in the text are forming a reasonable classification. Nevertheless, as pointed out in the previous section, the strength of fusion can be seen when taking a look at the disagreement between text and image classification (cf. figure 6.3). When one modality is confused by an unusual building shape or a strange text sequence, the other can support the classification.

The examples in the latter section show exactly this behavior despite the straightforward decision fusion approach. The text classifier can have difficulties classifying residential buildings. On the contrary, the computer vision models can achieve way better classification performances of the *residential* class (cf. table 6.3). A possible explanation is shown in figure 6.3 (B and C). Remote sensing images show that residential buildings have vegetation features like trees or green spaces adjacent to them or have relatively small roofs. For this reason, the vision models contribute by delivering strong classification results based on visual features.

The text classification can contribute as well. If the vision classifier is maybe distracted by trees shimmering on a rooftop, linguistic features could give the final hint (cf. figure 6.3, F). Moreover, example F is an excellent example of why multilingual texts should be included in building function classification. The depicted building has no English tweets surrounded at all. The used decision fusion method can compensate for a missing prediction. However, the building would remain a misclassification. By using multilingual content, the applied fusion method could resolve the wrong classification.

A disagreement between text and image classification could be evidence for a mixed-used building (cf. figure 6.3, C). For future research such examples could be filtered out for further analysis or combine a third perspective, e.g., a streetview perspective (cf. section 5.7.2 and figure 5.9). Maybe a specialized classifier could resolve the conundrum of mixed used buildings to a certain degree.

However, by taking a look at table 6.4 it can be noticed that even though fusing the best multilingual fastText model with the best vision model, the fusion result is almost equal to the superior multilingual BERT classifier. This result could be owed to the fusion method used in this work. Since the text classification prediction probabilities are first averaged building-wise to achieve a 1:1 relationship between tweets and buildings/images, correct classifications of single tweets could be canceled out by misclassifications. The same could happen during averaging the fused text probabilities with the image probabilities. For example, the *residential* class benefits least from adding the text modality, and the results stay the same.

In conclusion it can be mentioned that the overall results suggest that the fusion of remote sensing images and multilingual text messages is beneficial for building function classification. Even though the fusion method is straightforward, it can compensate misclassifications and improve the overall results.

6.6 Summary

This chapter shows the fusion of aerial remote sensing images and social media text messages. It was explained, how the vision models have been fine-tuned and the results have been introduced. The vision models achieved good individual performance. Here, the *residential* class sticks out with solid results. The additional performance brought the data fusion with social media text classification results. However, even though every text model can improve the building function classification results in general, the performance is almost equal between the fusion results of the deep learning text models. That said, the fused text classification results point out that models using subword information, i.e., fastText and BERT models, can outperform more naïve approaches. The outcome of the fusion experiment suggests promising results opening new insights to the building function classification at an individual building level.

7 Conclusion and Outlook

7.1 Conclusion

The journey from the first written symbols of humankind and cave paintings to tweets as citizen sensors and remote sensing images is long. Today, billions of geotagged social media posts from almost every place on earth are available. Therefore, using social media data in combination with remote sensing images to tackle the U.N. Sustainable Development Goals could be a valuable instrument. Human or citizen sensors may be one gear in the machinery of urban remote sensing.

For this reason, this dissertation analyzes the combination of remote sensing images and multilingual social media text messages from Twitter for individual building function classification in 42 cities. The focus is on natural language processing methods to classify the building functions into *commercial*, *residential*, and *other*. State-of-the-art methods like the neural language model BERT are used to generate features for the classification. Also, a self-trained multilingual Twitter skip-gram embedding is trained to generate multilingual word vectors.

In this dissertation, four primary research questions are investigated:

1. Can linguistic features derived from social media text messages from Twitter contribute to building function classification at an individual building level?
2. How can the multilingual reality in urban areas be represented best so that they are beneficial for building function classification?
3. What is the effect of unbalanced and balanced datasets on building function classification?
4. Finally, are visual features derived from very high-resolution remote sensing images complementary to linguistic features?

Despite the challenge regarding the accuracy of the geolocation of the tweets, the results reached via the text classification are positive. The best performing monolingual classifier is BERT. Depending on the datasets, it achieves accuracies from 0.55 to 0.59 and Kappa score range 0.248 to 0.332 (cf. tables 5.2 and 5.3). All deep learning models accomplish better results than the Naïve Bayes baseline in the majority of the cases. To conclude, the monolingual part of research question 1 can be answered *yes*.

For the multilingual part, i.e., research question 2, the LSTM model fitted with word vectors derived from a self-trained multilingual Twitter fastText embedding can perform almost on par with the large multilingual language Model BERT. The LSTM model gains accuracies from 0.54 to 0.57 and Kappa scores from 0.217 to 0.295. The multilingual

BERT model, on the other hand, achieves accuracies from 0.54 to 0.63 and Kappa scores from 0.298 to 0.336. Even though BERT attains a higher accuracy on the largest dataset, the performance converge to the fastText results when the datasets get smaller. The multilingual fastText models show almost identical performance but the multilingual BERT models yield slightly higher accuracies, higher Kappa scores, and class-wise higher F1 scores (cf. tables 5.2 and 5.3). Even though the own embedding was beaten by BERT, the findings give insights how a multilingual building function classification could be approached in the future. Namely, highly specialized self-trained embedding can produce feature vectors which are nearly as useful as the vectors produced by BERT for building function classification. This discovery is especially practical when data or hardware resources are sparse because the linguistic diversity in multicultural urban areas can be represented well with a self-trained multilingual Twitter word embedding and achieve almost on par classification results as BERT models.

To summarize the text classification, it can be said that models using subword information, i.e., the fastText models and contextualized information, i.e., BERT, are performing best. The self-trained multilingual Twitter fastText embedding appears to be a suitable alternative for multilingual BERT within the context of building function classification. It turned out to be a possible alternative for BERT.

Furthermore, research question 3 investigates if more balanced datasets, i.e., the limitation of the total number of tweets per building, impacting on text classification. Since landmark buildings could incorporate thousands of tweets, they could bias the text classification toward biased buildings and shadow buildings with much fewer tweets. Therefore, nine different mono and multilingual datasets are generated to explore if the text classification benefits from such a limitation. A different tweet-house-limit, λ, is computed to create the datasets. The λ value represents the mean number of tweets per building. The calculation is restricted by excluding buildings with a very high tweet count using nine upper and lower bounds. This process automatically leads to more balanced classes. The results indicate that the largest *full* datasets with restrictions show poorer Cohen's Kappa scores in comparison to smaller datasets. The models tend to overfit (cf. section 5.7.1 and figure 5.7) Furthermore, the drop in performance of smaller datasets is not as distinct as excepted. The *residential* class, with fewer tweets, profits the most by a tweet-house limitation. The class-wise results in table 5.3 show that if λ decrease, i.e., lowering the maximum tweets per building, the higher the F1 score of the *residential* class gets. Therefore, the usage of Twitter data without a limitation of tweets per building leads to decreased results for the minority class. When a tweet-house limit is established, the results become more balanced, and the drop in performance of the majority classes is moderate.

The applied computer vision models fine-tuned and Google aerial images show strong results compared to text classification. The best performing vision model is the VGG16 with an overall accuracy of 0.73 and a Kappa score of 0.588 (cf. table 6.2) The vision models perform better overall and class-wise. Opposed to the text classification, the *residential* class is the class with the highest classification performance (cf. table 6.3). These observations point directly to the usefulness of data fusion at a building function classification task. Remotely observed data and on-site information could be combined to support each other. Additionally, Google Maps images are available for everyone. Therefore, costly expenses for high-resolution remote sensing images can be avoided, and yet, good accuracies can be achieved.

Therefore, the fourth and final research question addressed in this Ph.D. thesis investigates whether visual and linguistic features are complementary. Fusing such diverse modalities as high-resolution aerial remote sensing images and Twitter text messages garnered positive results. The applied data fusion method, namely decision fusion of the classification probabilities, can benefit the building function classification task from this combination. The overall classification performance increases to an overall accuracy of 0.75 for the monolingual classification and 0.75 for the multilingual scenario. The two sensors back each other up, which leads to higher classification accuracies 6.4. Moreover, the simplicity of the fusion method evades the retraining of large models if something has changed, and predictions can be re-used in other experiments. This modularization encourages replicating the experiment in a research environment without top-notch GPU cloud servers.

This Ph.D. thesis found arguments for using multilingual georeferenced social media data as in situ sensors for building function classification for a better covering of the diverse linguistic reality in urban areas. It looks like the citizen sensor idea is genuinely a valuable source for urban remote sensing and building function classification. As discussed, the work with georeferenced social media data, especially with multilingual tweets, needs further refinements. Therefore, information recovered directly from a spot of interest could contribute to reaching the U.N. Sustainable Development Goals. Municipal administrations could use the gained data to improve infrastructure and, at the same time, the life of the citizens. However, although the big data revolution is definitely over (we now live amidst the big data age), climate change continues to transform. Therefore, the development of high-performing but economic algorithms and solutions is needed to tackle the U.N. Sustainable Development Goals *sustainably.*

7.2 Outlook

> *Without continual growth and progress, such words as improvement, achievement, and success have no meaning.*
>
> *—Benjamin Franklin*

To improve the citizen sensor idea further, some additional work can be done. In the future, a pre-trained large language model like BERT could be fine-tuned with tweets labeled with a building function to further improve the monolingual results. This model could be a *BuildingFunctionBERT* and added to the BERT expert model zoo. Such a model could act like a citizen sensor language model. Furthermore, the *other* class could be opened to a more fine-grained labeling scheme, e.g., "medical", "civic", "education", or "religious". This augmentation could elevate the internal feature representations of the individual classes, which might lead to improved building function classification. The multilingual embedding might be improved by using more data covering a whole year.

Seasonal words like *Christmas*, *Weihnachten*, or *Noël* would be mapped better in the linguistic feature space.

Although the straightforward decision fusion method is delivering good results, a drawback is the possible cancelation of predictions. By averaging the prediction probabilities, predictions could cancel each other out (cf. section 6.5). For this reason, future research could address a more sophisticated fusion methodology that causes higher performances and avoid the cancelation of predictions. The lack of feature fusion could induce a loss of information, and including or generating features by applying early or late fusion could cause an even higher classification score. Nevertheless, if the challenge of mixed-use buildings is not tackled, higher accuracies cannot be expected regardless of the fusion method.

The challenges of the mixed-used buildings again become clear, e.g., as pointed out by [20]. The results suggest that the text classification itself is working (cf. section 5.7.2). If a tweet refers to a Pizzeria at the ground level of a residential building, the tweet would be labeled as *residential* but classified as *commercial*. This finding could be seen as evidence that the text classification of individual building functions using tweets is working. As pointed out in section 5.7.2, a research perspective could be the investigation if extra optical sensor information at a street view level might be rewarding to detect mixed-use buildings, e.g., images like depicted in figure 5.9. A specialized image classifier trained only on a (human curated) mixed-used buildings dataset could turn the scale into a multilabel building function classification scenario with high performance.

By choosing light-weight algorithms like fastText or DistilBERT versions, the energy consumption might be lowered [188] to tackle the SDGs in a more environmentally friendly way. This idea could be an exciting research target worth investigating for the future within the context of urban remote sensing and building function classification. The insight found in this dissertation that light-weight algorithms such as fastText can generate feature vectors which are nearly as good as features derived from BERT could be a first starting point.

Furthermore, ethical concerns arise when using georeferenced social media data. Establishing a (geospatial) connection (e.g., during labeling) of georeferenced social media text and a concrete building can violate Twitter users' privacy. Since many users are not aware that their data is used for research [189], future research should include a privacy protection policy for the user[1]. Additional research is needed to address ethical concerns regarding building function classification using social media.

[1] In this thesis, for example, data in sample texts which refer to a natural person are replaced by a place holder (to the best of the author's knowledge).

Bibliography

[1] NASA. What is remote sensing?, 2021. Earth Data. Open Access for Open Science. visit: 13.1.2022, 17:59. URL: https://earthdata.nasa.gov/learn/backgrounders/remote-sensing.

[2] P. Ghamisi, B. Rasti, N. Yokoya, Q. Wang, B. Hofle, L. Bruzzone, F. Bovolo, M. Chi, K. Anders, R. Gloaguen, P. M. Atkinson, and J. A. Benediktsson. Multisource and Multitemporal Data Fusion in Remote Sensing: A Comprehensive Review of the State of the Art. *IEEE Geoscience and Remote Sensing Magazine*, 7(1):6–39, 3 2019. doi:10.1109/MGRS.2018.2890023.

[3] T. Mikolov, K. Chen, and G. Corrado. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at ICLR*, 2013.

[4] A. Kruspe, M. Häberle, E. J. Hoffmann, S. Rode-Hasinger, K. Abdulahhad, and X. X. Zhu. Changes in Twitter geolocations: Insights and suggestions for future usage. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 212–221, Online, November 2021. Association for Computational Linguistics. URL: https://aclanthology.org/2021.wnut-1.24.

[5] United Nations. Un-habitat strategic plan (2020–2023). 2019. URL: https://unhabitat.org/sites/default/files/documents/2019-09/strategic_plan_2020-2023.pdf.

[6] P. Rode, C. Keim, G. Robozza, P. Viejo, and J. Schofield. Cities and energy: Urban morphology and residential heat-energy demand. *Environment and Planning B: Planning and Design*, 41(1):138–162, 2014. URL: https://doi.org/10.1068/b39065, doi:10.1068/b39065.

[7] D. Wittmer and T. Lichtensteiger. Exploration of urban deposits: long-term prospects for resource and waste management. *Waste Management & Research*, 25(3):220–226, 2007. URL: https://doi.org/10.1177/0734242X07079183, doi:10.1177/0734242X07079183.

[8] S. Kulkarni, K. O'Reilly, and S. Bhat. No relief: lived experiences of inadequate sanitation access of poor urban women in India. *Gender & Development*, 25(2):167–183, 5 2017. doi:10.1080/13552074.2017.1331531.

[9] K. Mouratidis. Urban planning and quality of life: A review of pathways linking the built environment to subjective well-being. *Cities*, 115:103229, 8 2021. doi:10.1016/j.cities.2021.103229.

[10] I. Baud, M. Kuffer, K. Pfeffer, R. Sliuzas, and S. Karuppannan. Understanding heterogeneity in metropolitan india: The added value of remote sensing data for analyzing sub-standard residential areas. *International Journal of Applied Earth Observation and Geoinformation*, 12(5):359–374, 10 2010. `doi:10.1016/j.jag.2010.04.008`.

[11] A. Albert, J. Kaur, and M. C. Gonzalez. Using Convolutional Networks and Satellite Imagery to Identify Patterns in Urban Environments at a Large Scale. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 1357–1366. ACM, 2017. event-place: Halifax, NS, Canada. URL: `http://doi.acm.org/10.1145/3097983.3098070`, `doi:10.1145/3097983.3098070`.

[12] G. Metternicht, N. Mueller, and R. Lucas. *Digital Earth for Sustainable Development Goals*, pages 443–471. Springer Singapore, Singapore, 2020. `doi:10.1007/978-981-32-9915-3_13`.

[13] B. Howard, L. Parshall, J. Thompson, S. A. Hammer, J. Dickinson, and V. Modi. Spatial distribution of urban building energy consumption by end use. *Energy and Buildings*, 45:141–151, 2012.

[14] E. J. Hoffmann, Y. Wang, M. Werner, J. Kang, and X. X. Zhu. Model fusion for building type classification from aerial and street view images. *Remote Sensing*, 11(11), 2019. URL: `https://www.mdpi.com/2072-4292/11/11/1259`, `doi:10.3390/rs11111259`.

[15] S. Hu and L. Wang. Automated urban land-use classification with remote sensing. *International Journal of Remote Sensing*, 34(3):790–803, 2013. `doi:10.1080/01431161.2012.714510`.

[16] J. Xie and J. Zhou. Classification of urban building type from high spatial resolution remote sensing imagery using extended mrs and soft bp network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(8):3515–3528, 2017. `doi:10.1109/JSTARS.2017.2686422`.

[17] M. F. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 8 2007. `doi:10.1007/s10708-007-9111-y`.

[18] E. D'Andrea, P. Ducange, B. Lazzerini, and F. Marcelloni. Real-Time Detection of Traffic From Twitter Stream Analysis. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2269–2283, 2015. `doi:10.1109/TITS.2015.2404431`.

[19] E. Bokányi, D. Kondor, L. Dobos, T. Sebők, J. Stéger, I. Csabai, and G. Vattay. Race, religion and the city: twitter word frequency patterns reveal dominant demographic dimensions in the united states. *Palgrave Communications*, 2(1):16010, 4 2016. `doi:10.1057/palcomms.2016.10`.

[20] G. Dax and M. Werner. Information-optimal abstaining for reliable classification of building functions. *AGILE: GIScience Series*, 2:1–10, 2021. URL: `https://agile-giss.copernicus.org/articles/2/1/2021/`, `doi:10.5194/agile-giss-2-1-2021`.

[21] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2017.

[22] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(75537553):436–444, 5 2015. `doi:10.1038/nature14539`.

[23] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 1 2015. `doi:10.1016/j.neunet.2014.09.003`.

[24] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, 12 2017. `doi:10.1109/MGRS.2017.2762307`.

[25] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152:166–177, 6 2019. `doi:10.1016/j.isprsjprs.2019.04.015`.

[26] C. Persello, J. D. Wegner, R. Hänsch, D. Tuia, P. Ghamisi, M. Koeva, and G. Camps-Valls. Deep learning and earth observation to support the sustainable development goals, 2021. `arXiv:2112.11367`.

[27] S. Srivastava, J. E. Vargas-Muñoz, D. Swinkels, and D. Tuia. Multilabel building functions classification from ground pictures using convolutional neural networks. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, GeoAI'18, pages 43–46. Association for Computing Machinery, 11 2018. URL: `https://doi.org/10.1145/3281548.3281559`, `doi:10.1145/3281548.3281559`.

[28] J. Kang, M. Körner, Y. Wang, H. Taubenböck, and X. X. Zhu. Building instance classification using street view images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 145:44–59, 11 2018. `doi:10.1016/j.isprsjprs.2018.02.006`.

[29] X. Yang, C. Macdonald, and I. Ounis. Using word embeddings in twitter election classification. *Information Retrieval Journal*, 21(2-3):183–207, 2018. URL: `https://link.springer.com/article/10.1007/s10791-017-9319-5`.

[30] I. Breckner, H. Peukert, and A. Pinto. *The delicate search for language in spaces*, pages 209–226. In Siemund et al. [190], 2013. URL: `https://doi.org/10.1075/hsld.1`.

[31] I. Gogolin, S. Peter, M. E. Schulz, and J. Davydova. *Multilingualism, language contact, and urban areas: An introduction*, pages 1–15. In Siemund et al. [190], 2013. URL: `https://doi.org/10.1075/hsld.1`.

[32] J. Leimgruber. *The management of multilingualism in a city-state*, pages 227–256. In Siemund et al. [190], 2013. URL: `https://doi.org/10.1075/hsld.1`.

[33] G. Extra and K. Yağmur. Urban multilingualism in Europe: Mapping linguistic diversity in multicultural cities. *Journal of Pragmatics*, 43(5):1173–1184, 4 2011. `doi:10.1016/j.pragma.2010.10.007`.

[34] S. Kim, I. Weber, L. Wei, and A. Oh. Sociolinguistic analysis of twitter in multilingual societies. In *Proceedings of the 25th ACM conference on Hypertext and social media*, HT '14, pages 243–248. Association for Computing Machinery, 9 2014. URL: `https://doi.org/10.1145/2631775.2631824`, `doi:10.1145/2631775.2631824`.

[35] D. Mocanu, A. Baronchelli, N. Perra, B. Gonçalves, Q. Zhang, and A. Vespignani. The Twitter of Babel: Mapping World Languages through Microblogging Platforms. *PLOS ONE*, 8(4):e61981, 4 2013. `doi:10.1371/journal.pone.0061981`.

[36] J. Blommaert. Formatting online actions: #justsaying on Twitter. *International Journal of Multilingualism*, 16(2):112–126, 4 2019. `doi:10.1080/14790718.2019.1575832`.

[37] A. Ballatore and S. D. Sabbata. Los Angeles as a digital place: The geographies of user-generated content. *Transactions in GIS*, 24(4):880–902, 2020. `doi:10.1111/tgis.12600`.

[38] M. Häberle, M. Werner, and X. X. Zhu. Geo-spatial text-mining from Twitter – a feature space analysis with a view toward building classification in urban regions. *European Journal of Remote Sensing*, 52(sup2):2–11, 2019. URL: `https://www.tandfonline.com/doi/full/10.1080/22797254.2019.1586451`, `doi:DOI:10.1080/22797254.2019.1586451`.

[39] M. Häberle, M. Werner, and X. X. Zhu. Building Type Classification from Social Media Texts via Geo-Spatial Textmining. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 10047–10050, 2019. `doi:10.1109/IGARSS.2019.8898836`.

[40] K. C. Seto and J. M. Shepherd. Global urban land-use trends and climate impacts. *Current Opinion in Environmental Sustainability*, 1(1):89–95, 2009. `doi:https://doi.org/10.1016/j.cosust.2009.07.012`.

[41] H. Taubenböck and M. Wurm. Globale Urbanisierung - Markenzeichen des 21. Jahrhunderts. In *Globale Urbanisierung. Perspektive aus dem All.*, pages 5–10. Springer Spektrum, Heidelberg, Berlin, 1 edition, 2015.

[42] S. Salcedo-Sanz, P. Ghamisi, M. Piles, M. Werner, L. Cuadra, A. Moreno-Martínez, E. Izquierdo-Verdiguier, J. Muñoz-Marí, A. Mosavi, and G. Camps-Valls. Machine learning information fusion in earth observation: A comprehensive review of methods, applications and data sources. *Information Fusion*, 63:256–272, 11 2020. `doi:10.1016/j.inffus.2020.07.004`.

[43] M. Schmitt and X. X. Zhu. Data Fusion and Remote Sensing: An ever-growing relationship. *IEEE Geoscience and Remote Sensing Magazine*, 4(4):6–23, 12 2016. doi:10.1109/MGRS.2016.2561021.

[44] T. Esch, W. Heldens, and A. Metz. *Die Erde im Bild – Satelliten als Werkzeug zur Beobachtung der Landoberfläche*, pages 23–27. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015. doi:10.1007/978-3-662-44841-0_4.

[45] W. Fu, J. Ma, P. Chen, and F. Chen. *Remote Sensing Satellites for Digital Earth*, pages 55–123. Springer Singapore, 2020. URL: https://doi.org/10.1007/978-981-32-9915-3_3, doi:10.1007/978-981-32-9915-3_3.

[46] U. Sörgel. *Radar remote sensing of urban areas*. Springer, 2010.

[47] C. Mayer, J. Schaffer, T. Hattermann, D. Floricioiu, L. Krieger, P. A. Dodd, T. Kanzow, C. Licciulli, and C. Schannwell. Large ice loss variability at nioghalvfjerdsfjorden glacier, northeast-greenland. *Nature Communications*, 9(1):2768, 7 2018. doi:10.1038/s41467-018-05180-x.

[48] A. Ghulam, I. Porton, and K. Freeman. Detecting subcanopy invasive plant species in tropical rainforest by integrating optical and microwave (insar/polinsar) remote sensing data, and a decision tree algorithm. *ISPRS Journal of Photogrammetry and Remote Sensing*, 88:174–192, 2014. doi:https://doi.org/10.1016/j.isprsjprs.2013.12.007.

[49] S. Dutkiewicz, A. E. Hickman, O. Jahn, S. Henson, C. Beaulieu, and E. Monier. Ocean colour signature of climate change. *Nature Communications*, 10(1):578, 2 2019. doi:10.1038/s41467-019-08457-x.

[50] K. E. Joyce, S. E. Belliss, S. V. Samsonov, S. J. McNeill, and P. J. Glassey. A review of the status of satellite remote sensing and image processing techniques for mapping natural hazards and disasters. *Progress in physical geography*, 33(2):183–207, 2009.

[51] Z. Zhu. Change detection using landsat time series: A review of frequencies, preprocessing, algorithms, and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 130:370–384, 2017. doi:https://doi.org/10.1016/j.isprsjprs.2017.06.013.

[52] M. Rußwurm and M. Körner. Temporal vegetation modelling using long short-term memory networks for crop identification from medium-resolution multi-spectral satellite images. pages 11–19, 2017. URL: https://openaccess.thecvf.com/content_cvpr_2017_workshops/w18/html/Russwurm_Temporal_Vegetation_Modelling_CVPR_2017_paper.html.

[53] L. Shi, H. Taubenböck, Z. Zhang, F. Liu, and M. Wurm. Urbanization in china from the end of 1980s until 2010–spatial dynamics and patterns of growth using eo-data. *International Journal of Digital Earth*, 12(1):78–94, 2019.

[54] C. Geiß, T. Leichtle, M. Wurm, P. A. Pelizari, I. Standfuß, X. X. Zhu, E. So, S. Siedentop, T. Esch, and H. Taubenböck. Large-area characterization of urban morphology—mapping of built-up height and density using tandem-x and sentinel-2 data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(8):2912–2927, 2019.

[55] T. Esch, W. Heldens, A. Hirner, M. Keil, M. Marconcini, A. Roth, J. Zeidler, S. Dech, and E. Strano. Breaking new ground in mapping human settlements from space – the global urban footprint. *ISPRS Journal of Photogrammetry and Remote Sensing*, 134:30–42, 2017. doi:https://doi.org/10.1016/j.isprsjprs.2017.10.012.

[56] L. Wang, S. Wang, Y. Zhou, W. Liu, Y. Hou, J. Zhu, and F. Wang. Mapping population density in china between 1990 and 2010 using remote sensing. *Remote Sensing of Environment*, 210:269–281, 2018. doi:https://doi.org/10.1016/j.rse.2018.03.007.

[57] M. Paganini, I. Petiteville, S. Ward, G. Dyke, M. Steventon, J. Marry, and F. Kerblat. *Satellite earth observations in support of the sustainable development goals.* ESA Communication, special edition 2018 edition, 2018. URL: http://eohandbook.com/sdg/files/CEOS_EOHB_2018_SDG.pdf.

[58] H.-P. Plag and S.-A. Jules-Plag. A goal-based approach to the identification of essential transformation variables in support of the implementation of the 2030 agenda for sustainable development. *International Journal of Digital Earth*, 13(2):166–187, 2020. doi:10.1080/17538947.2018.1561761.

[59] J. Dong, G. Metternicht, P. Hostert, R. Fensholt, and R. R. Chowdhury. Remote sensing and geospatial technologies in support of a normative land system science: Status and prospects. *Current Opinion in Environmental Sustainability*, 38:44–52, 2019.

[60] T. Blaschke. *Collective Sensing: Fernerkundung, Sensorik in den Straßen, soziale Netzwerke und "die Crowd*, pages 267–269. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015. doi:10.1007/978-3-662-44841-0_31.

[61] I. E. R. Hernandez and W. Shi. A random forests classification method for urban land-use mapping integrating spatial metrics and texture analysis. *International Journal of Remote Sensing*, 39(4):1175–1198, 2 2018. doi:10.1080/01431161.2017.1395968.

[62] S. Srivastava, J. E. Vargas-Muñoz, and D. Tuia. Understanding urban landuse from the above and ground perspectives: A deep learning, multimodal solution. *Remote Sensing of Environment*, 228:129–143, 7 2019. doi:10.1016/j.rse.2019.04.014.

[63] Q. Yuan, H. Shen, T. Li, Z. Li, S. Li, Y. Jiang, H. Xu, W. Tan, Q. Yang, J. Wang, and et al. Deep learning in environmental remote sensing: Achievements and

challenges. *Remote Sensing of Environment*, 241:111716, 2020. doi:https://doi.org/10.1016/j.rse.2020.111716.

[64] B. V. Dasarathy. Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proceedings of the IEEE*, 85(1):24–38, 1 1997. doi:10.1109/5.554206.

[65] X. X. Zhu and R. Bamler. A sparse image fusion algorithm with application to pan-sharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 51(5):2827–2836, 2013. doi:10.1109/TGRS.2012.2213604.

[66] J. Wei, L. Wang, P. Liu, X. Chen, W. Li, and A. Y. Zomaya. Spatiotemporal fusion of modis and landsat-7 reflectance images via compressed sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 55(12):7126–7139, 2017. doi:10.1109/TGRS.2017.2742529.

[67] B. Rasti and P. Ghamisi. Remote sensing image classification using subspace sensor fusion. *Information Fusion*, 64:121–130, 2020. URL: https://www.sciencedirect.com/science/article/pii/S1566253520303146, doi:https://doi.org/10.1016/j.inffus.2020.07.002.

[68] C. Qiu, X. Tong, M. Schmitt, B. Bechtel, and X. X. Zhu. Multilevel feature fusion-based cnn for local climate zone classification from sentinel-2 images: Benchmark results on the so2sat lcz42 dataset. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:2793–2806, 2020. doi:10.1109/JSTARS.2020.2995711.

[69] N.-B. Chang and K. Bai. *Multisensor Data Fusion and Machine Learning for Environmental Remote Sensing*. CRC Press, 10 2017. doi:10.1201/9781315154602.

[70] J. Liu, T. Li, P. Xie, S. Du, F. Teng, and X. Yang. Urban big data fusion based on deep learning: An overview. *Information Fusion*, 53:123–133, 2020. doi:https://doi.org/10.1016/j.inffus.2019.06.016.

[71] M. Trepel. *Neuroanatomie. Struktur und Funktion*. Urban und Fischer, 5 edition, 2012.

[72] P. McLeod, K. Plunkett, and E. T. Rolls. *Introduction to Connectionist Modelling of Cognitive Processes*. Oxford University Press, New York, 1998.

[73] J. L. Elman. Finding Structure in Time. *Cognitive Science*, 14:179–211, 1990.

[74] D. E. Rumelhart, G. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(60886088):533–536, 10 1986. doi:10.1038/323533a0.

[75] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi:10.1109/5.726791.

[76] Y. LeCun, B. Boser, J. Denker, D. J. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. `doi:10.1162/neco.1989.1.4.541`.

[77] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[78] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2015.

[79] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[80] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2016.

[81] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. `doi:10.1007/s11263-015-0816-y`.

[82] A. Sharma, X. Liu, X. Yang, and D. Shi. A patch-based convolutional neural network for remote sensing image classification. *Neural Networks*, 95:19–28, 2017. `doi:https://doi.org/10.1016/j.neunet.2017.07.017`.

[83] Q. Zhang, Y. Wang, Q. Liu, X. Liu, and W. Wang. Cnn based suburban building detection using monocular high resolution google earth images. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 661–664, 7 2016.

[84] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[85] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 11 1997. `doi:10.1162/neco.1997.9.8.1735`.

[86] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[87] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, 10 2014. Association for Computational Linguistics. URL: `https://aclanthology.org/D14-1179`, `doi:10.3115/v1/D14-1179`.

[88] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103, 2014.

[89] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate, 2016. URL: `https://arxiv.org/abs/1409.0473`, `arXiv:1409.0473`.

[90] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6000–6010. Curran Associates Inc., 12 2017.

[91] J. Cheng, L. Dong, and M. Lapata. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561, 2016.

[92] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805 [cs]*, 10 2018. arXiv: 1810.04805. URL: `http://arxiv.org/abs/1810.04805`.

[93] S. D. Houston. *The First Writing: Script Invention as History and Process.* Cambridge University Press, 2004.

[94] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcouglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[95] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrvieval.* Cambridge University Press, 1 edition, 2008. URL: `http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf`.

[96] T. Y. Chong, R. Banchs, and E. S. Chng. An empirical evaluation of stop word removal in statistical machine translation. In *Proceedings of the Joint Workshop on Exploiting Synergies between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation (HyTra)*, pages 30–37. Association for Computational Linguistics, 4 2012. URL: `https://aclanthology.org/W12-0104`.

[97] H. Schütze. Dimensions of meaning. In *Supercomputing '92:Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, pages 787–796, 11 1992. `doi:10.1109/SUPERC.1992.236684`.

[98] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[99] T. Kiss and J. Strunk. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525, 2006. `doi:10.1162/coli.2006.32.4.485`.

[100] A. Taylor, M. Marcus, and B. Santorini. *The Penn Treebank: An Overview*, pages 5–22. Text, Speech and Language Technology. Springer Netherlands, 2003. URL: https://doi.org/10.1007/978-94-010-0201-1_1, doi:10.1007/978-94-010-0201-1_1.

[101] T. Kudo and J. Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71. Association for Computational Linguistics, 11 2018. URL: https://www.aclweb.org/anthology/D18-2012, doi:10.18653/v1/D18-2012.

[102] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, 6 2007. Association for Computational Linguistics. URL: https://aclanthology.org/P07-2045.

[103] J. Leskovec, A. Rajaraman, and J. D. Ullman. *Data Mining*, pages 1–18. Cambridge University Press, 2 edition, 2014. doi:10.1017/CBO9781139924801.002.

[104] K. Spärck Jones. A Statistical Interpretation of Term Specificity and its Application in Retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[105] J. R. Firth. A synopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis*, pages 1–32, 1957.

[106] H. Schütze. Word space. In *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, pages 895–902, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[107] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[108] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL: http://www.anthology.aclweb.org/D/D14/D14-1162.pdf.

[109] P. Bojanowski, E. Grave, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistic*, 5:135–146, 2017.

[110] T. Luong, R. Socher, and C. Manning. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on*

Computational Natural Language Learning, pages 104–113, Sofia, Bulgaria, 8 2013. Association for Computational Linguistics. URL: `https://aclanthology.org/W13-3512`.

[111] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL: `https://aclanthology.org/W03-0419`.

[112] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting Similarities among Languages for Machine Translation. *arXiv:1309.4168 [cs]*, 9 2013. arXiv: 1309.4168. URL: `http://arxiv.org/abs/1309.4168`.

[113] R. Tsarfaty, D. Seddah, Y. Goldberg, S. Kübler, M. Candito, J. Foster, Y. Versley, I. Rehbein, and L. Tounsi. Statistical Parsing of Morphologically Rich Languages (SPMRL). What, How and Wither. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 1–12, Los Angeles, CA, USA, 2010.

[114] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3483–3487, 2018. URL: `http://www.aclweb.org/anthology/L18-1550`.

[115] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[116] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, 7 2018. Association for Computational Linguistics. URL: `https://aclanthology.org/P18-1031`, `doi:10.18653/v1/P18-1031`.

[117] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. URL: `https://aclanthology.org/D16-1264`, `doi:10.18653/v1/D16-1264`.

[118] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. Glue: A multitask benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, 11 2018. Association for Computational Linguistics. URL: `https://aclanthology.org/W18-5446`, `doi:10.18653/v1/W18-5446`.

[119] Y. Peng, S. Yan, and Z. Lu. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the 2019 Workshop on Biomedical Natural Language Processing (BioNLP 2019)*, pages 58–65, 2019.

[120] S. Schweter and A. Akbik. Flert: Document-level features for named entity recognition, 2020. `arXiv:2011.06993`.

[121] P. Xia, S. Wu, and B. Van Durme. Which *bert? a survey organizing contextualized encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7516–7533, 2020.

[122] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[123] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online, 7 2020. Association for Computational Linguistics. URL: `https://aclanthology.org/2020.acl-main.195`, `doi:10.18653/v1/2020.acl-main.195`.

[124] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

[125] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[126] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism, 2020. `arXiv:1909.08053`.

[127] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.

[128] D. Marmanis, M. Datcu, T. Esch, and U. Stilla. Deep learning earth observation classification using imagenet pretrained networks. *IEEE Geoscience and Remote Sensing Letters*, 13(1):105–109, 1 2016. `doi:10.1109/LGRS.2015.2499239`.

[129] I. D. Stewart and T. R. Oke. Local climate zones for urban temperature studies. *Bulletin of the American Meteorological Society*, 93(12):1879–1900, 2012. URL: `https://journals.ametsoc.org/view/journals/bams/93/12/bams-d-11-00019.1.xml`, `doi:10.1175/BAMS-D-11-00019.1`.

[130] C. Qiu, L. Mou, M. Schmitt, and X. X. Zhu. Local climate zone-based urban land cover classification from multi-seasonal sentinel-2 images with a recurrent residual network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 154:151–162, 2019.

[131] C. Qiu, L. Mou, M. Schmitt, and X. X. Zhu. Fusing Multiseasonal Sentinel-2 Imagery for Urban Land Cover Classification With Multibranch Residual Convolutional Neural Networks. *IEEE Geoscience and Remote Sensing Letters*, 2020.

[132] Q. Hu, W. Wu, T. Xia, Q. Yu, P. Yang, Z. Li, and Q. Song. Exploring the Use of Google Earth Imagery and Object-Based Methods in Land Use/Cover Mapping. *Remote Sensing*, 5(11):6026–6042, 2013. `doi:10.3390/rs5116026`.

[133] A. Lin, X. Sun, H. Wu, W. Luo, D. Wang, D. Zhong, Z. Wang, L. Zhao, and J. Zhu. Identifying Urban Building Function by Integrating Remote Sensing Imagery and POI Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:8864–8875, 2021. `doi:10.1109/JSTARS.2021.3107543`.

[134] E. J. Hoffmann, M. Werner, and X. X. Zhu. Quality assessment of semantic tags in openstreetmap. *IOP Conference Series: Earth and Environmental Science*, 509:012025, 7 2020. `doi:10.1088/1755-1315/509/1/012025`.

[135] S. Lobry, J. Murray, D. Marcos, and D. Tuia. Visual question answering from remote sensing images. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 4951–4954, 7 2019. `doi:10.1109/IGARSS.2019.8898891`.

[136] L. Li, M. F. Goodchild, and B. Xu. Spatial, temporal, and socioeconomic patterns in the use of twitter and flickr. *Cartography and Geographic Information Science*, 40(2):61–77, 3 2013. `doi:10.1080/15230406.2013.777139`.

[137] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, 2017.

[138] A. Kruspe, M. Häberle, I. Kuhn, and X. X. Zhu. Cross-language sentiment analysis of European Twitter messages during the COVID-19 pandemic. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. Association for Computational Linguistics, 7 2020. URL: `https://aclanthology.org/2020.nlpcovid19-acl.14`.

[139] M. J. Kühn, D. Abele, T. Mitra, W. Koslow, A. Majid, K. Rack, M. Siggel, S. Khailaie, M. Klitz, S. Binder, L. Spataro, J. Gild, J. Kleinert, M. Häberle, L. Plötzke, C. D. Spinner, M. Stecher, X. X. Zhu, A. Basermann, and M. Meyer-Hermann. Assessment of effective mitigation and prediction of the spread of SARS-CoV-2 in Germany using demographic information and spatial resolution. *Mathematical Biosciences*, 339:108648, 9 2021. `doi:10.1016/j.mbs.2021.108648`.

[140] L. Mitchell, M. R. Frank, K. Decker Harris, P. Sheridan Dodds, and C. M. Danforth. The geography of happiness: Connecting twitter sentiment and expression, demographics, and objective characteristics of place. *PLoS ONE:e64417*, 8(5), 2013.

[141] W. He, S. Zha, and L. Li. Social media competitive analysis and text mining: A case study in the pizza industry. *International Journal of Information Management*, 33(3):464–472, 2013.

[142] F. Atefeh and W. Khreich. A Survey of Techniques for Event Detection in Twitter. *Computational Intelligence*, 31(1):132–164, 2015. `doi:10.1111/coin.12017`.

[143] D. Paul, F. Li, M. K. Teja, X. Yu, and R. Frost. Compass: Spatio temporal sentiment analysis of us election what twitter says! In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 1585–1594, New York, NY, USA, 2017. Association for Computing Machinery. `doi:10.1145/3097983.3098053`.

[144] Z. A. Hamstead, D. Fisher, R. T. Ilieva, S. A. Wood, T. McPhearson, and P. Kremer. Geolocated social media as a rapid indicator of park visitation and equitable park access. *Computers, Environment and Urban Systems*, 72:38–50, 11 2018. URL: `http://www.sciencedirect.com/science/article/pii/S0198971517303538`, `doi:10.1016/j.compenvurbsys.2018.01.007`.

[145] X. Chen, H. Vo, W. Yu, and F. Wang. A framework for annotating openstreetmap objects using geo-tagged tweets. *Geoinformatica*, (22):589–613, 2018.

[146] S. Sobolevsky, P. Kats, M. Sergey, M. Hoffman, B. Kettler, and C. Kontokosta. Twitter connections shaping new york city. 2018.

[147] J. Osorio-Arjona, J. Horak, R. Svoboda, and Y. García-Ruíz. Social media semantic perceptions on madrid metro system: Using twitter data to link complaints to space. *Sustainable Cities and Society*, 64:102530, 2021. `doi:https://doi.org/10.1016/j.scs.2020.102530`.

[148] F. Terroso-Saenz and A. Muñoz. Land use discovery based on Volunteer Geographic Information classification. *Expert Systems with Applications*, 140:112892, 2 2020. `doi:10.1016/j.eswa.2019.112892`.

[149] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4–5):993–1022, 2003.

[150] R. Huang, H. Taubenböck, L. Mou, and X. X. Zhu. Classification of Settlement Types from Tweets Using LDA and LSTM. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 6408–6411, July 2018. `doi:10.1109/IGARSS.2018.8519240`.

[151] E. S. Tellez, S. Miranda-Jiménez, M. Graff, D. Moctezuma, R. R. Suárez, and O. S. Siordia. A simple approach to multilingual polarity classification in twitter. *Pattern Recognition Letters*, 94:68–74, 7 2017. URL: `https://www.sciencedirect.com/science/article/pii/S0167865517301721`, `doi:10.1016/j.patrec.2017.05.024`.

[152] W. Ammar, G. Mulcaire, Y. Tsvetkov, G. Lample, C. Dyer, and N. A. Smith. Massively Multilingual Word Embeddings. *arXiv:1602.01925 [cs]*, February 2016. arXiv: 1602.01925. URL: `http://arxiv.org/abs/1602.01925`.

[153] R. Al-Rfou, B. Perozzi, and S. Skiena. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, 2013. URL: `http://www.aclweb.org/anthology/W13-3520`.

[154] M. Artetxe and H. Schwenk. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *arXiv:1812.10464 [cs]*, 12 2018. arXiv: 1812.10464. URL: `http://arxiv.org/abs/1812.10464`.

[155] V. Malykh, T. Khakhulin, and V. Logacheva. Robust word vectors: Context-informed embeddings for noisy texts. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, page 10, 2018. URL: `http://www.aclweb.org/anthology/W18-61#page=70`.

[156] X. Yang, R. McCreadie, C. Macdonald, and I. Ounis. Transfer Learning for Multi-language Twitter Election Classification. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ASONAM '17, pages 341–348, New York, NY, USA, 2017. ACM. URL: `http://doi.acm.org/10.1145/3110025.3110059`, `doi:10.1145/3110025.3110059`.

[157] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. Hernandez Abrego, S. Yuan, C. Tar, Y.-h. Sung, B. Strope, and R. Kurzweil. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online, 7 2020. Association for Computational Linguistics. URL: `https://aclanthology.org/2020.acl-demos.12`, `doi:10.18653/v1/2020.acl-demos.12`.

[158] B. Huang, B. Zhao, and Y. Song. Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery. *Remote Sensing of Environment*, 214:73–86, 2018. URL: `http://www.sciencedirect.com/science/article/pii/S0034425718302074`, `doi:https://doi.org/10.1016/j.rse.2018.04.050`.

[159] T. Hu, J. Yang, X. Li, and P. Gong. Mapping urban land use by using landsat images and open social data. *Remote Sensing*, 8(151), 2016. `doi:10.3390/rs8020151`.

[160] L. F. F. G. de Assis, B. Herfort, E. Steiger, F. E. A. Horita, and J. Porto de Albuquerque. A geographic approach for on-the-fly prioritization of social-media messages towards improving flood risk management. In *Proceedings of the 4th Brazilian Workshop Social Network Analysis and Mining (BraSNAM)*, pages 1–12, 2015.

[161] H. Wang, E. Skau, H. Krim, and G. Cervone. Fusing Heterogeneous Data: A Case for Remote Sensing and Social Media. *IEEE Transactions on Geoscience and Remote Sensing*, 56(12):6956–6968, 12 2018. doi:10.1109/TGRS.2018.2846199.

[162] G. Cervone, E. Sava, Q. Huang, E. Schnebele, J. Harrison, and N. Waters. Using Twitter for tasking remote-sensing data collection and damage assessment: 2013 Boulder flood case study. *International Journal of Remote Sensing*, 37(1):100–124, 1 2016. doi:10.1080/01431161.2015.1117684.

[163] H. Taubenböck, J. Staab, X. X. Zhu, C. Geiß, S. Dech, and M. Wurm. Are the Poor Digitally Left Behind? Indications of Urban Divides Based on Remote Sensing and Twitter Data. *ISPRS International Journal of Geo-Information*, 7(8):304, 8 2018. URL: https://www.mdpi.com/2220-9964/7/8/304, doi:10.3390/ijgi7080304.

[164] L. Kondmann, M. Häberle, and X. X. Zhu. Combining Twitter and Earth Observation Data for Local Poverty Mapping. In *NeuRIPS Machine Learning for the Developing World Workshop*, pages 1–5, 2020. URL: https://drive.google.com/file/d/12W7p4TBAlUV57EN-Iv4QofOdCNnr3tmp.

[165] Y. Zhang, Q. Li, H. Huang, W. Wu, X. Du, and H. Wang. The Combined Use of Remote Sensing and Social Sensing Data in Fine-Grained Urban Land Use Mapping: A Case Study in Beijing, China. *Remote Sensing*, 9(99):865, 9 2017. doi:10.3390/rs9090865.

[166] C. Fu, X.-P. Song, and K. Stewart. Integrating activity-based geographic information and long-term remote sensing to characterize urban land use change. *Remote Sensing*, 11(2424):2965, 1 2019. doi:10.3390/rs11242965.

[167] A. Leichter, D. Wittich, F. Rottensteiner, M. Werner, and M. Sester. Improved classification of satellite imagery using spatial feature maps extracted from social media. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XLII–4, pages 335–342. Copernicus GmbH, 9 2018. URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-4/335/2018/, doi:10.5194/isprs-archives-XLII-4-335-2018.

[168] G. Cheng, J. Han, L. Guo, Z. Liu, S. Bu, and J. Ren. Effective and Efficient Midlevel Visual Elements-Oriented Land-Use Classification Using VHR Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, 53(8):4238–4249, 2015. doi:10.1109/TGRS.2015.2393857.

[169] Y. R. Filion. Impact of Urban Form on Energy Use in Water Distribution Systems. *Journal of Infrastructure Systems*, 14(4):337–346, 2008.

[170] R. Ewing and F. Rong. The impact of urban form on U.S. residential energy use. *Housing Policy Debate*, 19(1):1–30, 2008. doi:10.1080/10511482.2008.9521624.

[171] Twitter. Sampled Stream. visit: 14.6.2021. URL: `https://developer.twitter.com/en/docs/twitter-api/tweets/sampled-stream/introduction`.

[172] Twitter. Tweet: Announcement to deactivate precise geo-location tagging in tweets, 6 2019. visit: 14.6.2021. URL: `https://twitter.com/twittersupport/status/1141039841993355264`.

[173] Twitter. Tweet: Supplement to announcement to deactivate precise geo-location tagging in tweets, 2019. visit: 14.6.2021. URL: `https://twitter.com/twittersupport/status/1142130343715078144`.

[174] Twitter. How to add your location to a Tweet. visit: 14.6.2021. URL: `https://help.twitter.com/en/using-twitter/tweet-location`.

[175] Y. Hu and R.-Q. Wang. Understanding the removal of precise geotagging in tweets. *Nature Human Behaviour*, 4(1212):1219–1221, 12 2020. `doi:10.1038/s41562-020-00949-x`.

[176] M. Owusu, M. Kuffer, M. Belgiu, T. Grippa, M. Lennert, S. Georganos, and S. Vanhuysse. Towards user-driven earth observation-based slum mapping. *Computers, Environment and Urban Systems*, 89:101681, 9 2021. `doi:10.1016/j.compenvurbsys.2021.101681`.

[177] X. X. Zhu, J. Hu, C. Qiu, Y. Shi, J. Kang, L. Mou, H. Bagheri, M. Häberle, Y. Hua, R. Huang, L. Hughes, H. Li, Y. Sun, G. Zhang, S. Han, M. Schmitt, and Y. Wang. So2Sat LCZ42: A Benchmark Data Set for the Classification of Global Local Climate Zones [Software and Data Sets]. *IEEE Geoscience and Remote Sensing Magazine*, 8(3):76–89, 9 2020. `doi:10.1109/MGRS.2020.2964708`.

[178] B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos. Class imbalance, redux. In *2011 IEEE 11th International Conference on Data Mining*, pages 754–763, 12 2011. `doi:10.1109/ICDM.2011.33`.

[179] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, 6(4):15:1–15:21, 12 2012. `doi:10.1145/2382577.2382579`.

[180] G. Deutscher. *Die Evolution der Sprache. Wie die Menschheit zu ihrer größten Erfindung kam.* C.H.Beck, 2008.

[181] F. Steuber, M. Schönfeld, and G. D. Rodosek. Topic modeling of short texts using anchor words. In *Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics*, WIMS 2020, pages 210–219. Association for Computing Machinery, 6 2020. URL: `https://doi.org/10.1145/3405962.3405968`, `doi:10.1145/3405962.3405968`.

[182] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017. `arXiv:1412.6980`.

[183] H. Parveen and S. Pandey. Sentiment analysis on twitter data-set using naive bayes algorithm. In *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pages 416–419, 2016. doi:10.1109/ICATCCT.2016.7912034.

[184] C. Tseng, N. Patel, H. Paranjape, T. Y. Lin, and S. Teoh. Classifying twitter data with naïve bayes classifier. In *2012 IEEE International Conference on Granular Computing*, pages 294–299, 2012. doi:10.1109/GrC.2012.6468706.

[185] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[186] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. URL: https://www.tensorflow.org/.

[187] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[188] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni. Green AI. *Commun. ACM*, 63(12):54–63, nov 2020. URL: https://doi.org/10.1145/3381831, doi:10.1145/3381831.

[189] C. Fiesler and N. Proferes. "participant" perceptions of twitter research ethics. *Social Media + Society*, 4(1):2056305118763366, 2018. URL: https://doi.org/10.1177/2056305118763366, arXiv:https://doi.org/10.1177/2056305118763366, doi:10.1177/2056305118763366.

[190] P. Siemund, I. Gogolin, M. E. Schulz, and J. Davydova, editors. *Multilingualism and Language Diversity in Urban Areas. Acquisition, identities, space, education.* John Benjamins, 2013. URL: https://doi.org/10.1075/hsld.1.