ENRICHING AND VALIDATING GEOGRAPHIC INFORMATION ON THE WEB

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades

DOKTOR DER NATURWISSENSCHAFTEN

Dr. rer. nat.

genehmigte Dissertation
von

Herrn M. Sc. Nicolas Tempelmeier

geboren am 21. Juni 1991 in Hannover

2022

Referent: Prof. Dr. techn. Wolfgang Nejdl
Korreferent: Prof. Dr. Elena Demivdova
Korreferent: Prof. Dr. Ziawasch Abedjan
Tag der Promotion: 14. Juni 2022

ABSTRACT

The continuous growth of available data on the World Wide Web has led to an unprecedented amount of available information. However, the enormous variance in data quality and trustworthiness of information sources impairs the great potential of the large amount of vacant information. This observation especially applies to *geographic information on the Web*, i.e., information describing entities that are located on the Earth's surface. With the advent of mobile devices, the impact of geographic Web information on our everyday life has substantially grown. The mobile devices have also enabled the creation of novel data sources such as *OpenStreetMap* (OSM), a collaborative crowd-sourced map providing open cartographic information. Today, we use geographic information in many applications, including routing, location recommendation, or geographic question answering.

The processing of geographic Web information yields unique challenges. First, the descriptions of geographic entities on the Web are typically not validated. Since not all Web information sources are trustworthy, the correctness of some geographic Web entities is questionable. Second, geographic information sources on the Web are typically isolated from each other. The missing integration of information sources hinders the efficient use of geographic Web information for many applications. Third, the description of geographic entities is typically incomplete. Depending on the application, missing information is a decisive criterion for (not) using a particular data source.

Due to the large scale of the Web, the manual correction of these problems is usually not feasible such that automated approaches are required. In this thesis, we tackle these challenges from three different angles. (i) *Validation of geographic Web information:* We validate geographic Web information by detecting vandalism in OpenStreetMap, for instance, the replacement of a street name with advertisement. To this end, we present the *OVID* model for automated vandalism detection in OpenStreetMap. (ii) *Enrichment of geographic Web information through integration:* We integrate OpenStreetMap with other geographic Web information sources, namely knowledge graphs, by identifying entries corresponding to the same world real-world entities in both data sources. We present the *OSM2KG* model for automated identity link discovery between OSM and knowledge graphs. (iii) *Enrichment of missing information in geographic Web information:* We consider semantic annotations of geographic entities on Web pages as an additional data source. We exploit existing annotations of categorical properties of Web entities as training data to enrich missing categorical properties in geographic Web information. For all of the proposed models, we conduct extensive evaluations on real-world datasets. Our experimental results confirm that the proposed solutions reliably outperform existing baselines.

Furthermore, we demonstrate the utility of geographic Web Information in two application scenarios. (i) *Corpus of geographic entity embeddings:* We introduce the *GeoVectors* corpus, a linked open dataset of ready-to-use embeddings of geographic entities. With GeoVectors, we substantially lower the burden to use geographic data in machine learning applications. (ii) *Application to event impact prediction:* We employ several geographic Web information sources to predict the impact of public events on road traffic. To this end, we use cartographic, event, and event venue information from the Web.

Keywords: *geographic data, Web data, Linked Open Data, spatio-temporal machine learning*

ZUSAMMENFASSUNG

Durch die kontinuierliche Zunahme verfügbarer Daten im World Wide Web, besteht heute eine noch nie da gewesene Menge verfügbarer Informationen. Das große Potential dieser Daten wird jedoch durch hohe Schwankungen in der Datenqualität und in der Vertrauenswürdigkeit der Datenquellen geschmälert. Dies kann vor allem am Beispiel von *geografischen Web-Informationen* beobachtet werden. Geografische Web-Informationen sind Informationen über Entitäten, die über Koordinaten auf der Erdoberfläche verfügen. Die Relevanz von geografischen Web-Informationen für den Alltag ist durch die Verbreitung von internetfähigen, mobilen Endgeräten, zum Beispiel Smartphones, extrem gestiegen. Weiterhin hat die Verfügbarkeit der mobilen Endgeräte auch zur Erstellung neuartiger Datenquellen wie *OpenStreetMap* (OSM) geführt. OSM ist eine offene, kollaborative Webkarte, die von Freiwilligen dezentral erstellt wird. Mittlerweile ist die Nutzung geografischer Informationen die Grundlage für eine Vielzahl von Anwendungen, wie zum Beispiel Navigation, Reiseempfehlungen oder geografische Frage-Antwort-Systeme.

Bei der Verarbeitung geografischer Web-Informationen müssen einzigartige Herausforderungen berücksichtigt werden. Erstens werden die Beschreibungen geografischer Web-Entitäten typischerweise nicht validiert. Da nicht alle Informationsquellen im Web vertrauenswürdig sind, ist die Korrektheit der Darstellung mancher Web-Entitäten fragwürdig. Zweitens sind Informationsquellen im Web oft voneinander isoliert. Die fehlende Integration von Informationsquellen erschwert die effektive Nutzung von geografischen Web-Information in vielen Anwendungsfällen. Drittens sind die Beschreibungen von geografischen Entitäten typischerweise unvollständig. Je nach Anwendung kann das Fehlen von bestimmten Informationen ein entscheidendes Kriterium für die Nutzung einer Datenquelle sein.

Da die Größe des Webs eine manuelle Behebung dieser Probleme nicht zulässt, sind automatisierte Verfahren notwendig. In dieser Arbeit nähern wir uns diesen Herausforderungen von drei verschiedenen Richtungen. (i) *Validierung von geografischen Web-Informationen*: Wir validieren geografische Web-Informationen, indem wir Vandalismus in OpenStreetMap identifizieren, zum Beispiel das Ersetzen von Straßennamen mit Werbetexten. (ii) *Anreicherung von geografischen Web-Information durch Integration*: Wir integrieren OpenStreetMap mit anderen Informationsquellen im Web (Wissensgraphen), indem wir Einträge in beiden Informationsquellen identifizieren, die den gleichen Echtwelt-Entitäten entsprechen. (iii) *Anreicherung von fehlenden geografischen Informationen*: Wir nutzen semantische Annotationen von geografischen Entitäten auf Webseiten als weitere Datenquelle. Wir nutzen existierende Annotationen kategorischer Attribute von Web-Entitäten als Trainingsdaten, um fehlende kategorische Attribute in geografischen Web-Informationen zu ergänzen. Wir führen ausführliche Evaluationen für alle beschriebenen Modelle durch. Die vorgestellten Lösungsansätze erzielen verlässlich bessere Ergebnisse als existierende Ansätze.

Weiterhin demonstrieren wir den Nutzen von geografischen Web-Informationen in zwei Anwendungsszenarien. (i) *Korpus mit Embeddings von geografischen Entitäten*: Wir stellen den *GeoVectors-Korpus* vor, einen verlinkten, offenen Datensatz mit direkt nutzbaren Embeddings von geografischen Web-Entitäten. Der GeoVectors-Korpus erleichtert die Nutzung von geografischen Daten in Anwendungen von maschinellen Lernen erheblich. (ii) *Anwendung zur Prognose von Veranstaltungsauswirkungen*: Wir nutzen Karten-, Veranstaltungs- und Veranstaltungsstätten-Daten aus dem Web, um die Auswirkungen von Veranstaltungen auf den Straßenverkehr zu prognostizieren.

Schlagwörter: *Geografische Daten, Web Daten, Linked Open Data, räumlich-zeitliches maschinelles Lernen*

ACKNOWLEDGMENTS

During my PhD studies, I have met and worked with many great researchers who supported me throughout my studies om various ways. I want to thank Prof. Wolfgang Nejdl for supervising my PhD thesis and for welcoming me to the L3S Research Center. I am very grateful for the committed mentorship and guidance of Prof. Elena Demidova, which involved countless detailed discussions and many hours of writing papers together. I thank Prof. Ziawasch Abedjan for his time and effort in examining my thesis. I would also like to thank Prof. Kurt Schneider for being part of the doctoral committee.

Further, I would like to express my warm thanks to the colleagues I have worked with during the past years, including projects and many joint research efforts. I thank my office mates Simon and Markus for a relaxed and friendly working atmosphere. I also want to thank our research group members, including but not limited to, Alishiba, Ashutosh, Rajjat, Ran, Stefan, and Thorben, for the friendly and productive joint work. I would like to thank Dimitar, Miroslav, and the others from the administrative and technical areas for their support.

I also want to thank my parents and my siblings for their lasting support not only throughout my academic career. Finally, I am deeply grateful for my wife Wiebke and her unconditional support, caring, and understanding even in the busiest times close to paper deadlines.

The works presented in the chapters of this theses were partially funded by the German Research Foundation (DFG) ("WorldKG", 424985896), the Federal Ministry of Education and Research (BMBF), Germany ("SimpleML", 01IS18054), ("Data4UrbanMobility", 02K15A040), the Federal Ministry for Economic Affairs and Climate Action (BMWK), Germany ("d-E-mand", 01ME19009B), ("CampaNeo", 01MD19007B), and the European Commission (EU H2020, "smashHit", 871477), ("AFEL", 687916).

FOREWORD

The contributions presented in this thesis have been published or are under review at the following conferences and journals:

In Chapter 3, we present a method for the validation of OpenStreetMap data by vandalism detection.

- Nicolas Tempelmeier and Elena Demidova. OVID: A Machine Learning Approach for Automated Vandalism Detection in OpenStreetMap. In *Proceedings of the 29th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'21, 4 pages, 2021. (poster) [TD21b]

- Nicolas Tempelmeier and Elena Demidova. Attention-Based Vandalism Detection in OpenStreetMap. (under submission)

In Chapter 4, we present an approach for enriching OpenStreetMap with identitiy links to Knowledge Graphs.

- Nicolas Tempelmeier and Elena Demidova. Linking OpenStreetMap with Knowledge Graphs - Link Discovery for Schema-Agnostic Volunteered Geographic Information. *Future Generation Computer Systems*, 116:349–364, 2021. (journal article) [TD21a]

In Chapter 5, we describe an approach for inferring missing information in Web markup.

- Nicolas Tempelmeier, Elena Demidova, and Stefan Dietze. Inferring Missing Categorical Information in Noisy and Sparse Web Markup. In *Proceedings of the Web Conference*, WWW'18, pages 1297–1306, 2018. (full paper) [TDD18]

In Chapter 6, we introduce the GeoVectors dataset that provides embeddings of geographic entities.

- Nicolas Tempelmeier, Simon Gottschalk, and Elena Demidova. GeoVectors: a Linked Open Corpus of OpenStreetMap Embeddings on World Scale. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, CIKM'21, 9 pages, 2021. (resource paper) [TGD21]

In Chapter 7, we use the geographic data for the application of event impact prediction.

- Nicolas Tempelmeier, Stefan Dietze, and Elena Demidova. Crosstown traffic - supervised prediction of impact of planned special events on urban traffic. *GeoInformatica*, 24(2):339–370, 2020. (journal article) [TDD20]
- Nicolas Tempelmeier, Anzumana Sander, Udo Feuerhake, Martin Löhdefink, and Elena Demidova. TA-Dash: An Interactive Dashboard for Spatial-Temporal Traffic Analytics. In *Proceedings of the 28th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'20, pages 409–412, 2020. (demonstration) [TSF+20]

The complete list of publications during my PhD includes the following publications:

Journal articles

- Nicolas Tempelmeier, Stefan Dietze, and Elena Demidova. Crosstown traffic - supervised prediction of impact of planned special events on urban traffic. *GeoInformatica*, 24(2):339–370, 2020. [TDD20]
- Nicolas Tempelmeier and Elena Demidova. Linking OpenStreetMap with Knowledge Graphs - Link Discovery for Schema-Agnostic Volunteered Geographic Information. *Future Generation Computer Systems*, 116:349–364, 2021. [TD21a]
- Nicolas Tempelmeier, Udo Feuerhake, Oskar Wage, and Elena Demidova. Mining Topological Dependencies of Recurrent Congestion in Road Networks. *ISPRS International Journal of Geo-Information*, 10(4):248, 2021. [TFWD21]

Conference papers

- Nicolas Tempelmeier, Elena Demidova, and Stefan Dietze. Inferring Missing Categorical Information in Noisy and Sparse Web Markup. In *Proceedings of the Web Conference*, WWW'18, pages 1297–1306, 2018. (full paper) [TDD18]
- Alishiba Dsouza, Nicolas Tempelmeier, and Elena Demidova. Towards Neural Schema Alignment for OpenStreetMap and Knowledge Graphs. In *Proceedings of the 20th International Semantic Web Conference*, ISWC'21, pages 56–73, 2021. (full paper) [DTD21]

- Ashutosh Sao, Nicolas Tempelmeier, and Elena Demidova. Deep Information Fusion for Electric Vehicle Charging Station Occupancy Forecasting. In *IEEE 24th International Conference on Intelligent Transportation Systems*, ITSC'21, 6 pages, 2021. (full paper) [STD21]

- Nicolas Tempelmeier, Simon Gottschalk, and Elena Demidova. GeoVectors: a Linked Open Corpus of OpenStreetMap Embeddings on World Scale. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, CIKM'21, 9 pages, 2021. (resource paper) [TGD21]

- Alishiba Dsouza, Nicolas Tempelmeier, Ran Yu, Simon Gottschalk, and Elena Demidova. WorldKG: A World-Scale Geographic Knowledge Graph. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, CIKM'21, 10 pages, 2021. (resource paper) [DTY+21]

- Simon Gottschalk, Nicolas Tempelmeier, Günter Kniesel, Vasileios Iosifidis, Besnik Fetahu, and Elena Demidova. Simple-ML: Towards a Framework for Semantic Data Analytics Workflows. In *International Conference on Semantic Systems*, SEMANTiCS'19, pages 359–366, 2019. (short paper) [GTK+19]

Poster and demonstration papers

- Nicolas Tempelmeier, Udo Feuerhake, Oskar Wage, and Elena Demidova. ST-Discovery: Data-Driven Discovery of Structural Dependencies in Urban Road Networks. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'19, pages 488–491, 2019. (poster) [TFWD19]

- Nicolas Tempelmeier and Elena Demidova. OVID: A Machine Learning Approach for Automated Vandalism Detection in OpenStreetMap. In *Proceedings of the 29th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'21, 4 pages, 2021. (poster) [TD21b]

- Nicolas Tempelmeier, Anzumana Sander, Udo Feuerhake, Martin Löhdefink, and Elena Demidova. TA-Dash: An Interactive Dashboard for Spatial-Temporal Traffic Analytics. In *Proceedings of the 28th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'20, pages 409–412, 2020. (demonstration) [TSF+20]

Workshop papers

- Nicolas Tempelmeier, Yannick Rietz, Iryna V. Lishchuk, Tina Kruegel, Olaf Mumm, Vanessa Miriam Carlow, Stefan Dietze, and Elena Demidova. Data4UrbanMobility: Towards Holistic Data Analytics for Mobility Applications in Urban Regions. In *Companion of The 2019 World Wide Web Conference*, WWW'19, pages 137–145, 2019. [TRL+19]

Under submission

- Nicolas Tempelmeier and Elena Demidova. Attention-Based Vandalism Detection in OpenStreetMap

List of Figures

List of Tables

Contents

1 Introduction

Geographic information has always played a vital role for human civilizations. The quality of the geographic information, e.g., in the form of maps, provides insights into the advances of historical societies. Early examples of geographic information, e.g., the *Babylonian Map of the World* are dating back to the 6th century BC [RT10]. Today, the availability of geographic information has positive effects on many aspects of our everyday life. For instance, we may use geographic information for route planning and travel time estimation for commuting to work or prepare for bad weather by considering the weather forecast for our residential area.

The advent of the *World Wide Web* together with the development of mobile devices such as smartphones has further increased the impact of geographic information on our lives and paved the way for novel applications. For example, location-based social networks such as Foursquare[1] allow sharing the own position by indicating visits of point of interests. The image hosting service Flickr[2] includes optional geographic location information on images. In this sense, the World Wide Web has emerged as universal middleware for many location-based applications.

Throughout the last decades, several sources of geographic information on the Web evolved. First, so-called *Volunteered Geographic Information* (VGI) is a type of user-generated content that was enabled by the development of Web 2.0 technologies [Goo07]. In VGI, volunteers collect, describe and publish geographic information. The most prominent example of VGI is the OpenStreetMap[3] (OSM) project. OSM was funded in 2004 to address the lack of free cartographic information of the United Kingdom [Ope21a]. OSM relies on data contributed by volunteers (so-called *mappers*) that manually specify geographic information about, e.g., their living regions. Today, OSM is a rich source of publicly available geographic Web information used in countless applications such as routing algorithms or Web map services. Besides

[1] https://foursquare.com/
[2] https://www.flickr.com/
[3] https://www.openstreetmap.org/

OSM, other examples of VGI include GPS recordings of animal movements[4], textual knowledge bases describing geographic entities such as Wikipedia[5], or more structured knowledge bases such as knowledge graphs, e.g., Wikidata[6].

Second, websites providing geographically referenced content constitute another source of geographic Web information. Exemplary geographic content includes travel and restaurant recommendations or announcements of scheduled public events like concerts and fairs. Often, these websites highlight geographic information in a machine-readable way using so-called *semantic markup*. The machine-readable information enables geographic queries in Web search engines, such as '*Find concerts near me tomorrow*", to retrieve relevant Websites and ultimately increases the visibility of the Websites. At the same time, (machine-readable) geographic website contents in principle offer a high potential value for many more geographic data-driven applications.

Whereas geographic Web information is a highly relevant data source for location-based applications, unique challenges arise from the data acquisition paradigms of VGI and from collecting geographic website content. First, there are rarely guarantees for the correctness of the information. Especially in VGI, contributions can often be anonymous, lowering the hurdle for intentionally and unintentionally spread of wrong information [Bal14]. Second, different sources of geographic information on the Web are rarely integrated, hindering the efficient use of the information in downstream applications [TD21a]. Third, the descriptions of geographic entities on the Web are often incomplete, such that common individual entity properties are regularly not available [YFGD16]. Fourth, the sheer amount of available geographic information on the Web obstructs the manual correction of the abovementioned problems. Therefore, the scale of the Web requires automated approaches to address these challenges [BEM+13].

While historical pursuits of collecting geographic information may have focussed on filling blank spots on maps, the modern challenges require intelligent algorithms to provide high-quality geographic information. In this thesis, we develop several approaches to enrich and validate geographic information on the Web to fill selected information technology age "blank spots" in geographic data availability.

1.1 Research Questions

Despite the growing importance and availability of geographic information on the Web, the data quality heavily varies with respect to geographic regions and data sources [BNZ14, MRP16]. This varying data quality can be exemplary observed in OpenStreetMap. In June 2021, the size of data available for the country of Germany summed up to 3.3 GB, while only 2.5 GB of data was available for the entire

[4] https://www.movebank.org/cms/movebank-main
[5] https://www.wikipedia.org/
[6] https://www.wikidata.org/

South American continent. While the variance in data quality may hinder the use of geographic Web Information, the available data still has a high potential value to serve as training data for machine learning models. This thesis investigates how to enrich and validate several aspects of geographic Web information by exploiting the available data in supervised machine learning models.

In the first step, we aim to validate existing geographic Web information in OpenStreetMap. Data correctness is a vital prerequisite for the majority of data-driven applications. To this end, we aim to remove wrong and potentially harmful information from OSM. In particular, we aim to detect vandalism within OSM, leading to the first research question.

> **RQ1:** How to create a machine learning model to detect vandalism in volunteered geographic information sources on the Web, such as OpenStreetMap?

Vandalism detection in OpenStreetMap is critical and remarkably challenging due to the large scale of the dataset, the sheer number of contributors, various vandalism forms, and the lack of annotated data to train machine learning algorithms.

After validating the existing information, we aim to enrich geographic Web Information by adding new information. In principle, many modern applications require the combination of different datasets. For instance, a routing application requires the combination of road network data and current traffic data to facilitate travel time predictions. Therefore, the usefulness of a dataset is often limited by its capability to be integrated with other datasets. Tim Berners-Lee, the inventor of the World Wide Web, formulated the five-star open data deployment scheme from this observation [BL06]. In this scheme, a dataset can only reach the highest data quality level if it is integrated with other datasets. This integration is typically achieved with so-called *links* modeling relations of entities across different datasets.

While OpenStreetMap provides rich cartographic information, it lacks non-cartographic properties, e.g., the population evolution of a city. Contrarily, popular knowledge graphs such as Wikidata and DBpedia are potentially rich sources of such information but lack comprehensive cartographic information. Therefore, OpenStreetMap and knowledge graphs have a high potential to complement each other. However, links between OSM and knowledge graphs are still rare, hindering the effective combination of these data sources. The need for enrichment with additional links leads to the following research question.

> **RQ2.1:** How to discover identity links between OpenStreetMap nodes and knowledge graph entities?

The problem of link discovery is particularly challenging due to the lack of a strict schema and heterogeneity of the user-defined node representations in OSM. The heterogeneity of the OSM node descriptions includes a varying level of detail as well as inconsistent annotations. This heterogeneity imposes major challenges in capturing OSM nodes semantics in feature representations for machine learning algorithms.

From this observation, we formulate the following research question:

> **RQ2.2:** How to capture the semantics of OpenStreetMap objects despite varying data quality and the lack of a consistent schema?

So far, we have considered OpenStreetMap capturing cartographic information and popular knowledge graphs capturing encyclopedic information as data sources. While both sources provide a detailed description of geo-entities, they lack current or dynamic information, e.g., about public events that are going to take place. Therefore, we approach semantic Web markup as a third source of geographic Web information. Semantic Web markup provides machine-readable descriptions of entities occurring on Web pages, for instance, when and where public events are taking place. Throughout the past years, Semantic Web markup has seen widespread adoption since markup typically increases the visibility of Web pages for search engines. The increased visibility constitutes a substantial incentive for website authors to provide markup annotations. However, due to the diversity of websites, semantic Web markup is typically not used consistently, resulting in sparse entity annotation. This sparsity raises the need for an enrichment approach to add missing information. We consider this problem in our last research question:

> **RQ3:** How to infer missing categorical attributes of geographic Web entities?

As a first step towards enriching semantic Web markup data, we focus on categorical information. Challenges arise from the noisy, sparse, and inconsistent usage of Web markup introduced by the immense diversity of Web pages.

1.2 Contributions

To address the research questions formulated in the previous section, we present several contributions in this thesis. Figure 1.1 presents an overview of the individual contributions and their mutual relations. We exploit several sources of Geographic Web Information to make contributions in the areas of *validation of geographic Web information*, *enrichment of geographic Web information*, and for selected *applications*.

Validation of Geographic Web Information

This thesis introduces a validation model that identifies vandalism in OpenStreetMap, which is currently one of the most prominent sources of volunteered geographic information.

- We address **RQ1** in Chapter 3. We propose the *Ovid* (OpenStreetMap Vandalism Detection) model, a novel machine learning approach for vandalism detection in OpenStreetMap. Furthermore, we extract a dataset of real-world vandalism incidents from OpenStreetMap's edit history for the first time and provide

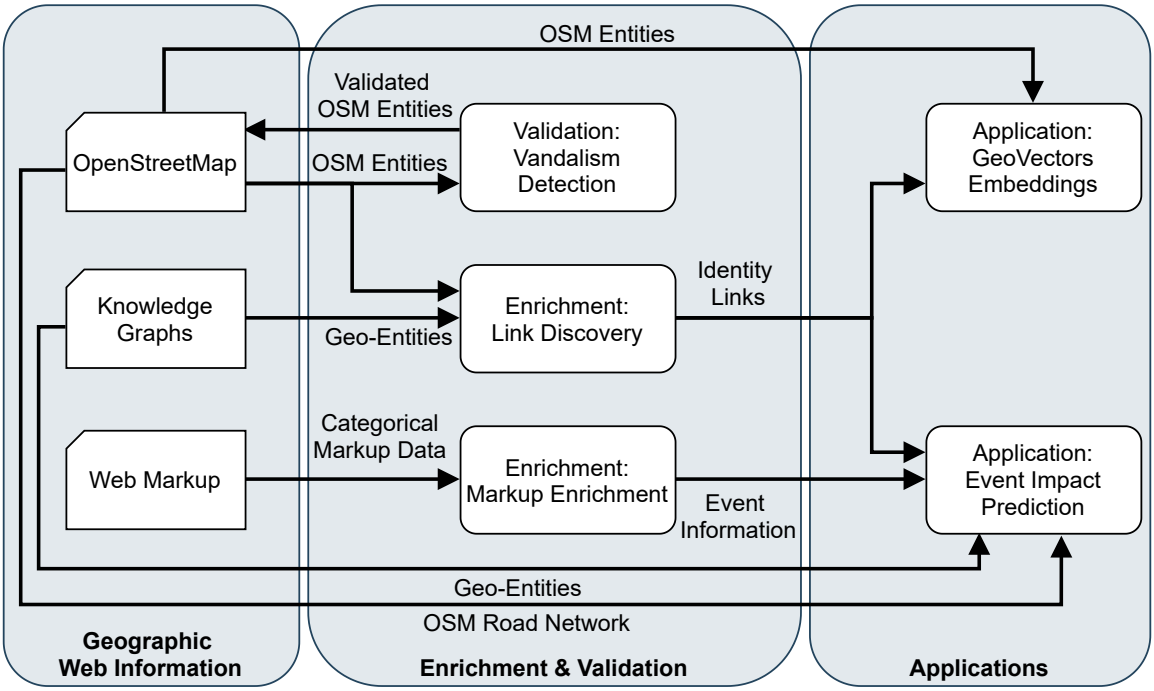


Figure 1.1. Contribution summary of this thesis. We use geographic Web information in the forms OpenStreetMap, knowledge graphs, and Web markup. We propose an approach to validate OpenStreetMap data through vandalism detection. We propose two approaches to enrich geographic Web information, i.e., we describe an algorithm for link discovery between OSM and knowledge graphs and one method to enrich categorical Web markup data. We describe two applications that benefit from the improved geographic knowledge, namely the construction of the GeoVectors corpus that provides usable embeddings of geographic entities and the prediction of event impact on road traffic.

this dataset as open data. Our evaluation results on real-world vandalism data demonstrate that the proposed Ovid method outperforms the baselines by eight percentage points regarding the F1 score on average.

Enrichment of Geographic Web Information

We tackle the challenge of enriching geographic Web Information from two directions. First, we enrich OpenStreetMap by discovering identity links between OpenStreetMap and knowledge graphs such as Wikidata and DBpedia in Chapter 4.

- We address **RQ2.1** by introducing the *OSM2KG* model - a novel link discovery approach to predict identity links between OSM nodes and geographic entities in a knowledge graph. OSM2KG combines a supervised machine learning

model with a geographic candidate generation step to determine identity links effectively. Our experiments, conducted on several OSM datasets as well as the Wikidata and DBpedia knowledge graphs, demonstrate that OSM2KG can reliably discover identity links. OSM2KG achieves an F1 score of 92.05% on Wikidata and of 94.17% on DBpedia on average.

- We address **RQ2.2** by proposing a novel latent, compact representation of OSM nodes that captures semantic node similarity in an embedding. OSM2KG adopts this latent representation to train a supervised model for link prediction.

Second, in Chapter 5, we enrich semantic Web markup by inferring missing categorical information, for instance, the types of events.

- We address **RQ3** by presenting a supervised classification model that uses Web-specific features such as domain information to augment missing categorical information. We demonstrate superior performance compared to both naive baselines and specialized state-of-the-art methods for type inference and achieve F1 scores of 79% and 83% in two experimental tasks.

Applications

We illustrate the use of geographic Web information with the two applications of geographic embedding generation and event impact prediction.

- In Chapter 6 we present GeoVectors, a unique, world-scale corpus containing ready-to-use embeddings of over 980 million geographic entities in 180 countries. We create a semantic description of the GeoVectors corpus, including identity links to the Wikidata and DBpedia knowledge graphs, and provide a SPARQL endpoint as a semantic interface.

- In Chapter 7 we introduce a novel metric to measure the spatial and temporal impact of special public events on road traffic. Then, we propose a supervised regression model that exploits event information, OSM road network information, and additional geo-entity information to predict spatial and temporal event impact.

1.3 Thesis Outline

The remainder of this thesis is structured as follows: In Chapter 2 we provide an overview of the relevant background areas for this thesis. Specifically, we describe the Resource Description Framework in Section 2.1, sources of geographic information on the Web in Section 2.2, and spatio-temporal machine learning in Section 2.3.

Then, in the subsequent three chapters, we present one approach to validate and two approaches to enrich geographic information on the Web. In Chapter 3, we validate geographic Web information by introducing an approach for vandalism detection in OpenStreetMap. To this end, we propose the OVID model, a supervised binary classification model, in Section 3.4 and evaluate it in Section 3.6. Next, in Chapter 4 we enrich geographic Web information by discovering identity links between OpenStreetMap and knowledge graphs. We present the OSM2KG model in Section 4.5 and conduct an extensive evaluation in Section 4.7. Then, in Chapter 5 we introduce another enrichment approach for geographic Web information by inferring missing categorical information in Web markup, e.g., public event information. We describe the challenges resulting from the markup data quality in Section 5.3 and describe the enrichment model in Section 5.5.

The next two chapters describe two applications that benefit from the improved geographic Web data quality. First, in Chapter 6 we present the GeoVectors corpus, a world-scale resource providing ready-to-use embeddings of geographic entities extracted from OpenStreetMap. We describe the embedding generation process in Section 6.4. We use identity links between OSM and knowledge graphs to provide semantic access to the resource described in Section 6.5. Then, in Chapter 7, we address the problem of event impact prediction on road traffic. To this end, we exploit event information, road network data from OSM, and additional geo-entity information. We provide a formalization of spatial and temporal event impact in Sections 7.4 and 7.5. Finally, we conclude the thesis in Chapter 8 by summarizing the findings and discussing future research directions.

2 Background

This chapter presents the most relevant concepts for collecting and processing geographic information on the Web. First, we describe best practices for publishing data on the Web and introduce the *Linked Open Data* paradigm and the *resource description framework*. Then, we provide an overview of geographic information sources on the Web including *OpenStreetMap*, *knowledge graphs* and *semantic Web markup*. Finally, we discuss architectures for processing geographic Web information, i.e., architectures for *spatio-temporal machine learning models*.

2.1 Linked Open Data & The Resource Description Framework (RDF)

Linked Open Data (LOD), also referred to as Linked Data, is a data publishing paradigm [HB11]. The core idea of Linked Open Data is to connect distributed data sources on the Web by establishing links between entities of the respective data sources. LOD aims to create a connected Web of data providing structured access to seamlessly connected data sources and thereby increasing the potential utility of all included data sources [BHB09, HB11].

An essential concept to establish the Web of Data is the use of *International Resource Identifiers* (IRIs) [SRM+14]. IRIs are absolute references to resources, i.e., references to arbitrary objects or concepts, with a global scope. The most prominent examples of IRIs are uniform resource locators (URLs), typically used to reference Web pages. Due to the global scope of IRIs, IRIs distinctly identify resources across different datasets. These global identifiers enable references between different datasets, e.g., by indicating which resources refer to the same real-world object.

From a technical perspective, the Resource Description Framework (RDF) has become the key technology for publishing Linked Open Data. RDF is a framework for specifying machine-readable information about arbitrary entities [SRM+14]. The

World Wide Web Consortium (W3C) endorsed RDf as W3C recommendation in the year 1999 [Wor99]. The RDF specification version 1.0 appeared in the year 2004. In version 1.0, RDF was intended as language to "*represent metadata about Web resources*" [MMM04]. Ten years later, in 2014, version 1.1 of the RDF specification was released. Version 1.1 broadened RDF's scope to a general "*framework for expressing information about resources.*" [SRM+14]. Here, the term *resource* is a general placeholder and may refer to arbitrary real-world entities as well as abstract concepts.

RDF describes resources in the form of so-called *triples*. Each triple consists of a *subject*, a *predicate*, and an *object*. The simplest format to express RDF statements is *N-Triples*. A single N-Triple consists of white space separated subject, predicate, and object terminated by a "." [Bac14]. For instance, the information that Hanover is a city could be expressed via the following N-Triple:

```
<Hanover> <is a> <city> .
```

Where `<Hanover>` is the subject, `<is a>` is the predicate, and `<city>` is the object. In an RDF triple, all three subject, predicate, and object can be an IRI.

Running Example

Throughout this chapter, we illustrate different representations of geographic objects by the example of the city of Hanover, Germany. We adopt the RDF representation of Hanover in the GeoNames[1] database. The GeoNames project captures the names of geographical places worldwide, e.g., cities, countries, or mountains. Listing 2.1 presents an excerpt of Hanover's RDF representation in GeoNames. The first two lines define prefixes for the used vocabularies. `gn` refers to the GeoNames vocabulary, while `wgs84_pos` refers to the W3C basic vocabulary for representing longitude and latitude in a WGS84[2] coordinate system. Prefixes abbreviate redundant parts of IRIs to allow the easier reading and writing of RDF statements. Each line represents a single RDF triple. The subject of all triples is Hanover's IRI in GeoNames `https://www.geonames.org/6559065`. Also, all predicates are IRIs referring to well-defined properties. For instance, `gn:name` (spelled out `http://www.geonames.org/ontology#name`) indicates the name of the subject, whereas `wgs84_pos:lat` and `wgs84_pos:long` indicate the geographic coordinates in the form of latitude and longitude.

[1] https://www.geonames.org/

[2] The World Geodetic System (WGS) is the standard coordinate system for coordinates on the earth surface. WGS84 denotes the current version [wgs14].

Listing 2.1: Exerpt of Hanover's RDF representation from the GeoNames database.

```
@prefix gn: <http://www.geonames.org/ontology#> .
@prefix wgs84_pos <http://www.w3.org/2003/01/geo/wgs84_pos#> .

<https://www.geonames.org/6559065> gn:name 'Hannover' .
<https://www.geonames.org/6559065> gn:officialName 'Hannover' .
<https://www.geonames.org/6559065> gn:countryCode 'DE'.
<https://www.geonames.org/6559065> gn:population 536925 .
<https://www.geonames.org/6559065> wgs84_pos:lat 52.37362 .
<https://www.geonames.org/6559065> wgs84_pos:long 9.73711 .
```

2.2 Geographic Information on the Web

In this section, we discuss essential sources of geographic information on the Web. In particular, we give an overview on *OpenStreetMap*, geographic *knowledge graphs*, and *semantic Web markup*.

2.2.1 OpenStreetMap

The OpenStreetMap (OSM) project collects and provides free geographic data covering the whole world [Ope21b, AZMH15]. Steve Coast founded OSM at the University College London in 2004 [HW08]. Today, OSM is the most prominent example of volunteered geographic information (VGI) [Goo07]. OSM is a collaborative project that collects all information from community members, so-called "contributors", that voluntarily provide geographic data under the *Open Database License* (ODbL)[3]. In principle, OSM aims at capturing data about all objects with a geographic extent. Typical OSM objects include physical objects like roads, rivers, or forests, but also conceptual objects like country boundaries. Figure 2.1 shows the OSM Web interface that provides an accessible way to explore and visualize the map data.

OSM has recently evolved as the key source of openly accessible VGI for many parts of the world. The amount of information available in OpenStreetMap is continuously growing. For instance, the number of GPS points captured by OSM increased from $7.4 \cdot 10^9$ in 2019 to $8.6 \cdot 10^9$ in 2021. Figure 2.2 presents a visualization of OSM's growth from 2006 to 2021 with respect to the number of registered users in Figure 2.2a and the number of captured GPS points in Figure 2.2b.

Today, OSM data is used in a plethora of machine learning applications such as road traffic analysis [KGG20], remote sensing [VMSTF21], and geographic entity disambiguation [TD21a]. Other data-driven OSM applications include map tile generation [HW08], routing [HR16], and knowledge graph generation [SLHA12].

[3] https://opendatacommons.org/licenses/odbl/

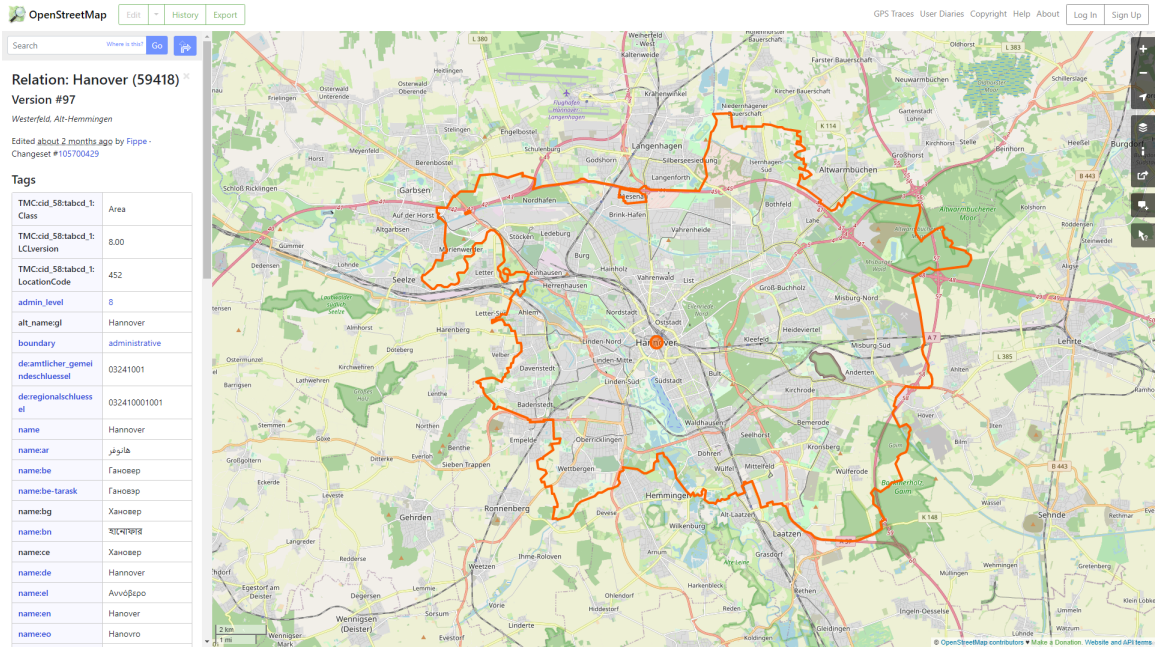
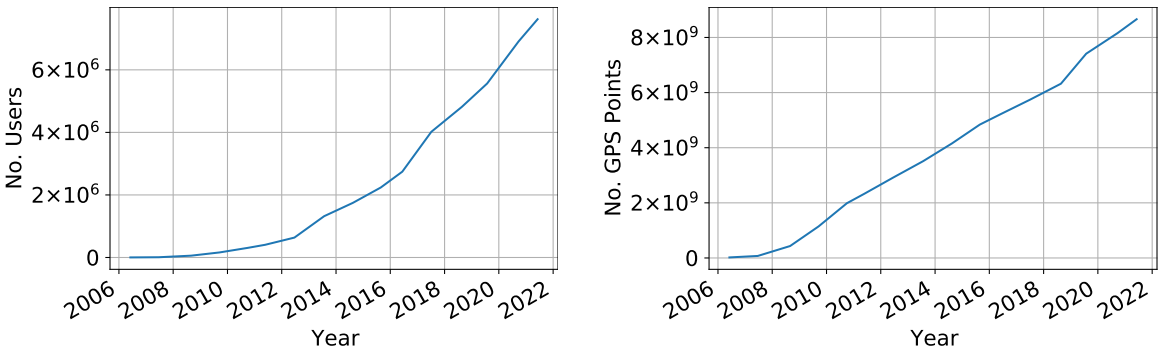


Figure 2.1. OpenStreetMap Web view of Hanover, Germany. ©OpenStreetMap contributors, ODbL.



(a) No. OSM Users (b) No. GPS Points

Figure 2.2. Growth of OpenStreetMap with respect to number of registered users and number of captured GPS points.

The OpenStreetMap Data Model

OpenStreetMap categorizes its geographic objects into three types. *Nodes* represent geographic points (e.g., mountain peaks) with the position specified by latitude and longitude. *Ways* represent lines (e.g., roads) composed of a sequence of nodes. *Relations* are composed of nodes and ways and describe more complex objects, e.g., national borders. Relations can also include sub-relations. Formally, we define an OSM object as follows:

Definition 2.1 (OSM object)**.** *An* OSM object *is defined as* $o = \langle id, type, loc, tags, ver \rangle$*, where:*

- id *is an object identifier.*
- $type \in \{\texttt{Node}, \texttt{Way}, \texttt{Relation}\}$ *indicates the object type.*
- loc *indicates the geographic location of the object. The location can either be a point (Node), a line (Way), or a set of points and lines (Relation).*
- $tags$ *is a set of "tags" describing object characteristics. Each tag* $\langle k, v \rangle \in tags$ *is represented as a key-value pair with the key* k *and a value* v*.*
- ver *is the version number of the object. The version number corresponds to the number of revisions of the object* o*.*

An OSM object o *can be distinguished by its identifier* $o.id$ *together with its type* $o.type$*.*

An OSM object may exhibit an arbitrary number of so-called *tags*, i.e., key-value pairs, that describe the semantics of the object. For instance, the tag `place=city` indicates that an OSM object annotated with this tag represents a city. OSM does not provide a fixed taxonomy of keys or range restrictions for the values but encourages its users to use established key-value combinations and to follow a set of best practices[4]. For example, the node labels are often available under the "`name`" key, whereas the labels in different languages can be specified using the "`name:code=`" convention[5]. However, OSM contributors are free to introduce new tags to OSM on their behalf[6]. As a consequence, the information representation and the level of detail provided for individual OSM objects are very heterogeneous [TR15]. This heterogeneity may make the use of OSM data potentially challenging.

Figure 2.3 illustrates this heterogeneity by depicting the mean and the standard deviation of the number of tags for selected object types. In particular, we consider the most frequent entity types in an OSM snapshot of Germany from 2018 such as cities, train stations, castles, and mountains. We observe that the number of tags

[4] https://wiki.openstreetmap.org/wiki/Map_features
[5] https://wiki.openstreetmap.org/wiki/Multilingual_names
[6] https://wiki.openstreetmap.org/wiki/Any_tags_you_like

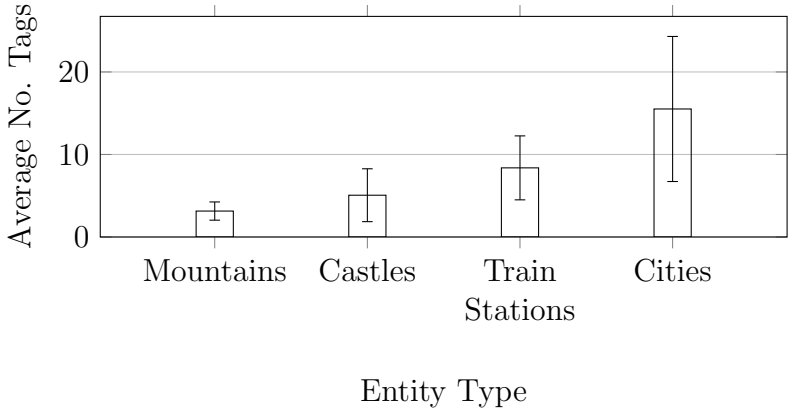


Figure 2.3. Average number of tags per object type in an OpenStreetMap snapshot of Germany from 2018. Error bars indicate the standard deviation.

varies significantly with the entity type. Moreover, the standard deviation is relatively high (between 35% and 63%) for all entity types. While for some entity types (e.g., mountains) the variation in the absolute number of tags is rather small, other types (e.g., cities) exhibit more substantial variations, meaning that some cities possess more detailed annotations compared with the rest.

Running Example

For the running example, we present Hanover's representation in OSM in Listing 2.1. The relation with the ID 59418 represents Hanover. The geographic location (loc) is a polygon also depicted in Figure 2.1 as a red line. The polygon consists of several ways, i.e., lines, that define the polygon boundaries. The tags describe various properties including traffic information systems identifier (`TMC:cid_58:tabcd_1:LocationCode`), administrative information (`de:amtlicher_gemeindeschluessel`), the name in different languages, and links to the corresponding Wikipedia page.

Listing 2.1. Representation of Hanover in OSM (©OpenStreetMap contributors, ODbL).

ID	59418
type	Relation
ver	97
loc	
member type	member ID
Node	1651888734
Way	23729287
Way	239327869
Way	138493032
⋮	⋮
Way	834738892
tags	
key	value
TMC:cid_58:tabcd_1:Class	Area
TMC:cid_58:tabcd_1:LCLversion	8.00
TMC:cid_58:tabcd_1:LocationCode	452
admin_level	8
alt_name:gl	Hannover
boundary	administrative
de:amtlicher_gemeindeschluessel	03241001
de:regionalschluessel	032410001001
name	Hannover
name:ar	هانوفر
name:be	Ганновер
⋮	⋮
name:zh	汉诺威
type	boundary
wikidata	Q1715
wikipedia	de:Hannover

2.2.2 Knowledge Graphs

Knowledge Graphs (KGs) have recently become a popular way to model large or heterogeneous data collections [HBC+21]. In 2012, Google coined the term "Knowledge Graph' by introducing the Google knowledge graph [Sin12]. Academia and industry quickly adapted the concept due to its ability to capture complex relations in arbitrary domains [NGJ+19, HBC+21]. Today, the major technology companies (e.g., Microsoft, Google, Facebook, eBay, and IBM [NGJ+19]) have adopted knowledge graphs, and research on knowledge graphs constitutes a vital topic in the Semantic Web field.

Knowledge Graphs use a graph structure to model entities and their mutual relations. That is, entities constitute the nodes, and the relations form the edges in the graph. In contrast to tabular data, where the columns induce a fixed schema for each row, knowledge graphs provide a lightweight way to model arbitrary relations and nonlinear structures. Figure 2.4 presents an example of a simple knowledge graph. The nodes represent different entities, i.e., cities (Hanover, Hildesheim), a federal-state (Lower Saxony), and a climate zone (Oceanic Climate). The edges express relations between the entities, i.e., that Hanover and Hildesheim are neighboring cities and that both are located in Lower Saxony. Finally, Hanover, Hildesheim, and Lower Saxony have the same climate classification, i.e., oceanic climate.

Today, several established open knowledge graphs exist. The most popular knowledge graphs capture general knowledge [HBC+21]. Prominent examples are the Wikidata [VK14], the DBpedia [LIJ+15], the YAGO [SKW07], or the Freebase [BEP+08] knowledge graphs. These graphs are extracted from textual knowledge bases [LIJ+15, SKW07], e.g., Wikipedia, or are manually constructed by volunteers [VK14, BEP+08]. Another class of knowledge graphs aims to capture domain-specific information. Examples of geographic knowledge graphs are the LinkedGeoData KG [SLHA12] and WorldKG [DTY+21] extracted from OpenStreetMap and the YAGO2GEO KG that

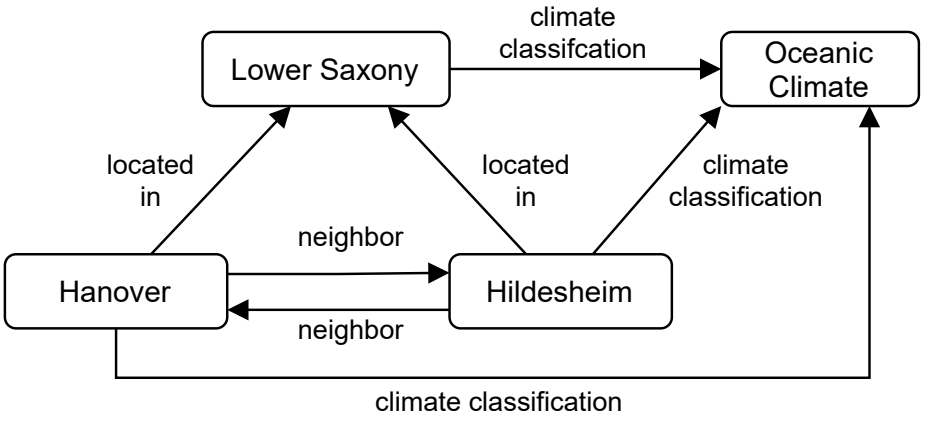


Figure 2.4. Simple knowledge graph example.

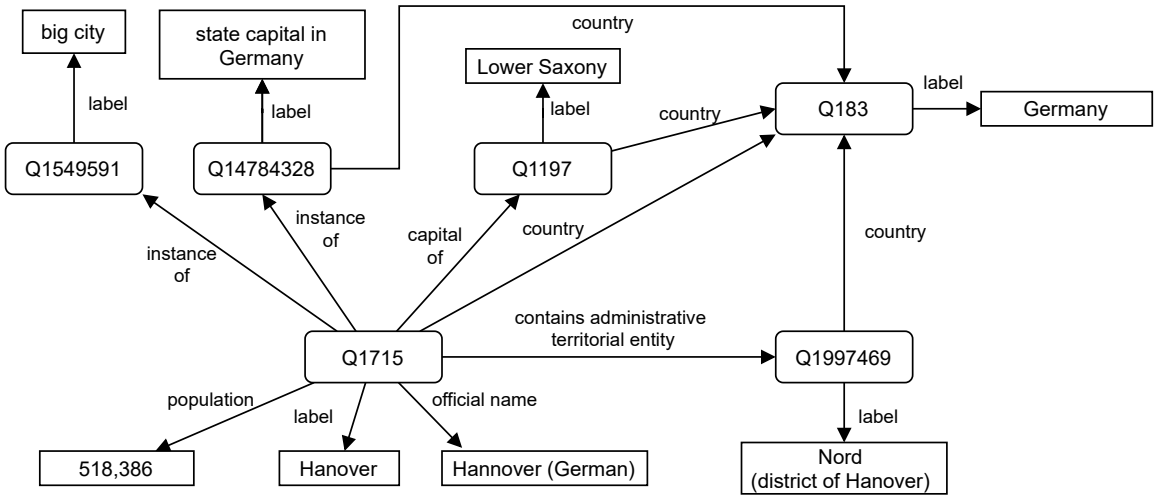


Figure 2.5. Excerpt of Hanover's representation in the Wikidata knowledge graph.

extends the YAGO2 KG with geographic information [KMK19]. Popular applications of knowledge graphs include but are not limited to information retrieval, recommender systems, or question answering [HBC+21, NGJ+19].

Running Example

In the running example, we consider the representation of Hanover in the Wikidata knowledge graph[7]. Figure 2.5 presents an excerpt of the node representing Hanover (Q1715) and direct neighbors. Rectangles with rounded corners indicate entities, while rectangles with sharp corners represent literal values. All entities can be identified using Wikidata IDs that also serve as IRIs. The label edges indicate the colloquial name of the nodes. For instance, the label of Q1715 is "Hanover" and the label of Q1549591 is "big city". Several literal values indicate properties of Hanover, such as the population and the official name. The "instance of"-edges provide type information about Q1715, i.e., Hanover is of type "big city" and "state capital in Germany". Furthermore, the remaining edges indicate Hanover's relation to other entities that, in turn, have mutual relations. For instance, Hanover has a "capital of"-relation to the German federal state Lower Saxony and a "contains administrative territorial entity' to the city district Nord. Both Lower Saxony and the district Nord have a country-relation with the node representing Germany such that multiples connections between Lower Saxony and Nord exist. This example illustrates the capability of knowledge graphs to model such nonlinear relation patterns.

[7] https://www.wikidata.org/wiki/Q1715

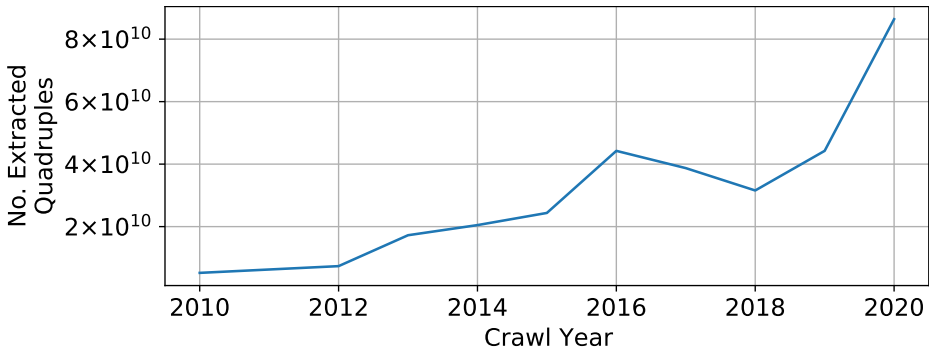


Figure 2.6. No. triples extracted from semantic markup in the Web Data Commons corpus with respect to the year of the Web crawl.

2.2.3 Semantic Markup

Semantic markup embeds semi-structured data about entities within Websites. Several formats recommend by the W3C such as Microdata, RDFa [Wor08], or JSON Linked Data (JSON-LD) [Wor20] exist and can be used to annotate the semantic entity information. The embedded markup formats can be easily transformed into other RDF formats, such as RDF quadruples (N-Quads)[Wor14]. RDF quadruples extend the RDF triples concept introduced in Section 2.1 by adding a graph label to each triple. The graph label specifies the scope of the quadruple. In the case of Web markup, the graph label often indicates the origin, i.e., the URL from which the triple was extracted.

By today, semantic markup data is available at an unprecedentedly large scale, which can be exemplarily observed on the *Web Data Commons* (WDC) [MPB14] corpus. WDC[8] offers a large-scale corpus of RDF quadruples extracted from the *Common Crawl*[9]. The Common Crawl is a non-profit initiative that provides open accessible Web crawl datasets. Figure 2.6 presents the number of extracted quadruples in the WDC datasets with respect to the crawl year. We observe a near-constant growth of extracted quadruples of the years. The growth indicates the increasing adoption of semantic markup. For instance, the crawl of October 2016 contains $3.41 \cdot 10^9$ URLs, of which 50% exhibit markup from which over $8.6 \cdot 10^{10}$ triples were extracted. In contrast, the crawl of November 2015 contains $3.18 \cdot 10^9$ URLs, of which only 39% exhibit markup, resulting in only about $4.4 \cdot 10^{10}$ extracted triples [BMP21].

The most prominent use case of semantic markup is *Search Engine Optimization* (SEO) [Mik15], i.e., to provide machine-readable information about website contents to search engines. While being leveraged to facilitate interpretation and retrieval of Websites by most major search engines, markup data is also helpful for maintaining and augmenting knowledge graphs. Additional Web markup applications include

[8] http://webdatacommons.org/
[9] http://commoncrawl.org/

Google Rich Snippets, Pinterest Rich Pins and search features for Apple Siri [GBM16]. The *Schema.org* vocabulary is a joint initiative from major search engines such as Bing, Google, Yahoo! and Yandex that provides a joint vocabulary and is the most commonly deployed vocabulary for semantic markup on the Web [BEM+13]. In the following we abbreviate the prefix of the *schema.org* vocabulary by `s:`, e.g., `s:City`.

Running Example

Due to the still limited adaption of semantic markup describing geographic entities, we manually augment an excerpt of Hanover's Wikipedia infobox[10] with the corresponding schema.org annotations for the running example. Listing 2.1 presents a representation of Hanover with semantic Microdata annotations marked in brown color.

First, the `itemtype` attribute of the `tbody` specifies the entity type `s:City`, indicating that the table describes a city. Next, the individual rows of the table describe particular properties. The `itemprop` attribute provides the property name, e.g., `name`, `containedInPlace`, or `url`. Properties can also refer to sub-entities. For instance, the `temprop="geo"` attribute indicates the "geo" property describing the geographic extent of an entity. An own entity of type `s:GeoCoordinates` provides the latitude and longitude information. Finally, the `s:PropertyValue` type can provide additional properties currently missing in schema.org. In the example, we define the "Elevation" and "Population" properties.

Listing 2.3: Example of a Microdata representation of Hanover. Brown color indicates Microdata markup.

```
<table>
  <tbody itemscope itemtype="https://schema.org/City">
    <tr>
      <th colspan="2" class="infobox-above">
        <span class="wrap" itemprop="name">Hanover</span>
      </th>
    </tr>
    <tr>
      <th scope="row" class="infobox-label">Country</th>
      <td class="infobox-data">
        <span itemprop="containedInPlace">Germany</span>
      </td>
    </tr>
    <tr class="mergedrow">
      <th scope="row" class="infobox-label">State</th>
      <td class="infobox-data">
        <span itemprop="containedInPlace">Lower Saxony</span>
```

[10] https://en.wikipedia.org/wiki/Hanover

```
      </td>
    </tr>
    <tr class="mergedtoprow" itemprop="geo" itemscope
        itemtype="https://schema.org/GeoCoordinates">
      <th scope="row" class="infobox-label">Coordinates</th>
      <td class="infobox-data">52  22'N 9  43'E</td>
      <meta itemprop="latitude" content="52.366667" />
      <meta itemprop="longitude" content="9.716667" />
    </tr>
    <tr class="mergedtoprow" itemscope
        itemtype="https://schema.org/PropertyValue">
      <th scope="row" class="infobox-label">Elevation</th>
      <td class="infobox-data">55 m (180 ft)</td>
      <meta itemprop="name" content="Elevation" />
      <meta itemprop="value" content="55m" />
    </tr>
    <tr class="mergedtoprow" itemscope
        itemtype="https://schema.org/PropertyValue">
      <th colspan="2" class="infobox-header">Population</th>
      <td class="infobox-data">534,049</td>
      <meta itemprop="name" content="Population" />
      <meta itemprop="value" content="534,049" />
    </tr>
    <tr class="mergedrow">
      <th scope="row" class="infobox-label">Website</th>
      <td class="infobox-data">
        <a href="https://www.hannover.de/" itemprop="url">
            www.hannover.de</a>
      </td>
    </tr>
  </tbody>
</table>
```

2.3 Spatio-Temporal Machine Learning

This section discusses the technical background in the area of spatio-temporal machine learning. Spatio-temporal machine learning algorithms exploit both the spatial and temporal dimensions and often learn patterns from geographic data. In this thesis, we only consider spatial data located on the earth's surface. Therefore, we will interchangeably use the terms "spatial", "geographic", and "geospatial" throughout this thesis.

Figure 2.7 presents a typical spatio-temporal machine learning pipeline based on the description of Wang et al. [WCY20]. The pipeline starts with raw spatio-temporal

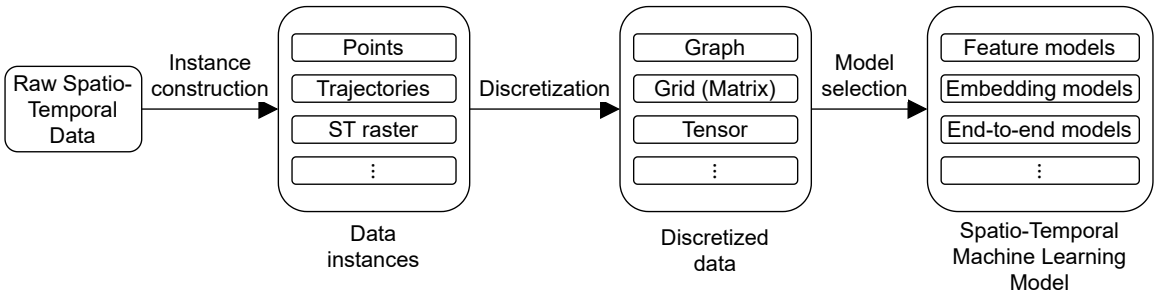


Figure 2.7. Typical spatio-temporal machine learning pipeline based on [WCY20].

data as input. First, the *instance construction* step transforms the input into well-defined spatio-temporal data types, e.g., points or trajectories. Then, the *discretization* step maps the spatio-temporal data into aggregated representations, e.g., graphs or grids. Finally, the *model selection* step aims to select an appropriate machine learning model to exploit the discretized data. In the following sections, we detail the individual steps of the pipeline.

2.3.1 Instance Construction

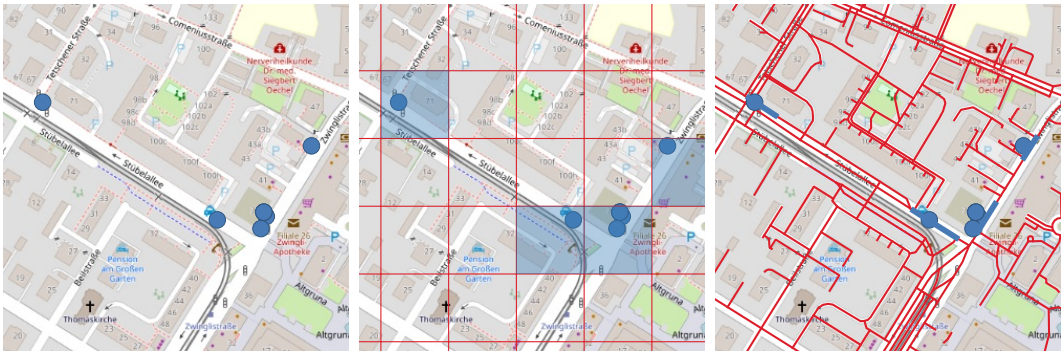
The instance construction step uses raw spatio-temporal data and transforms it into well-defined spatio-temporal data types.

In general, spatio-temporal data comes in many forms and can be collected with diverse sensors, including GPS sensors [AYW+16, CYH+18], induction loops for traffic detection [PDGS15, Chu12, AHC14, FPEG17] weather radars [HSZ20], or satellite imagery [TDEP18]. This section discusses the spatial-temporal data types and the corresponding data sources most relevant for this thesis.

Points: Points are instances whose location information is given by a single pair of latitude/longitude coordinates. Points can represent various real-world measurements for a particular location and a particular time, such as the taking place of a public event [TDD20], traffic incidents [PDGS15], stationary traffic speed recordings [Chu12, AHC14, FPEG17], or crime incidents [WCY20].

Trajectories: Trajectories are sequences of points, commonly representing the movement of objects through space and time. Each point of the trajectory captures the object's position at a particular point in time, such that subsequent points are temporal successors. Trajectory data is typically collected with sensors that are attached to the moving object [WCY20]. These sensors measure the position of the object at constant time intervals, also called the sampling rate [CFC+09]. Furthermore, some sensors provide additional measurements like the current velocity, acceleration, and electric field strength. The most prominent trajectory sensors are GPS sensors that are often integrated with navigation devices.

Other spatio-temporal instance types include spatio-temporal polygons, geograph-



(a) Raw GPS points (b) Grid Discretization (c) Graph Discretization

Figure 2.8. Traffic data discretization example. Map image ©OpenStreetMap contributors, ODbL.

ically referenced satellite images, and spatio-temporal raster data.

2.3.2 Spatio-Temporal Data Discretization

Data discretization is the processing of mapping continuous data points to a categorical data distribution [BOS09], e.g., mapping GPS coordinates to streets. The categories are also often referred to as *bins*. Depending on the task, categorical data distributions can enable more effective training of machine learning models. In the context of spatio-temporal data, the discretization process typically consists of separate spatial and temporal discretization steps.

Considering *spatial discretization*, two discretization techniques are commonly used. Figure 2.8 contrasts *grid discretization* and *graph discretization* of GPS point data. In *grid discretization*, a virtual uniform spatial raster indicates the discretization bins. Spatial geometries that fall into individual cells of the raster are mapped to the respective cells. On the one hand, grid discretization constitutes a simple and often effective way to discretize spatial data. On the other hand, grid discretization may introduce a loss of accuracy for the following reasons. First, the mapping to grid cells shadows the exact position, where the size of the grid cells is an upper bound for imprecision. Second, locations that fall into the same grid cell are not necessarily related to each other. For instance, a rural road and a highway with a parallel course might fall into the same grid cell but may not be reachable from each other.

In *graph discretization*, the nodes or edges of a spatial graph serve as bins. Spatial graphs are a flexible way to model many spatial distance relations. For instance, road networks can be modeled as spatial graphs, where the edges represent roads and the nodes represent junctions. Other examples include sensor networks or neighboring relations of city districts. A popular example, of graph discretization is the assignment of car trajectory to roads (so-called *map matching* [YG18]). Graph discretization is often more accurate than grid discretization and may enable more fine granular

analyses. On the downside, the graph construction and the mapping of spatial data to the individual graph components is typically more complex than the mapping to grid cells. Furthermore, modeling spatial data as a graph does not make sense for all kinds of spatial data.

Temporal discretization is usually carried out by mapping exact time stamps to discrete-time bins. The particular application determines the granularity of the bins. For instance, traffic forecasting might require more precise bins, e.g., 15-minute bins, whereas weather forecasting might allow for less granular bins, e.g., three-hour bins. Periodic patterns might also require determining reoccurring bins, such as weekdays or the time of day. For instance, rush hour traffic usually occurs in the morning and the afternoon of weekdays, but not during the night on weekends.

2.3.3 Spatio-Temporal Model Selection

The spatio-temporal model selection step aims to determine the best-suited machine learning model, such as classification, regression, or unsupervised models.

Traditional machine learning models, such as random forests, support vector machines, or decision trees have been applied to many spatio-temporal machine learning problems. These models require a feature engineering process and rely on explicit feature representations of spatial and temporal relations. Compared to neural networks, these models require fewer data to facilitate training but usually only achieve lower performance than neural models. However, traditional machine learning models are still a good alternative if only little training data is available.

Neural networks have superseded traditional machine learning models as state of the art for a wide range of spatio-temporal problems during the last years, e.g., traffic speed prediction [ZCM+20] and next location recommendation [LLL21]. Compared to traditional machine learning models, neural networks do not rely as much on feature engineering but are often able to automatically discover patterns in spatio-temporal data [GBC16]. On the downside, neural models usually require large amounts of training data. We distinguish between two classes of neural models: (i) models learning spatial and temporal patterns *separately* and (ii) model learning spatial and temporal patterns *jointly*.

Models *separately* learning spatial and temporal relations first apply a set of spatial layers and then employ the temporal layers on top of the spatial layers [WCY20]. The *spatial* layers aim to determine and aggregate spatial patterns in the data, for instance, mutually dependent road segments, congestion propagation, or relations between geographic regions. To this end, the spatial layer computes intermediate spatial representations of each time step individually, forming a *spatial time series*. Commonly used spatial layers include convolutional neural networks (often used for grid data) and graph neural networks. Recently, *geographic representation learning* algorithms have emerged as an additional method to determine spatial relations. These algorithms can be seen as preceding unsupervised feature extraction step to

transform geographic data into continuous numerical representations. Current geographic representation learning algorithms include stacked auto encoders [MZWL18], generative models [HMLS20], and word2vec-like models [FCAC17, YHMH19]. The *temporal* layers aim to determine temporal patterns in the spatial representations of the individual time steps, for instance, trends or periodic behavior. To this end, *recurrent neural networks* with an *encoder-decoder* architecture are commonly used. First, the encoder learns a latent representation of the spatial time series. Then, the decoder uses the latent representation to predict future values of the time series incrementally. Typical recurrent neural architectures include long short-term memory cells and gated recurrent units are com.

Lately, models *jointly* learning spatio-temporal relations have emerged. These models typically consist of spatio-temporal blocks that are repeatedly stacked on top of each other. Each block is capable of determining both spatial and temporal patterns. By removing the strict separation, these models aim to address the mutual dependence of spatial and temporal patterns. Consider a traffic jam as an example for mutual spatio-temporal dependencies. The length of a traffic jam grows with a longer duration of the jam. At the same time, the traffic jam will last longer if more roads are affected. Recent architectures for jointly learning spatio-temporal relations include convolutional neural networks [ZCM+20] and attention networks [LLL21].

3 Validation of OpenStreetMap Data through Vandalism Detection

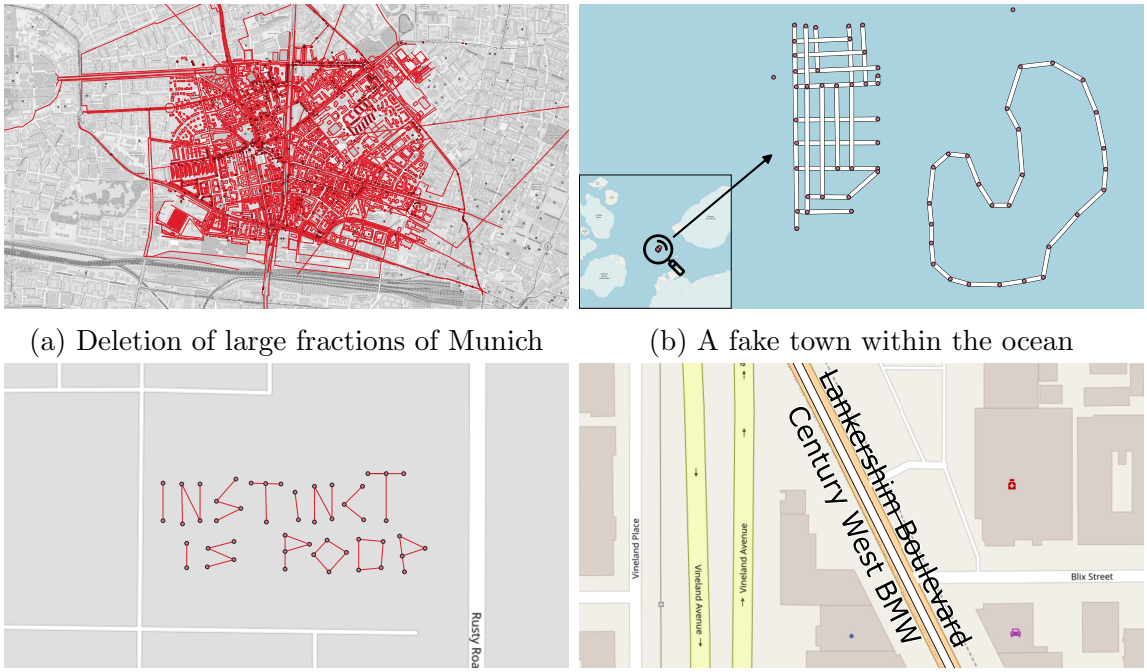
As a first step to increase data quality, we aim to validate the existing geographic information on the Web in this chapter before we later add missing information using the enrichment approaches described in Chapter 4 and 5. As discussed in Section 2.2.1, OpenStreetMap is currently one of the most important sources of geographic Web information. Therefore, in this chapter, we aim to validate OpenStreetMap information. In particular, we aim to detect vandalism in OpenStreetMap, i.e., wrong or prohibited information.

The manual correction of such information, e.g., by OSM contributors, is a costly process due to the scale of OpenStreetMap. Existing approaches for vandalism detection in crowd sourced knowledge bases used machine learning models to facilitate automated vandalism detection, e.g., in Wikidata. However, the heterogeneity of OSM objects and the lack of annotated training data imposes substantial challenges to the application of machine learning models. In **RQ1** we ask how to create such a machine learning model to detect vandalism in volunteered geographic information, such as OSM. In this chapter, we address **RQ1** by introducing the Ovid model, a machine learning model for vandalism detection in OpenStreetMap.

3.1 Introduction

The amount of geospatial information in OpenStreetMap is continuously growing. For instance, the number of nodes captured by OSM increased from $5.9 \cdot 10^9$ in March 2020 to $6.7 \cdot 10^9$ in March 2021. With the OSM growth, quality assurance becomes essential but also increasingly challenging. Recently, the problem of vandalism detection in OSM has attracted interest of researchers [TTdR20, VMSTF21] and OSM contributors[1]. For example, the OSM community identified several cases of vandal-

[1] https://wiki.openstreetmap.org/wiki/Vandalism



(a) Deletion of large fractions of Munich

(b) A fake town within the ocean

(c) "Graffiti" on the map

(d) Road name replaced by an advertisement

Figure 3.1. Real-world examples of different vandalism forms in OSM. Map data: ©OpenStreetMap contributors, ODbL.

ism in the context of the location-based mobile game Pokémon Go [JNHQ20], in which users added wrong information to the map to gain an advantage in the game. Today, OSM provides a data basis for various real-world applications, including navigation systems and geographic information systems (GIS). Detecting and removing vandalism cases is essential to preserve the credibility and trust in OSM data.

The problem of vandalism detection in OSM is particularly challenging due to the large scale of the dataset, the high number of contributors (over 7.6 million in June 2021), the variety of forms vandalism can take, and the lack of annotated data to train machine learning algorithms. Figure 3.1 presents four real-world vandalism examples. The vandalism forms in OSM include the arbitrary deletion of map regions, creating non-existing cities in the middle of an ocean, drawing texts using geometric shapes, and overwriting street names and other objects with advertisements and offensive content. Vandalism detection methods need to consider various aspects such as the geographic context, typical user behavior, and content semantics to identify potentially malicious edits effectively. The diversity of vandalism appearances and relevant features constitutes a significant challenge for automated vandalism detection.

Whereas the existing literature has considered OSM vandalism previously, only a few automated approaches for vandalism detection in OSM exist. An early approach proposed in [NGZ12] adopts a rule-based method to identify suspicious edits. This approach is subject to numerous manually tuned thresholds. In [TTdR20], the

authors proposed a random forest-based method that detects vandalized buildings. This approach is limited to the building domain and does not capture other various OSM vandalism forms. Furthermore, due to the shortage of benchmark datasets with real vandalism examples, existing studies typically utilize synthetic data and lack evaluation in real-world settings.

In this chapter, we present the Ovid (OpenStreetMap Vandalism Detection) model - a novel supervised machine learning approach to detect a variety of vandalism forms in OSM effectively. We propose a neural network architecture that adopts multi-head attention to select the most relevant edits within an individual changeset, i.e., a set of edits performed by a user within one session. Furthermore, we propose an original feature set that captures different aspects of OSM vandalism, such as user experience and contribution content. We train and evaluate Ovid on real-world vandalism occurrences in OSM, manually identified by the OSM community. To enable training of supervised machine learning models, we create a new ground truth dataset by extracting reverted entries from the OSM history. Although reverts indicating vandalism are available in OSM, identifying specific geographic entities affected by vandalism from reverts is not trivial as OSM does not specify which exact changeset is being corrected by the revert. Therefore, we develop an extraction procedure to extract vandalism occurrences accurately. Our evaluation results on two real-world datasets demonstrate that Ovid outperforms existing approaches by 8.14 percent points in F1 score and 5.41 percent points in terms of accuracy on average.

Contributions. In this chapter, we address **RQ1** and make the following contributions:

- We present Ovid – a novel machine learning method for vandalism detection in OpenStreetMap. Ovid relies on a neural network architecture that adopts a multi-head attention mechanism to summarize information indicating vandalism from OpenStreetMap changesets effectively.
- We propose a set of original features that capture changeset, user, and edit information to facilitate effective vandalism detection.
- We extract a dataset of real-world vandalism incidents from the OpenStreetMap edit history for the first time.
- We conduct an evaluation on the extracted real-world vandalism dataset and demonstrate the effectiveness of the proposed Ovid method, outperforming the baselines by eight percentage points regarding the F1 score on average.

The rest of the chapter is organized as follows: We discuss related work in Section 3.2. Then, in Section 3.3, we formally define the problem of vandalism detection in OpenStreetMap. In Section 3.4 we introduce the proposed Ovid model. We describe the experimental setup in Section 3.5. Following that, in Section 3.6 we present and discuss the evaluation results using two real-world OpenStreetMap datasets. Finally, we provide a discussion in Section 3.7.

3.2 Related Work

This section discusses the related work in the areas of vandalism detection in crowd-sourced knowledge bases and vandalism in OpenStreetMap.

Vandalism Detection in Crowd-Sourced Knowledge Bases. The existing literature investigated vandalism detection in crowd-sourced knowledge bases such as OpenStreetMap in several studies. Neis et al. early proposed *OSM-Patrol* [NGZ12], a rule-based system to detect vandalism in OpenStreetMap. OSM-Patrol determines a vandalism score for each edit. The score includes features such as user reputation, object type, or the number of established tags used in the edit. While OSM-Patrols aims at classifying individual edits, we classify entire changesets. In this chapter, we use OSM-Patrol as a baseline. Our experimental results confirm that Ovid outperforms this baseline.

More recently, another line of research has investigated the validation of building shapes within OpenStreetMap. Xie et al. developed a convolutional neural network that extracts building shapes from remote sensing imagery. The authors then compare the extracted shapes with shapes from OSM [XZX+19] to validate the buildings in OSM. The *OSMWatchman* approach uses a supervised random forest model for the detection of vandalism on buildings in OSM [TTdR20]. However, the authors evaluated OSMWatchman on synthetically created vandalism incidents only. Whereas the approach presented in [XZX+19] is too specific for our experimental setting, we compare Ovid to OSMWatchman as a baseline. In our evaluation, we demonstrate that Ovid outperforms OSMWatchman with respect to all considered metrics on real-world vandalism datasets.

Heindorf et al. investigated the problem of vandalism detection in the Wikidata knowledge graph [HPSE16, HPSE15, HSEP19]. They developed the *Wikidata Vandalism Detection* (WDVD) model that uses a random forest classification model together with user-based features (e.g., number of previous contributions) and text-based content features (e.g., the ratio of uppercase letters). We compare to WDVD as a baseline and show that our proposed Ovid model outperforms WDVD concerning F1 score and accuracy.

Another class of approaches aims at detecting vandalism of textual knowledge bases such as Wikipedia [PSG08, KSS15]. These approaches use Wikipedia-specific features (e.g., edits of meta-pages) and features tailored to natural language texts (e.g., the fraction of pronouns in a text). In contrast, OSM provides object descriptions as key-value pairs that do not normally contain long natural language texts, such that this class of models is not applicable to OSM data.

Vandalism in OpenStreetMap. Previous research has investigated the characteristics of vandalism in OpenStreetMap. Antoniou et al. identified vandalism as a threat to the OSM data quality in a recent survey on volunteered geographic information [ASF+17]. Quin et al. analyzed OSM user bans and further specified several threat categories such as *nefariousness, obstinance, ignorance*, and *mechani-*

cal problems [QB19]. Similarly, Ballatore et al. proposed a typology of vandalism in OpenStreetMap and coined the term "carto-vandalism" [Bal14]. They categorize vandalism incidents in the types *play, ideological, fantasy, artistic, industrial*, and *spam* carto-vandalism. The authors point out the potential use of automated tools such as machine learning for vandalism detection. Mooney et al. analyzed high frequently edited objects and found so-called "edit wars" in OpenStreetMap [MC12]. Edit wars are disputes of two or more contributors in which the contributors repeatedly revert each other's contributions. Edit wars are considered vandalism or bad-editing by the OSM community[2].

Recently, the OSM community observed an increased amount of vandalism aiming to manipulate the location-based game "Pokémon Go" that uses OpenStreetMap data. Juhász et al. found that the OSM community has manually corrected most of these incidents [JNHQ20].

These studies highlight the importance of mitigating vandalism in OpenStreetMap. Our Ovid model, proposed in this chapter, can lower the effort required for vandalism correction in OSM in the future.

3.3 Problem Definition

In this chapter, we target the problem of identifying vandalism changesets in OpenStreetMap. To this end, we extend the OSM formalization introduced in Section 2.2.1.

OSM allows for updates in the form of edits of individual OSM objects. An edit can either *create* new objects or *modify* or *delete* existing objects. More formally, we define an OSM edit as follows:

Definition 3.1 (Edit)**.** *An* edit *is defined as* $e = \langle o, op, ver, t \rangle$*, where:*

- o *is an OpenStreetMap object.*
- $op \in \{\texttt{create}, \texttt{modify}, \texttt{delete}\}$ *is the operation performed on the object* o*.*
- ver *is the new version number* $(o.ver + 1)$ *after the edit is performed.*
- t *is the time when the edit took place.*

Edits are submitted to OSM in the form of *changesets*. Changesets bundle multiple edits created by a single user during a short time period.

Definition 3.2 (Changeset)**.** *A* changeset *is defined as* $c = \langle E, t, u, co \rangle$*, where:*

- E *is a set of OSM edits that belong to the changeset.*

[2]See: https://wiki.openstreetmap.org/wiki/Vandalism

- *t is the changeset commit time.*
- *u is the user who committed the changeset.*
- *co is a comment describing the changeset contents.*

We denote the set of all changesets by C. We define the vandalism detection task in OpenStreetMap, as detecting the changesets that contain wrong or prohibited (e.g., discriminating or offensive) content.

Definition 3.3 (Vandalism Detection)**.** *Vandalism detection is the task of identifying changesets that constitute vandalism by either deleting correct information or adding wrong or prohibited information. We aim to learn a function* $\hat{y} : C \mapsto \{True, False\}$ *that assigns vandalism labels to changesets.*

In OSM, examples of vandalism include the deletion of existing towns, creating non-existing roads, adding advertisements, or replacing object names with offensive terms. Some vandalism examples are illustrated in Figure 3.1.

3.4 The Ovid Model

This section presents the Ovid (OpenStreetMap Vandalism Detection) model. Ovid is a supervised binary classification model that discriminates between regular and vandalism OSM changesets. The model consists of a supervised artificial neural network, including three main components: Feature Extraction, Feature Refinement & Aggregation, and Prediction. Figure 3.2 provides an overview of the Ovid model architecture. We adopt features in three categories. First, changeset features capture meta-information of the individual changesets, e.g., the editor software. Second, user features provide information regarding previous editing activities of the changeset author, e.g., the number of prior contributions. Third, edit features encode information describing the individual changes within the changeset, e.g., if an object was added, modified, or deleted. Since a single changeset may consist of multiple edits, Ovid relies on a multi-head attention mechanism to aggregate the edits and identify information relevant for vandalism detection. Finally, a sequence of prediction layers integrates the features and facilitates the detection of vandalism changesets. We present the main components of the Ovid approach in the following sections in more detail.

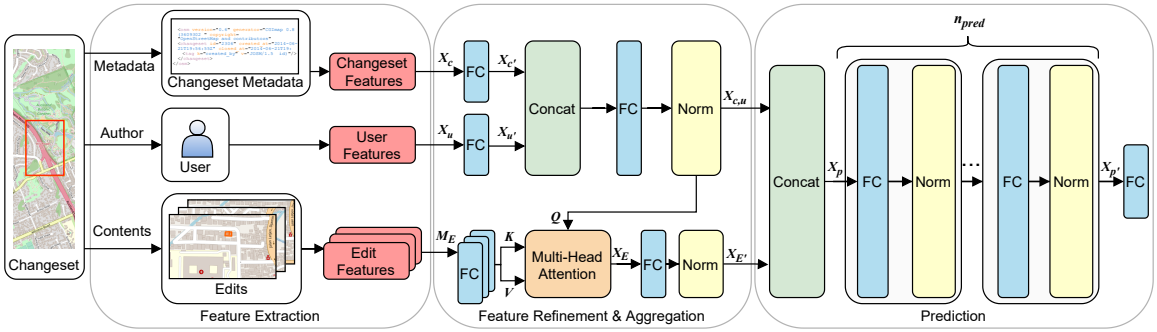


Figure 3.2. OVID model architecture. The changeset and user feature refinement part, as well as the prediction layers, are composed of fully connected (FC), normalization (norm), and concatenation (concat) layers. Multi-head attention layers aggregate the features from multiple edits in a changeset into a single feature vector. Map images: ©OpenStreetMap contributors, ODbL.

3.4.1 Feature Extraction

This section describes the extraction of the changeset, user, and edit features. Table 3.1 presents a summary of the features.

Changeset Features

The changeset features provide information regarding the changeset metadata. We denote the changeset feature vector by X_c. For a changeset c, the feature vector consists of the following individual features.

No. creates, no. modifications, no. deletes, no. edits. We capture the changeset size by the number of created, modified, and deleted objects and the total number of edits in the changeset $|c.E|$.

Min/max latitude/longitude, bounding box size. We capture the changeset's geographic extent by considering the minimum and maximum latitude and longitude among the changeset entries. Furthermore, we include the size of the overall geographic bounding box of the changeset. We expect that a large geographic extent may indicate vandalism.

Editor application. Several editor applications can create changesets for OSM. Basic editors are easy to use and, therefore, more likely to be used for vandalism. Following this intuition, we include the editor application as a categorical feature. We apply 1-hot encoding such that an individual dimension represents a specific editor application. For a specific editor application, we set the corresponding dimension to 1 and left all other dimensions as 0.

Has imagery used. OSM contributors may specify whether they used aerial images for a specific changeset, which intuitively can make a changeset more trustworthy. We hence include this information as a binary variable.

Comment length. Contributors can provide a comment *c.co* to document the contents of a changeset. Intuitively, a long description may indicate trustworthy changes. Therefore, we use the number of characters in a comment as a feature.

Table 3.1. OVID features overview

Feature	Type	Reference
Changeset Features		
No. creates	numerical	[Jas18]
No. modifications	numerical	[Jas18]
No. deletes	numerical	[Jas18]
No. edits	numerical	*original*
Max. latitude	numerical	*original*
Max. longitude	numerical	*original*
Min. latitude	numerical	*original*
Min. longitude	numerical	*original*
Bounding box size	numerical	*original*
Editor application	categorical	*original*
Has imagery used	binary	*original*
Comment length	numerical	[HPSE16]
User Features		
No. past creates	numerical	[NGZ12]
No. past modifications	numerical	*original*
No. past deletes	numerical	*original*
No. contributions	numerical	[HPSE16, TTdR20]
No. top-12 keys used	numerical	[NGZ12]
Account creation date	numerical	[HPSE16]
No. active weeks	numerical	[TTdR20]
Edit Features		
Edit operation	categorical	[NGZ12]
Object type	categorical	[NGZ12]
Object version number	numerical	[HPSE16]
No. previous authors	numerical	[HPSE16, TTdR20]
Time to previous version	numerical	[NGZ12, TTdR20]
No. tags	numerical	[TTdR20]
No. tags added	numerical	*original*
No. tags deleted	numerical	*original*
No. valid tags	numerical	[NGZ12]
No. previous valid tags	numerical	[NGZ12]
Name changed	binary	*original*

Finally, we concatenate all changeset features to obtain the changeset feature vector $X_c \in \mathbb{R}^{d_c}$, where d_c denotes the dimension of X_c.

User Features

We utilize user features to capture the previous activity of the changeset author $c.u$, as a more experienced user may be more trustworthy than a new user. Given a changeset c and its author $c.u$, we denote the user feature vector by X_u.

No. past creates, no. past modifications, no. past deletes. The user experience plays a vital role in quantifying the users' credibility [NGZ12]. Therefore, we count the number of objects that the user $c.u$ added to OSM and use this number as a feature. In addition, we also consider the number of objects edited by the user $c.u$ to capture another aspect of the user experience. We count the number of objects that $c.u$ has modified and deleted previously and use each count as a feature.

No. contributions. We count the overall number of objects contributed by the user and include this number in our model.

No. top-12 keys used. Following [NGZ12], we use the top-12 most frequent keys used in OSM and determine how often the user added one of the top-12 keys to an entry. As of May 2020, the top-12 most frequent keys are *building, source, highway, name, natural, surface, landuse, power, waterway, amenity, service,* and *oneway*. The top-12 most essential keys are likely to be present in the history of credible OSM users. In contrast, an absence of top-12 tags in the user history might indicate harmful behavior. We include the number of top-12 keys previously utilized by the user $c.u$ as a feature.

Account creation date, no. active weeks. To quantify the temporal scope of user experience, we consider the timestamp of the user account creation date and the number of weeks in which the user has contributed at least one changeset.

We concatenate all features to obtain the user feature vector $X_u \in \mathbb{R}^{d_u}$, where d_u denotes the dimension of X_u.

Edit Features

The edit features capture information regarding the individual edits contained in a changeset. As a single changeset c may contain many edits, we first extract the features for every individual edit $e \in c.E$. We extract the following features from each edit e:

Object type, edit operation. The object type $e.o.type \in \{\texttt{Node}, \texttt{Way}, \texttt{Relation}\}$ and the edit operation $e.op \in \{\texttt{create}, \texttt{modify}, \texttt{delete}\}$ provide basic information about the type of the edited object and the editing operation applied. Some object types might be easier to vandalize than others. For example, it is easier to move the single node representing the South Pole than to move the complex

relation representing Antarctica. We use one-hot encoding to represent both features.

Object version number, no. previous authors. If the edit changes an already existing object, we capture information about the object edit history by considering the version number $e.ver$ and the number of distinct previous authors. A high version number might indicate controversial objects that are a subject of so-called edit wars[3]. For instance, the country affiliation of some regions might be controversial[4].

Time to the previous version. We measure the time between the current and the last object version as the difference between the corresponding timestamps. If no prior version of the object exists, we set this feature to zero.

No. tags. The tags provide semantic information about the object. A high number of tags may indicate an established OSM object. Therefore, we consider the total number of tags $|e.o.tags|$, the number of added tags, and the number of tags deleted in the edit as features.

No valid tags, no. previous valid tags. The OpenStreetMap Wiki provides a description of established key-value pairs as so-called map feature list[5]. Following [NGZ12], we count the number of tags that appear in the map feature list as the number of valid tags. We assume that the use of a valid key-value combination indicates proper editing behavior. Likewise, we determine the number of valid tags in the previous version of the edited object if a previous version exists. Otherwise, we set this feature to zero.

Name changed. Vandalizing object names is an effective way to create visible fake information in OSM. Therefore, we create a binary feature indicating whether the new name of an object differs from its previous name. If the object does not have a name or a prior version, we set this feature to 0.

For each edit $e \in c.E$, we concatenate the features into the edit feature vector $X_e \in \mathbb{R}^{d_e}$, where d_e denotes the dimension of X_e.

3.4.2 Feature Refinement & Aggregation

In this step, we first refine the changeset and user features and then aggregate the edit features to obtain a single feature vector for each changeset.

We refine the changeset and user feature vectors by passing them to the fully connected layers $X_{c'} = FC_{d_h}(X_c)$ and $X_{u'} = FC_{d_h}(X_u)$ with:

$$FC_{d_h}(X_i) = ReLU(X_i W_i + b_i),$$

where $W_i \in \mathbb{R}^{d_i \times d_h}$ denotes a weight matrix, $b_i \in \mathbb{R}^{d_h}$ a bias vector, d_h is the hidden layer size, and $ReLU$ denotes the Rectified Linear Unit activation function [GBC16].

[3] https://wiki.openstreetmap.org/wiki/Disputes
[4] https://wiki.openstreetmap.org/wiki/Disputed_territories
[5] https://wiki.openstreetmap.org/wiki/Map_features

We concatenate the changeset and user features and apply a fully connected layer with normalization:

$$X_{c,u} = norm(FC([X_{c'}, X_{u'}])),$$

where $norm(\cdot)$ denotes layer normalization [BKH16] that scales the layer output based on the mean and standard deviation of the individual neurons' activation values.

For OVID, we aim at selecting the edits most relevant to identify vandalism in the corresponding changeset. To this extent, we aggregate the individual features of all edits in the same changeset. We combine the feature vectors X_e for each $e \in c.E$ into the edit feature matrix $M_e \in \mathbb{R}^{d_e \times |c.E|}$. We apply the same fully connected layer to each edit $M_{e'} = FC_{d_h}(M_e)$ to obtain the refined edit features $M_{e'}$. To aggregate the features of the individual edits into a single feature vector, we adopt the multi-head attention mechanism proposed by [VSP+17]. Intuitively, the multi-head attention mechanism computes a weighted sum of the edit features, where the model learns the so-called attention weights representing the importance of specific edits.

Formally, the attention mechanism distinguishes between a *query* Q, *keys* K and *values* V. Attention selects the most relevant values V for the query Q based on the similarity between Q and the keys K. As we aim at selecting the edits most relevant to identify vandalism in the corresponding changeset, we represent the refined changeset and user features as the query in the attention model and the refined edit features as keys and values: $Q = X_{c,u}$, $K = M_{e'}$, and $V = M_{e'}$. The *Attention* function is defined as:

$$Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V,$$

where Q denotes a query vector, K a key matrix, V a value matrix, and d_k is the dimension of one row (one key) of the key matrix. The term QK^T computes the similarity between the query vector Q and the individual keys in the key matrix K. Then, the softmax function transforms the similarities to a probability distribution representing the attention weights. The scaling factor $\sqrt{d_k}$ prevents the softmax from having extremely small gradients during back propagation [VSP+17]. Finally, the multiplication of the attention weights with the value matrix V yields the weighted sum of the values.

Multi-Head attention extends attention by using multiple (n_{head}) attention heads. Each head learns to focus on different edit feature combinations, e.g., the object type together with the semantic description provided by the tags. Formally, each attention head computes its own attention function:

$$Multi\text{-}Head(Q, K, V) = [\text{head}_1, ..., \text{head}_{n_{head}}]W^o,$$
$$\text{head}_i = Attention(QW_i^Q, KW_i^K, VW_i^V),$$

with the projection matrices $W_i^Q \in \mathbb{R}^{d_h \times d_h}$, $W_i^K \in \mathbb{R}^{d_h \times d_h}$, $W_i^V \in \mathbb{R}^{d_h \times d_h}$, and $W^O \in \mathbb{R}^{n_{head} \cdot d_h \times d_h}$.

We compute an aggregated edit feature vector using multi-head attention: $X_E = \textit{Multi-Head}(X_{c,u}, M_{e'}, M_{e'})$. Finally, we refine the edit feature vector using a fully connected layer with the layer normalization $X_{E'} = norm(FC_{d_h}(X_E))$.

Some changesets, e.g., automatic imports, may contain a high number of edits (up to 50,000 in our datasets) such that the contribution of an individual edit is negligible. Therefore, we introduce an upper threshold $th_{e,max}$ for the maximum number of edits within a changeset. If the number of edits exceeds $th_{e,max}$, we set $X_{E'} = 0$ and rely on the user and changeset features.

3.4.3 Prediction

We facilitate the detection of vandalism changesets by combining the refined changeset and user features with the aggregated edit features into a single feature vector $X_p = [X_{c,u}, X_{E'}]$. We repeat fully connected layers with layer normalization n_{pred} times and use a final fully connected layer with a single output dimension and sigmoid activation function to make predictions of the binary vandalism label:

$$X_{p'} = norm(FC_{d_h}(X_p))^{n_{pred}},$$

$$\hat{y} = sigmoid(X_{p'}W_{p'} + b_{p'}),$$

with $W_{p'} \in \mathbb{R}^{1 \times d_h}$ and $b_{p'} \in \mathbb{R}$. We consider a changeset as vandalism if $\hat{y}$ exceeds the classification threshold: $\hat{y} > th_{class}$. We use the established threshold for the sigmoid function $th_{class} = 0.5$ [GBC16]. We investigate the influence of th_{class} on the precision and recall of the classification model later in the evaluation in Section 3.6.3.

We normalize all input features by removing the mean and scaling to unit variance. We use the ADAM optimizer to train the network using a binary cross-entropy loss and apply L2 regularization to all layers:

$$L = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i) + \lambda \sum_{W_i \in W} ||W_i||^2,$$

where N denotes the number of training examples, W the set of all weight matrices, λ is the regularization weight, and $|| \cdot ||^2$ is the L^2 norm.

3.5 Evaluation Setup

This section describes the datasets, baselines, metrics, and hyperparameter optimization used in the evaluation.

Table 3.2. Dataset statistics for OSM-Reverts and OSM-Manual

Dataset property	OSM-Reverts	OSM-Manual
No. vandalism changesets	18,276	2,018
No. distinct users	8,768	1,686
Median create operations per changeset	5	3
Median modify operations per changeset	1	1
Median delete operations per changeset	1	1
Median nodes per changeset	6	4
Median ways per changeset	2	2
Median relations per changeset	1	1
Median edits per changeset	10	8
Median edits/vandalism changeset	10	4
Median edits/negative changeset	10	11
Timespan	2014-2019	2014-2019

3.5.1 Datasets

We conduct the experiments on two real-world datasets, *OSM-Reverts* and *OSM-Manual*, described in the following. Table 3.2 summarizes selected statistics of both datasets.

OSM-Reverts. We create a ground truth dataset by considering the changesets reverting vandalism in the OpenStreetMap history from 2014 to 2019. The correctness of the ground truth is essential for the training of supervised models. Therefore, we aim at high precision in the ground truth extraction process.

Determining specific vandalism changesets repaired by the reverts is not trivial. Whereas some reverts mention the corresponding changesets in the comment explicitly, others simply delete, update or insert geographic objects to repair the vandalism. The revert comments are highly heterogeneous and do not always explicitly specify the vandalism changeset.

The extraction process consists of the following steps: First, we extract the revert changesets that fix vandalism changesets. We only consider changesets that mention "vandalism" in their comments. Second, we determine the vandalism changesets corrected by the revert. If a revert changeset explicitly mentions a specific changeset, we consider the mentioned changeset as vandalism. Otherwise, we review the objects that are the subject of the revert. If the revert deletes an object and only one user contributed to the object, we consider changesets contributing to this object to be vandalism. It is hard to attribute the revert to a specific changeset in other cases. As we aim at high precision, we do not include such underspecified cases in our ground truth.

To create negative examples (i.e., changesets that do not represent vandalism), we remove the identified vandalism changesets and the reverts from the OSM changeset

history. The overall fraction of the vandalism changesets is small compared to all OSM changesets. Therefore, we obtain negative examples by randomly sampling changesets from the filtered OSM history. We randomly sample the same number of changesets as the vandalism changesets from the reduced changeset history to create negative examples and obtain a balanced dataset. As a result, we obtain a dataset with 18,276 training examples. We make our dataset available as open data to make our results reproducible and facilitate further research.[6] We split the dataset into training (70%), validation (10%), and test (20%) sets. To avoid bias towards individual OSM users, we ensure that the training, validation, and test sets are disjunct concerning OSM users. We train the models on the training and validation set and evaluate the results on the test set.

OSM-Manual. In 2018, the OSM community manually identified approximately one thousand vandalism incidents, including spam and forbidden imports.[7] We use these changesets as positive examples. Analogous to OSM-Reverts, we create negative examples by randomly sampling changeset from the OSM history. In total, the OSM-Manual dataset includes 2,018 examples. The number of examples in the OSM-Manual dataset is too small to train machine learning models. Thus, we use OSM-Manual only as a test set. We evaluate vandalism detection approaches with OSM-Manual by training the models on the entire OSM-Reverts dataset and then using OSM-Manual as a test set.

3.5.2 Baselines

We compare our model with the following baselines:

Random. This naïve baseline chooses the vandalism label at random.

OSMPatrol. This model is an early approach to detect vandalism in OSM [NGZ12]. OSMPatrol is a rule-based system that aims to identify vandalism at the level of OSM edits. For each edit, the baseline computes a vandalism score based on a combination of rules considering user information and edit features, e.g., the object version number. Each rule compares an individual feature to a threshold, where the thresholds are model parameters. In our settings, we utilize this baseline to label changesets. We consider a changeset c as vandalism if the baseline identifies at least one edit $e \in c.E$ in this changeset as vandalism. We apply an exhaustive grid search to find the optimal thresholds for the individual rules.

OSMWatchman. This model was recently proposed to detect vandalism on buildings in OSM [TTdR20]. OSMWatchman uses a random forest classifier that utilizes content features (e.g., number of tags), context features (e.g., time to the previous version), and user features (e.g., number of contributions). We apply random

[6] The OSM-Reverts dataset is available at: https://github.com/NicolasTe/Ovid

[7] The OSM-Manual dataset is available at: https://github.com/jremillard/osm-changeset-classification

search to optimize the hyperparameters of the random forest model.

WDVD. The Wikidata Vandalism Model was proposed to detect vandalism in the Wikidata knowledge graph [HPSE16]. This baseline uses a random forest classifier with text-based features, e.g., the ratio of upper case letters and user-based features, to detect vandalism. We apply random search to optimize the hyperparameters of the random forest model.

GloVe+CNN. OSM community members developed this baseline to detect suspicious changesets [Jas18]. This baseline transforms OSM changesets into pseudo-natural language sentences describing the changeset contents. For instance, the sentences can include information regarding the number of created and deleted objects and object tags. The baseline then uses pre-trained GloVe word embeddings [PSM14] to obtain numerical representations of the sentences. The numerical representations serve as an input for a convolutional neural network. We apply random search to optimize the hyperparameters of the neural network.

3.5.3 Metrics

To evaluate the performance of the different vandalism detection approaches, we compute the following metrics:

- **Precision.** The fraction of correctly classified vandalism instances among all instances classified as vandalism.
- **Recall.** The fraction of correctly classified vandalism instances among all vandalism instances.
- **F1 score.** The harmonic mean of recall and precision.
- **Accuracy.** The fraction of correctly classified instances among all instances.

As the F1 score reflects recall and precision and the accuracy measures the overall classification performance, we consider the F1 score and the accuracy as the most relevant metrics for this study.

3.5.4 Hyperparameter Tuning & Training

We optimize the hyperparameters of Ovid by applying the random search algorithm. Table 3.3 summarizes the hyperparameter search space. We train Ovid using the ADAM optimizer [KB15] and dropout layers. We use 100 epochs and apply the early stopping with patience strategy [GBC16].

Table 3.3. Hyperparameter search space of Ovid

Parameter	Description	Search Space
$th_{e,max}$	Maximum number of edits per changeset	$\{10, 20, 30\}$
n_{pred}	Number of prediction layers	$\{1, 2, 3, 4, 5\}$
n_{head}	Number of attention heads	$\{5, 10, 15, 20\}$
d_h	Hidden layer size	$\{12, 24, 36, 48\}$
δ	Dropout rate	$\{0.4, 0.5, 0.6, 0.7\}$
λ	Regularization weight	$\{0.005, 0.01, 0.02\}$

3.6 Evaluation

The evaluation aims to assess the effectiveness of the proposed Ovid approach for vandalism detection. Furthermore, we analyze the contribution of Ovid's feature categories in an ablation study and investigate the capability to customize our approach concerning precision and recall.

3.6.1 Vandalism Detection Performance

Table 3.4 summarizes the overall vandalism detection performance of the Random, OSMPatrol, OSMWatchman, WDVD, and GloVe+CNN baselines as well as our proposed Ovid approach. Table 3.4a and 3.4b respectively report the performance on the OSM-Reverts and the OSM-Manual datasets, while Table 3.4c provides the average scores.

Overall, we observe that in terms of F1 score and accuracy, Ovid achieves the best performance on both datasets. On average, Ovid achieves 8.14 percent points improvement in F1 score and 5.41 percent points improvement in accuracy compared to the best performing baseline.

The OSMPatrol baseline achieves the best recall on both datasets. However, OSMPatrol's precision scores are close to 50%, which corresponds to the performance of the random choices by the naïve Random baseline. The recall and precision scores reveal that OSMPatrol assigns almost all changesets to the vandalism class, resulting in ultimately low accuracy of 55.47% on average. The low accuracy scores indicate that supervised machine learning models like Ovid are better suited to detect vandalism than the OSMPatrol system that relies on manually specified rules.

WDVD achieves the best precision on the OSM-Reverts dataset, but only reaches a recall score of 64.79%. WDVD mainly relies on user features. The high precision score indicates that user features can effectively identify a fraction of the malicious changesets. However, the low recall score indicates that user features are insufficient to capture the diverse vandalism forms. Low recall means that many vandalism cases

Table 3.4. Vandalism detection performance with respect to precision, recall, F1 score and accuracy [%]. Best scores are marked bold.

(a) OSM-Reverts

Approach	Precision	Recall	F1	Accuracy
Random	49.89	50.27	50.08	49.92
OSMPatrol	53.94	**96.29**	69.15	57.06
OSMWatchman	77.60	70.74	74.01	75.18
WDVD	**81.52**	64.79	72.20	75.07
GloVe+CNN	81.46	72.93	76.96	78.18
Ovid	80.35	83.02	**81.66**	**81.37**

(b) OSM-Manual

Approach	Precision	Recall	F1	Accuracy
Random	49.46	49.95	49.70	49.45
OSMPatrol	52.20	**91.77**	66.55	53.87
OSMWatchman	57.75	32.11	41.27	54.31
WDVD	74.88	46.98	57.73	65.61
GloVe+CNN	**82.16**	19.62	31.68	57.68
Ovid	69.86	70.76	**70.31**	**70.12**

(c) Average

Approach	Precision	Recall	F1	Accuracy
Random	49.68	50.11	49.89	49.69
OSMPatrol	53.07	**94.03**	67.85	55.47
OSMWatchman	67.68	51.43	57.64	64.74
WDVD	78.20	55.88	64.97	70.34
GloVe+CNN	**81.81**	46.27	54.32	67.93
Ovid	75.11	76.89	**75.99**	**75.75**

will remain undetected when using this baseline. In contrast, Ovid that considers user, changeset, and edit features, achieves 83.02% recall on OSM-Reverts.

The GloVe+CNN baseline achieves the best precision on OSM-Manual but only achieves a recall score of 19.62% resulting in a relatively low F1 score of 31.68%. GloVe+CNN uses information from the changeset and the edits, but does not consider user information. Comparing GloVe+CNN to WDVD, we observe that WDVD achieves higher recall and F1 scores on OSM-Manual than GloVe+CNN. This result indicates the importance of the user features for the OSM-Manual dataset. Ovid that combines user and content features achieves a high recall (70.76%) on OSM-Manual while maintaining a comparably high precision (69.86%).

The OSMWatchman baseline achieves a moderate performance considering all metrics on OSM-Reverts but fails to maintain the performance level on OSM-Manual. OSMWatchman uses a combination of user and content features. However, the low F1 score on OSM-Manual of 41.27% indicates that OSMWatchman's specific feature set does not generalize to the OSM-Manual dataset.

Comparing the performance across the datasets, we generally observe higher scores on OSM-Reverts than on OSM-Manual. As OSM-Manual is too small to train supervised machine learning models, we trained all models on OSM-Reverts for both datasets, as described in Section 3.5.1. The difference in the performance indicates that the datasets exhibit slightly different underlying distributions. The difference in the distribution may, for instance, result from different vandalism forms contained in the datasets. As expected, we observe the best performance when we train and evaluate the models on the train and test datasets obtained from the same distribution, i.e., OSM-Reverts. However, the performance on OSM-Manual indicates the ability of the model to generalize to unseen data. The GloVe+CNN baseline, which achieved the second-best performance on OSM-Reverts, fails to generalize to OSM-Manual and only achieves an F1 score of 31.68%. In contrast, we observe that Ovid achieves higher than 70% F1 score and accuracy. This observation indicates that Ovid's proposed features and model architecture better generalize to unseen data than the baselines.

3.6.2 Ablation Study

We conduct an ablation study to assess the contribution of Ovid's feature categories. To this end, we remove individual parts of our model and measure the vandalism detection performance on the OSM-Reverts and OSM-Manual datasets. We consider the following configurations for the ablation study:

- Ovid$_{-Changeset}$: We remove $X_{c'}$, i.e., the changeset features and the corresponding refinement layer.
- Ovid$_{-User}$: We remove $X_{u'}$, i.e., the user features and the corresponding refinement layer.

- OVID$_{-Edits}$: We remove $X_{E'}$, i.e., the edit features and the corresponding multi-head attention and refinement layers.
- OVID$_{-Changeset,Edits}$: We remove $X_{c'}$ and $X_{E'}$, i.e., the changeset and edit features and the corresponding layers.
- OVID$_{-User,Edits}$: We remove $X_{u'}$ and $X_{E'}$, i.e., the user and edit features and the corresponding layers.

We do not remove $X_{c'}$ and $X_{u'}$ simultaneously since the edit features aggregation component requires at least one of $X_{c'}$ or $X_{u'}$ to provide the query vectors for the multi-head attention. Table 3.5a presents the scores on OSM-Reverts, while 3.5b provides the scores on OSM-Manual.

On the OSM-Reverts dataset, we observe that we can not remove any feature category without reducing the vandalism detection performance. In other words, every feature category of our proposed model contributes to the vandalism detection performance. Considering configurations removing only one feature category, we observe the highest difference in accuracy for OVID$_{-User}$. This configuration leads to a slight increase in recall (3.22 percent points) but a higher decrease in precision (12.89 percent points), which signals that the user features are especially beneficial for precision. For OVID$_{-Edits}$, we observe moderate losses on both recall and precision. We obtain similar results for OVID$_{-Changeset}$ results, i.e., a moderate difference in precision and recall of approximately 1.8 percent points. The loss on all metrics indicates the general usefulness of the edit and changeset information. Removing two feature categories simultaneously further reduces the accuracy compared to removing only one category, indicating that the categories capture different aspects of the OSM changesets and complement each other. For OVID$_{-User,Edits}$, we observe the lowest accuracy scores of 61.62%, whereas we still obtain 77.58% accuracy for OVID$_{-Changeset,Edits}$. The still moderate accuracy of OVID$_{-Changeset,Edits}$ highlights the user features' importance since this configuration solely relies on the user information.

Moving on to the OSM-Manual dataset, we obtain patterns similar to OSM-Reverts for the OVID$_{-Changeset}$ and OVID$_{-User}$. For OVID$_{-User}$, we observe a performance drop, especially regarding precision. In contrast to OSM-Reverts, OVID$_{-Edits}$ achieves an increased classification accuracy of 1.58 percent points. As discussed in Section 3.6.1, we expect that the underlying distributions of OSM-Reverts and OSM-Manual datasets can slightly differ. In particular, the median number of edits per vandalism changeset in OSM-Reverts is ten, while this number in OSM-Manual is only four, as shown in Table 3.2. Consequently, OVID cannot use the edit features to their full advantage on OSM-Manual, where less edit information is available. As a result, the edit features have a slight negative impact on the vandalism detection performance on the OSM-Manual dataset. Nevertheless, OVID still achieves the best performance in F1 score and accuracy compared to the baselines.

In summary, we observe that the changeset and user features provide valuable

Table 3.5. Vandalism detection performance with respect to precision, recall, F1 and accuracy of OVID when removing individual components. Best scores are marked bold.

(a) OSM-Reverts

Model	Precision	Recall	F1	Accuracy
OVID	**80.35**	83.02	**81.66**	**81.37**
OVID$_{-Changeset}$	78.42	81.33	79.85	79.49
OVID$_{-User}$	67.46	**86.24**	75.71	72.34
OVID$_{-Edits}$	78.68	78.98	78.83	78.81
OVID$_{-Changeset,Edits}$	75.69	81.22	78.36	77.58
OVID$_{-User,Edits}$	59.54	72.38	65.34	61.62

(b) OSM-Manual

Model	Precision	Recall	F1	Accuracy
OVID	69.86	**70.76**	70.31	70.12
OVID$_{-Changeset}$	59.03	49.55	53.88	57.58
OVID$_{-User}$	53.83	62.74	57.94	54.46
OVID$_{-Edits}$	**73.70**	67.49	**70.46**	**71.70**
OVID$_{-Changeset,Edits}$	66.94	55.8	60.86	64.12
OVID$_{-User,Edits}$	56.02	62.24	58.97	56.69

contributions for the vandalism detection task on both datasets. The edit features and their corresponding layers $X_{E'}$ are highly beneficial for the datasets with a higher number of edits per changeset like OSM-Reverts.

3.6.3 Precison/Recall Trade-off

This section discusses the trade-off between precision and recall in OVID. OVID can be customized to address the requirements of specific scenarios regarding precision and recall by adjusting the classification threshold th_{class}. th_{class} specifies the minimum activation value of OVID's last sigmoid prediction layer at which OVID classifies a changeset as vandalism.

Figure 3.3 presents the OVID's precision/recall diagram considering the OSM-Reverts and OSM-Manual datasets. At the most left, all changesets are classified as non-vandalism such that OVID achieves 100% precision but 0% recall. At the most right, all changesets are classified as vandalism, resulting in 50% precision and 100% recall. Note that the precision does not drop to 0% due to the class balance within the datasets.

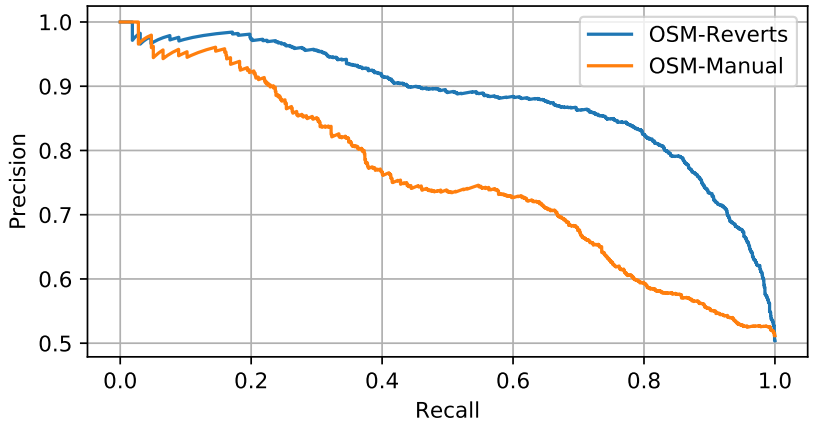


Figure 3.3. Precision/recall diagram of Ovid.

In general, we observe higher precision and recall scores on OSM-Reverts than on OSM-Manual. On OSM-Reverts, Ovid can obtain very high precision of 97% at the cost of only achieving 20% recall. Similarly, we obtain 92% precision at 20% recall for OSM-Manual. A high-precision configuration can potentially be used to run a fully automated vandalism detection system that blocks detected changesets directly. Lowering precision leads to a rapid increase in recall on OSM-Reverts. At 80% precision, Ovid already achieves 83% recall. For OSM-Manual, Ovid maintains approximately 70% precision at 70% recall. High recall of vandalism cases may be favorable to maintain high data quality, but may lead to accidental blocking of correct changesets (false positives). A high recall configuration of Ovid can be used to generate vandalism candidates in a human-in-the-loop approach, in which vandalism candidates are verified manually by the OSM community.

3.7 Discussion

In this chapter, we tackled the problem of validating geographic Web information by presenting an approach for vandalism detection in OpenStreetMap. Validation of OpenStreetMap is crucial since OSM is currently the most prominent source of volunteered geographic information and is widely adopted by many real-world applications. Whereas existing OSM vandalism detection approaches, such as [NGZ12], try to identify vandalism on the edit level, i.e., for isolated objects, we propose to consider vandalism on the changeset level. Considering whole changesets enables the identification of vandalism in composed OSM objects, such as ways and relations.

We proposed the Ovid (OpenStreetMap Vandalism Detection) model, a novel supervised machine learning approach for vandalism detection in OpenStreetMap. Ovid relies on a neural network architecture that adopts a multi-head attention mechanism to summarize information indicating vandalism from OpenStreetMap changesets ef-

fectively. To facilitate automated vandalism detection, we introduce a set of original features that capture changeset, user, and edit information.

To the best of our knowledge, there was no openly available dataset of OSM vandalism allowing for the training of supervised machine learning models before our work. Therefore, we constructed a new ground truth dataset for vandalism detection in OSM by systematically analyzing vandalism-related reverts in the OpenStreetMap history. Based on the reverts, we extract vandalism examples from the OpenStreetMap history resulting in a novel dataset containing over 18,000 vandalism examples.

Our experiments on real-world datasets demonstrate that Ovid can effectively detect OSM vandalism. Ovid achieves an F1 score of 75.99 % and an accuracy of 75.75% on average, which corresponds to 8.14 percent points increase in F1 score and a 5.41 percent point increase in accuracy compared to the best performing baselines. Furthermore, the experiments on the smaller, held-out OSM-Manual dataset show Ovid's capabilities to generalize to previously unseen data.

Ovid demonstrates the potential of machine learning models for validation of geographic information on the Web. We investigated different precision/recall configurations in detail in Section 3.6.3. Whereas the current precision does not allow for the fully autonomous deployment of Ovid, we believe there is enormous potential to substantially ease vandalism detection in a human-in-the-loop setting.

4 Enriching OpenStreetMap with Links to Knowledge Graphs

In this chapter, we approach the problem of enriching geographic information on the Web. Analogous to Chapter 3, we employ OpenStreetMap as an essential source for such information. A very effective way to enrich information is to integrate it with other information sources and thereby unlock large chunks of information provided by the additional sources at once. Knowledge graphs are rich sources of semantic information but are rarely integrated with OpenStreetMap. Therefore, we consider knowledge graphs as an additional information source and integrate them with OpenStreetMap by determining identity links between OSM objects and knowledge graph entities in this chapter. We present a supervised machine learning model for link discovery addressing **RQ2.1** that asks how to identify such links. Furthermore, **RQ2.2** asks how to capture the semantics of OSM objects despite OSM's heterogeneity. This chapter addresses this research question by introducing an unsupervised embedding model learning latent representations of OSM objects.

4.1 Introduction

Knowledge graphs (KGs), i.e., graph-based knowledge bases [FBMR18], including Wikidata [VK14], DBpedia [LIJ+15], YAGO [HSBW13] and EventKG [GD19] are a rich source of semantic information for geographic entities, including for example cities and points of interest (POIs). This information, typically represented according to the RDF data model, has a high and so far, mostly unexploited potential for semantic enrichment of OSM nodes. An interlinking of OSM nodes and geographic entities in knowledge graphs can bring semantic, spatial, and contextual information to its full advantage and facilitate, e.g., geographic question answering [PSB+18] and semantic trip recommendation [HSW19].

The interlinking of OSM and knowledge graphs has recently attracted interest in

the Wikidata[1] and OSM[2] communities. Our analysis results, presented in Section 4.3, illustrate that the coverage of the existing interlinking between the OSM nodes and Wikidata entities varies significantly across entity types and geographic regions. For example, in a recent OSM snapshot of Germany (referred to as `OSM-DE`), cities are linked more often (73%) than less popular entities like mountains (5%). For another example, there are 42% more linked OSM nodes in the OSM snapshot of Germany than in that of Italy (`OSM-IT`). In practice, the interlinking of OSM nodes with semantic reference sources such as Wikidata or DBpedia is typically conducted manually by volunteers (and sometimes companies, see, e.g., [Gan16]).

The problem of OSM link discovery is particularly challenging due to the heterogeneity of the OSM node representations. Other factors affecting the effectiveness of OSM node disambiguation in the context of link discovery include place name ambiguity and limited context [GPLC18]. Furthermore, geographic coordinates in the VGI sources such as OSM often represent the points of community consensus rather than being determined by objective criteria [SLHA12] and can thus vary significantly across sources. For example, the average geographic distance between the coordinates of the corresponding entities in Germany in the OSM and Wikidata datasets is 2517 meters. This example illustrates that geographic coordinates alone are insufficient to effectively discover identity links between the corresponding entities in VGI sources.

Although research efforts such as the LinkedGeoData project [SLHA12] and Yago2Geo [KMK19] have been conducted to lift selected parts of OSM data in the Semantic Web infrastructure to facilitate link discovery, these efforts typically rely on manually defined schema mappings. Maintenance of such mappings does not appear feasible or sustainable, given the large scale, and openness of the OSM schema. Therefore, link discovery approaches that can address the inherent heterogeneity of OSM datasets are required.

In this chapter, we propose the novel OSM2KG link discovery approach to establish identity links between the OSM nodes and equivalent geographic entities in a knowledge graph. OSM2KG addresses OSM's heterogeneity problem through a novel latent representation of OSM nodes inspired by the word embedding architectures [MSC+13]. Whereas embeddings have recently gained popularity in several domains, their adoption to volunteered geographic information in OSM is mostly unexplored. In contrast to state-of-the-art approaches to link discovery in OSM (such as [KMK19, SLHA12]), OSM2KG does not require any schema mappings between OSM and the reference knowledge graph.

The core of the OSM2KG approach is a novel latent representation of OSM nodes that captures semantic node similarity in an embedding. OSM2KG learns this latent, compact node representation automatically from OSM tags. To the best of our knowledge, OSM2KG is the first approach to address the heterogeneity of the OSM data by a novel embedding representation. This embedding representation is created

[1] https://www.wikidata.org/wiki/Wikidata:OpenStreetMap
[2] https://wiki.openstreetmap.org/wiki/Proposed_features/Wikidata

in an unsupervised fashion and is task-independent. The embedding systematically exploits the co-occurrence patterns of the OSM's key-value pairs to capture their semantic similarity. Building upon this embedding, along with spatial and semantic information in the target knowledge graph, OSM2KG builds a supervised machine learning model to predict missing identity links. To train the proposed link prediction model, we exploit publicly available community-created links between OSM, Wikidata, and DBpedia as training data.

Contributions. In this chapter, we address **RQ2.1** and **RQ2.2**. The key contribution of this chapter is the novel OSM2KG link discovery approach to infer missing identity links between OSM nodes and geographic entities in knowledge graphs, including:

- A novel unsupervised embedding approach to infer latent, compact representations that capture semantic similarity of heterogeneous OSM nodes.
- A supervised classification model to effectively predict identity links, trained using the proposed latent node representation, selected knowledge graph features, and existing links.
- We describe an algorithm for link discovery in the OSM datasets that uses the proposed supervised model and the latent representation to effectively identify missing links.
- We evaluate the proposed approach on three real-world OSM datasets for different geographic regions, along with the Wikidata and DBpedia knowledge graphs. OSM2KG achieves an F1 score of 92.05% on Wikidata and of 94.17% on DBpedia on average, which corresponds to a 21.82 percentage points increase in F1 score on Wikidata compared to the best performing baselines.

The remainder of this chapter is organized as follows. In Section 4.2, we discuss related work relevant to OSM2KG. In Section 4.3, we motivate our approach by discussing the representation of geographic information in OSM and Wikidata and the existing interlinking between these sources. Then, in Section 4.4, we formally introduce the link discovery problem addressed in this chapter. In Section 4.5, we present the proposed OSM2KG approach. Following that, we describe the evaluation setup in Section 4.6 and provide and discuss our evaluation results in Section 4.7. Finally, we provide a discussion in Section 4.8.

4.2 Related Work

In this section, we discuss related work in the areas of link discovery, entity linking, linking geographic data, and geospatial link discovery.

Link Discovery. is the task of identifying semantically equivalent resources in different data sources [NHNR17]. Nentwig et al. [NHNR17] provide a recent survey of link discovery frameworks, with prominent examples, including Silk [VBGK09] and LIMES [NA11].

In particular, the Wombat algorithm, integrated within the LIMES framework [SNL17], is a state-of-the-art approach for link discovery in knowledge graphs. Link discovery approaches that operate on Linked Data typically expect datasets in Resource Description Framework (RDF) format, having a schema defined by an underlying ontology and data exhibiting graph structure. This assumption does not apply to the OSM data represented as key-value pairs.

Besides the syntactic and structural differences, LIMES relies on several assumptions that severely limit its applicability to OSM datasets. First, LIMES assumes a one-to-one mapping between properties. In contrast, the required mappings between the Wikidata properties and the OSM keys are 1:n, as a Wikidata property can correspond to several OSM keys. For example, the "instanceOf" property in Wikidata corresponds to "place," "natural," "historic," and many other keys in OSM. Second, LIMES requires all instances to contain all considered properties. Therefore, LIMES is limited to utilize only frequently used properties, such as the name and the geo-coordinates. To this end, LIMES is not suited to utilize the information from other infrequent properties for mapping. Finally, the current LIMES implementation does not adequately support a combination of different data types, such as strings and geo-coordinates. Given these differences, the application of LIMES to the OSM data is de-facto restricted to the name matching. We utilize Wombat/LIMES as a baseline for the evaluation.

In the context of individual projects such as LinkedGeoData and Yago2Geo [SLHA12, KMK19], a partial transformation of OSM data to RDF was conducted using manually defined schema mappings for selected keys. In contrast, the proposed OSM2KG approach adopts an automatically generated latent representation of OSM data.

Entity linking. (also referred to as entity disambiguation) is the task of linking mentions of real-world entities in unstructured sources (e.g., text documents) to equivalent entities in a knowledge base. A recent survey on entity linking approaches is provided in [SWH15]. Entity linking approaches typically adopt Natural Language Processing (NLP) techniques and use the context of the entity mentions, such as phrases or sentences. However, such a context is not available in OSM, where textual information is mainly limited to node labels (typically available as a specialized name tag). One of the most popular state-of-the-art models to automatically annotate mentions of DBpedia entities in natural language text is *DBpedia Spotlight* [DJHM13]. DBpedia Spotlight adopts NLP techniques to extract named entities (including locations) from text and uses a context-aware model to determine the corresponding DBpedia entities. This approach serves as a baseline in our experiments, whereas we use the name tag of an OSM node as its textual representation.

Linking geographic data. The most relevant projects in the context of our work

are LinkedGeoData [SLHA12] and Yago2Geo [KMK19]. LinkedGeoData is an effort to lift OSM data into semantic infrastructure. This goal is addressed through deriving a lightweight ontology from the OSM tags and transforming OSM data to the RDF data model. LinkedGeoData interlinks OSM nodes represented as RDF with geo-entities in external knowledge sources such as DBpedia and GeoNames. Yago2Geo aims at extending the knowledge graph YAGO2 [HSBW13] with geographic knowledge from external data sources. To this extent, identity links between YAGO2 and OSM are computed. Both interlinking approaches rely on manually defined schema mappings and heuristics based on name similarity and geographic distance. The dependence of both approaches on manual schema mappings restricts the coverage of mapped entity types and can also negatively affect link maintenance. In contrast, the OSM2KG approach proposed in this chapter extracts latent representations of OSM nodes fully automatically. The LinkedGeoData and Yago2Geo interlinking approaches serve as baselines in our experiments.

The applications of linked geographic data include, for example, the training of comprehensive ranking models [DA16] or the creation of linked data based gazetteers [CAS+16].

Geospatial link discovery. [SFFN17, ASN18, SDSN17, SK16] refers to the problem of creating topological relations across geographic datasets. These links express the topographic relations between entities (e.g., "intersects" and "overlaps"). For example, [SK16] presented the problem of discovery of spatial and temporal links in RDF datasets. In Radon [SDSN17], efficient computation of topological relations between geospatial resources in the datasets published according to the Linked Data principles was presented. In contrast, in this work, we focus on link discovery for identity links.

4.3 Motivation

While tags in OSM primarily describe the semantics of OSM objects, the tags can also be used to specify identity links across datasets, e.g., to link OSM nodes to the equivalent entities in a knowledge graph. For example, the link between the OSM node representing the city of Berlin and its Wikidata counterpart is established via the tag "`wikidata=Q64`" assigned to the OSM node. Here, "Q64"[3] denotes the identifier of the corresponding Wikidata entity. Recent studies indicate that the level of details provided for the individual OSM nodes is very heterogeneous [TR15]. Contextual information, e.g., regarding the historical development of the city population, is typically not available in OSM. Furthermore, the individual keys and tags do not possess any machine-readable semantics, which further restricts their use in applications.

Country-specific OSM snapshots are publicly available[4]. In the following, we

[3] https://www.wikidata.org/wiki/Q64

[4] OSM snapshots can be found at http://download.geofabrik.de.

Table 4.1. Number of nodes, tags and distinct keys in the country-specific OSM snapshots (`OSM-[country]`) and their respective subsets linked to Wikidata (`Wikidata-OSM-[country]`).

(a) France

	OSM-FR	Wikidata-OSM-FR	Ratio
No. Nodes	390,586,064	21,629	0.01%
No. Nodes with Name	1,229,869	20,507	1.67%
No. Tags	27,398,192	199,437	0.73%
No. Distinct Keys	6,009	1,212	20.17%

(b) Germany

	OSM-DE	Wikidata-OSM-DE	Ratio
No. Nodes	289,725,624	24,312	$< 0.01\%$
No. Nodes with Name	1,681,481	23,979	1.43%
No. Tags	37,485,549	212,727	0.56%
No. Distinct Keys	12,392	1,700	13.72%

(c) Italy

	OSM-IT	Wikidata-OSM-IT	Ratio
No. Nodes	171,576,748	18,473	0.01%
No. Nodes with Name	557,189	18,420	3.31%
No. Tags	18,850,692	122,248	0.65%
No. Distinct Keys	4,349	892	20.51%

refer to the country-specific snapshots as of September 2018 as the `OSM-[country]` dataset. For instance, the snapshot for Germany is referred to as "`OSM-DE`". The linked sets `Wikidata-OSM-FR`, `Wikidata-OSM-DE`, and `Wikidata-OSM-IT` are the subsets of the `OSM-[country]` datasets obtained by extracting all nodes that link to Wikidata entities from the respective OSM snapshot. Table 4.1 provides an overview of the number of nodes, nodes with name, tags, and distinct key contained in the `OSM-[country]` datasets and the respective linked sets `Wikidata-OSM-[country]`. As we can observe, only a small fraction of nodes, tags, and distinct keys from the overall datasets appear in the linked sets. Furthermore, nearly all nodes contained in one of the linked sets exhibit a name tag.

Knowledge graphs such as Wikidata [VK14], DBpedia [LIJ+15], and YAGO [HSBW13] are a rich source of contextual information about geographic entities, with Wikidata currently being the largest openly available knowledge graph linked to OSM. In September 2018, Wikidata contained more than 6.4 million entities for which geographic coordinates are provided. Overall, the geographic information in OSM and contextual information regarding geographic entities in the existing knowledge graphs are highly complementary. As an immediate advantage of the existing effort to manually interlink OSM nodes and Wikidata entities, the names of the linked OSM nodes

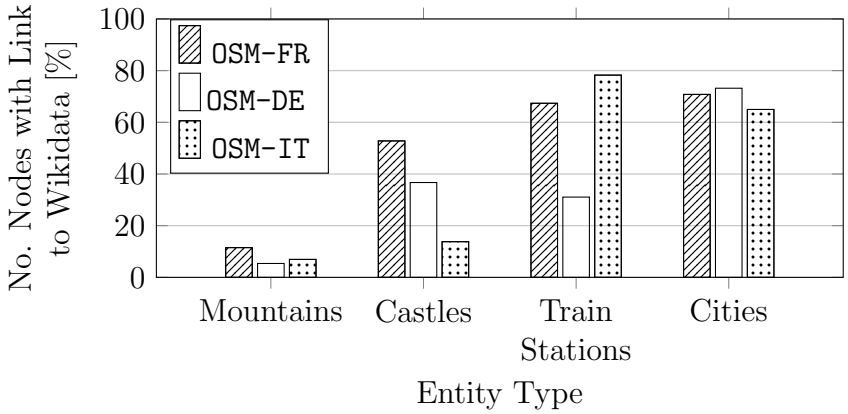
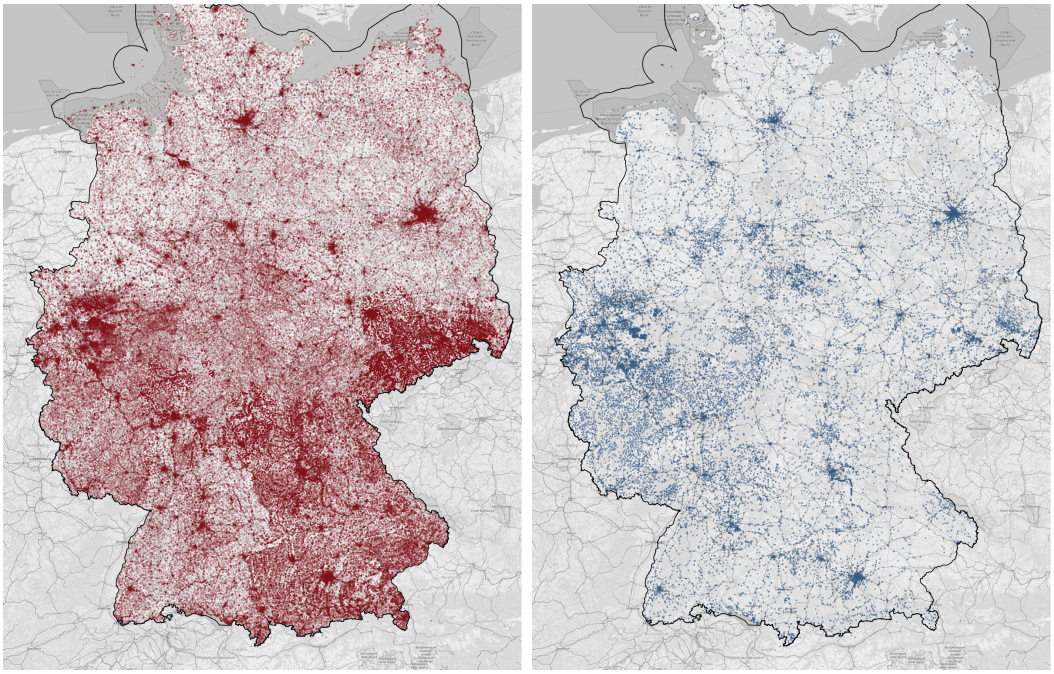


Figure 4.1. Percentage of frequent OSM node types with links to Wikidata entities within the OSM datasets for Germany (`OSM-DE`), France (`OSM-FR`), and Italy (`OSM-IT`) as of September 2018.

have become available in many languages [Gan16].

The links between the OSM nodes and geographic entities in Wikidata are typically manually annotated by volunteers and community efforts and are still only rarely provided. Figure 4.1 illustrates the percentage of the four most frequent geographic entity types (i.e., cities, train stations, mountains, and castles) that link to Wikidata from the OSM datasets for Germany, France, and Italy. Here, entity types are obtained from Wikidata using existing links between the OSM nodes and Wikidata entities. As we can observe, the cities are linked most frequently, with a link coverage of approximately 70% for all datasets. The link coverage of the other entity types is significantly lower, with mountains having the smallest coverage across these four categories with approximately 5% in Germany. Figure 4.2 provides a visual comparison of the number of Wikidata entities located in Germany and the number of Wikidata entities to which links from OSM exist. While a significant fraction of links is still missing, existing links manually defined by volunteers reveal a high potential for being used as training data for supervised machine learning to increase link coverage automatically.

In summary, volunteered geographic information is a continually evolving large-scale source of heterogeneous spatial data, whereas knowledge graphs provide complementary, contextual information for geographic entities. The links between VGI and knowledge graphs are mainly manually specified and are still only rarely present in the OSM datasets. The existing links represent a valuable source of training data for supervised machine learning methods to automatically increase the link coverage between OSM and knowledge graphs. This interlinking can provide a rich source of openly available semantic, spatial, and contextual information for geographic entities.



(a) Wikidata (b) `Wikidata-OSM-DE`

Figure 4.2. Wikidata geo-entities located within Germany and Wikidata geo-entities linked by OSM. Map image: ©OpenStreetMap contributors, ODbL.

4.4 Problem Statement

In this chapter, we target the problem of identity link discovery between the nodes in a semi-structured geographic corpus such as OSM with equivalent entities in a knowledge graph. To this end, we extend the OSM data model described in Section 2.2.1.

Definition 4.1. ***Knowledge graph:*** *Let E be a set of entities, R a set of labelled directed edges and L a set of literals. A* knowledge graph $\mathcal{KG} = \langle E \cup L, R \rangle$ *is a directed graph where entities in E represent real-world entities and the edges in $R \subseteq (E \times E) \cup (E \times L)$ represent entity relations or entity properties.*

In this work, we focus on interlinking entities in a knowledge graph that possess geographic coordinates, i.e., longitude and latitude. We refer to such entities as *geo-entities*. Typical examples of geo-entities include cities, train stations, castles, and others.

Definition 4.2. ***Geo-entity:*** *A* geo-entity $e \in E$ *is an entity for which a relation $r \in R$ exists that associates e with geographic coordinates, i.e., a longitude $lon \in L$ and a latitude $lat \in L$.*

For instance, a geo-entity representing the city of Berlin may be represented as follows (the example illustrates an excerpt from the Wikidata representation of Berlin):

Entity	**Property**	**Entity/Literal**
$Q64$	*name*	*Berlin*
$Q64$	*instance of*	*Big City*
$Q64$	*coordinate location*	$52^{1\prime}N,\ 13^{3\prime}E$
$Q64$	*capital of*	*Germany*

We denote the subset of nodes representing geo-entities in the knowledge graph $\mathcal{KG}$ as $E_{geo} \subseteq E$.

Definition 4.3. ***Geographic corpus:*** *A* geographic corpus $\mathcal{C}$ *is a set of OSM nodes, i.e. OSM objects (defined in Section 2.2.1) with type Node.*

A node $n \in \mathcal{C}$, $n = \langle id, \texttt{Node}, loc, tags, ver \rangle$ *contains an identifier (id), a location (loc), a set of tags (tags), and version number (ver). Each tag* $t \in tags$ *is represented as a key-value pair with the key* k *and a value* v*:* $t = \langle k, v \rangle$.

For instance, the city of Berlin is represented as follows (the example illustrates an excerpt from the OSM representation):

id	240109189
loc	52.5170365, 13.3888599
type	`Node`
version	137
`name=`	*Berlin*
`place=`	*city*
`capital=`	*yes*

Let $sameAs(n, e) : \mathcal{C} \times E_{geo} \mapsto \{true, false\}$ be the predicate that holds iff $n \in \mathcal{C}$ and $e \in E_{geo}$ represent the same real-world entity. We assume that a node $n \in \mathcal{C}$ corresponds to at most one geo-entity in a knowledge graph $\mathcal{KG}$. Then the problem of link discovery between a knowledge graph $\mathcal{KG}$ and a geographic corpus $\mathcal{C}$ is defined as follows.

Definition 4.4. ***Link discovery:*** *Given a node* $n \in \mathcal{C}$ *and the set of geo-entities* $E_{geo} \subseteq E$ *in the knowledge graph* $\mathcal{KG}$*, determine* $e \in E_{geo}$ *such that* $sameAs(n, e)$ *holds.*

In the example above, given the OSM node representing the city of Berlin, we aim to identify the entity representing this city in E_{geo}.

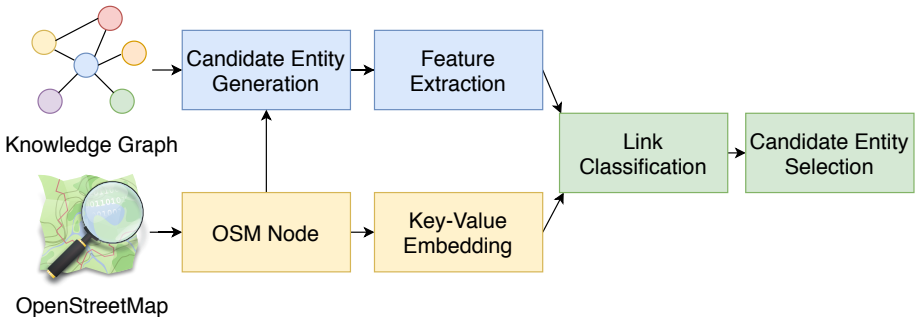


Figure 4.3. OSM2KG Link discovery pipeline overview.[5]

4.5 OSM2KG Approach

The intuition of the proposed OSM2KG approach is as follows:

1. Equivalent nodes and entities are located in geospatial proximity. Therefore, OSM2KG adopts geospatial blocking to identify candidate entities in large-scale datasets efficiently.

2. OSM nodes are schema-agnostic and heterogeneous. Therefore, OSM2KG relies on an unsupervised model to infer latent, compact node representation that captures semantic similarity.

3. Equivalent nodes and entities can indicate common representation patterns. Therefore, OSM2KG adopts a supervised classification model for link prediction.

Figure 4.3 presents the OSM2KG link discovery pipeline. First, in the blocking step, for each node $n \in \mathcal{C}$ in the geographic corpus $\mathcal{C}$, a set of candidates $E' \subseteq E_{geo}$ is generated from the set of geo-entities E_{geo} contained in the knowledge graph. In the next feature extraction step, representations of the node n and the relevant entities E' from the knowledge graph are extracted. A latent representation of the node $n \in \mathcal{C}$ is a *key-value embedding* that is learned in an unsupervised fashion. Representations of the knowledge graph entities in E' are generated using selected knowledge graph features. Furthermore, distance and similarity metrics for each candidate pair ($n \in \mathcal{C}$, $e \in E'$) are computed. Following that, each candidate pair is processed by a supervised machine learning model during the link classification step. The model predicts if the pair represents the same real-world entity and provides a confidence score for the link prediction. Finally, an identity link for the pair with the highest confidence among the positively classified candidate pairs for the node n is generated. In the following, we discuss these steps in more detail.

[5]The OSM logo is a trademark of the OpenStreetMap Foundation, and is used with their permission. We are not endorsed by or affiliated with the OpenStreetMap Foundation.

4.5.1 Candidate Entity Generation

Representations of a real-world geographic entity in different data sources may vary; this can be especially the case for the geographic coordinates in VGI, where the reference points represent typical points of community consensus rather than an objective metric [SLHA12]. The blocking step is based on the intuition that geographic coordinates of the same real-world entity representation in different sources are likely to be in a short geographic distance.

Given a node $n \in \mathcal{C}$ contained in a geographic corpus and a knowledge graph $\mathcal{KG} = \langle E \cup L, R \rangle$, with a set of geo-entities $E_{geo} \subseteq E$, in the blocking step we compute a set of candidate geo-entities $E' \subseteq E_{geo}$ from $\mathcal{KG}$, i.e., the geo-entities potentially representing the same real-world entity as n.

The set of candidates E' for a node n consists of all geographic entities $e \in E_{geo}$ that are in a short geographic distance to n. In particular, we consider all entities within the distance specified by the blocking threshold th_{block}:

$$E' = \{e \in E_{geo} \mid distance(n, e) \leq th_{block}\},$$

where $distance(n, e)$ is a function that computes the geographic distance between the node n and a geo-entity e. Here the geographic distance is measured as *haversine distance* [KK03].

Note that E' can be computed efficiently by employing spatial index structures such as R-trees [Gut84]. The value of the threshold th_{block} can be determined experimentally (see Section 4.7.5).

4.5.2 Key-Value Embedding for the Geographic Corpus

In this work, we propose an unsupervised approach to infer novel latent representations of nodes in a geographic corpus. This representation aims at capturing the semantic similarity of the nodes by utilizing typical co-occurrence patterns of OSM tags. Our approach is based on the intuition that semantic information, like for example entity types, can be inferred using statistical distributions [PB14]. To realize this intuition in the context of a geographic corpus such as OSM, we propose a neural model inspired by the skip-gram model for word embeddings by Mikolov et al. [MSC+13]. This model creates latent node representations that capture the semantic similarity of the nodes by learning typical co-occurrences of the OSM tags.

In particular, we aim to obtain a latent representation of the node $n = \langle id, \texttt{Node}, loc, tags, ver \rangle, n \in \mathcal{C}$ that captures the semantic similarity of the nodes. To this extent, we propose a neural model that encodes the set of key-value pairs T describing the node in an embedding representation. Figure 4.4 depicts the architecture of the adopted model that consists of an input, a projection, and an output layer. The *input layer* encodes the identifier $n.id$ of each node n. In particular, vector representations are obtained by applying one-hot-encoding of the identifiers, i.e., each

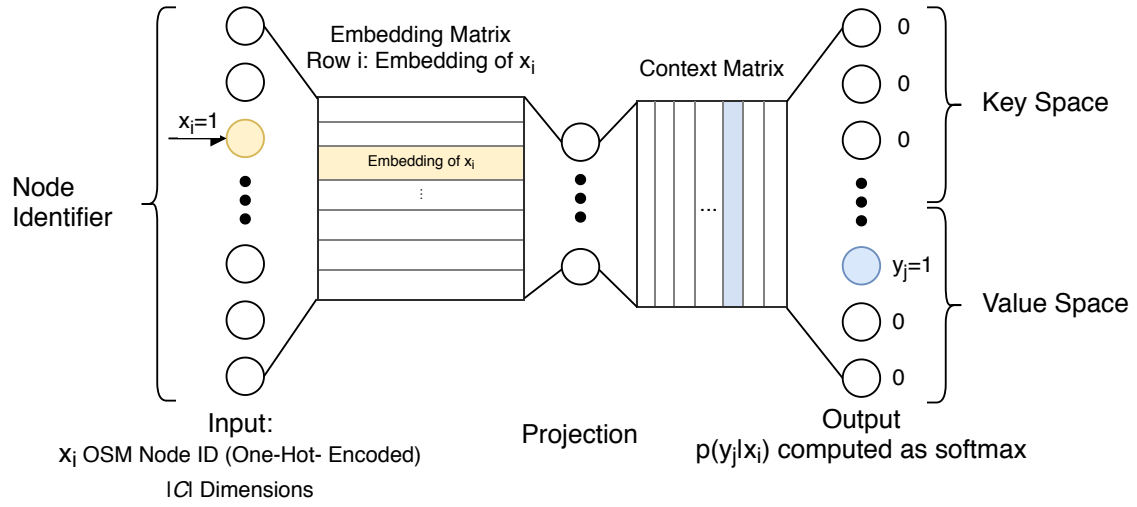


Figure 4.4. Architecture of the key-value embedding model. The input layer 1-hot encodes the node identifiers. The embedding matrix transforms the input to the latent representation in the projection layer. The output layer maps the latent representation to the encoded keys and values by applying the softmax function.

identifier $n.id$ corresponds to one dimension of the input layer. The corresponding entry of the vector representation is set to 1, while other entries are set to 0. The *projection layer* computes the latent representation of the nodes. The number of neurons in this layer corresponds to the number of dimensions in the projection, i.e., the embedding size. The *output layer* maps the latent representation to the encoded keys and values using softmax [GBC16]. The key-value pairs $\langle k, v\rangle \in n.tags$ for each node n are encoded by applying one-hot-encoding to both keys and values separately. As the set of values might be highly diverse, we only consider the top-k most frequent values to be represented as an individual dimension. The non-frequent values are unlikely to be indicative for semantic similarity, whereas the information of the presence of a rare value can be discriminative. Thus, all non-frequent values are mapped to a single dimension.

The embedding aims to generate a similar representation for the nodes with similar properties, independent of their location. Therefore, we do not include location information, such as geographic coordinates, in the embedding. Note that the value of name tags are typically not part of the embedding, as names typically have rare values.

The objective of the proposed model is to maximize the following log probability:

$$\frac{1}{|C|}\sum_{n\in\mathcal{C}}\sum_{\langle k,v\rangle\in n.tags}\log p(k|n.id)+\log p(v|n.id).$$

Here, the term $\log p(k|n.id)+\log p(v|n.id)$ expresses the node's log probability with the identifier $n.id$ to be annotated with the key-value pair $\langle k, v\rangle$, i.e. $\langle k, v\rangle \in n.tags$.

The probabilities are calculated using softmax. The training of the network aims at minimizing the key-value based loss function. This way, nodes that exhibit similar keys or values are assigned similar representations in the projection layer. Thus, we use the activation of the projection layer as a latent representation of each respective OSM node. This representation captures the latent semantics of the keys and values of the node. We refer to this feature as *KV-embedding*. We learn the *KV-embedding* for each OSM node. The training is conducted without any supervision. The resulting node representation is task-independent.

4.5.3 Feature Extraction from KG

This step aims at extracting features for the entities $e \in E'$, where E' denotes the set of candidate geo-entities in the knowledge graph for the target node $n \in \mathcal{C}$. We adopt the following features:

Entity Type: Entities and nodes that belong to the same category, for instance "city" or "train station", are more likely to refer to the same real-world entity than the candidates of different types. In the knowledge graph, we make use of the *rdf:type*[6] property as well as knowledge graph specific properties (e.g. *wikidata:instanceOf*) to determine the type of e. To encode the type, we create a vector of binary values in which each dimension corresponds to an entity type. For each type of e, the corresponding dimension is set to "1" while all other dimensions are set to "0". Concerning the target node n, the node type is not expected to be explicitly provided in a geographic corpus. Nevertheless, we expect that the *KV-embedding* of the geographic corpora implicitly encodes type information, based on the intuition that types can be inferred using statistical property distributions [PB14].

Popularity: A similar level of entity popularity in the respective sources can provide an indication for matching. Popular entities are likely to be described with a higher number of relations and properties than less popular entities. To represent entity popularity, we employ the number of edges starting from e in $\mathcal{KG}$ as a feature. More formally: $popularity(e) = |\{(e, x) \in R \mid x \in E \cup L\}|$. We expect that the *KV-embedding* implicitly encodes the node popularity information in the geographic corpora as popular nodes have a higher number of tags.

4.5.4 Similarity and Distance Metrics

This step aims at extracting features that directly reflect the similarity between an OSM node $n \in \mathcal{C}$ and a candidate geo-entity $e \in E'$. To this extent, we utilize name similarity and geographical distance.

Name Similarity: Intuitively, a geo-entity and an OSM node sharing the same name are likely to represent the same real-world object. Therefore, we encode the

[6] rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns

similarity between the value of the *name* tag of an OSM node $n \in \mathcal{C}$ and the *rdfs:label*[7] of a geo-entitiy $e \in E'$ as a feature. We compute the similarity using the Jaro-Winkler distance [Win99], also adopted by [SLHA12]. The Jaro-Winkler distance assigns a value between [0,1], where 0 corresponds to no difference and 1 to the maximum dissimilarity. If a *name* tag or a *rdfs:label* is not available for a particular pair (n, e), the value of this feature is set to 1.

Geo-Distance: Based on the intuition that nodes and candidate entities that exhibit smaller geographic distance are more likely to refer to the same real-world entity, we employ geographic distance as a feature. To this extent, we utilize the logistic distance function proposed in [SLHA12]:

$$\textit{geo-distance}(n, e) = 1/(1 + exp(-12d'(n, e) + 6)),$$

with $d' = 1 - d(n, e)/th_{block}$, where d denotes the so-called *haversine distance* [KK03] between n and e and takes the spheroid form of the earth into account. th_{block} denotes the threshold that defines the maximum geographic distance at which the candidates are considered to be similar. To facilitate efficient computation, the th_{block} threshold is also utilized in the blocking step, described in Section 4.5.1. The intuition behind the logistic distance function is to allow for smaller differences of the geographic positions and to punish more significant differences. The Geo Distance feature directly encodes the geospatial similarity between the node n and the candidate geo-entity e.

4.5.5 Link Classification

We train a supervised machine learning model to predict whether the target node $n \in \mathcal{C}$ and a candidate geo-entity represent the same real-world entity. Each target node n and the set of candidates E' for this node are transformed into the feature space. Each node-candidate pair is interpreted as an instance of a supervised machine learning model by concatenating the respective feature vectors. For training, each pair is then labelled as correct or incorrect, where labels are obtained from the existing links to the knowledge graph within the OSM corpus $\mathcal{C}$. Note that the number of pairs labelled as incorrect (i.e., negative examples) is typically higher than the number of correct pairs. To allow an efficient training of classification models, we limit the number of incorrect candidates for each node n to 10 candidates via random sampling. To address the imbalance of classes within the training data, we employ oversampling to level out the number of instances per class. In particular, we employ the state-of-the-art SMOTE algorithm [CBHK02]. The data is then normalized by removing the mean and scaling to unit variance. We use the normalized data as input to the classification model. We consider the following models: Random Forest, Decision Tree, Naïve Bayes, and Logistic Regression. We discuss the model performance in Section 4.7.3. We optimize the hyperparameters using random search [BB12].

[7] rdfs: http://www.w3.org/2000/01/rdf-schema

Finally, the candidate entity selection is based on the assumption that the knowledge graph contains at most one geo-entity equivalent to the target node. If at least one node within E' is classified as correct (with a confidence $> 50\%$), a link between node n and $e_{max} \in E'$ is created, where e_{max} denotes the entity with the highest confidence score of the model. If all entities are labelled as incorrect, no link for the node n is created.

4.5.6 Algorithm for Link Discovery

Finally, Algorithm 1 details the process of link discovery. The algorithm integrates the above described steps, namely *candidate entity generation* (line 1), *feature extraction* (lines 2-7), *link classification* (lines 8-11) and *candidate entity selection* (lines 12-16). Table 4.2 presents a description of the functions used in the algorithm.

Algorithm 1: Link Discovery

Input : Node $n \in \mathcal{C}$
Knowledge graph $\mathcal{KG}$

Output: Entity $e_{link} \in \mathcal{KG}$ that should be linked to n `null` if no matching entity was found

```
1  E' ← generateCandidates(n, KG)
2  features ← []
3  features[n] ← KV-embedding(n)
4  forall e ∈ E' do
5  |  features[e] ← KG-features(e, KG)
6  |  features[e] ← features[e] ∪ similarity-features(e, n)
7  end
8  confidences ← []; forall e ∈ E' do
9  |  confidences[e] ← link-classification(features[n], features[e])
10 end
11 e_link ← argmax_{e∈E'}(confidences[e])
12 if classifiedAsCorrect(e_link) then
13 |  return e_link
14 else
15 |  return null
16 end
```

Table 4.2. Description of functions used in Algorithm 1.

Function Name	Returned Result	Section
`generateCandidates`	Candidate entities from $\mathcal{KG}$ nearby n	4.5.1
`KV-embedding`	Latent representation of n	4.5.2
`KG-features`	Feature representation for e	4.5.3
`similarity-features`	Similarity features between e and n	4.5.4
`link-classification`	Confidence score for (n, e)	4.5.5
`classifiedAsCorrect`	True iff a link between (n, e) is classified to be correct	4.5.5

4.5.7 Implementation

In this section, we provide implementation details of the OSM2KG components. We implemented our overall experimental framework and the proposed algorithm in Java 8. We stored the evaluation results in a PostgreSQL[8] database (version 9.6). In a pre-processing step, we extracted relevant data from OpenStreetMap using Python (version 3.6) and the osmium[9] library (version 2.14). We extracted relevant knowledge graph entities from Wikidata with geographic coordinates using pyspark[10] (version 2.2). The geographic data was stored in a PostgreSQL database (version 9.6) and indexed using the PostGIS[11] extension (version 2.3). The feature extraction is implemented in Java 8 within our experimental framework. We implemented the extraction of the KV-embedding in Python 3.6, using Tensorflow[12] version 1.14.1. The machine learning algorithms were implemented in Python 3.7 using the scikit-learn[13] (version 0.21) and the imbalanced-learn[14] (version 0.5) libraries. To facilitate the reproducibility, we make our code available under the open MIT license in a GitHub repository[15].

4.6 Evaluation Setup

In this section, we describe the datasets, metrics, baselines and OSM2KG configurations utilized in the evaluation.

[8] https://www.postgresql.org/
[9] https://osmcode.org/libosmium/
[10] https://spark.apache.org/docs/latest/api/python/pyspark.html
[11] https://postgis.net/
[12] https://www.tensorflow.org/
[13] https://scikit-learn.org/stable/
[14] https://imbalanced-learn.readthedocs.io/en/stable/api.html
[15] https://github.com/NicolasTe/osm2kg

Table 4.3. The number of geographic entities, distinct types and average statements per geo-entity in the considered knowledge graphs.

Knowledge Graph	No. Geo-Entities	No. Distinct Types	Average No. Edges/Entity
Wikidata	6,465,081	13,849	24.69
DBpedia-FR	317,500	185	18.33
DBpedia-DE	483,394	129	31.60
DBpedia-IT	111,544	11	31.13

4.6.1 Datasets and Metrics

We conduct our evaluation on three large-scale OSM datasets for France, Germany, and Italy as well as the Wikidata and DBpedia knowledge graphs.

Knowledge Graphs: In our experiments, we consider the Wikidata snapshot from September 2018, as well as DBpedia in its German, French and Italian editions, snapshots from August 2019, as the target knowledge graphs. *Wikidata* [VK14] is a publicly available collaborative knowledge graph. Wikidata is the central repository for structured information of the Wikimedia Foundation and the currently largest openly available knowledge graph. *DBpedia* [LIJ+15] is a knowledge graph that extracts structured data from the information of various Wikimedia projects, e.g., the Wikipedia[16] encyclopedia. DBpedia is provided in language-specific editions. We refer to each language-specific edition of DBpedia as `DBpedia-[language]`. Table 4.3 presents the number of available geographic entities as well as the number of distinct types and the average number of edges per geo-entity in each knowledge graph. Note that we consider geo-entities in the knowledge graphs with valid geographic coordinates, i.e., coordinates that can be located on the globe.

OpenStreetMap: We consider OSM datasets extracted from the three largest country-specific OSM snapshots as of September 2018. In particular, we consider the snapshots of Germany, France, and Italy. We denote the country-specific snapshots as `OSM-[country]`. Furthermore, we extract all nodes that exhibit a link to a geo-entity contained in Wikidata or DBpedia. For DBpedia, we consider links to the DBpedia version of the language that corresponds to the country of the individual OSM snapshot, since the existing links in the country-specific snapshots target the respective language-specific edition of DBpedia in all cases for the considered datasets. We denote the considered link datasets as `[KG]-OSM-[language]`. For instance, `DBpedia-OSM-FR` denotes the dataset that interlinks the OSM snapshot of France with the French DBpedia.

Table 4.4 provides an overview of the number of existing links between OSM and the knowledge graphs. The existing links between the OSM datasets and knowledge

[16] https://www.wikipedia.org

Table 4.4. The number of existing links between OpenStreetMap, Wikidata and DBpedia. `OSM-[country]` denote the country-specific snapshots of OSM as of September 2018. The existing links serve as ground truth for the experimental evaluation.

Knowledge Graph	OSM-FR	OSM-DE	OSM-IT
Wikidata	21,629	24,312	18,473
DBpedia-FR	12,122	-	-
DBpedia-DE	-	16,881	-
DBpedia-IT	-	-	2,353

graphs in these link datasets serve as ground truth for the experimental evaluation of all link discovery approaches considered in this chapter.

To assess the performance of link discovery approaches, we compute the following metrics:

Precision: The fraction of the correctly linked OSM nodes among all nodes assigned a link by the considered approach.

Recall: The fraction of the OSM nodes correctly linked by the approach among all nodes for which links exist in the ground truth.

F1 score: The harmonic mean of recall and precision. In this chapter, we consider the F1 score to be the most relevant metric since it reflects both recall and precision.

We apply the 10-fold cross-validation. We obtain the folds by random sampling the links from the respective link datasets. For each fold, we train the classification model on the respective training set. We report the macro average over the folds of each metric.

4.6.2 Baselines

We evaluate the link discovery performance of OSM2KG against the following unsupervised and supervised baselines:

BM25: This naive baseline leverages the standard BM25 text retrieval model [MRS08] to predict links. We created an inverted index on English labels of all geo-entities (i.e., for all $e \in E_{geo}$) in a pre-processing step to apply this model. Given the target node n, we query the index using the value of the name tag of n to retrieve geo-entities with similar labels. We query the index using either the English name tag of the node n (if available) or the name tag without the language qualifier. We create the link between n and the entity with the highest similarity score returned by the index. If the name tag is not available, we do not create any link.

SPOTLIGHT: This baseline employs the *DBpedia Spotlight* [DJHM13] model to determine the links. *DBpedia Spotlight* is a state-of-the-art model to perform entity linking, i.e., to link named entities mentioned in the text to the DBpedia knowledge

graph. Given an OSM node n, we use the name tag of this node in the language native to the specific OSM dataset as an input to the DBpedia Spotlight model in the same language edition. The model returns a set of DBpedia entities out of which we choose the entity with the highest confidence score. To increase precision, we restrict the DBpedia Spotlight baseline to return only entities of type *dbo:Place*[17]. DBpedia entities are resolved to the equivalent Wikidata entities using existing *wikidata:about* links.

GEO-DIST: This baseline predicts the links solely based on the geographic distance, measured as haversine distance. For a target OSM node n, the link is created between n and $e_{min} \in E_{geo}$, where

$$e_{min} = argmin_{e \in E_{geo}}(distance(n, e)).$$

Here, $distance(n, e)$ is a function that computes the haversine distance between the OSM node n and the geo-entity e.

LGD: This baseline implements a state-of-the-art approach of interlinking OSM with a knowledge graph proposed in the *LinkedGeoData* project [SLHA12]. The LGD baseline utilizes a combination of name similarity computed using the *Jaro–Winkler* string distance and geographic distance. It aims at computing links with high precision. For each OSM node n a link between n and $e \in E_{geo}$ is generated if the condition $\frac{2}{3}s(n, e) + \frac{1}{3}g(n, e, th_{block}) > th_{str}$ is fulfilled, where $th_{str} = 0.95$ Here, $s(n, e)$ denotes the Jaro-Winkler distance between the value of the name tag of n and the label of e. If the name tag is not available, an empty string is used to compute the distance. $g(n, e, th_{block})$ is a logistic geographic distance function specified in [SLHA12]. The parameter th_{block} denotes the maximum distance between a geo-entity and the node n. In our experiments, we use $th_{block} = 20000$ meter to allow for high recall.

LGD-SUPER: We introduce supervision into the LGD baseline by performing exhaustive grid search for $th_{block} \in \{1000, 1500, 2500, 5000, 10000, 20000\}$ meter and $th_{str} \in \{0.05 \cdot i \mid i \in \mathbb{N}, 1 \leq i \leq 20\}$. We evaluate each combination on the respective training set and pick the combination that results in the highest F1 score.

YAGO2GEO: This method was proposed in [KMK19] to enrich the YAGO2 knowledge graph with geospatial information from external sources, including OpenStreetMap. Similar to LGD, this baseline relies on a combination of the Jaro-Winkler and geographic distance. In particular, a link between an OSM node n and $e \in E_{geo}$ is established if $s(n, e) < th_{str}$ and $distance(n, e) < th_{block}$ with $th_{str} = 0.82$, $th_{block} = 20000$ meter. $s(n, e)$ denotes the Jaro-Winkler distance between the value of the name tag of n and the label of e, and $distance(n, e)$ denotes the geographic distance between e and n.

YAGO2GEO-SUPER: We introduce supervision into the YAGO2GEO baseline by performing exhaustive grid search for $th_{block} \in \{1000, 1500, 2500, 5000, 10000, 20000\}$ meter and $th_{str} \in \{0.05 \cdot i \mid i \in \mathbb{N}, 1 \leq i \leq 20\}$. We evaluate each combination

[17] dbo: DBpedia Ontology

on the respective training set and pick the combination that results in the highest F1 score.

LIMES/Wombat: The Wombat algorithm, integrated within the LIMES framework [SNL17], is a state-of-the-art approach for link discovery in knowledge graphs. The algorithm learns rules, so-called link specifications, that rate the similarity of two entities. The rules conduct pairwise comparisons of properties, which are refined and combined within the learning process. As LIMES requires the data in the RDF format, we transformed the OSM nodes into RDF triples, in which the OSM id represents the subject, the key represents the predicate, and the value represents the object. We further added *geo:lat*[18] and *geo:long* properties representing geographic coordinates of the OSM nodes. LIMES requires all entities to contain all considered properties. Therefore we limit the properties to the geographic coordinates *geo:lat*, *geo:lon* as well as the name tag in OSM and the *rdfs:label*[19] in the knowledge graph. We use the default similarity metrics of LIMES, namely Jaccard, trigram, 4-grams, and cosine similarity and accept all links with a similarity score higher or equal to 0.7. Note that LIMES does not distinguish between data types when using machine learning algorithms. Therefore, it is not possible to simultaneously use string similarity and spatial similarity metrics (e.g. Euclidean distance).

4.6.3 OSM2KG Configurations

We evaluate our proposed OSM2KG approach in the following configuration: Random Forest as classification model (according to the results presented later in Section 4.7.3, Random Forest and Decision Tree perform similarly on our datasets), dataset-specific embedding size of 3-5 dimensions (Section 4.7.5), and a blocking threshold of 20 km for `DBpedia-OSM-IT` and 2.5 km for all other datasets (Section 4.7.5).

Furthermore, we evaluate our proposed approach in the following variants:

OSM2KG: In this variant, we run OSM2KG as described in Section 4.5 using the features KV-embedding, Name Similarity, Geo Distance, Entity Type, and Popularity. To obtain latent representations of the OSM nodes, we train unsupervised embedding models as described in Section 4.5.2 on each of the `OSM-FR`, `OSM-IT`, `OSM-DE` datasets. During training, we consider the top-k most frequent values with k=1000 to be represented in the value space and compute 1000 epochs using a learning rate of $\alpha = 1.0$. We make the key-value embeddings of OpenStreetMap nodes created in our experiments publicly available[20]. These key-value embeddings provide a task-independent compact representation of OSM nodes.

[18] geo: http://www.w3.org/2003/01/geo/wgs84_pos
[19] rdfs: http://www.w3.org/2000/01/rdf-schema
[20] http://l3s.de/~tempelmeier/osm2kg/key-value-embeddings.zip

OSM2KG-TFIDF: To better understand the impact of the proposed embedding method on the link discovery performance, in this variant, we exchange the proposed KV-embedding with a simple TF-IDF representation of the keys and values (i.e., term frequency and inverse document frequency). To this extent, we computed the TF-IDF values of the top 1000 most frequent keys and values for each OSM dataset. In this representation, each of the keys and values is described by a single dimension, resulting in a 1000-dimension vector. All other features, such as Name Similarity, Geo Distance, Entity Type, and Popularity remain the same.

4.7 Evaluation

The main goal of the evaluation is to assess the link discovery performance of OSM2KG compared to the baselines. Moreover, we analyze the effectiveness of the classification model and the proposed features and perform parameter tuning.

4.7.1 Link Discovery Performance

Table 4.5 summarizes the overall link discovery performance results of the BM25, SPOTLIGHT, Geo-Dist, LGD, LGD-super, YAGO2GEO, YAGO2GEO-super, and LIMES/Wombat baselines as well as our proposed approach in the OSM2KG and OSM2KG-TFIDF variants. Table 4.5a reports the results of the experiments conducted on the link datasets from Wikidata, while Table 4.5b reports the result on the DBpedia datasets. We report the macro averages of the 10-fold cross-validation conducted on the corresponding link dataset concerning the precision, recall, and F1 score. In our experiments, we observed that the micro averages behave similarly.

Overall, we observe that in terms of F1 score, OSM2KG performs best on all Wikidata datasets, where it achieves an F1 score of 92.05% on average and outperforms the best performing LGD-super baseline by 21.82 percentage points. Furthermore, we observe that OSM2KG achieves the best performance concerning the recall on all datasets. Moreover, OSM2KG maintains high precision, i.e., 94.62% on Wikidata and 97.94% on DBpedia, on average. Regarding the DBpedia datasets, we observe that OSM2KG outperforms the baselines on `DBpedia-OSM-FR` and `DBpedia-OSM-IT`, whereas the difference to the LGD-super baseline is much smaller, compared to Wikidata. On `DBpedia-OSM-DE`, LGD-super archives a slightly higher F1 score, compared to OSM2KG. This result indicates that, in contrast to Wikidata, the respective DBpedia and OSM datasets are well-aligned in terms of names and geographic coordinates, such that simple heuristics utilizing name similarity and geographic distance can already yield good results in many cases. In contrast, the task of link discovery in Wikidata is more challenging. In these settings, the advantages of the OSM2KG approach become clearly visible.

Table 4.5. Macro averages for precision, recall and F1 score [%], best scores are bold. Statistically significant (according to paired t-tests with $p < 0.05$) F1 score results of OSM2KG compared to all baselines and OSM2KG-TFIDF are marked with *.

(a) Link prediction performance on the Wikidata datasets

Approach	Wikidata-OSM-FR			Wikidata-OSM-DE			Wikidata-OSM-IT			Average		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
BM25	45.22	42.59	43.86	47.28	41.60	44.26	44.49	41.67	43.04	45.66	41.95	43.72
SPOTLIGHT	65.17	32.26	43.15	69.79	51.03	58.95	54.79	26.89	36.08	63.25	36.73	46.06
GEO-DIST	74.46	74.46	74.46	62.16	62.16	62.16	72.80	72.80	72.80	69.81	69.81	69.81
LGD	**100.00**	44.09	61.20	**100.00**	47.46	64.37	**100.00**	43.59	60.71	**100.00**	45.05	62.09
LGD-SUPER	**100.00**	53.25	69.50	**100.00**	55.34	71.25	**100.00**	53.79	69.95	**100.00**	54.13	70.23
YAGO2GEO	63.66	44.98	52.71	64.48	48.61	55.43	58.40	47.36	52.30	62.18	46.98	53.48
YAGO2GEO-SUPER	78.49	47.38	59.09	73.49	48.96	58.76	72.25	48.73	58.20	74.74	48.36	58.69
LIMES/WOMBAT	74.03	17.50	28.31	78.54	17.01	27.97	65.28	17.22	27.25	72.62	17.25	27.84
OSM2KG-TFIDF	95.06	90.60	92.77	93.67	86.37	89.87	93.98	87.07	90.39	94.24	88.01	91.01
OSM2KG	95.51	**91.90**	**93.67***	93.98	**88.29**	**91.05***	94.39	**88.68**	**91.45***	94.62	**89.63**	**92.05**

(b) Link prediction performance on the DBpedia datasets

Approach	DBpedia-OSM-FR			DBpedia-OSM-DE			DBpedia-OSM-IT			Average		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
BM25	70.04	69.32	69.68	47.28	76.84	75.58	44.49	41.67	43.04	53.94	62.61	62.77
SPOTLIGHT	72.40	49.42	58.74	79.08	62.31	69.70	85.38	56.17	67.76	78.95	55.97	65.40
GEO-DIST	85.94	85.94	85.94	66.49	66.49	66.49	86.17	86.17	86.17	79.53	79.53	79.53
LGD	**100.00**	61.81	76.40	**100.00**	60.72	75.56	**100.00**	64.94	78.74	**100.00**	62.49	76.90
LGD-SUPER	**100.00**	88.18	93.72	**100.00**	84.56	**91.63**	**100.00**	86.90	92.99	**100.00**	86.55	92.78
YAGO2GEO	77.52	70.40	73.78	87.41	75.84	81.22	94.74	78.47	85.84	86.56	74.90	80.28
YAGO2GEO-SUPER	84.74	82.47	83.59	93.62	80.14	86.36	97.46	81.28	88.64	91.94	81.30	86.19
LIMES/WOMBAT	82.34	60.33	69.64	79.00	68.00	73.09	97.38	70.89	82.05	86.24	66.41	74.93
OSM2KG-TFIDF	98.68	95.35	96.99	95.61	84.93	89.95	98.46	89.83	93.95	97.91	90.04	93.63
OSM2KG	99.06	**96.25**	**97.63***	95.65	**85.83**	90.47	99.11	**90.13**	**94.41**	97.94	**90.74**	**94.17**

The BM25 and SPOTLIGHT baselines adopt name similarity for matching, whereas SPOTLIGHT can also make use of the knowledge graph context, including entity types. As we can observe, BM25 shows relatively low performance in terms of both precision (on average 45.66% (Wikidata) and 53.94% (DBpedia)) and recall (on average 41.95% (Wikidata) and 62.61% (DBpedia)). The SPOTLIGHT baseline can improve on BM25 regarding precision and F1 score on Wikidata and DBpedia datasets. However, the absolute precision and F1 scores of SPOTLIGHT, with the maximum F1 score of 65.40% on Wikidata, are not competitive. Overall, we conclude that name similarity, as adopted by these baselines, is not sufficient for effective link prediction.

The LGD and LGD-SUPER baselines that combine name similarity and geographic distance achieve the best precision of 100% on all datasets. However, the LGD baselines suffer from lower recall. LGD-SUPER achieves on average 54.13% recall on Wikidata and 86.55% recall on DBpedia, overall resulting in lower F1 scores on average compared to OSM2KG. The YAGO2GEO baseline that uses similar features as LGD achieves higher recall scores than LGD (46.98% on Wikidata, 74.90% on DBpedia on average) but cannot maintain the high precision of LGD (on average 62.18% on Wikidata, 86.56% on DBpedia). Overall, YAGO2GEO achieves lower F1 scores compared to OSM2KG.

Regarding the supervised baselines, Table 4.6 presents the parameters learned by LGD-SUPER and the YAGO2GEO-SUPER during the training process. We observe that YAGO2GEO-SUPER learns more restrictive parameters, whereas LGD-SUPER allows for less restrictive threshold values. This result indicates that the ranking function of LGD-SUPER that combines geographic distance and name similarity is more robust than the ranking function of YAGO2GEO-SUPER. YAGO2GEO-SUPER uses geographic distance exclusively for blocking and ranks the candidates based solely on the name similarity. We observe that both baselines achieve a reasonably good performance on the DBpedia datasets. On the contrary, both baselines can not reach comparable performance on the Wikidata datasets and result in 70.23% F1 score for LGD-SUPER, and 58.69% F1 score for YAGO2GEO-SUPER, on average.

GEO-DIST, which solely relies on the geographic distance, achieves an F1 score of 69.81% on Wikidata, and 79.53% on DBpedia on average. Although a significant fraction of the OSM nodes can be correctly linked solely based on the geographic distance, still a significant fraction of nodes (on average 30.19% for Wikidata and 20.74% for DBpedia) can not be appropriately linked this way. We observe that the lower performance of GEO-DIST corresponds to densely populated areas (e.g., large cities), where we expect knowledge graphs to have a higher number of entities, making disambiguation based on geographic distance ineffective. OSM2KG overcomes this limitation and outperforms the GEO-DIST baseline by 22.24 percentage points (Wikidata) and 14.64 percentage points (DBpedia) on average concerning F1 score.

The LIMES/WOMBAT baseline that aims to learn rules for link discovery in a supervised fashion does not achieve competitive performance on any considered dataset and results in 27.84% F1 score for Wikidata and 74.93% F1 score for DBpedia on

Table 4.6. Parameters learned by the LGD-SUPER and the YAGO2GEO-SUPER baselines

Dataset	LGD-SUPER		YAGO2GEO-SUPER	
	th_{block}	th_{str}	th_{block}	th_{str}
Wikidata-OSM-FR	1500	0.1	1000	0.70
Wikidata-OSM-DE	2000	0.1	2000	0.80
Wikidata-OSM-IT	1500	0.1	1000	0.70
DBpedia-OSM-FR	1000	0.1	1000	0.30
DBpedia-OSM-DE	5000	0.1	2000	0.75
DBpedia-OSM-IT	20000	0.3	1500	0.30

average. One of the main reasons for such low performance is that LIMES/WOMBAT requires all entities to contain all considered properties. As none of the OSM tags is mandatory, this baseline is de-facto limited to only frequently used properties, such as the name and the geo-coordinates. These properties alone are insufficient to extract the rules leading to competitive performance in the link discovery task on these datasets.

Comparing the performance of OSM2KG across the datasets, we observe that scores achieved on the WikidataOSM-FR and DBpedia-OSM-FR datasets (93.67%, and 97.63% F1 score) are higher than on the other language editions. This result can be explained through a more consistent annotation of the nodes within the OSM-FR dataset. For instance, in OSM-FR eight key-value combinations appeared more than 2000 times, whereas in OSM-DE and OSM-IT only two to four combinations are that frequent.

Comparing the overall link discovery performance on the DBpedia and Wikidata datasets, we observe that higher F1 scores are achieved on DBpedia by all considered approaches. Furthermore, the LGD-SUPER and YAGO2GEO-SUPER baselines that utilize only geographic distance and name similarity heuristics can reach high performance on DBpedia (up to 92.78% F1 score on average). In contrast, their maximal performance on Wikidata is limited to 70.23% F1 score. This result indicates that, in general, geographic coordinates and entity names of OSM are better aligned with DBpedia than with Wikidata. This result also suggests that the link discovery task is more difficult on Wikidata. Our OSM2KG approach is particularly useful in these settings, where we achieve 21.82 percentage points increase in F1 score compared to the best performing LGD-SUPER baseline.

4.7.2 Comparison to OSM2KG-TFIDF

Comparing the performance of OSM2KG with the OSM2KG-TFIDF variant, we observe that the embedding of OSM2KG leads to better performance (1.04 percentage

points of F1 score for Wikidata and 0.54 percentage points of F1 score for DBpedia on average).

We observe a statistically significant difference between the F1 scores of OSM2KG and Osm2kg-tfidf on all Wikidata datasets and DBpedia-OSM-FR (paired t-tests with $p < 0.01$). Through a manual inspection of exemplary instances, we found that OSM2KG especially improves over OSM2KG-tfidf on discovering links for nodes with name information and nodes corresponding to Wikidata types with a small number of instances. For example, a node corresponding to a private school[21] was wrongly assigned to a trade school[22] instead of the entity[23]. In this example, the name of the OSM node and the geo-entity are identical. We believe that through the high number of dimensions in the TF-IDF representation, the *name* dimension and the corresponding *name similarity* might lose importance, even though the name is typically a very effective feature in the context of link discovery. From the Random Forest models, we observe that the *name similarity* achieves a lower mean decrease impurity [LWSG13] in Osm2kg-tfidf than in OSM2KG, indicating the lower contribution of the feature. Moreover, the *KV-embedding* poses a distributed representation of the OpenStreetMap tags. We believe that especially for Wikidata types with a small number of instances the distributed representation might be more robust, whereas in a TF-IDF representation single tags could introduce bias towards types with a higher number of instances. In the example above, the tag `toilets:wheelchair=yes` is likely to co-occur with both the private school and trade school types but might be biased towards the more populated type.

We do not observe statistically significant differences between OSM2KG and OSM2KG-tfidf on the `DBpedia-OSM-DE` and `DBpedia-OSM-IT` datasets. On these datasets, baselines that exclusively make use of geographic distance and name similarity such as LGD-super achieve the best or close-to-best F1 score. Therefore, the individual importance of the *KV-embedding* or the TF-IDF feature is not as high as for the other datasets.

Furthermore, the proposed *KV-embedding* provides a compact representation that consists of only 3-5 dimensions, whereas the corresponding TF-IDF representations consist of 1000 dimensions. Figure 4.5 contrasts the average memory consumption across the folds of the random forest models of OSM2KG and Osm2kg-tfidf. We observe that the usage of the *KV-embedding* generally results in a lower memory footprint than the TF-IDF variant, which becomes particularly visible for larger datasets. The difference is largest on the `Wikidata-OSM-FR` dataset, where the *KV-embedding* (0.7 GB) requires only 5% of memory compared to the TF-IDF variant (14 GB). We observe the smallest difference on `DBpedia-OSM-IT`. This dataset has the smallest number of instances (2353), resulting in the small memory difference between the models (0.1 GB).

[21] https://www.openstreetmap.org/node/2733503641
[22] https://www.wikidata.org/wiki/Q828825
[23] https://www.wikidata.org/wiki/Q2344470

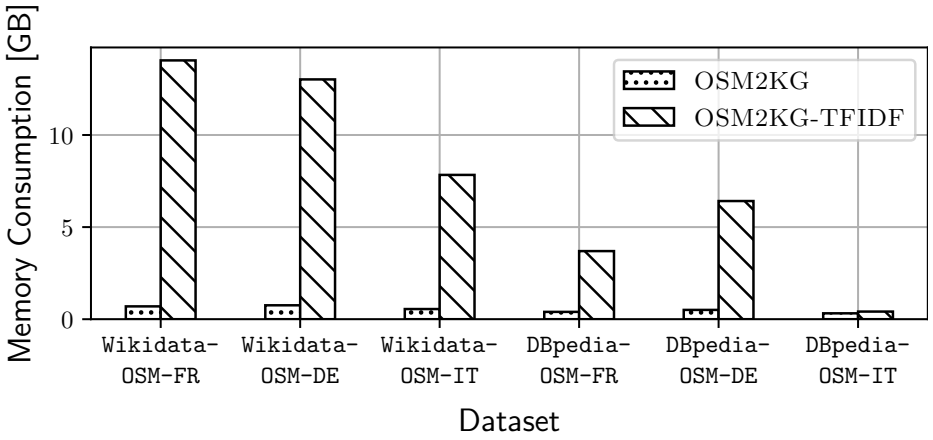


Figure 4.5. Average memory consumption across folds of the training of the Random Forest models used by OSM2KG and OSM2kg-tfidf.

We conclude that KV-embedding is an effective, concise, and task-independent way to represent the OSM information. We believe that this representation makes OSM data more usable for models that may suffer from the curse of dimensionality or memory limitations.

4.7.3 Classification Model Performance

Table 4.7 presents the F1 scores achieved by OSM2KG with respect to each dataset while varying the classification model. In particular, we evaluate the performance of Random Forest, Decision Tree, Naïve Bayes, and Logistic Regression. As we can observe, the performance of the classification models is consistent among the datasets. Random Forest and Decision Tree achieve similar F1 scores and show the best performance, i.e., on average 92.05% (Wikipedia), 94.17% (DBpedia) F1 score using Random Forest, and 92.21% (Wikidata), 93.77% (DBpedia) using Decision Tree. According to a paired t-test, the observed differences between the Random Forest and Decision Tree are not statistically significant on our datasets. In contrast, the performance of Naïve Bayes and Logistic Regression is much lower, i.e., they achieve on average only 66.99% (Wikidata), 80.93% (DBpedia) F1 score using Naïve Bayes and 67.54% (Wikidata), 87.49% (DBpedia) using Logistic Regression.

We conclude that non-linear classification models such as Random Forest and Decision Tree are better suited to the problem we address than the linear models. This result also suggests that the classification problem is not linearly separable. In our experiments in Section 4.7.1, we made use of Random Forest classification models.

Table 4.7. Comparison of OSM2KG F1 scores [%] with respect to the classification model, best scores are bold.

(a) Wikidata

Classifier	Wikidata-OSM-FR	Wikidata-OSM-DE	Wikidata-OSM-IT	Wikidata-Average
Random Forest	93.67	91.05	**91.45**	92.05
Decision Tree	**94.45**	**91.17**	91.01	**92.21**
Naïve Bayes	70.88	63.64	66.45	66.99
Logistic Regression	65.36	66.40	70.87	67.54

(b) DBpedia

Classifier	DBpedia-OSM-FR	DBpedia-OSM-DE	DBpedia-OSM-IT	DBpedia-Average
Random Forest	**97.63**	**90.47**	94.41	**94.17**
Decision Tree	97.12	89.62	**94.56**	93.77
Naïve Bayes	76.69	77.69	88.40	80.93
Logistic Regression	86.84	86.93	88.71	87.49

4.7.4 Feature Evaluation

In this section, we assess the feature contributions of OSM2KG. To assess the contribution of the single features to link discovery, we conducted a leave-one-out feature evaluation. In particular, we removed each feature individually from the feature set and determined the difference in F1 score to quantify the feature importance.

Table 4.8 shows the differences in the F1 score of the OSM2KG model when a single feature is left out compared to the F1 score achieved when the entire feature set is used. Since no difference is negative, except for `DBpedia-OSM-IT`, we conclude that all features typically contribute to better classification performance. *Geo Distance* results in the most substantial difference of 13.99 percentage points on average for Wikidata. On DBpedia, *Geo Distance* results in the second-largest difference of 4.56 percentage points on average. The most considerable difference for DBpedia results from the *Name* feature, with 5.38 percentage points on average. For Wikidata, the *Name* feature results in a variation of 2.98 percentage points on average. The importance of the *Name* feature on DBpedia indicates that the names of the OSM and DBpedia datasets are well-aligned. This result confirms our observations in Section 4.7.1, where we discussed the performance of the LGD-SUPER baseline that utilizes both features.

The *KV-embedding* feature shows the second-largest difference on Wikidata (3.75 percentage points) and the third-largest difference on DBpedia (1.30 percentage points) on average. As expected, the contribution of this feature is higher for the more complex link discovery task in Wikidata, as opposed to DBpedia, where simple heuris-

Table 4.8. Differences in OSM2KG F1 score [percentage points] when leaving out single features using Random Forest.

(a) Wikidata

Left out Feature	Wikidata-OSM-FR	Wikidata-OSM-DE	Wikidata-OSM-IT	Wikidata-Average
KV-embedding	2.80	3.91	4.53	3.75
Geo Distance	15.28	14.72	11.98	13.99
Entity Type	0.71	2.00	2.77	1.83
Popularity	0.29	1.07	0.94	0.77
Entity Type & Popularity	1.67	9.30	6.94	5.97

(b) DBpedia

Left out Feature	DBpedia-OSM-FR	DBpedia-OSM-DE	DBpedia-OSM-IT	DBpedia-Average
KV-embedding	1.94	1.96	0	1.30
Geo Distance	2.81	2.19	8.67	4.56
Entity Type	0.45	0.54	-0.08	0.30
Popularity	0.29	0.28	-0.02	0.18
Entity Type & Popularity	0.84	1.50	-0.08	0.75

tics may suffice. As an extreme example, we do not observe any contribution of *KV-embedding* for DBpedia-OSM-IT. As discussed before, simple heuristics (e.g., geographic distance and name similarity) are sufficient to achieve relatively high performance on this dataset.

The *Entity Type* and *Popularity* show the smallest differences, where *Entity Type* has slightly larger differences than *Popularity*. For the Wikidata datasets, we observe that the individual contributions of the features are rather small, i.e. 1.83 percentage points (*Entity Type*) and 0.77 percentage points (*Popularity*) on average. When leaving both features out, we observe a difference of 5.97 percentage points on average. We conclude that the information encoded in both features is partly redundant. Furthermore, this relatively large difference indicates feature importance. We conclude that for Wikidata datasets the information of the *Entity Type* is especially useful when combined with the *Popularity* feature. On the contrary, for the DBpedia datasets, we observe that the contribution of the *Popularity* feature is nearly identical to the joint contribution of *Entity Type* and *Popularity*. For DBpedia-OSM-IT we observe negative contributions for both features. Again, this indicates that geographic distance and name similarity are sufficient for link discovery in this dataset.

Although *Entity Type* and *Popularity* are correlated in many cases, they can provide complementary information for some instances. Intuitively, the joint information can help to disambiguate entities similar concerning one of the features, but dissimilar regarding the other. For example, two railway stations of different sizes are likely to be described with a different number of statements, whereas the type is identical.

In such cases, in addition to the Entity Type, Popularity can help to disambiguate entities better.

4.7.5 Parameter Tuning

We evaluate the influence of the parameters such as embedding size and the blocking threshold value on the performance of OSM2KG.

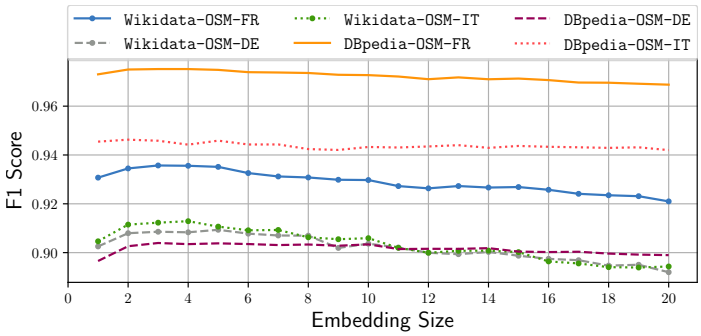


Figure 4.6. Influence of the embedding size on F1 score of the Random Forest classifier.

Embedding Size

The embedding size corresponds to the number of dimensions (i.e. neurons) in the projection layer of the neural model presented in Section 4.5.2. Figure 4.6 shows F1 scores obtained with respect to the number of dimensions of the *KV-embedding* achieved by the Random Forest classifier on all datasets.

We observe similar trends for all datasets except for `DBpedia-OSM-IT`. Overall, we can observe a growth of the F1 score of the classifier with an increasing number of dimensions, between one and four dimensions for all datasets. We conclude that embeddings with an insufficient number of dimensions are not able to capture all relevant information. When the number of dimensions increases, more information can be encoded, which leads to better performance. As we can observe, the curve achieves its maximum at three dimensions for the `Wikidata-OSM-FR`, and `DBpedia-OSM-FR` datasets, at four dimensions for `WikidataOSM-IT` and at five dimensions for the `Wikidata-OSM-DE` and `DBpedia-OSM-DE` datasets. Further increase of the embedding size does not lead to an increase in performance. On the contrary, the performance can drop, indicating that no additional beneficial information is obtained by adding further dimensions.

For `DBpedia-OSM-IT`, we observe a near-constant performance around 94% F1 score of the classifier. As discussed in Section 4.7.4, here the contribution of the KV-embedding is not as high as for the other datasets. Thus, the variation of the embedding size does not result in any significant performance changes for this dataset.

Overall, we conclude that 3-5 dimensions are most suited for the datasets that make effective use of the KV-embedding feature. Thus, we adopted the following number of dimensions: `Wikidata-OSM-FR`: 3, `Wikidata-OSM-DE`:5, `Wikidata-OSM-IT`: 4, `DBpedia-OSM-FR`: 3, `DBpedia-OSM-DE`: 5, `DBpedia-OSM-IT`: 4.

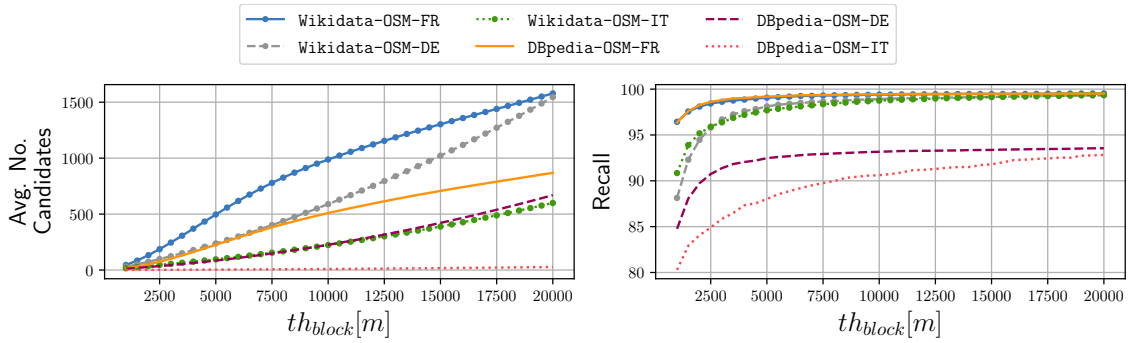


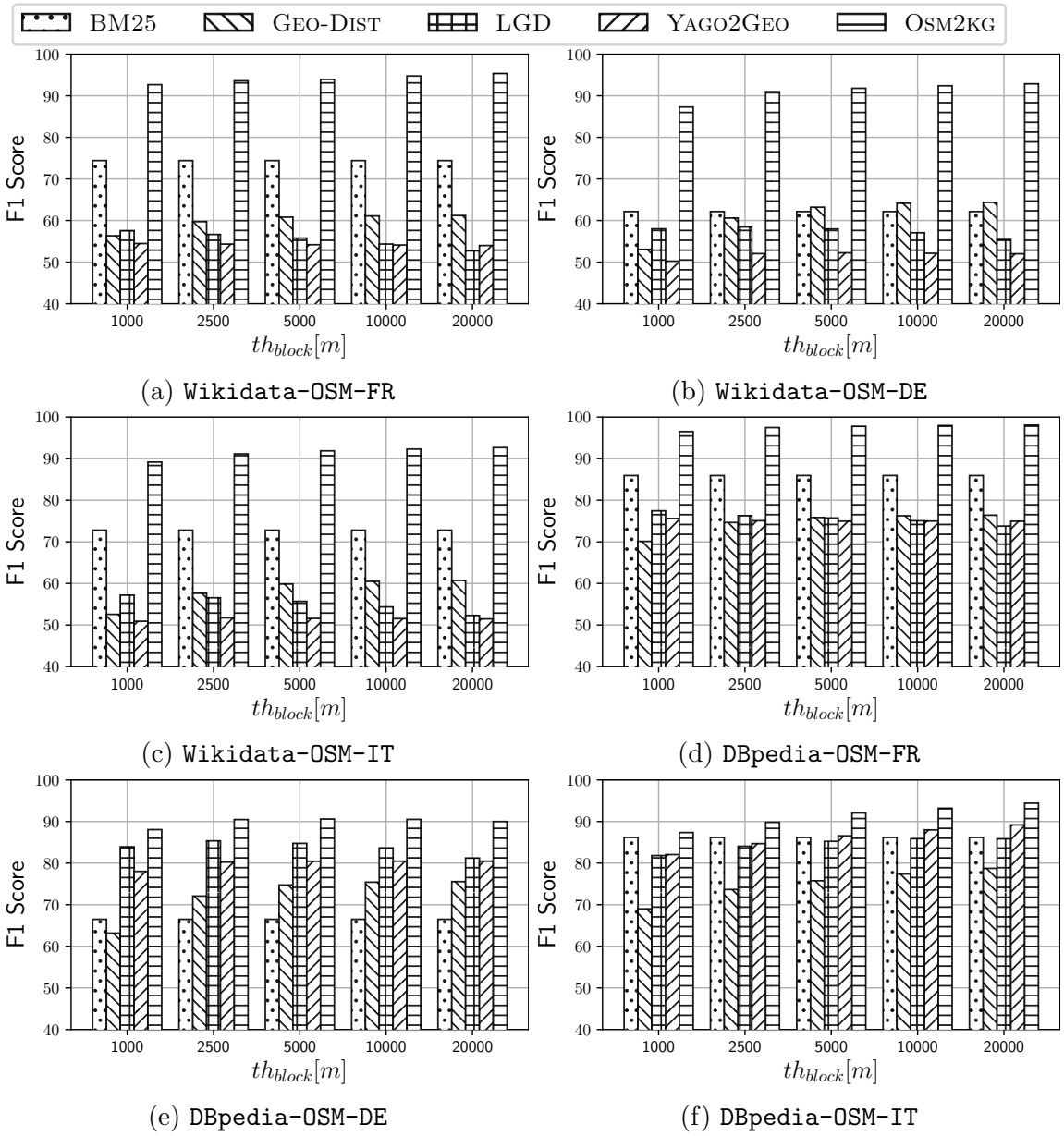
Figure 4.7. Influence of the threshold th_{block} on the average number of candidates and recall of the blocking step.

Blocking Threshold

The blocking threshold th_{block} represents the maximal geographic distance considered for candidate entity generation, as discussed in Section 4.5.1. For a single OSM node, all knowledge graph entities that are closer than th_{block} are considered as candidates. The value of th_{block} can be determined experimentally by evaluating the recall of the blocking step.

Figure 4.7 shows the influence of th_{block} on the average number of candidates and the recall of the blocking step. Considering the average number of candidates, we observe a linear-like rise (i.e., the slope of the curve is nearly constant) of the number of candidates concerning th_{block} for all datasets, whereas the datasets differ in slope. Due to the low geographic density of the `DBpedia-OSM-IT` dataset, the corresponding slope is especially low. Concerning recall, we observe that the curve starts with a steady incline, but quickly saturates with an increasing th_{block}. We conclude that in most cases, the correct candidate exhibits a geographic distance of about 2.5 km. Thus, in our experiments, we chose $th_{block} = 2.5$ km. This threshold value allows for more than 85% recall of correct candidates for the DBpedia datasets and 95% recall for the Wikidata datasets in the blocking step, while effectively limiting the number of candidates. For `DBpedia-OSM-IT`, we adopt a different th_{block} threshold of 20 km to increase recall on this dataset.

Figure 4.8 presents the F1 scores regarding the blocking threshold value th_{block}. To make the impact of geospatial blocking comparable across the considered approaches, we assess the effect of the blocking step on the overall link discovery performance. To this extent, we added an additional blocking step to the BM25 and Geo-



(a) `Wikidata-OSM-FR` (b) `Wikidata-OSM-DE`

(c) `Wikidata-OSM-IT` (d) `DBpedia-OSM-FR`

(e) `DBpedia-OSM-DE` (f) `DBpedia-OSM-IT`

Figure 4.8. Link discovery performance concerning th_{block} value for OSM2KG and the baselines that can include a blocking step. X-axis presents the value of th_{block} in meter. Y-axis presents the F1 score.

Dist baselines and evaluate the models BM25, Geo-Dist, LGD, Yageo2Geo and OSM2KG with the blocking thresholds $th_{block} \in \{1, 2.5, 5, 10, 20\}$ km. As we can observe, the general link discovery performance is not very sensitive to the th_{block} value. However, if th_{block} value is chosen too low, e.g. 1 km, the link discovery performance can drop, as shown in Figure 4.8b. Overall, an optimal threshold value depends on the model as well as on the dataset. For example, LGD may benefit from a lower blocking threshold value, as shown in Figure 4.8e, whereas Geo-Dist works better with a higher threshold (Figure 4.8f). For OSM2KG we do not observe any significant impact for values of $th_{block} \geq 2.5$ km for most datasets. For the supervised variants of the baselines LGD and Yago2Geo, LGD-super and Yago2Geo-super, we observe that the appropriate threshold can be determined during the training process. The performance of the Geo-Dist baseline is degraded with the limitation of the additional blocking step, as this limitation does not contribute to precision, but potentially limits recall of this baseline. The BM25 baseline benefits from the blocking step but is still clearly outperformed by OSM2KG. In summary, as presented by Figure 4.8, we observe that OSM2KG outperforms all baselines for all values of the blocking threshold th_{block} on all considered datasets concerning F1 score.

4.7.6 Error Analysis

We conducted an error analysis through manual inspection of a random sample of 100 nodes for which OSM2KG identified no correct link for each of the Wikidata datasets. Table 4.9 presents the resulting error distribution. As we can observe, the most common reason for errors is a too restrictive candidate selection leading to an empty candidate set (in 49.67% of cases), followed by the selection of wrong candidates (in 32.67% of cases) and quality issues in Wikidata such as duplicate entities (in 13.67%) as well as wrong links in the ground truth data (in 4%). Note that the restrictive candidate selection is subject to the choice of the blocking threshold value. For this study, the threshold was chosen in such a way that 95% recall of the blocking step was achieved. In a few cases (3% on average), the candidate set is not empty, but the correct candidate is not included in this set. This issue can be addressed by an adaptive increase of the threshold for the nodes without any candidates.

Furthermore, we observe that the selection of wrong candidates in most cases happens within the regions with a high geographic density of Wikidata entities, e.g., in cities where single houses can represent entities, resulting in a large candidate set. To further increase the precision of OSM2KG, a dedicated, supervised model for geographically dense regions can be trained. Such a model can follow a more restrictive policy, e.g., by requiring higher confidence to establish a link.

Finally, the detection of duplicate entities and wrong ground truth links indicates the potential to adopt OSM2KG for de-duplication of geo-entities in Wikidata to increase data quality. These observations provide a basis for an incremental tuning of OSM2KG in future work.

Table 4.9. Distribution of error types on nodes for which no correct link could be found by OSM2KG.

Error Type	Wikidata-OSM-FR	Wikidata-OSM-DE	Wikidata-OSM-IT	Avg.
No candidate found	41%	54%	54%	49.67%
Wrong candidate selected	39%	37%	22%	32.67%
Duplicate entity in Wikidata	17%	4%	20%	13.67 %
Wrong link in ground truth	3%	5%	4%	4.00%

4.7.7 Evaluation Summary

Approaches that mainly rely on name similarity heuristics and do not leverage any geospatial features are not suitable for effective link prediction for the OSM nodes. We can observe this by considering the relatively low performance of the BM25 and SPOTLIGHT baselines, where SPOTLIGHT achieved F1 scores of 46.06% (Wikidata) and 65.40% (DBpedia), on average. Geospatial features such as geographic distance are a reliable indicator to match OSM nodes with knowledge graph entities in our datasets. This observation is confirmed by the GEO-DIST baseline, which reached F1 scores of 69.81% (Wikidata) and 79.53% (DBpedia) by solely considering the geographic distance. However, in a significant fraction of cases, geospatial information alone is insufficient to disambiguate OSM nodes effectively. Heuristics using a combination of the name similarity and geospatial features, and in particular the supervised LGD-SUPER baseline, can achieve competitive performance on the DBpedia datasets. However, they are insufficient for link discovery in more complex datasets, such as Wikidata, where the entity names are not well-aligned with OSM.

The proposed OSM2KG approach combines the latent representation of OSM nodes that captures the semantic similarity of the nodes with geospatial information and is highly effective for link prediction. OSM2KG is of particular advantage for link discovery between OSM and Wikidata, where it significantly outperforms the baselines concerning the recall and F1 score. Overall, we observe that the proposed latent node representation as *key-value embedding* combined with geospatial distance is an effective way to facilitate link discovery in a schema-agnostic volunteered geographic dataset such as OSM. This representation, with only 3-5 dimensions, is compact and task-independent.

4.8 Discussion

In this chapter, we enriched geographic Web information by adding links to OpenStreetMap. We proposed OSM2KG, a novel link discovery approach to predict identity links between OpenStreetMap nodes and geographic entities in knowledge

graphs.

OSM2KG combines latent representations of OSM nodes, knowledge graph features, and a supervised classification model to effectively predict identity links across OSM and knowledge graphs. The representation of OSM nodes is highly heterogeneous. We tackle the problem of OSM data heterogeneity by introducing an unsupervised key-value embedding capturing the semantics of OSM nodes. In contrast, knowledge graphs provide well-defined schemas, i.e., ontologies. We capitalize on these schemas by extracting selected features, e.g., indicating the entity type or popularity. Based on the feature representations, we use a binary supervised classification model predicting whether an OSM node and a knowledge graph entity represent the same real-world entity.

We conducted an extensive evaluation on three large-scale OSM datasets for Germany, France, and Italy and Wikidata and DBpedia knowledge graphs. Our experiments demonstrate that the proposed OSM2KG approach can reliably discover identity links. OSM2KG achieves an F1 score of 92.05% on Wikidata and of 94.17% on DBpedia on average, which corresponds to a 21.82 percentage points increase in F1 score on Wikidata compared to the best performing baselines. We showed the superior effectiveness of the key-value embeddings compared to traditional TF-IDF feature representations of OSM nodes regarding link classification performance and memory consumption. We investigated the contribution of the individual features and found that all features help in the link discovery process. Finally, we provided a detailed analysis of parameter tuning considering the embedding size, the blocking threshold, and the choice of the classification model.

Limitations for the proposed link discovery model can arise from the candidate generation step, where we consider the set of entities for which geographic coordinates are available in the knowledge graph only. A promising direction for future research is to discover identity links between OSM nodes and geographic entities for which geographic coordinates are not available in the knowledge graph. In this chapter, we focused the discussion and evaluation of OSM2KG on Wikidata and DBpedia as target knowledge graphs due to their openness, popularity, and availability of training data (i.e., the links between these knowledge graphs and OSM). Nevertheless, the proposed OSM2KG approach is applicable to other knowledge graphs, provided a set of identity links between OSM and the target knowledge graph is available for training the OSM2KG classifier.

5 Enriching Missing Information in Web Markup

In this chapter, we broaden the scope of the thesis by introducing an additional data source, i.e., semantic Web markup. While semantic Web markup is not limited to describing geographic entities only, many markup entities relate to geographic places. For instance, *events* usually provide venues or coordinates where they take place. Semantic Web markup data is often sparse, such that important entity properties may be missing. **RQ3** asks how to add such missing information to Web markup entities. We tackle this problem by presenting an enrichment approach that infers missing categorical information of Web markup entities.

5.1 Introduction

Semi-structured, entity-centric knowledge has become a key component for the interpretation of Web documents and enable, e.g., effective Web search. Recently, Web markup facilitated through standards such as RDFa [Wor08], Microdata[1] and Microformats[2] has become prevalent on the Web, driven by initiatives such as *schema.org*. Such semi-structured Web annotations are a potentially rich source of geographic Web information.

There is an upward trend of Web markup adoption, where the proportion of pages containing markup increased from 5.76% to 39% between 2010 and 2016. To this extent, markup data provides an unprecedented and growing source of explicit entity annotations to be used when interpreting and retrieving Web documents, to complement annotations otherwise obtainable through traditional information extraction pipelines, or to train information extraction methods. In addition, while traditional KGs capture large amounts of factual knowledge, they still are incomplete, i.e. coverage and completeness vary heavily across different types or domains. In particular,

[1] https://www.w3.org/TR/microdata/
[2] http://microformats.org

there is a large percentage of less popular (long-tail) entities and properties that are usually insufficiently represented [BEM+13]. In this context, markup also provides essential input when incrementally augmenting and maintaining KGs [YGFD17], in particular when attempting to complement information about long-tail properties and entities [YGF+19].

The specific characteristics of statements extracted from embedded Web markup pose particular challenges [YFGD16]. Whereas coreferences are very frequent (for instance, in the WDC 2013 corpus, 18,000 entity descriptions of type *schema.org:Product* are returned for the query `'Iphone 6'`), these are not linked through explicit statements. In contrast to traditional densely connected RDF graphs, markup statements mostly consist of isolated nodes and small subgraphs, each usually made up of small sets of statements per entity description. In addition, extracted RDF markup statements are highly redundant and are often limited to a small set of highly popular predicates, such as *schema.org:name*, complemented by a long tail of less frequent statements. Moreover, data extracted from markup contains a wide variety of errors [MRP16], ranging from typos to the frequent misuse of vocabulary terms [aHP15]. Hence, individual markup extracted from a particular Web document or crawl usually contains very limited or unreliable information about a particular entity. According to our analysis, out of 26 million annotated events in the WDC 2016 corpus, less than 257,000 (0.96%) indicate a more specific event subtype and 59% nodes provide less than six statements. This strongly limits the meaningfulness of Web markup, in particular for entities that cannot be mapped to a representation in an existing knowledge graph.

In this chapter, we introduce an approach to automatically infer missing categorical information for particular entities obtained from Web markup. Building on the Web-scale availability of markup, and hence, the abundance of potential training data for the task, we introduce a supervised method to efficiently infer missing categorical information from existing entity markup describing coreferring or similar entities. Our experiments address the inference of entity (sub-)types, as well as inference of arbitrary non-hierarchical predicates, such as movie genres. We demonstrate superior performance compared to both naïve baselines and specialized state-of-the-art methods for type inference and achieve F1 scores of 79% and 83% in two experimental tasks.

Contributions. In this chapter, we address **RQ3** and make the following contributions

- We present a novel supervised classification model to infer missing categorical information in Web markup.
- We introduce an algorithm to derive training data from unbalanced Web markup data.
- We conduct an extensive evaluation, comparing our approach to both naïve and

state-of-the-art baselines.

The rest of this chapter is organized as follows: We discuss related work in Section 5.2. Then, in Section 5.3, we motivate our approach by analyzing the distribution of selected markup properties. Next, we formally define the problem in Section 5.4. In Section 5.5, we introduce our supervised classification model. In Section 5.6, we define the evaluation setup. Following that, in Section 5.7, we present the evaluation results. Finally, we provide a discussion in Section 5.8.

5.2 Related Work

In this section, we discuss related work in the areas of knowledge graph completion and schema inference for traditional knowledge graphs along with works focused directly on Web markup.

Knowledge Graph Completion. Existing approaches to knowledge graph (KG) completion and dataset profiling including its applications to schema inference have been summarized in recent survey articles [YFGD16, EBB+18]. These approaches include in particular entity type inference, relation prediction and relation validation. In the context of KG completion, entity type inference is most commonly addressed as a multi-class prediction problem. [PB13] makes use of properties and conditional probabilities to infer entity types, building the baseline for our approach. Schemex is an approach to extract and index schema information from Linked Open Data (LOD) [KGSS12]. In YAGO+F instance-based matching enables to enrich Freebase entities with YAGO concepts [DON13]. [GKSS13] made use of Schemex to analyze schema information of LOD and found that properties provide information about subject types. In this chapter, we use properties as features for inferring missing categorical information in general. [GM15] predicts relations between two nodes by leveraging random walk inference methods using sub-graphs to improve the path ranking algorithm (PRA), initially proposed in [LC10]. [WLL+16] also builds on PRA and extends it to a multitask learning approach.

All of the works discussed above have been applied to traditional KGs such as DBpedia, NELL and YAGO. In contrast, in this chapter we aim at inferring information on the Web markup data. Web markup is distinguished from the aforementioned knowledge graphs by specific characteristics, i.e. annotations are often very sparse or noisy, vocabularies are not used correctly in many cases and the overall RDF graph is connected very loosely [DTY+17, MPB14]. For these reasons, existing KG completion methods are not likely to perform well on Web markup. For instance, KG completion approaches based on graph topology (e.g. relation prediction discussed above) rely on the presence of relations, which are not widely available in markup.

Various approaches employ embeddings for KG completion in traditional knowledge graphs. [WMWG17] conducted a survey on KG embeddings for applications such as link prediction, entity classification and triple classification. [WWG15] makes

use of embeddings and rules. [LLS+15] propose the *TransR* model that builds separate entity and relation embeddings to compute the plausibility of missing triples. [SCMN13] predicts relations between entities by employing neural tensor networks. Embeddings techniques have not yet been applied to Web markup yet lend themselves as direction for future research.

Web Markup. Several recent studies focused on analyzing the characteristics, evolution and coverage of markup [TD16, SGY+16, DTY+17] and on addressing specific tasks in the context of Web markup. Meusel et al. proposed heuristics that can be employed to fix common errors in Web markup [MPB14, aHP15]. In this chapter, we apply the heuristics proposed in [aHP15] for pre-processing and data cleansing. [YGFD17, YGZ+16] provide pipelines for data fusion and entity summarization on Web markup, involving heuristics, clustering and supervised approaches for entity matching and classification of markup statements. [YGF+19] builds on these works by utilizing fused markup data to augment existing knowledge bases, showing the complementarity of markup data and its potential to significantly complement information from traditional reference KGs.

While these works demonstrate the use of markup data, they suffer from the sparsity of individual nodes. The inference approach proposed in our work can augment markup nodes and is likely to boost the performance on both fusion and KG augmentation tasks. In particular, considering the impact of the use of controlled vocabularies on data reuse [EGT+17], we anticipate that inference of crucial categorical information can facilitate reuse of markup data.

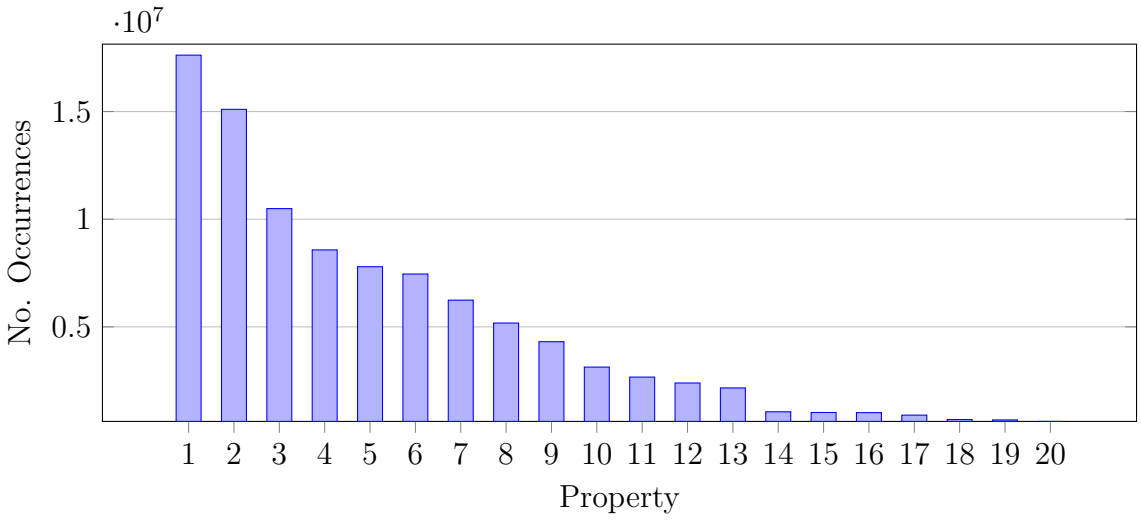
5.3 Motivation

By today, Web markup data is available at an unprecedentedly large scale, resulting in a high potential value for data-driven algorithms In the following we illustrate the challenges and the potential of leveraging Web markup data at the example of the *Web Data Commons* (WDC) [MPB14] corpus. We abbreviate the prefix of the *schema.org* vocabulary by *s:*, e.g. *s:Movie*. We refer to the WDC corpus from October 2016 as the WDC 2016 corpus.

While Web markup constitutes an unprecedented source of semi-structured knowledge, markup is usually sparse and highly redundant, consisting of vast amounts of coreferences and (near) duplicate statements [YGFD17]. The description of individual entities extracted from Web markup is usually sparse, such that only a fraction of the properties foreseen by *schema.org* for a specific type is provided. Specific nodes often only provide a label and the entity type. Table 5.1 provides an overview of the number of quadruples per single node for the specific types (*s:Event*, *s:Movie*) in the WDC 2016 corpus. The property distribution follows a power law, where a small set of terms is very prevalent, yet the majority of properties is hardly used across the Web. Figure 5.1 shows the top-20 most frequently used properties of movies,

Table 5.1. Number of quadruples per node for specific types in WDC 2016.

Type	Total No. Quadruples	Total No. Nodes	Quadruples				Distinct Properties			
			Min.	Max	Avg.	Median	Min.	Max.	Avg.	Median
s:Event	$1.58 \cdot 10^8$	$2.66 \cdot 10^7$	1	2889	5.55	5	1	32	5.31	5
s:Movie	$1.25 \cdot 10^8$	$1.62 \cdot 10^7$	1	4547	7.71	6	1	26	5.77	6

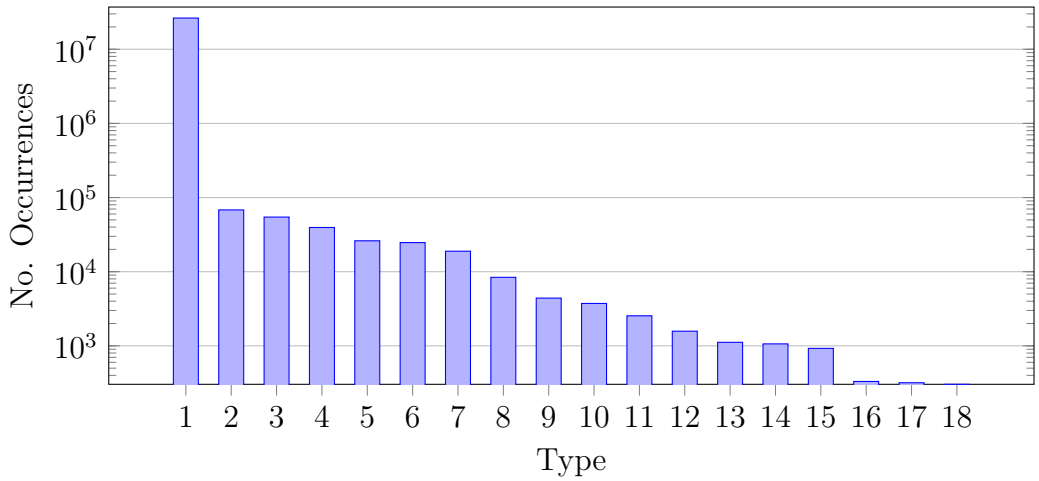


1:*s:actor*	6:*s:genre*	11:*s:aggregateRating*	16:*s:actor*
2:*s:name*	7:*s:duration*	12:*s:actors*	17:*s:dateCreated*
3:*s:image*	8:*s:datePublished*	13:*s:keywords*	18:*s:thumbnailUrl*
4:*s:url*	9:*s:director*	14:*s:productionCompany*	19:*s:inLanguage*
5:*s:description*	10:*s:interactionCount*	15:*s:contentRating*	20:*s:review*

Figure 5.1. Top-20 most frequent properties for the type *s:Movie* in WDC 2016. The second entry of *s:actor* is caused by erroneous annotations in Web markup.

highlighting that certain properties occur very often (e.g. *s:actor*) while others are provided rarely, such as *s:productionCompany*. Sparsity is exacerbated by the lack of connectivity of markup data, where controlled vocabularies, taxonomies, and essentially, links among nodes are hardly present. Previous studies [DTY+17] on a specific markup subset find that, out of a set of 46 million quadruples involving transversal, i.e. non-hierarchical properties, approximately 97% actually refer to literals rather than URIs, that is object nodes. These findings underline that markup data largely consists of rather isolated nodes, which are linked through common schema terms (as provided by *schema.org*) at best, but commonly lack relations at the instance level. In particular for categorical information, such as movie genres or product categories, this poses a crucial challenge when it comes to interpreting such information.

A particular instantiation of the aforementioned problem is the use of unspecific types. Figure 5.2 illustrates the number of instances of events annotated with respective event subtypes. Note that assignment of multiple types is theoretically possible,



1: *s:Event*	6: *s:VisualArtsEvent*	11: *s:FoodEvent*	16: *s:SaleEvent*
2: *s:PublicationEvent*	7: *s:TheaterEvent*	12: *s:DanceEvent*	17: *s:ExhibitionEvent*
3: *s:MusicEvent*	8: *s:ComedyEvent*	13: *s:SocialEvent*	18: *s:BusinessEvent*
4: *s:ScreeningEvent*	9: *s:SportsEvent*	14: *s:ChildrensEvent*	
5: *s:EducationEvent*	10: *s:LiteraryEvent*	15: *s:Festival*	

Figure 5.2. Number of occurrences of schema.org event types in WDC 2016 (Y-axis is logarithmic).

but rarely used in practice (i.e. less than 0.1% of events have multiple types). Apparently, most of the instances are assigned the generic type *s:Event*, while only 0.96% of nodes use more specific types like *s:TheaterEvent* or *s:Festival*, hindering data interpretation.

Whereas individual markup nodes are usually sparsely annotated, markup as a whole provides a rich source of data, where in particular for categorical, i.e. discrete, properties a wide variety of instances can be drawn from the long tail. For instance, referring to Figure 5.2, while only 0.96% of all event nodes are typed with a meaningful subtype, this still corresponds to a set of 257,000 nodes available as training data to build supervised models to classify the remaining 26 million insufficiently typed events. Hence, we follow the intuition that markup data can significantly benefit from supervised approaches, which learn categorical or discretized properties as a means to infer missing categorical information for sparsely annotated nodes, i.e. to enrich markup entities. Overall, augmentation of sparse Web markup nodes can contribute to the improvement of the interpretability of the markup, the enrichment of knowledge graphs, and hence, to the effectiveness of the applications using the markup. This includes search and Web page classification, where in particular categorical and type information is essential to correctly interpret resources.

5.4 Problem Statement

In this chapter, we aim at inferring missing categorical information in data sourced from Web markup. For a given corpus of Websites $\mathcal{C}$, $\mathcal{Q}_\mathcal{C}$ denotes the set of *RDF quadruples* of the form (s, p, o, u) extracted from the corpus, where s, p, o represent an RDF triple, i.e. a statement, of the form subject, predicate and object and u represents the URL of the Web document, from which the triple has been extracted.

A *vocabulary* V consists of a set of *types* T and *properties* P. A particular property $p_i \in P$ has a declared domain $d(p_i)$ that defines the set of expected types $T_i \subseteq T$ a subject involved in the same triple with p_i is meant to be an instance of. The range $r(p_i)$ of a property p_i defines the expected types an object involved in the same triple as p_i is meant to be an instance of.

For instance, within the *schema.org* vocabulary, the domain of the property *translator*[3] is defined as instances of type *Event*[4] and *CreativeWork*[5], while the declared range is defined as instances of type *Organization*[6] and *Person*[7].

Definition 5.1. *Given a vocabulary V, a set of quadruples $\mathcal{Q}_\mathcal{C}$, for a particular node representing a subject $s_i \in \mathcal{Q}_\mathcal{C}$, this we aim at predicting quadruples $q = (s_i, p_i, o_i, u_i)$ which are: (a) not present in the markup corpus ($q \notin \mathcal{Q}_\mathcal{C}$), (b) valid according to the definition of vocabulary V, and (c) a valid statement about subject s_i in the context of u_i.*

The last requirement of the aforementioned definition is experimentally evaluated according to a ground truth G, where an example is described in Section 5.6.1.

Note that this chapter focuses on *categorical* properties, i.e. we consider properties where the corresponding range $r(p)$ is *finite*. For instance, consider the following markup triple, extracted from the URL http://www.imdb.com/title/tt0109830/?ref_=tt_trv_cnn describing the movie "Forrest Gump":

$$\left[\begin{array}{l|l} s: & \_:nodea73846c741abe988abf1c682f1fe26e7 \\ p: & rdf{:}type \\ o: & s{:}Movie \end{array}\right]$$

For the specific subtask of predicting movie genres (Section 5.7), we aim at predicting the quadruple involving the following triple (URL omitted) stating the genre of the movie:

$$\left[\begin{array}{l|l} s: & \_:nodea73846c741abe988abf1c682f1fe26e7 \\ p: & s{:}genre \\ o: & "Drama" \end{array}\right]$$

[3] http://schema.org/translator
[4] http://schema.org/Event
[5] http://schema.org/CreativeWork
[6] http://schema.org/Organization
[7] http://schema.org/Person

5.5 Supervised Inference Approach

The characteristics of the data at hand suggest that, for most subjects s_i which are to be augmented, e.g. the movie mentioned in the previous example, sufficient training data can be obtained (Section 5.3). That means, we anticipate that a sufficient number of entity descriptions (instances) exist, which share the same missing categorical property p_i, e.g. a movie genre in the example above. Thus, we approach the inference problem as a supervised classification problem, where nodes which share the sought after property p_i are used as training data to build a model for the prediction of respective statements. This section describes our approach, namely the steps taken for data cleansing, feature extraction and building classification models.

5.5.1 Data Cleansing

Based on studies on common errors on deployed microdata [aHP15], we applied the following heuristics proposed in [aHP15], to improve the quality of the dataset by fixing the following errors:

Wrong namespaces: Many terms that deviate from the correct *schema.org* namespace can be corrected by adding missing slashes, changing *https://* to *http://*, removing additional substrings between *http://* and *schema.org* and fixing capitalization errors.

Undefined properties and types: The use of wrong capitalization of property and type names leads to the presence of undefined terms in markup data. We corrected the capitalization by using the capitalization defined by the *schema.org* vocabulary.

Applying these heuristics aids the feature extraction and classification steps described below by providing a larger amount of training data as well as by improving feature quality.

5.5.2 Feature Extraction

This section describes the considered features for our task and the applied feature extraction.

pld/tld: Based on the assumption that many Web domains are specialized on particular topics, e.g. concerts or documentary films, we employ domain-based features. The intuition is that any particular *pay-level-domain* (pld) and/or *top-level-domain* (tld) usually correlates with particular categorical properties, such as the types of covered events. Thus, for each node, we extract the pld and the tld from the URL of the Web page. For instance, taking into account the task of predicting event subtypes,

consider the quadruple:

$$\left[\begin{array}{l|l} s: & \_:node396540c21b6fa0388c7293ebe216583 \\ p: & rdf:type \\ o: & s:Event \\ u: & <http://www.touristlink.com/india/cat/events.html> \end{array}\right]$$

From this quadruple we extract the pld "touristlink.com" and the tld ".com" from u and use these as features to predict the subtype "*s:MusicEvent*" of the described event. The plds and tlds are mapped into feature space via *1-hot-encoding*, resulting in one dimension for each pld and each tld.

node-vocab: The intuition behind this feature is that there is a correlation between the used vocabulary terms and the specific classes we aim to predict. For example, a composer (*s:composer*) is more likely to be provided for a music event (*s:MusicEvent*) than for a sports event. Following this intuition, Paulheim et al. [PB13] proposed an approach for entity type prediction using vocabulary term correlations. To this extent, they made use of the outgoing and incoming statements of the node n for type prediction of n in knowledge graphs (i.e. statements that have n either in the subject or the object position, respectively). In case of Web markup, it may not be feasible to determine all incoming statements for a given subject at Web scale. Therefore, in this chapter, we make use of the outgoing statements only and use these statements to predict categorical properties of the entity described through the node n. More specifically, for all quadruples Q_n involving subject n, we extract all *schema.org* terms used as predicate. For each node n, we compute a frequency vector, where each dimension corresponds to a vocabulary term t_i and each value is the normalised number of times t_i occurs in a quadruple with n as a subject. The frequencies are normalised using the l^2 (euclidean) norm.

Example 1. *For the node s and URL u*

$$\left[\begin{array}{l|l} s: & \_:node3957c770b4f7c0bd1a17805dd8ca406 \\ u: & <https://gdssummits.com/nghealthcare/us/> \end{array}\right]$$

the following tuples are present:

$$\left[\begin{array}{l|l} p: & rdf:type \\ o: & <http://schema.org/BusinessEvent> \end{array}\right]$$
$$\left[\begin{array}{l|l} p: & s:Event/name \\ o: & "NG Healthcare Summit US"@en \end{array}\right]$$
$$\left[\begin{array}{l|l} p: & s:Event/location \\ o: & "Omni Barton Creek Resort \& Spa, Austin, Texas"@en \end{array}\right]$$

These tuples result in the following node-vocab: {rdf:type:1, s:Event/name:1, s:Event/location:1}.

Note that we concatenated the predicate and the type used as the domain of the predicate. This way we ensure that: (a) types as well as terms are considered and (b) the connection between a predicate and its observed domain is preserved. The latter appears useful, considering that *schema.org* terms are used in a variety of contexts, often in ways other than recommended by the vocabulary definition, e.g. by violating domain and range definitions [DTY+17].

page-vocab: The vocabulary used on a Web page within which a subject appears intuitively correlates with categorical classes associated with nodes on the respective page. For instance, Websites discussing music albums are more likely to also contain music events rather than sports events. To take this context into account, we consider all *schema.org* vocabulary terms that appear as predicates on the same Web page as the node under consideration as a feature. Similar to the node-vocab, we create a frequency vector normalized using the l^2 (euclidean) norm.

Example 2. *Assume that in addition to the quadruples in Example 1, the following triples are present on the same Web page:*

$$\begin{bmatrix} s & \text{\_:nodea9ff152514bcfb63c2714bc1336b2b3} \\ p: & \text{s:Organization/url} \\ o: & \text{<http://www.gdsinternational.com>} \end{bmatrix}$$

$$\begin{bmatrix} s & \text{\_:node4ccbf7f34c95f14168f5fdb47b73ab} \\ p: & \text{rdf:type} \\ o: & \text{s:BusinessEvent} \end{bmatrix}$$

Then the terms from these quadruples are added to the node-vocab to form the page-vocab: {rdf:type:2, s:Event/name:1, s:Event/location:1, s:Organization/url:1}.

After computing the individual features, all features are concatenated to form a single feature vector. Finally, the feature vectors are normalized, i.e. the mean is removed and the features are scaled to unit variance. The feature vectors serve as input for supervised machine learning approaches that are detailed in Section 5.5.3.

5.5.3 Classification Models

We compare the use of the following classifiers:

Naïve Bayes: A Gaussian Naïve Bayes classifier that assumes that the likelihood of the features follows a Gaussian distribution. Since the features are normalized (i.e. may have negative values), a multinomial Naïve Bayes can not be applied. Naïve Bayes classifiers are known to be adoptable to many classification tasks.

Decision Tree: A classifier that successively divides the feature space to maximize a given metric (e.g. Gini Impurity, Information Gain). Decision Trees are able to identify discriminative features within high-dimensional data.

Random Forest: A classifier that utilizes an ensemble of uncorrelated decision trees. Random Forests can utilize a large amount of training data that is likely to be found in Web crawls.

SVM: A Support Vector Machine with a linear kernel. SVMs have been applied to a large variety of classification problems.

5.6 Evaluation Setup

While our approach is independent of the respective categorical information to be inferred, we conducted an evaluation in two specific tasks: (1) predicting subtypes of *s:Event* instances, and (2) predicting genres (*s:genre*) of *s:Movie* instances.

5.6.1 Datasets

Training and test datasets were extracted from the Web Data Commons dataset of October 2016.

Event Classification: This task deals with the prediction of event subtypes. *Schema.org* distinguishes between 19 different event subtypes, such as *s:BusinessEvent* or *s:SportsEvent*. Given a generic event, the goal of this task is to predict the correct subtype of the event, i.e. to predict the object of the *rdf:type* statement.

Movie Genre Classification: *Schema.org* allows annotation of movie genres via the *s:genre* property. The goal of this classification task is to predict statements describing the *s:genre* of respective movies. Since it is possible to assign multiple genres to a single movie by defining multiple *s:genre* properties, the classification of movie genres is a *multi-label problem*, i.e. a single movie entity can belong to multiple genre classes. We address this multi-label problem by extracting individual datasets for each genre upon which a binary classifier for each genre is trained.

Balancing and Sampling

We extracted quadruples that exhibit the respective property of interest by selecting quadruples which describe nodes of *rdf:type s:Event* (*s:Movie*) and are annotated with a more specific event subtype in the case of events and the *s:genre* predicate for movies. This results into a single *Events* dataset (containing instances of all considered subtypes) and an individual dataset for each movie genre. As illustrated in Figure 5.2, the class distribution is uneven.

To obtain a balanced dataset that is sufficiently large for training of a machine learning algorithm, we applied the following steps. For *Events*, we picked the top-7 classes with the highest number of instances. We introduced an additional class containing all events not included in the top-7 classes. The classes were balanced by limiting the size of all classes to c_e, which is the size of the smallest class. For

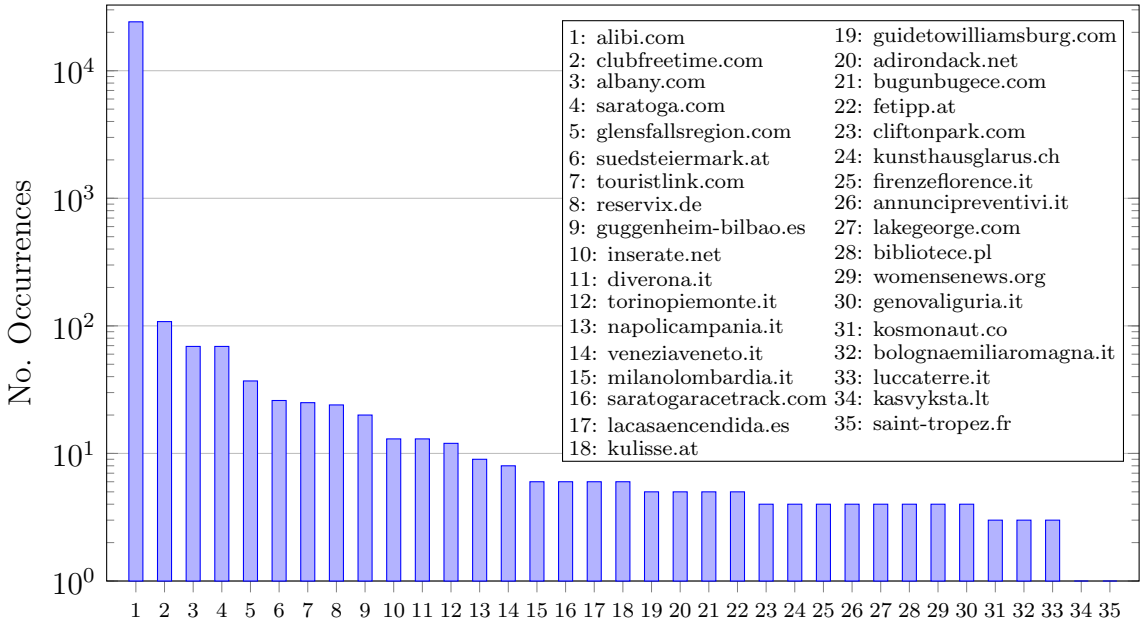


Figure 5.3. tld/pld-distribution of *s:VisualArtsEvent*s. The Y-axis is logarithmic.

Movies, we extracted 7 individual datasets corresponding to the top-7 most frequent movie genres. Each individual genre dataset includes all instances of the particular genre as well as all the remaining instances, which are labeled as "Other". The size of each genre datasets is limited to c_m, which is the size of the smallest class among all 7 datasets.

We employed two different sampling strategies:

1) *Stratified Random Sampling* simply chooses c_e (c_m) instances of each class at random from the whole dataset.

2) *pld-Aware Sampling*: Figure 5.3 depicts the pld distribution of *s:VisualArtsEvent*s. The distribution follows a power law, such that a small set of plds provides the majority of events. Random sampling may result in dropping some of the plds with fewer events and overfitting towards the patterns exhibited by very prominent plds. Therefore, we employ a sampling approach that ensures representation of long-tail entities in the sample. To this extent, we calculate a *fair share* in the sample by dividing the number of instances by the numbers of plds. We add all instances from plds that have fewer instances than the *fair share*. This process is repeated with recalculating the *fair share* with respect to the number of missing instances until the dataset contains c_e (c_m) instances of each class, where c_e (c_m) is the number of instances of the smallest class in the case of events (movies). If all remaining plds contain more instances than the *fair share*, each pld contributes the *fair share* to the final sample.

After the sampling, we split each resulting dataset in an individual training and test set (80% / 20% of the instances).

Labeling & Ground Truth

We follow a dataset-specific strategy to obtain class labels, i.e. a ground truth for training and testing. For assigning event types, we rely on the event subtypes defined within the *schema.org* type hierarchy. The class labels for events are thus explicitly given by the *rdf:type*-statements.

With respect to the prediction of movie genres, no controlled vocabulary is used consistently, whereas literals are used widely. Therefore, we map the literals to a unified genre taxonomy. We make use of the 22 genres defined by the International Movie Database (IMDB)[8]. To obtain the class labels, we check for string containment of the IMDB genre names in the literal values of the *s:genre* properties. If a genre name is a substring of the aforementioned property the genre is assigned as class label to the respective instance. Note that it is possible for one instance to exhibit multiple labels since multiple genre names may be substrings of a single *s:genre* property and, in addition, single instances may have multiple *s:genre* properties. Intuitively, this process leads to reasonable class labels for the majority of instances, such that a sufficiently large amount of correctly labeled training data can be obtained. Yet, we also anticipate a certain amount of noise. The cleansed and labeled datasets are made publicly available[9].

Table 5.2 provides an overview of the size of the extracted datasets as well as the amount of included plds. The event datasets are denoted by *Events* and contain the following classes: *PublicationEvent, MusicEvent, ScreeningEvent, ComedyEvent, TheaterEvent, EducationEvent, VisualArtsEvent, Other*. For movie genres, the genre-specific datasets are denoted by the first three letters of the respective genre as follows $\{Drama, Comedy, Action, Thriller, Romance, Documentary, Adventure\} = \{Dra, Com, Act, Thr, Rom, Doc, Adv\}$. $Movies$ refers to average values for all genres. The sampling method is denoted by the subscript, where s represents *stratified random sampling* and p represents *pld-aware sampling*.

5.6.2 Metrics

To evaluate the performance of the different classifiers, we compute the following metrics:
Precision: The fraction of the correctly classified instances among the instances assigned to one class.
Recall: The fraction of the correctly assigned instances among all instances of the class.
F1 score: The harmonic mean of recall and precision. In this chapter, we consider the F1 score to be the most relevant metric since it reflects both recall and precision.

[8] http://www.imdb.com/genre/
[9] The datasets can be found at http://markup.l3s.de.

Table 5.2. Overview of the dataset size and contained plds. Movie genres are abbreviated by their first three letters. An own dataset for each genre is extracted since each genre is treated as a binary classification problem.

Dataset	Size	Distinct plds	Avg. Instances/pld
$Events_s$	67,744	1,482	45.71
$Events_p$	67,744	2,064	32.82
Dra_s	239,030	360	663.97
Dra_p	239,030	476	502.16
Com_s	239,030	342	698.92
Com_p	239,030	476	502.16
Act_s	239,030	361	662.13
Act_p	239,030	476	502.16
Thr_s	239,030	342	698.92
Thr_p	239,030	476	502.16
Rom_s	239,030	347	688.85
Rom_p	239,030	476	502.16
Doc_s	239,030	337	709.29
Doc_p	239,030	476	502.16
Adv_s	239,030	340	703.03
Adv_p	239,030	476	502.16
$Movies_s$	239,030	347	689.30
$Movies_p$	239,030	476	502.16

5.6.3 Baselines

We compare our approach to the following baselines:

RANDOM: This baseline chooses a class at random.

SD-TYPE: This baseline leverages conditional probabilities to infer the subject types using the *SD-Type* approach [PB13]. The probabilities are based on the incoming and outgoing statements of a particular node. Since *SD-Type* was not originally designed to be applied to Web markup, we adapted it by only considering outgoing statements. This is motivated by the fact that a complete set of incoming statements can not be obtained for Web markup, where links might (but are unlikely to) originate from any Web page.

KG-B: This baseline employs a knowledge graph to obtain class labels. The *s:name* of a subject is used as input for *DBpedia Spotlight* [DJHM13] to obtain candidate entities from *DBpedia* (*dbp*). If the markup is annotated in one of the 12 languages supported by Spotlight[10], the corresponding Spotlight model is used. For all other cases we employ the English Spotlight model. Labels obtained from DBpedia may be different from labels found in Web markup (e.g. the genre of the movie "Forrest Gump" is stated to be *Drama* and *Comedy* in DBpedia, but marked as *Drama*

[10] http://www.dbpedia-spotlight.org/faq

and *Romance* on *imdb.com*). In order to avoid noisy and costly matching process, we address this issue by considering all candidates with a confidence of at least 0.5 as true positives as long as the matching entity shows the correct type (*dbp:Event or s:Event* respectively *dbp:Movie* or *s:Movie*), independent of whether or not the entity actually shows the expected categorical property. If no candidate with a suitable type is found, the instance is assigned to the "Other"-class. Note that this simplification significantly boosts the performance of this otherwise naive baseline, yet serves the purpose of illustrating the lack of sufficient coverage (Section 5.7).

5.7 Evaluation

This section presents the results on the classification performance, the influence of the sampling methods and the individual features.

5.7.1 Classification Performance

Table 5.3 summarizes the overall results of the baselines (Random, SD-Type, KG-B) as well as our proposed classification models (Naïve Bayes, Decision Tree, Random Forest, SVM) for event type (Table 5.3a) and movie genre classification (Table 5.3b). For both tasks, we report the macro averages of the results with respect to precision, recall and F1 scores for both *stratified random sampling* and *pld-aware sampling*. We observe that, for *Movies*, Random Forest, closely followed by Decision Tree, performs best across all evaluation metrics, except for precision/$Movies_s$, where it is slightly outperformed by KG-B. This is caused by the underlying assumption of the KG-B baseline that any entity match is considered as successful information inference, which unfairly boosts the baseline performance, in particular for popular entities. For *Events*, Random Forest shows the highest Recall and F1, closely followed by Decision Tree, whereas highest precision is achieved by Naïve Bayes in this case. The use of a single Decision Tree already results in relatively high F1 scores, e.g. 81.86% for $Movies_p$. Considering a Random Forest as an ensemble of Decision Trees, we conclude that additional trees only slightly improve the outcome (F1 of 83.14%). The SD-Type baseline achieves F1 scores of 56.99% for *Events*. This significant difference in performance between the baseline and our approach reflects the fundamental difference between knowledge graphs and data sourced from markup and the need to consider features beyond the structural connections of entity descriptions when dealing with markup data. For both *Events* and *Movies*, KG-B assigns the vast majority of the instances to the "Other"-class, resulting in high recall and low precision for the aforementioned class. Due to the design of the baseline, all classes different from "Other" exhibit 100% precision but very low recall, which ultimately results in low F1 scores after computing the macro average across classes.

Table 5.3. Macro averages for precision, recall, and F1 score [%] over all datasets.

(a) Event type classification

Classifier	$Events_s$			$Events_p$		
	Precision	Recall	F1	Precision	Recall	F1
Random	12.72	12.71	12.71	12.81	12.82	12.81
SD-Type	58.35	49.56	40.98	58.71	62.83	56.99
KG-B	39.06	12.59	02.96	39.06	12.59	02.96
Naïve Bayes	**86.04**	44.24	40.04	**84.06**	50.51	47.78
Decision Tree	70.60	70.26	70.15	78.70	77.78	77.25
Random Forest	73.34	**72.46**	**71.67**	80.75	**79.71**	**79.59**
SVM	75.51	70.10	67.64	81.45	78.67	77.34

(b) Movie genre classification

Classifier	$Movies_s$			$Movies_p$		
	Precision	Recall	F1	Precision	Recall	F1
Random	50.00	50.00	50.00	49.87	49.87	49.86
SD-Type	61.67	58.92	56.36	68.34	67.77	67.62
KG-B	**76.52**	55.70	44.82	76.94	57.16	47.42
Naïve Bayes	69.06	50.29	33.98	61.55	50.39	34.19
Decision Tree	72.95	72.88	72.85	82.01	81.89	81.86
Random Forest	74.62	**74.49**	**74.46**	**83.27**	**83.16**	**83.14**
SVM	72.84	72.42	72.27	81.75	81.37	81.27

For *Movies*, Table 5.3b reports the average scores of the individual genre-specific classifiers. It is worth to mention that the boundary of the classes (genres) might be fuzzy, e.g. it could be hard to differentiate a movie of genre "Thriller" from a movie of genre "Action". Since the classification of each genre is formulated as a binary classification problem, the RANDOM-baseline performance is close to 50% for all classes. The highest F1 score achieved by SD-Type is 67.62%, indicating that the subject properties used by this baseline might not be sufficient to classify movie genres precisely. Overall performance of the KG-B baseline is better in this task, driven by higher recall for instances of type movie, which are better represented in knowledge bases. Similar to our observations in the event classification task, Random Forest performs best, closely followed by Decision Tree. The F1 score of 83.14% for Random Forest significantly outperforms the baselines (paired t-test with $p < 0.01$) when comparing Random Forest against the baselines in all configurations. Overall, Random Forest classification using the features proposed in this chapter clearly outperforms the baselines in both tasks.

Table 5.4. Hyperparameters considered for optimization.

Classifier	Parameter	Range
Decision Tree	Criterion	Gini Impurity, Information Gain
	Min.Impurity Decrease	[0,1]
Random Forest	Criterion	Gini Impurity, Information Gain
	Min.Impurity Decrease	[0,1]
	No. Estimators	[5,20]
SVM	Penalty	[0,5]
	Stopping Tolerance	$[0,10^{-3}]$

Classification Hyperparameter

For each classifier used with an exception of the Naïve Bayes classifier, we determine the parameters that maximize the F1 score by employing the random search algorithm proposed by Bergstra and Bengio [BB12]. The Naïve Bayes classifier does not exhibit parameters that could be optimized. Table 5.4 gives an overview of the parameters that were considered during the optimization, whereas Table 5.5 summarizes the hyper-parameters that were determined using random search. All previously shown performance results were obtained using the specified hyper-parameters.

5.7.2 Influence of Sampling Methods

In this section, we discuss the influence of the different sampling methods. Since the Random Forest classifier achieves the best results, we investigate the effects of sampling methods on our Random Forest configuration.

Figure 5.4 shows the F1 scores with respect to the sampling method for *Events* and the individual *Movies* genre datasets. The use of *pld-aware sampling* yields up to 17% percentage points better results than the use of *stratified random sampling*.

We observe that the use of a more diverse training set (i.e. a dataset including more data from long-tail domains e.g. obtained through the *pld-aware sampling*) has a significant and beneficial effect on the classification outcome (paired t-test with $p < 0.03$).

Table 5.5. Summary of classifier hyperparameters determined with random search for the following parameters: Crit: Criterion, Imp: Min. Impurity Decrease, No: No. Estimators, Pen: Penalty, Tol: Stopping Tolerance.

Dataset	Decision Tree		Random Forest			SVM	
	Crit	**Imp**	**Crit**	**Imp**	**No**	**Pen**	**Tol**
$Events_s$	ent.	0.192	ent.	0.892	13	3.53	0.0043
$Events_p$	gini	0.527	ent.	0.892	13	1.88	0.0098
Dra_s	gini	0.360	ent.	0.938	16	0.66	0.0037
Dra_p	gini	0.360	gini	0.414	18	0.66	0.0037
Com_s	gini	0.360	gini	0.414	18	0.66	0.0037
Com_p	ent.	0.608	gini	0.160	20	0.66	0.0037
Act_s	gini	0.360	gini	0.160	20	0.66	0.0037
Act_p	ent.	0.558	gini	0.160	20	0.66	0.0037
Thr_s	gini	0.360	ent.	0.482	16	0.66	0.0037
Thr_p	ent.	0.608	ent.	0.482	13	0.66	0.0037
Rom_s	ent.	0.608	ent.	0.482	13	0.66	0.0037
Rom_p	ent.	0.287	gini	0.160	20	0.66	0.0037
Doc_s	ent.	0.192	gini	0.160	20	0.66	0.0037
Doc_p	ent.	0.099	gini	0.068	16	0.66	0.0037
Adv_s	ent.	0.287	gini	0.160	20	0.66	0.0037
Adv_p	ent.	0.287	gini	0.068	16	0.66	0.0037

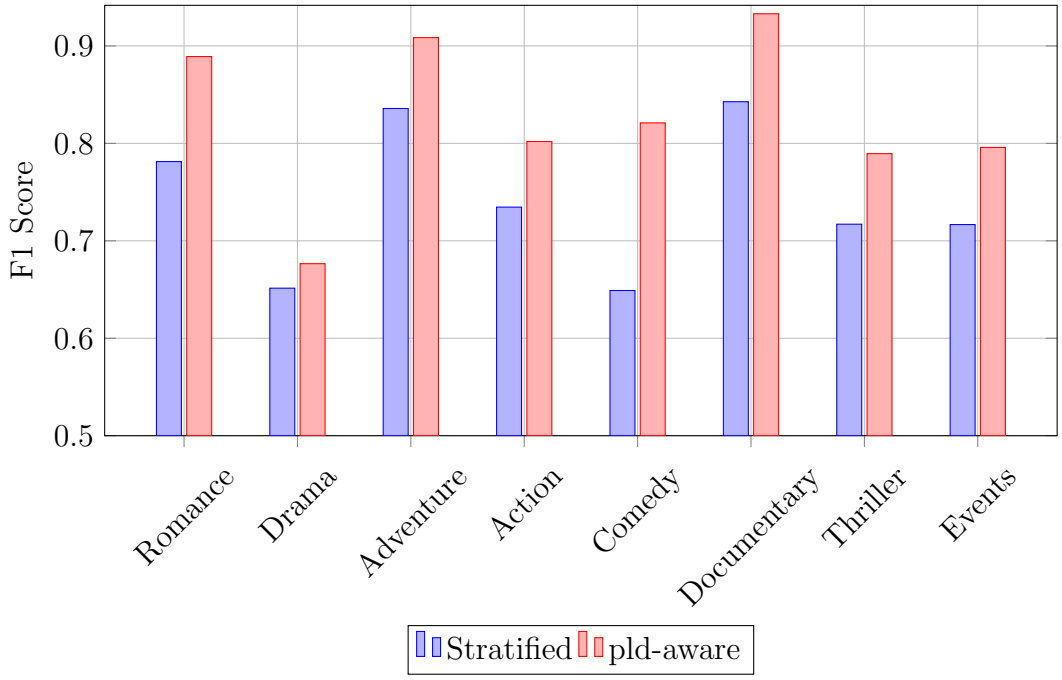


Figure 5.4. F1 scores macro averages [%] for the Random Forest classifier with respect to dataset and sampling method.

5.7.3 Influence of Features

Next, we discuss the influence of the proposed features. We focus on the best performing classifier (Random Forest) to explore the effects of varying the features.

Table 5.6 presents the F1 scores obtained through Random Forest on the *Events* dataset with respect to different feature combinations. Our results indicate that the influence of features varies strongly dependent on the respective types and classes. This seems intuitive, given that some classes might be more specifically characterized by certain features, such as a set of plds. The *tld/pld* features alone result in a reasonable performance for *Events* but not for *Movies*. This indicates that the source of the markup node is stronger correlated with its actual type or category for *Events* than for *Movies*. This seems intuitive, given that event-centred Websites tend to be more focused on certain event types than movie-centred Websites are focused on particular genres. However, these observations are likely to vary strongly dependent on the actual classification task. In contrast, the *node-vocab* alone is not sufficient to determine the event subtype with high F1 score. This observation corresponds to the insufficient performance of the *SD-Type* baseline.

The combination of *tld/pld* and *node-vocab* results only in a slight improvement of the results for $Events_p$. A dependence between the two features seems intuitive as pages extracted from the same pld are likely to be maintained by the same organization and thus typically use the same set of *schema.org* terms. For instance, an event database is likely to assign the same set of properties to each event resulting in a characteristic *node-vocabulary* for the events of a single pld. Since the *page-vocab* considers the terms that occur on the whole page, the number of considered terms is higher, which results in better chances to find usage of the same terms on other Web pages. This is reflected by the fact that both combinations of *tld/pld, page-vocab* and *page-vocab, node-vocab* lead to an improvement while the performance of *tld/pld, node-vocab* is roughly the same as *tld/pld* only. The combination of all three features yields in a slight decrease of the F1 score compared to *tld/pld, page-vocab* only, indicating once more that the information contributed by *node-vocab* is already provided by *tld/pld*.

Table 5.7 shows average F1 scores using the Random Forest classifier on the *Movies* datasets. In contrast to the *Events* datasets we can achieve relatively good performance by employing only the *node-vocab* feature. Another difference is that we can observe a slightly larger margin between the exclusive use of *node-vocab* and *tld/pld*. This indicates that markup of movies of certain genres tend to exhibit the same *schema.org* terms. Any combination of two or more features results in similar outcomes (with approximately 1 percentage point difference). In both domains we can see a substantial difference in the performance with respect to the sampling methods for all feature combinations. *pld-aware sampling* consistently achieves higher F1 scores than *stratified random sampling*, leading to the conclusion that individual features and feature combinations benefit from *pld-aware sampling*.

Table 5.6. Random Forest F1 scores macro averages [%] for different feature combinations (*Events* datasets).

Features	$\boldsymbol{Events_s}$	$\boldsymbol{Events_p}$
tld/pld	65.30	76.29
page-vocab	62.57	79.8
node-vocab	60.66	68.09
tld/pld,page-vocab	71.01	80.03
tld/pld,node-vocab	65.38	77.65
page-vocab,node-vocab	71.70	80.27
tld/pld,page-vocab,node-vocab	71.67	79.59

Table 5.7. Random Forest F1 scores macro averages [%] for different feature combinations (*Movies* datasets).

Features	$\boldsymbol{Movies_s}$	$\boldsymbol{Movies_p}$
tld/pld	66.32	80.56
page-vocab	72.27	82.14
node-vocab	73.96	82.18
tld/pld,page-vocab	72.59	82.68
tld/pld,node-vocab	74.28	82.94
page-vocab,node-vocab	74.33	82.59
tld/pld,page-vocab,node-vocab	74.46	83.14

5.7.4 Evaluation Summary

Our experiments illustrated that traditional knowledge graph completion approaches that are not specifically designed for Web markup data may not be directly applicable to this kind of data, mainly due to the sparsity of individuals and the lack of connectivity in Web markup. Moreover, we observed that it is not sufficient to consider only node-specific features such as *node-vocab* to infer missing categorical information in Web markup. In contrast, contextual features such as *tld/pld* and *page-vocab* provide important information to infer missing statements.

In particular, our experiments demonstrated that contextual features such as *tld/-pld* and *page-vocab* are discriminative for both tasks under consideration. These features are effective because many Websites focus on a particular topic, e.g. theater or music events. We observed that the *page-vocab* feature is especially useful in both tasks, as it describes the context of the particular node in a more specific way. Whereas the use of the *tld/pld* feature can naturally only be applied to instances from known plds, i.e. plds that are contained in the training data, performance drops are expected when classifying data from unknown plds. However, our results indicate that features representative for certain kinds of plds, such as *page-vocab*, can serve as

a potent substitute able to efficiently classify markup from unknown sources.

Limitations arise from the focus on two particular tasks only. We anticipate variation in performance of particular features when applying this approach to other kinds of categorical information. Similarly, considering that our ground truth has been constructed by relying on markup nodes where the sought-after information was present already on the Web, one might argue that this constraint has led to a bias towards markup nodes of generally higher quality. Additional experiments on an unconstrained and randomly selected ground truth will investigate this assumption further as part of future work.

5.8 Discussion

In this chapter, we unlocked an additional source of geographic Web information, i.e., semantic Web markup. We provided a detailed analysis of the intrinsic challenges resulting from the typical distribution of Web markup data. In particular, we observed noise, sparsity, and bias towards few prevalent properties and entity types. In contrast to knowledge graphs, markup nodes are typically isolated and not integrated with other entities within the Web page or from different data sources.

We enriched the markup by interpreting noisy and sparse Web markup and automatically inferring categorical information for particular entities. We augmented sparse markup nodes with information, which often is essential when interpreting markup and the corresponding Web pages. We leveraged a large amount of publicly available data as training data for a supervised machine learning approach. We employed Web markup specific features such as *tld/pld*, *node vocabulary* and *page vocabulary* and conducted an extensive evaluation of different classification algorithms, sampling methods and feature sets.

Our proposed configuration outperforms existing baselines significantly, with Random Forest providing the most consistent performance across classes and datasets. Our experiments, conducted on properties of events and movies, show a performance of 79% and 83% F1 score correspondingly, significantly outperforming existing baselines. We demonstrated that supervised inference can enrich entity-centric categorical information, which is essential when interpreting markup or websites in general.

6 GeoVectors: A Linked Corpus of OpenStreetMap Embeddings

After improving the quality of geographic Web information with validation and enrichment approaches, we illustrate the utility of such information. In this chapter, we present a first application, i.e., the GeoVectors corpus. GeoVectors provides ready-to-use embeddings of OpenStreetMap objects to enable the efficient development of machine learning applications using geographic Web information. GeoVectors builds on two ideas presented in Chapter 4. First, analogous to the use of key-value embeddings to address **RQ2.2**, we use unsupervised embeddings to capture the semantic of heterogeneous OSM objects. Second, we use identity links to integrate the GeoVectors corpus with established knowledge graphs.

6.1 Introduction

Today, OSM data is used in a plethora of machine learning applications such as road traffic analysis [KGG20], remote sensing [VMSTF21], and geographic entity disambiguation [TD21a]. However, as observed in Chapter 3 and 4, the effective use of OSM data with machine learning algorithms is not trivial. Factors including 1) a varying number of tags and details for specific geographic entities, 2) the lack of a well-defined ontology resulting in numerous tags with unclear semantics, and 3) missing values for any given property, substantially hinder the feature extraction for broader OSM usage in machine learning applications.

A central prerequisite to facilitate the effective and efficient use of geographic data in machine learning models is the availability of suitable representations of geographic entities. Recently, latent representations (embeddings) have been shown to have several advantages in machine learning applications, compared to traditional feature engineering, in a variety of domains [LWSY18, XYW+16, WL17]. First, embeddings can capture semantic entity similarity not explicitly represented in the data. Second,

embeddings facilitate a compact representation of entity characteristics, overall resulting in a significant reduction of memory consumption [TD21a]. Whereas much work has been performed to provide pre-trained embeddings for textual data and knowledge graphs [WZJ20, WMWG17], only a few attempts, such as [KS17], aimed to provide such latent representations for geographic entities and captured selected entities only. From the technical perspective, the creation of OSM embeddings is particularly challenging due to the large scale of OSM (more than 1430 GB of data as of June 2021[1]) and the OSM data format (*"protocolbuffer binary format"*[2]), requiring powerful computational infrastructure and dedicated data extraction procedures. Furthermore, the semi-structured data format of OSM tags requires specialized embedding algorithms to capture the semantics of entity descriptions. As a result of these challenges, currently, no datasets that capture latent representations of OSM entities exist.

The GeoVectors corpus of embeddings presented in this chapter is a significant step to enable the efficient use of extensive geographic data in OSM by machine learning algorithms. GeoVectors facilitates access to these embeddings using semantic technologies. We utilize established representation learning techniques (word embeddings and geographic representation learning) to capture various aspects of OSM data. We demonstrate the utility of the GeoVectors corpus in two case studies covering the tasks of type assertion and link prediction in knowledge graphs. GeoVectors follows the *5-Star Open Data* best practices [BL06] in data publishing and reuses existing vocabularies to lift OpenStreetMap entities into a semantic representation. We provide a knowledge graph that semantically represents the GeoVectors entities and interlinks them with existing resources such as Wikidata, DBpedia, and Wikipedia. With the provision of pre-computed latent OSM representations, we aim to substantially ease the use of OSM entities for machine learning algorithms and other applications.

To the best of our knowledge, currently, there are no dedicated resources that provide extensive reusable embeddings for geographic entities at a scale comparable to GeoVectors. The absence of comprehensive geographic data following a strict schema makes it particularly challenging to process geographic data in machine learning environments. We address these problems by providing models capable of embedding arbitrary geographic entities in OSM. Moreover, we enable easy reuse by making both models and encoded data publicly available.

Contributions. The main contributions of this chapter are as follows:

- We provide GeoVectors – a world-scale corpus of embeddings covering over 980 million geographic entities in 188 countries using two embedding models and capturing the semantic and the geographic dimensions of OSM entities.
- We introduce an open-source embedding framework for OSM to facilitate the

[1] https://wiki.openstreetmap.org/wiki/Planet.osm
[2] https://wiki.openstreetmap.org/wiki/PBF_Format

reusable embedding of up-to-date entity representations[3].

- We provide a knowledge graph to enable semantic access to GeoVectors.

The remainder of this chapter is organized as follows. In Section 6.2, we discuss related work. Then, In Section 6.3, we discuss the predicted impact of GeoVectors. In Section 6.4, we present the embedding generation framework. Next, in Section 6.5, we present the GeoVectors knowledge graph. We describe the characteristics of the GeoVectors corpus in Section 6.6. We illustrate the usefulness of GeoVectors in two case studies in Section 6.7 and discuss availability and utility in Section 6.8. Finally, in Section 6.9, we provide a discussion.

6.2 Related Work

This section discusses related work in the areas of word embeddings and knowledge graph embeddings.

Word Embeddings: A multitude of natural language processing algorithms adopts word embedding models for downstream tasks. [WZJ20] conducted a recent survey on neural word embeddings algorithms. Recent approaches like BERT [DCLT19], and ELMo [PNI+18] exploit the context information, e.g., the word order in sentences, to infer latent representations. In contrast, the fastText algorithm [JGBM17] infers the latent representation of each word individually. As OSM tags describing geographic entities neither have any natural order nor form any sentences, we choose fastText over BERT and ELMo to create embeddings.

Knowledge Graph Embeddings: Knowledge graph embeddings have recently evolved as an important area to facilitate latent representations of entities and their relations [WMWG17, MJC+20]. General-purpose knowledge graphs like Wikidata [VK14], DBpedia [LIJ+15], and YAGO [HSBW13], and even specialized KGs like EventKG [GD19] and LinkedGeoData [SLHA12] typically only include the most prominent geographic entities. Compared to OpenStreetMap, the number of geographic entities captured in such knowledge graphs is relatively low [TD21a]. For instance, as of June 2021, Wikidata contained less than 8.5 million entities with geographic coordinates, while OpenStreetMap contained more than 7 billion entities. The specific geographic entities or entity types, e.g., roads or shops, might not be relevant or prominent enough to be captured by the general-purpose knowledge graphs. Nevertheless, these entities play an essential role for various downstream applications, for instance, for land use classification [SVA+17] or in the prediction of mobility behavior [WYSG17]. Consequently, pre-trained embeddings of popular knowledge graphs, such as Wikidata [LWS+19] or DBpedia, lack coverage of geographic entities required by spatio-temporal analytics applications. In contrast, the GeoVectors embeddings

[3] https://github.com/NicolasTe/GeoVectors

proposed in this chapter specifically target geographic entities and ensure adequate coverage in the resulting dataset.

6.3 Predicted Impact

GeoVectors is a new resource. This section discusses the predicted impact of GeoVectors regarding the advances of state of the art in geographic embedding datasets, geographic information retrieval, machine learning applications, knowledge graph embeddings and broader adoption of semantic web technologies.

Advances of the state of the art: We advance the state of the art by providing the first large-scale corpus of pre-trained geographic embeddings. We carefully select established representation learning techniques to capture both the semantic dimension (What entity type does the OSM entity represent?) and the spatial dimension (Where is the entity located?) and adapt these techniques to OSM data to create meaningful latent representations. The GeoVectors corpus is the first dataset that captures the entire OpenStreetMap, thus offering the data on the world scale. Therefore, GeoVectors is significantly larger than any existing geographic embedding resources. For instance, the Geonames embedding [KS17] provides a dataset containing less than 358 thousand entities, whereas GeoVectors contains over 980 million entities.

Impact on geographic information retrieval: Geographic information retrieval (GIR) is a field focussing on addressing geographic information needs [PCJ+18]. Recent GIR approaches build on geographic embeddings to address several use cases, including tag recommendation for urban complaint management [GHW+19], geographic question answering [CGMS21], and POI categorization [TKS20]. While these approaches demonstrate the utility of geographic embeddings for GIR tasks, the laborious generation process hinders the adaption of geographic embeddings for other GIR tasks such as geographic named entity recognition, next location recommendation, or geographic relevance ranking. In this context, the availability of large-scale and accessible geographic embeddings is a vital prerequisite to stimulate research in the GIR field. The GeoVectors corpus presented in this chapter addresses these requirements by providing ready-to-use geographic embeddings of the entire OpenStreetMap.

Impact on machine learning applications: Existing machine learning applications use geographic data to address numerous use cases including location recommendation [LWSY18, XYW+16], human mobility prediction [WL17], and travel time estimation [WLFY21]. The variety of use cases highlights the general importance of geographic information for machine learning models. However, these approaches conduct a costly feature extraction process or learn supervised embeddings of geographic entities on task-specific datasets for specific tasks. In this context, the availability of easy-to-use representations of geographic entities at scale provided by GeoVectors is crucial to enabling and easing the further development of geographic machine learning models and geographic algorithms.

Impact on knowledge graph embeddings: Knowledge graph embeddings generated without the specific focus on geographic entities have shown success in a large variety of knowledge graph inference and enrichment tasks, including type assertions and link prediction [Pau17]. We envision that GeoVectors can further enhance the quality of embeddings used in the context of these tasks: While geographic entities are part of many popular knowledge graphs such as Wikidata and DBpedia, their specific characteristics are still rarely considered. Existing approaches typically focus on the graph structure, but rarely on the property values assigned to the single nodes [KKL+19]. However, both tags and coordinates of geographic entities bear valuable semantics. Specifically, the geographic interpretation of coordinates may heavily lift the role of coordinates in knowledge graph embeddings. In the future, the GeoVectors embeddings can directly support knowledge graph inference and enrichment and creation of geographically aware embeddings from other sources.

Impact on adoption of semantic web technologies: In the context of the Semantic Web, a variety of models and applications, including link prediction, creation of domain-specific knowledge graphs [GD19] and Question Answering for event-centric questions [CGD20] make use of geographic data. Semantic technologies have been applied to a variety of domains that require spatio-temporal data, including crime localization, transport data, and historical maps [RP20, SCCC20, SKD+20]. Furthermore, with the increased availability of mobile devices, location-based algorithms such as next location recommendation or trip planning evolved. Recently, SPARQL extensions for integrated querying of semantic and geographic data have been proposed [HSJ20]. In this context, the availability of easy-to-use representations of geographic entities at scale is crucial to enable further development of semantic models and geographic algorithms and their adoption in real-world scenarios. Increasing availability of geographic data accessible through semantic technologies, as facilitated by GeoVectors, and seamless integration of this data with other semantic data sources in the Linked Data Cloud can attract interested users from various disciplines and application domains, including geography, mobility, and smart cities.

6.4 Framework for Embedding Generation

The GeoVectors framework facilitates the generation of OSM embeddings that capture geographical (*GV-NLE*) and semantic (*GV-Tags*) similarity of OSM entities. In this section, we first describe the OSM data model. Then, we provide an overview of the GeoVectors embedding generation process and present embedding algorithms that generate the proposed *GV-NLE* and *GV-Tags* embeddings.

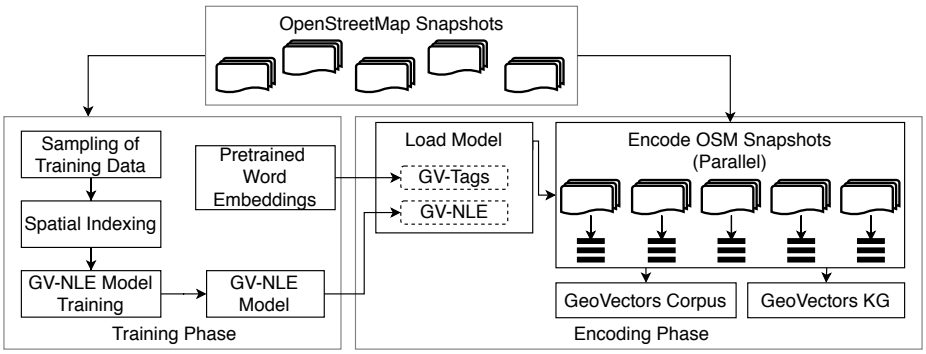


Figure 6.1. Overview of the embedding generation process.

6.4.1 GeoVectors Embedding Generation Overview

The GeoVectors embeddings reflect semantic and geographic relations of OSM entities, where semantic relations capture semantic entity similarity, expressed through shared annotations, and geographic relations capture geographic entity proximity. In general, the relevant relation type is application-dependent. Therefore, we compute two embedding datasets, one capturing geographic and the other semantic similarity of OSM entities:

- (1) *GV-NLE* is our geographic embedding model based on the Neural Location Embeddings (NLE) [KS17] – an approach to capture the spatial relations of geographic entities.
- (2) *GV-Tags* is our semantic embedding model based on fastText [JGBM17] – a state-of-the-art word embedding model that we apply on the OSM tags.

The embedding generation process that takes as input a set of OSM snapshots and generates the GeoVectors corpus and the GeoVectors knowledge graph is illustrated in Figure 6.1. We divide this process into the *training* phase in which we train the GV-NLE model and the *encoding* phase in which we apply embedding models to encode OSM entities.

The training of an embedding model is typically significantly more expensive than the application of the model. Due to the large scale of OpenStreetMap (as of June 2021, OSM contains more than 7 billion entities), the training of embedding models on the entire corpus is not feasible. Therefore, in the *training* phase, we sample a subset of OSM entities from OSM snapshots to serve as training data. Formally, we define an *OSM snapshot* s taken at a time t in a region r by $s = \langle O, t, r \rangle$, where O is a set of OSM objects within the specified region r at this time. We discuss the sampling process in Section 6.4.2. Based on the sampled data, we train our embedding models. To generate semantic embeddings, we utilize existing pre-trained word embedding models.

In the *encoding* phase, we first load the trained embedding model and then pass all individual entities from an OSM snapshot to the model. The application of the model can be parallelized by applying the model to each snapshot separately. The model encodes the OSM entities and stores the generated embedding vectors into an easily processable, tab-separated value file.

We provide an open-source implementation of the embedding framework, including the pre-trained embedding models[4]. This framework enables the computation of up-to-date embeddings of individual OpenStreetMap snapshots. We also generate the GeoVectors knowledge graph that enables semantic access to GeoVectors and is described in detail in Section 6.5.

We performed the entire extraction and embedding process on a server with 6 TB of memory and 80 Intel(R) Xeon(R) Gold 5215M 2.50GHz CPU cores. Our framework required about four days for data extraction, model training, and data encoding.

6.4.2 Sampling of OSM Training Data for Embedding Algorithms

At the beginning of the training phase, we extract a representative entity subset to use as a training set. To ensure representativeness, we employ the following conditions: First, the training set should have a balanced geographic distribution to avoid biases towards specific geographic regions. Second, the training set should only include meaningful OSM entities. For instance, many OSM nodes do not provide any tags and only represent spatial primitives for composite entities, such as ways and relations. Such nodes do not correspond to real-world entities and, taken isolated, do not convey any meaningful information. Therefore, we exclude nodes without tags from the training data.

Algorithm 2 presents the sampling process to obtain training data. The input of the algorithm consists of a minimum number n of training samples to be collected and a corpus of OpenStreetMap snapshots $\mathcal{S}$ (e.g., country-specific snapshots). First, we calculate the total geographic area covered by all snapshots using the `geo_area`$(s.r)$ function (line 1), where $s.r$ denotes the region of the OSM snapshot s. To enforce a uniform geographic distribution, we calculate the number of samples extracted from a single snapshot regarding its geographic size. For each snapshot, we determine the number of samples n_s to be extracted proportionally to the geographic area of the snapshot (line 4). Then, the `scan_snapshot` function divides the snapshot into *linked*, *tagged* and *other* entities (line 5). Linked entities provide an identity link to external datasets. As identity links typically indicate good data quality, our algorithm includes all linked entities. Tagged entities provide at least one tag. Other entities are entities that neither provide an identity link nor a tag. Next, the algorithm samples all linked entities (even if their number exceeds n_s) into the result set $\mathcal{T}$ (line 6). If

[4] https://github.com/NicolasTe/GeoVectors

Algorithm 2: Sample Training Data

Input : $\mathcal{S}$: OpenStreetMap snapshots
n: Minimum number of training examples
Output: $\mathcal{R}$: Set of training examples

```
1  total_area ← ∑_{s∈S} geo_area(s.r);
2  R ← {};
3  forall s ∈ S do
4  |  n_s ← n · geo_area(s.r) / total_area;
5  |  linked, tagged, other ← scan_snapshot(s);
6  |  T ← linked;
7  |  if |T| < n_s then
8  |  |  T ← T ∪ sample(tagged, (n_s − |T|));
9  |  end
10 |  if |T| < n_s then
11 |  |  T ← T ∪ sample(other, (n_s − |T|));
12 |  end
13 |  R ← R ∪ T;
14 end
15 return R
```

the size of $\mathcal{T}$ does not reach n_s, the function `sample` uniformly selects at maximum $n_s - |\mathcal{T}|$ random samples from the tagged entities (lines 7-9). If n_s is still not reached, we sample the remaining examples from the other entities (lines 10-12). Finally, the algorithm returns the union of all snapshot-specific training examples $\mathcal{R}$ (lines 13-15).

6.4.3 *GV-NLE* Embedding of OSM Entity Locations

The GV-NLE model builds on the neural location embedding (NLE) model [KS17] that captures the geographic relations of a set of geographic entities in a latent representation. The NLE method is an established method to create reusable geographic embeddings. GV-NLE extends the NLE model with a suitable encoding algorithm to encode previously unseen OSM entities.

Training: GV-NLE first constructs a weighted graph representing OSM entities and their mutual distances. The OSM entities form the nodes of the graph. The edges encode the geographic distance between OSM entities. For each node n, GV-NLE constructs edges between n and the k geographically nearest neighbor nodes. Following [KS17], we set $k = 50$. The edge weights represent the haversine distance between two nodes in meters, which measures the geographic distance of two points while taking the earth's curvature into account. To facilitate an effective distance computation between OSM entities, we employ a Postgres database that provides spatial indexes. Based on the graph, a weighted DeepWalk algorithm [PAS14] learns

the latent representations of the OSM nodes. GV-NLE computes a damped weight $w' = max(1/ln(w), e)$, where w denotes the original edge weight, ln the natural logarithm, and e Euler's number. The use of damped weights further prioritizes short distances between the nodes. The normalized damped weights serve as a probability distribution for the transition probabilities of the random walk within the DeepWalk algorithm.

Encoding: As the original NLE algorithm does not generalize to unseen entities, i.e., entities that are not part of the training set, we extend the NLE model with a suitable encoding algorithm. The idea of the GV-NLE encoding is to infer a representation of an entity from its geographically nearest neighbors. We calculate the weighted average of the latent representation of the geographically nearest $k = 50$ entities in the training set.

$$GV\text{-}NLE(o) = \frac{1}{\sum_{o' \in N_o} w(o, o')} \sum_{o' \in N_o} w(o, o') \cdot NLE(o')$$

Here, o denotes an OSM entity, $NLE(o')$ denotes the latent representation of an entity o' according to the NLE algorithm, N_o denotes the set of the k geographically nearest OSM entities in the training set. We define the weighting term $w(o, o')$ as

$$w(o, o') = ln(1 + \frac{1}{dist(o, o')})$$

where $dist(o, o')$ denotes the geographic distance between o and o'. $w(o, o')$ assigns a higher weight to geographically closer entities. We apply a logarithm function to soften high weights of very close entities.

6.4.4 *GV-Tags* Embedding of OSM Entity Tags

To infer the *GV-Tags* representations, we adopt fastText, a state-of-the-art word embedding model that infers the latent representation of single words individually [JGBM17]. As the tags of OSM entities do not have any natural order, we chose fastText to embed them.

Training: Pre-trained word vectors are available at the fastText website[5]. As most of the OSM keys are in English, we chose the 300-dimensional English word vectors trained on the Common Crawl, and Wikipedia [GBG+18].

Encoding: To encode an OSM entity o, we utilize the individual word embeddings of the keys and values that form the entity tags $o.T$. We map entities without any tags to a vector of zeros.

$$GV\text{-}Tags(o) = \begin{cases} \frac{1}{2|o.T|} \cdot \sum_{\langle k,v \rangle \in o.T} ft(k) + ft(v), & if\ |o.T| > 0 \\ \{0\}^{300}, & \text{otherwise.} \end{cases}$$

[5] https://fastText.cc/

Here, $\{0\}^{300}$ denotes a 300-dimensional vector of zeros, and $ft(x)$ denotes the fastText word embedding of x.

6.5 GeoVectors Knowledge Graph

Semantic access to the GeoVectors embeddings is of utmost importance to facilitate the use of the dataset in downstream semantic applications. Therefore, GeoVectors is accompanied by a knowledge graph that models the embedding metadata. This metadata facilitates interlinking of the embeddings with established knowledge graphs such as Wikidata and DBpedia using existing entity links. This way, the GeoVectors embeddings can be used to enrich geographic entities in these knowledge graphs. The GeoVectors knowledge graph includes more than 28 million triples and is made available under a public SPARQL endpoint[6].

The GeoVectors knowledge graph is based on three established vocabularies. We utilize the LinkedGeoData [SLHA12] and the Basic Geo vocabulary[7] to model the spatial dimension of geographic entities, as well as the PROV Ontology [BCC+13] for modeling data provenance, i.e., where the geographic entities were extracted from and what they represent. Figure 6.2 illustrates the schema of the GeoVectors knowledge graph, including its prefixes and namespaces.

Each geographic entity in the knowledge graph is typed as `geovec:EmbeddedSpatialThing`, which encapsulates the classes `geo:SpatialThing` and `prov:Entity`. We group the relevant properties shown in Figure 6.2 regarding these three classes:

- `geo:SpatialThing`: Each geographic entity is either a node, a way or a relation and assigned to the respective *LinkedGeoData* class. In addition, the GeoVectors knowledge graph provides the entity's latitude and longitude.
- `prov:Entity`: For tracking the origins of an embedding, each geographic entity is linked to the dataset it is extracted from (`prov:Collection`). Through versioning of these datasets, the GeoVectors corpus and the GeoVectors knowledge graph can be extended in future versions.
- `geo:EmbeddedSpatialThing`: The geographic entities are linked to other resources representing the same (`owl:sameAs`) or a related resource (`dcterms:related`) in Wikidata, DBpedia and Wikipedia.

Listing 6.1 presents the triples describing the geographic entity representing the city of Berlin. These triples provide the geolocation of Berlin, references to its counterparts in Wikidata, DBpedia, Wikipedia and OpenStreetMap, as well as provenance information (the embeddings were extracted from an OSM snapshot from November

[6] http://geovectors.l3s.uni-hannover.de/sparql
[7] https://www.w3.org/2003/01/geo/wgs84_pos

2020). Access to the *GV-Tags* and *GV-NLE* embedding is enabled through the Zenodo DOIs 10.5281/zenodo.4321406 and 10.5281/zenodo.4957746 pointing to the *GV-Tags* and *GV-NLE* embeddings, the entity type (`lgd:Node`) and its identifier (240109189).

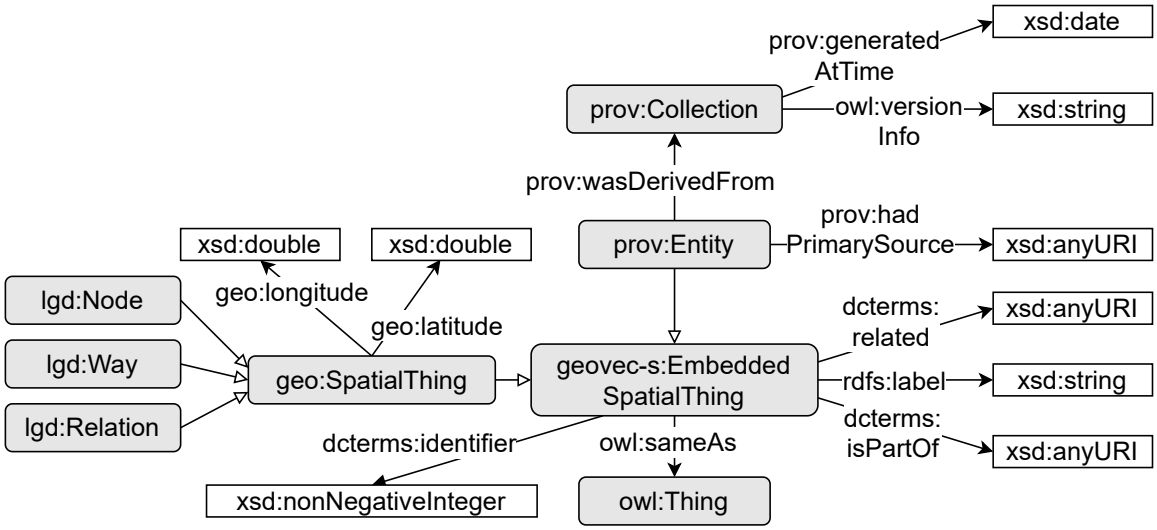


Figure 6.2. Schema, prefixes and namespaces of the GeoVectors knowledge graph. ⇾ marks a `rdfs:subClassOf` relation, → denotes the domain and range of a property.

Listing 6.1: RDF representation of Berlin in the GeoVectors Knowledge Graph.

```
geovec:v2_n_240109189 a geovec-s:EmbeddedSpatialThing;
  a lgd:Node;
  geo:longitude "13.3888599"^^xsd:double;
  geo:latitude "52.5170365"^^xsd:double;
  dcterms:identifier 240109189;
  rdfs:label "Berlin";
  dcterms:isPartOf <https://doi.org/10.5281/zenodo.4321406> ;
  dcterms:isPartOf <https://doi.org/10.5281/zenodo.4323008> ;
  owl:sameAs <https://www.wikidata.org/wiki/Q64>;
  dcterms:related
    <https://de.wikipedia.org/wiki/Berlin>;
  dcterms:related
    <http://de.dbpedia.org/resource/Berlin >;
  prov:hadPrimarySource
```

```
    <https://www.openstreetmap.org/node/240109189>;
  prov:wasDerivedFrom geovec:v2/collection.
geovec:v2/collection a prov:Collection;
  prov:generatedAtTime "2020-11-10"^^xsd:date;
  owl:versionInfo "1.0".
```

6.6 GeoVectors Embedding Characteristics

In GeoVectors V1.0, we extracted representations of nodes, ways, and relations from OpenStreetMap snapshots at country-level from October 2020[11]. We capture all OSM entities having at least one tag. Entities without any tags typically represent geometric primitives that isolated carry no semantics. Compound OSM entities such as ways and relations typically subsume such geometric primitives and are better suited for the representation. Table 6.1 summarizes the number of extracted representations regarding their geographic origin. In addition, Figure 6.3 provides a visualization of the geographic coverage of the GeoVectors corpus. Overall, we observe high geographic coverage. In total, GeoVectors contains representations of over 980 million OpenStreetMap entities.

The most significant fraction of extracted representations is located in Europe (430 million), followed by North America (240 million) and Asia (150 million). The number of representations per region follows the distribution of available volunteered information in OpenStreetMap, most prominent in the regions mentioned above. Nevertheless, GeoVectors provides a considerable amount of entity representations for the remaining regions, e.g., 97 million entities for Africa. We believe that this amount of data is sufficient for many real-world applications.

Table 6.1. Number of OSM entities contained in GeoVectors by region.

Continent	No. Nodes	No. Ways	No. Relations	Total
Africa	$9.6 \cdot 10^6$	$8.7 \cdot 10^7$	$2.4 \cdot 10^5$	$9.7 \cdot 10^7$
Antarctica	$6.9 \cdot 10^3$	$8.4 \cdot 10^4$	$9.2 \cdot 10^3$	$1.0 \cdot 10^5$
Asia	$1.5 \cdot 10^7$	$1.8 \cdot 10^5$	$6.7 \cdot 10^5$	$1.5 \cdot 10^8$
Australia/Oceania	$5.2 \cdot 10^6$	$7.6 \cdot 10^6$	$1.7 \cdot 10^5$	$1.3 \cdot 10^7$
Europe	$9.6 \cdot 10^7$	$3.2 \cdot 10^8$	$5.6 \cdot 10^6$	$4.3 \cdot 10^8$
Central-America	$4.4 \cdot 10^5$	$4.1 \cdot 10^6$	$1.6 \cdot 10^4$	$4.6 \cdot 10^6$
North-America	$5.1 \cdot 10^7$	$1.9 \cdot 10^8$	$1.8 \cdot 10^6$	$2.4 \cdot 10^8$
South-America	$8.3 \cdot 10^5$	$2.6 \cdot 10^7$	$3.9 \cdot 10^5$	$3.5 \cdot 10^7$
Total	$1.8 \cdot 10^8$	$7.8 \cdot 10^8$	$9.1 \cdot 10^6$	$9.8 \cdot 10^8$

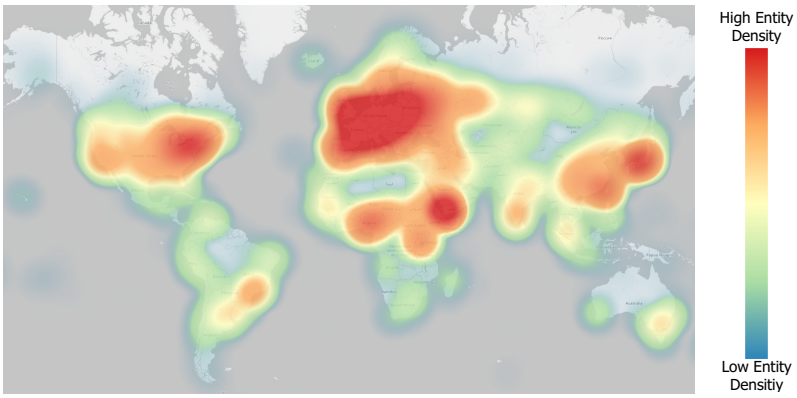


Figure 6.3. Heatmap visualization of geographic embedding coverage. Map image: ©OpenStreetMap contributors, ODbL.

6.7 Case Studies

To illustrate the utility of the GeoVectors embeddings, we have conducted two case studies dealing with the type assertion and link prediction tasks. These case studies were selected to demonstrate how widely adopted machine learning models can benefit from the GeoVectors embeddings based on semantic and geographic entity similarity. Other potential use cases include but are not limited to next trip recommendation, geographic information retrieval, or functional region discovery.

In both case studies, we use the same widely adopted classifiers: The Random Forest model is a standard random forest classifier. We use the implementation provided by the scikit-learn library[8] with the default parameters. The Multilayer Perceptron model is a simple feed-forward neural network. The hidden network layers have the dimensions [200, 100, 100] and use the ReLu activation function. The classification layer uses the softmax activation function. The network is trained using the Adam optimizer and a categorical cross-entropy loss. We use the default parameters from the Keras API[9]. As the purpose of the case studies is to demonstrate the utility of GeoVectors, rather than achieving the highest possible effectiveness of the models, we adopt the default model hyper-parameters without any further optimization.

6.7.1 Case Study 1: Type Assertion

The goal of this case study is to assign Wikidata classes to OSM entities, which aligns well with the established task of completing type assertions in knowledge graphs [Pau17]. We expect that this case study particularly benefits from the semantic dimension of the OSM entities as captured by the *GV-Tags* embeddings.

[8] https://scikit-learn.org/
[9] https://keras.io/

Test and training dataset generation: To obtain a set of relevant Wikidata classes, we first extract all OSM entities that possess an identity link to Wikidata. All Wikidata classes that are assigned to at least 10,000 OSM entities are selected for this case study. This way, we obtain 32 Wikidata classes, including "church building"[10] and "street"[11], as well as more fine-grained classes such as "village of Poland"[12]. Finally, we balance the classes by applying random under-sampling and split the data into a training set (80%, 285k examples) and a test set (20%, 71k examples).

Performance: Table 6.2 presents the classification performance of the Random Forest and Multilayer Perceptron models using *GV-Tags* and *GV-NLE* in terms of precision, recall and F1-score.

As expected, we observe that the *GV-Tags* embeddings achieve a better performance than the *GV-NLE* embeddings concerning all metrics. In particular, *GV-Tags* achieves an F1-score of 85.95% and 83.43% accuracy using the Multilayer Perceptron model. The Random Forest model using *GV-NLE* embeddings reaches an F1-score of 50.17%. This result can be explained by a few classes such as "village of Poland" that are correlated with a location. The results of this case study confirm that the semantic proximity information is appropriately captured by the *GV-Tags* embeddings.

6.7.2 Case Study 2: Link Prediction

This case study aims to assign OSM entities to their countries of origin. This task is a typical example of link prediction, where the missing object of an RDF triple is identified [Pau17]. We expect that this case study particularly benefits from the *GV-NLE* embeddings based on geographic proximity.

Test and training dataset generation: To obtain a set of countries, we sample OSM entities from the country-specific snapshots[13] as described in Algorithm 2 and preserve the origin information. In analogy to case study 1, we select all countries with at least 10,000 examples and obtain 88 different countries. Again, we balance the examples by applying random under-sampling and split the data into a training set (80%, 687k examples) and a test set (20%, 171k examples).

Performance: Table 6.3 presents the classification performance of the Random Forest and Multilayer Perceptron models using *GV-Tags* and *GV-NLE* in terms of precision, recall, and F1-score. As expected, we observe that *GV-NLE* achieves a better performance than the *GV-Tags* embeddings concerning all metrics on this task. In particular, the *GV-NLE* embeddings achieve an F1-score of 96.03% and 94.80% accuracy using the Multilayer Perceptron classification model. In contrast, the *GV-Tags* embeddings achieve an F1-score of only 29.91% and 20.20%

[10] https://www.wikidata.org/wiki/Q16970
[11] https://www.wikidata.org/wiki/Q79007
[12] https://www.wikidata.org/wiki/Q3558970
[13] Country-specific snapshots are available at https://download.geofabrik.de/.

Table 6.2. Precision, recall and F1-score (macro averages) and accuracy [%] of type assertion.

	GV-Tags				*GV-NLE*			
	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
Random Forest	92.80	69.07	77.37	69.06	70.97	41.53	50.17	41.53
Multilayer Perceptron	90.18	83.41	85.95	83.43	63.70	36.68	41.69	36.66

Table 6.3. Precision, recall and F1-score (macro averages) and accuracy [%] of link prediction.

	GV-Tags				*GV-NLE*			
	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy
Random Forest	84.38	20.25	29.91	20.28	99.08	89.79	93.67	89.78
Multilayer Perceptron	86.68	17.21	25.39	17.23	96.03	94.89	95.39	94.89

accuracy on this task because the OSM tags of an OSM entity are rarely related to its country of origin. The results of this case study confirm that the *GV-NLE* embeddings appropriately capture geographic proximity.

6.8 Availability & Utility

The GeoVectors website[14] provides a dataset description, the embedding framework as well as pointers to the following resources:

- GeoVectors embeddings: We provide permanent access to the GeoVectors embeddings and the trained models on Zenodo under the Open Database License[15]. To facilitate efficient reuse, we provide embeddings in a lightweight *TSV* format.
- The GeoVectors knowledge graph described in Section 6.5 can be queried through a public SPARQL endpoint that is integrated into the GeoVectors website[4]. In addition, we provide an interface for the label-based search of knowledge graph resources. The resources can be accessed both via HTML pages and via machine-readable formats. A machine-readable VoID description of the dataset is provided and integrated into the knowledge graph. New dataset releases will imply knowledge graph updates, where each release is accompanied by a new instance of `prov:Collection`.
- The GeoVectors embedding generation framework presented in Section 6.4 is available as open-source software on GitHub[2] under the MIT License.

[14] http://geovectors.l3s.uni-hannover.de/
[15] https://opendatacommons.org/licenses/odbl/

In the beginning of Chapter 6, we have presented the benefits of using geographic embeddings in a variety of domains [LWSY18, XYW+16, WL17]. With GeoVectors, we aim at providing access to easily reusable embeddings of geographic entities that can directly support tasks in these and other domains. Due to the task-independent nature of our embedding generation framework, we envision high generalizability of GeoVectors in a variety of application scenarios.

For demonstrating the effectiveness of the GeoVectors embeddings in different scenarios, we have conducted two case studies presented in Section 6.7, which illustrate that the GeoVectors embeddings adequately capture both the semantic and geographic similarity of OSM entities. Therefore, we believe that GeoVectors eases the use of OSM data and is of potential use for many machine learning and semantic applications that rely on geographic data. Finally, the GeoVectors framework can be reused to infer embeddings from arbitrary OSM snapshots.

For sustainability and compliance with up-to-date OSM data, we plan yearly releases of new embeddings versions.

6.9 Discussion

In this chapter, we presented GeoVectors – a linked open corpus of OpenStreetMap embeddings. GeoVectors is entirely constructed from geographic Web information. GeoVectors contains embeddings of over 980 million OpenStreetMap entities in 180 countries that capture both their semantic and geographic entity similarity.

GeoVectors constitutes a unique resource of geographic entities concerning its scale and its latent representations. We conducted two case studies, i.e., type assertion and link prediction, that demonstrated the capability of the latent representations to capture semantic and geographic properties for machine learning models.

Further, we follow best practices in data publishing and integrate GeoVectors with other data sources of geographic information on the Web using identity links. The GeoVectors knowledge graph provides a semantic description of the corpus and includes identity links to Wikidata and DBpedia. We further provide an open-source implementation of the proposed GeoVectors embedding framework that enables the dynamic encoding of up-to-date OpenStreetMap snapshots for specific geographic regions.

7 Application to Event Impact Prediction

This chapter presents a second application scenario for geographic Web information. In particular, we consider the problem of predicting the impact of public special events on road traffic. Such predictions are especially challenging because they require the combination of various information sources. First, event information is necessary to recognize the taking place of an event. Web markup, discussed in Chapter 5 is a possible source of event information. Second, road network information is essential to analyze and quantify the event's impact. OpenStreetMap, discussed in Chapter 3 and 4, is a prominent source for road network data. Third, context information, e.g., the event venue capacity, is vital to facilitate accurate predictions. Such information is often provided by knowledge graphs and is accessible via links as discussed in Chapter 4. We combine these information sources to measure the event impact on road traffic. Finally, we facilitate the prediction of event impact using a supervised machine learning model.

7.1 Introduction

Mobility behavior in urban areas is influenced by a wide variety of factors, such as seasonal and time-dependent patterns, weather conditions, construction sites, and, in particular, planned special events such as concerts, fairs or sports matches. In practice, mobility service providers, urban planners or citizens rely on basic heuristics to estimate mobility behavior, usually taking into account temporal differences only, i.e. other prevalent factors are still widely ignored when planning routes or estimating mobility needs in the short-, medium-, or long-term [JLB16].

Data about mobility behavior as well as influential factors is being generated at unprecedented scale. This includes floating car data (FCD) generated by built-in GPS devices, route planning requests to public transportation apps as well as navigation systems, or data reflecting contextual factors, such as weather conditions

or planned special events. In particular, Web mining can surface unprecedented data to capture, understand, explain and predict mobility behavior. Whereas data becomes increasingly available, it is however, usually incomplete and highly heterogeneous, posing significant challenges with respect to cleansing, smoothing or integration when aiming to build accurate mobility models.

Recent research has recognized the potential arising from the widespread availability of mobility data, e.g. to investigate the impact of traffic incidents on road networks [PDGS15] and to predict the impact of major urban events on public transportation usage [RBRP17] or on road traffic [KMN14]. However, while such events differ significantly with respect to their scale, venue, scope, type, or audience, limited research exists on understanding the impact of event characteristics on mobility needs in urban areas. In addition, no established models are available for computing the event-induced load, i.e. impact on particular units or subgraphs of a transportation graph, such as a road network.

In this chapter, we present a supervised approach to predict the spatial and temporal dimensions of the impact of planned special events on road traffic. We utilize a range of features to characterize events, mobility behavior as well as urban infrastructure. We apply our approach to historic data about mobility behavior containing over 195 million records as well as data about influential events in the city of Hanover (Germany), spanning a time period from October 2017 to January 2018 in which 150 major urban events took place. Our results demonstrate that, we consistently outperform both naïve and established baselines, demonstrating an error reduction of up to 27% with respect to mean absolute error (MAE).

Applications of this work include augmentation of route planning apps and journey recommendations with event-driven traffic predictions, including areas and specific paths in a transportation graph affected by the urban events, or the long-term understanding of event impact on mobility patterns to enable more precise planning of urban mobility infrastructures and services.

Contributions. The key contributions of this chapter include:

- a novel formalization and metrics for computing the spatial and temporal dimensions of impact of planned special events on road traffic,
- an algorithm to identify subgraphs of the transportation graph that are typically affected by planned special events,
- the assessment of features and supervised models that are well suited for both the available data and the considered regression problems,
- insights into the characteristics of event impact and experimentally identified thresholds able to distinguish event-induced traffic load as opposed to periodic or temporal traffic fluctuations.

The remaining chapter is organized as follows: Section 7.2 discusses related work. Then, Section 7.3 introduces the overall problem of predicting the impact of planned special events. Section 7.4 proposes dedicated metrics for the prediction of the spatial dimension of event impact on road traffic, while Section 7.5 proposes respective metrics for the temporal dimension of event impact. Next, Section 7.6 introduces our supervised approach for impact prediction, i.e. the considered features and regression models. Section 7.7 introduces our evaluation setup, while Section 7.8 provides a case study on typically event-affected road networks. Performance results and feature analysis for the spatial and temporal dimensions of event impact are presented in Section 7.9 and Section 7.10, respectively. Section 7.11 summarizes key findings and limitations and briefly discusses future work, while Section 7.12 provides an overall discussion.

7.2 Related Work

In the following we present related work in the areas of impact of planned special events on road traffic and public transportation, analysis of urban road networks, road traffic forecasting and impact of incidents in more detail.

Impact of planned special events on road traffic. To the best of our knowledge, the only approach that directly addresses the task of prediction of impact of planned special events on urban traffic tackled in this chapter is the *category-based modelling approach* (CBMA) proposed by Kwoczek et al. [KMN14]. CBMA is a simple average model, which predicts the spatial impact of events on road traffic by computing averages for each event category. The impact is defined as an average traffic delay on streets located within the 500m distance of the event venue. In contrast, in this chapter we develop specialized machine learning models for impact prediction. We adopt [KMN14] as a baseline in our experiments and demonstrate superior capacity of our models with respect to [KMN14].

Further aspects of event impact. Several research works addressed other aspects of planned special events in urban areas and analyzed the corresponding traffic situations. Whereas the problems addressed by these articles are related to traffic in the context of events, they address more specific aspects and, in contrast to our work, do not aim to capture the overall event impact. For example, in [KMN15] the authors employed an artificial neural network to identify road segments typically affected by events that take place in a particular venue. Lécué et al. employed semantic technologies to develop STAR-CITY, a system for traffic prediction and reasoning [LTH+14], used for spatio-temporal analysis of the traffic status as well as for the exploration of contextual information such as nearby events. [LWFZ18] investigated the general predictability of location-based social network data. They conducted a case study on Foursquare datasets, which is a service on which users can indicate their geographic location, i.e. user can indicate that they are at a certain event venue. The

authors do not focus on a specialized prediction task, but provide general insights on working with the aforementioned data. [JLB16] provides a checklist with measures to improve traffic flow in the context of events, e.g. through reserving parking spots or guiding the traffic on particular roads. These works do not aim at predicting an overall event impact and thus are orthogonal to our research.

Impact of planned special events on public transportation. Planned special events can also impact public transportation. Pereira et al. investigated non-habitual overcrowding of public transportation by using information from social networks and specialized event websites [PRPB15]. They proposed a probabilistic model that divides an overcrowding behavior into explanatory components. [RBRP17] proposed a Bayesian additive model that predicts the total number of public transportation trips to event venues. [NHG17] detects events from social media by employing a hashtag-based algorithm. The event information extracted from the social media is then used for prediction of the public transportation flow. These studies focus on public transportation and are orthogonal to the prediction of event impact on road traffic addressed in this chapter.

Analysis of urban road networks. Urban road networks have been subject to many studies aiming to identify problematic areas using traffic information. Studies focusing on congestion proposed methods for measuring and tracking congestion [RR12], [ALVI16], and identifying propagation of congestion patterns [LJZ17, NLC17]. [JFF16, HZY+15] proposed approaches for the detection of so-called *urban black holes*, i.e. traffic anomalies with a greater inflow than outflow. [LGL17] investigated the maximum capacity of urban street networks. They introduced a formal upper bound of the capacity and found that the capacity is independent of individual routing strategies. [KLS18] employed topic modelling to analyze urban street networks. They proposed the concept of interactional regions, i.e. regions that commonly bound routes within the street networks. [WWL16] investigated the use of external datasets like POIs, location-based social media, weather and incident data to find explanations for traffic data. They found that POIs data is correlated with regular traffic patterns, while location-based social media can be used to explain irregular traffic patterns. In contrast to this chapter, none of these methods aims at determining impact of planned special events on road traffic.

Road traffic forecasting and impact of incidents. Road traffic forecasting (see a recent survey conducted by Vlahogianni et al. [VKG14] for an overview) aims at predicting traffic flow on particular roads on a short term. [PDS12] tackled the problem of traffic speed prediction in the presence of incidents by introducing the hybrid H-ARIMA model, which combines a historic average model and the well-established ARIMA model [BJ90]. [PDGS15] classified different kinds of traffic incidents and used this classification to predict the impact of each incident type on road traffic. [ADG+14] used Support Vector Regression and clustering of spatial and temporal patterns to predict traffic speed for individual units. [WZX14] made use of sparse FCD data and a context-aware tensor decomposition approach to estimate travel

times for road segments for which no FCD is available. The information is then used to estimate the required travel time for a given route. [MYS+17] proposed a framework for the city-wide inference of traffic volume. They made use of a semi-supervised learning algorithm that can be used with sparse loop detector data as well as taxi GPS data. Similar, [WPC+16] employed a Hidden Markov Model to estimate traffic speeds of a road network based on sparse floating car data where the speed to be estimated on a single road is considered as a hidden state. [MYWW15] make use of FCD data to predict congestion, while [KXS+16] tackles this problem by using GPS data from mobile phones. [MG12] predicted the impact of highway incidents and proposed a model to predict false reports of incidents. The authors of [KNC08] investigated the duration of freeway incidents. They identified variables that influence the duration of the incidents and propose a Rule-Based-Tree-Model to predict the duration. The approaches discussed above focus on short-term traffic predictions and on incidents and congestion that can be clearly assigned to the individual road segments. In contrast, in this chapter we aim to predict the more diffuse impact of planned special events on complex urban road networks, that is subject to the infrastructure and event characteristics rather than to the short-term traffic fluctuations.

Road traffic forecasting using deep learning. Recently, various approaches emerged that adopt deep learning techniques (i.e. deep neural networks) for short-term traffic forecasting. The architectures include feed-forward networks, deep believe networks and long short term memory networks (see, e.g., [LLWL18, PS17, SKK16, LXZ+18]). Whereas deep learning approaches become increasingly popular in the context of road traffic forecasting and impact of incidents, they require large amount of training data. This requirement makes deep learning hardly applicable to the problem of impact prediction for planned special events, as the number of large-scale events in a particular city is typically limited to a range of a few hundred events per year. In contrast, the approach proposed in this chapter facilitates accurate predictions of event impact given a limited amount of training data.

7.3 Problem Definition

Planned special events (such as concerts, fairs, or sports matches) can negatively impact urban traffic. Some parts of the transportation network can experience usage exceeding the actual network capacity, leading to traffic congestion with slower traffic speed and longer trip times.

Intuitively, this problem may occur within temporal proximity to the event and spatial proximity to the event venue, i.e. when event participants arrive at or leave the venue. However, event impact on the transportation graph can vary significantly according to the event and venue characteristics as well as further factors.

The aim of this chapter is to predict event impact of planned special events on the transportation graph, in particular with respect to the spatial and temporal di-

mensions.

More formally, a *transportation graph* is a directed multi-graph $TG := (V, U)$ that represents the road network. V is a set of nodes (representing the crossings of the road network) and U is a set of edges (also referred to as *units* in this chapter), i.e. the road segments. Each node $v \in V$ is assigned coordinates. Each edge $u \in U$ is assigned a speed limit $lim(u)$.

In particular we target the following problems:

1. Prediction of the *spatial dimension of event impact* ($impact_{spatial}(ve, t_j, TG)$): Given an event that takes place at venue ve, we aim to predict the maximal distance from the event venue where event impact on the transportation graph TG in terms of a restricted traffic flow can be observed at time point t_j.

2. Identification of the *typically affected subgraphs*: Given an event venue ve, we aim to determine the subgraph of the transportation graph TG that is commonly affected by planned special events taking place at this venue.

3. Prediction of the *temporal dimension of event impact* ($impact_{temporal}(ve, t_j, TG)$): Given an event that takes place at venue ve, we aim to predict the average delay that can be observed at time point t_j on the units of TG that are typically affected in the presence of events in ve.

Furthermore, we aim to analyze the factors that can facilitate such predictions for both impact dimensions.

In this chapter, we focus our discussion on the impact of the incoming event traffic. The impact of the outgoing event traffic can be considered analogously, provided that information about the respective event end times is available. In the following, we define metrics to estimate the spatial and the temporal dimensions of event impact, discuss the corresponding factors and build supervised prediction models.

7.4 Spatial Dimension of Event Impact

One of the problems addressed in this chapter is the prediction of the spatial dimension of event impact. In particular, we aim to build a supervised regression model to predict $impact_{spatial}(ve, t_j, TG)$, where ve is the event venue, TG is the transportation graph and t_j is the time point for which the prediction is requested. First, we introduce the necessary concepts and considerations, which are then used in Section 7.4.3 to define our metric for the spatial dimension of event impact. This metric builds the basis for the actual prediction task that is addressed as a regression problem later in Section 7.6.

In order to measure the spatial dimension of event impact on the transportation graph, we take the following considerations into account:

- First, events may directly impact the traffic flow on the units (edges) of the transportation graph TG through an increased load. Thus, we define a metric for the **unit load** $ul(u, t_j) \in [0, 1]$ that measures the normalized load on unit u at time point t_j.
- Second, an increased unit load observed on the units of the transportation graph may or may not be (partially) induced by a particular event. Therefore, we introduce the concept of an **affected subgraph of** TG that indicates event-induced load, i.e., unit load that can be partially attributed to the event.
- Third, we identify subgraphs of TG that are **typically affected** by events in particular venues.
- Finally, we define a metric for the **spatial dimension of event impact** at time point t_j as a measure of the longest distance from event venue ve to unit u of the transportation graph where event-induced load is observed at t_j.

In the following we present these steps in more detail. Notations frequently used in this chapter are summarized in Table 7.1.

7.4.1 Unit Load and Average Unit Load

In the following, a unit $u \in TG$ refers to an edge of the transportation graph TG. We define the unit load $ul(u, t_j) \in [0, 1]$ as the relative speed reduction at unit u with respect to the speed limit $lim(u)$ of the corresponding edge in the transportation graph in Equation 7.1:

$$ul(u, t_j) = \frac{lim(u) - speed(u, t_j)}{lim(u)}, \tag{7.1}$$

where $speed(u, t_j)$ represents the actual traffic speed at u at time point $t_j \in \mathcal{T}$, where $\mathcal{T}$ is the set of all time points. Here, '1' corresponds to the maximal speed reduction (i.e. a congestion when traffic is fully halted), and '0' corresponds to the normal usage, where the traffic can reach the maximal speed allowed by the speed limit.

We represent typical unit load for unit u on a particular week day and day time via average unit load $ul_{avg}(u, t_j)$. Given a transportation graph TG and a time point t_j, let $UL(u, t_j)$ be a set containing all unit loads for u on the same week day and day time as t_j: $UL = \{ul(u, t) | tod(t) = tod(t_j), dow(t) = dow(t_j)\}$, where $dow(t_j)$ and $tod(t_j)$ map t_j to its week day and time of day, correspondingly. The average unit load for unit u at t_j is then defined as $ul_{avg}(u, t_j) = avg(UL(u, t_j))$.

7.4.2 Event-Induced Load an Affected Subgraphs

High load observed on a particular unit of the transportation graph at a certain time point is not always caused by special events but may be due to recurring temporal

Table 7.1. Notation summary.

Notation	Description
ve	Event venue
$\mathcal{T}$	Set of all time points
TG	Transportation graph
u	A unit (i.e. an edge) of TG
$speed(u, t_j)$	Traffic speed at unit u at time point t_j
$ul(u, t_j) \in [0, 1]$	Unit load on unit u at time point t_j
$ul_{avg}(u, t_j) \in [0, 1]$	Average unit load on unit u corresponding to t_j
$ud(u, t_j)$	Unit delay. Delay of travel time on unit u at time point t_j
$iqr(u, t_j) \in \{true, false\}$	Indicates whether $ul(u, t_j)$ is an outlier with respect to IQR
$dist_{geo}(ve, u)$	Geographic distance between venue ve and u
$dist_{temp}(t_i, t_j)$	Temporal distance between two time points
TAS_{ve}	Typically affected subgraph of ve
th_{ul}	Unit load threshold
th_{temp}	Temporal proximity threshold
th_{ta}	Minimum percentage of events at which units are considered typically affected
$impact_{spatial}(ve, t_j, TG)$	Spatial impact of an event in venue ve on TG at t_j
$impact_{temporal}(ve, t_j, TG)$	Temporal impact of an event in venue ve on TG at t_j

patterns, incidents or other factors. Therefore, we consider the event-induced unit load, i.e. unit load $ul(u, t_j)$ that can be partially attributed to an event by considering several indicators including: (i) temporal proximity of t_j to event start time t_e; (ii) geographic proximity between unit u and event venue ve; (iii) the (unusually) high load on unit u; (iv) connectedness of unit u with other units that indicate event-induced load in the transportation graph.

Temporal proximity. The intuition behind this indicator is that event-induced traffic is likely to occur within close temporal proximity to the start or the end time of the event, when event participants arrive at or leave the venue. Temporal distance $dist_{temp}(t_e, t_j)$ is the length of the time interval between the time point $t_j \in \mathcal{T}$ (at which the specific load on unit u is observed) and the event time $t_e \in \mathcal{T}$. Here, t_e the is the event start time (event end time works analogously). We say that time point t_j is within the temporal proximity of event e if the temporal distance between t_j and the event time t_e is within an interval given by th_{temp}: $dist_{temp}(t_e, t_j) \leq th_{temp}$, where th_{temp} is a parameter in our approach.

Geographic proximity. The intuition behind this indicator is that event-induced traffic is likely to start and end within the close geographic proximity to the event venue. The $dist_{geo}(ve, u)$ denotes the geographic distance (i.e. Euclidean distance that takes the Earth's curvature into account) between the venue ve represented

through its coordinates and the unit u. For the distance computation, we represent the unit through the coordinates of its adjacent nodes in the transportation graph. In particular, we distinguish the $\textit{min-dist}_{geo}(ve, u)$ and the $\textit{max-dist}_{geo}(ve, u)$, dependent on the geographic location of the corresponding node. We say that unit u is within the geographic proximity of the venue ve if $\textit{min-dist}_{geo}(ve, u) \leq th_{geo}$. The threshold th_{geo} is a parameter, defined as the walking distance in our approach. Based on the heuristics proposed in [Per29] we choose a walking distance of $th_{geo} = 500\ meter$ in the following.

Affected unit. Intuitively, when event-induced traffic impacts the road network, some units can indicate an (unusually) high load. To capture this intuition we propose a method based on the outlier analysis. In this context, we define a Boolean metric $\textit{affected}(u, t_j) \in \{true, false\}$, where "true" denotes that unit u is affected at time point t_j, i.e. this unit indicates an (untypically) high load degree. We consider two indicators of unit affectedness: (a) Absolute unit load: We consider u to be affected at time point $t_j \in \mathcal{T}$ if its load exceeds the pre-defined unit load threshold th_{ul}, i.e. $ul(u, t_j) \geq th_{ul}$. We discuss the influence of the thresholds in the experimental evaluation in Section 7.9.1. (b) Outliers: To identify the outliers, i.e. the units whose load at a given time point deviates significantly from their typical load at comparable times (i.e. the same week day and day time), we employ the *interquartile range* (IQR)-rule [KZ00]. We define a Boolean metric $iqr(u, t_j) \in \{true, false\}$ that denotes if the unit load $ul(u, t_j)$ is an outlier according to the IQR-rule in Equation 7.2.

$$iqr(u, t_j) = \begin{cases} True, & if\ ul(u, t_j) > Q_1(u, t_j) + 1.5 \cdot (Q_3(u, t_j) - Q_1(u, t_j)) \\ False, & \text{otherwise} \end{cases} \tag{7.2}$$

where $Q_n(u, t_j)$ denotes the nth quartile of the unit load on unit u with respect to the week day and day time. Equation 7.3 combines both load indicators presented above. We say that unit u is affected at t_j if at least one of both conditions holds, i.e. either the unit load exceeds the pre-defined load threshold th_{ul}, or the unit load is an outlier according to the IQR-rule.

$$\textit{affected}(u, t_j) \equiv (ul(u, t_j) \geq th_{ul}) \ \vee \ iqr(u, t_j). \tag{7.3}$$

Affected path. Given a venue ve and a time point t_j, Equation 7.4 defines an affected path p as a connected subgraph of the transportation graph TG, such that all units of this subgraph are affected at this time point and at least one unit u_n of this subgraph is within the geographic proximity of venue ve. Moreover, to prevent affected paths to mainly capture regular traffic patterns, we require the unit u_d most distant from the event venue within this subgraph to deviate from the regular traffic pattern. More formally:

$$
\begin{aligned}
\textit{affected}(p, ve, t_j) \equiv \\
&\forall u_i \in p : \textit{affected}(u_i, t_j) \\
&\wedge \exists u_n \in p : \textit{min-dist}_{geo}(ve, u_n) \leq th_{geo} \\
&\wedge u_d = \underset{u_m}{\text{argmax}}\{u_m \in p | \textit{max-dist}_{geo}(ve, u_m)\} \\
&\wedge iqr(u_d, t_j).
\end{aligned}
\tag{7.4}
$$

Affected subgraph. Event-induced traffic that starts or ends in a geographic proximity of the venue is likely to be propagated along the transportation graph. To capture this intuition, we define the concept of an affected subgraph. Equation 7.5 defines an affected subgraph (ASG) of venue ve at time point t_j as the subgraph of TG that contains all respective affected paths:

$$
\begin{aligned}
&\textit{affected}(ASG, ve, t_j) \equiv \\
&\quad \forall u_i \in ASG : (\exists p_i \subseteq TG : u_i \in p_i \wedge \textit{affected}(p_i, ve, t_j)) \\
&\quad \wedge \neg(\exists p_k \subseteq TG : \exists u_k \in p_k : (\textit{affected}(p_k, ve, t_j) \wedge u_k \notin ASG)).
\end{aligned}
\tag{7.5}
$$

The first line of Equation 7.5 specifies that an affected subgraph ASG only contains the units u_i that are part of at least one affected path p_i. The second line ensures that all affected paths are part of ASG (i.e. no affected path p_k exists that is not part of ASG).

7.4.3 Metric for the Spatial Dimension of Event Impact

In general, spatial distance can be measured using different metrics, including Euclidean distance, as well as path-based or grid-based distance, whereas the suitability of the metric depends on the particular application scenario. In this chapter, we adopt Euclidean distance to make the notion of the spatial dimension of event impact comparable across venues and also better explainable to the end users. As road networks are diverse, the length of the paths measured on the directed road network on different paths within a given radius from the venue may vary. Euclidean distance is a grid-independent indicator. This indicator compensates the differences in the network topology, provides an external view on the event-related traffic around the venue, makes it comparable across venues and also allows for an intuitive explanation. For example, Euclidean distance facilitates statements such as "The event e that takes place at venue ve will impact the area within x km from the venue". This way, the affected area around the venue can be avoided by the vehicles not directly involved in the event.

Given an event e starting at t_e with $dist_{temp}(t_e, t_j) \leq th_{temp}$ at venue ve, we define the spatial dimension of event impact $impact_{spatial}(ve, t_j, TG)$ of e on the transporta-

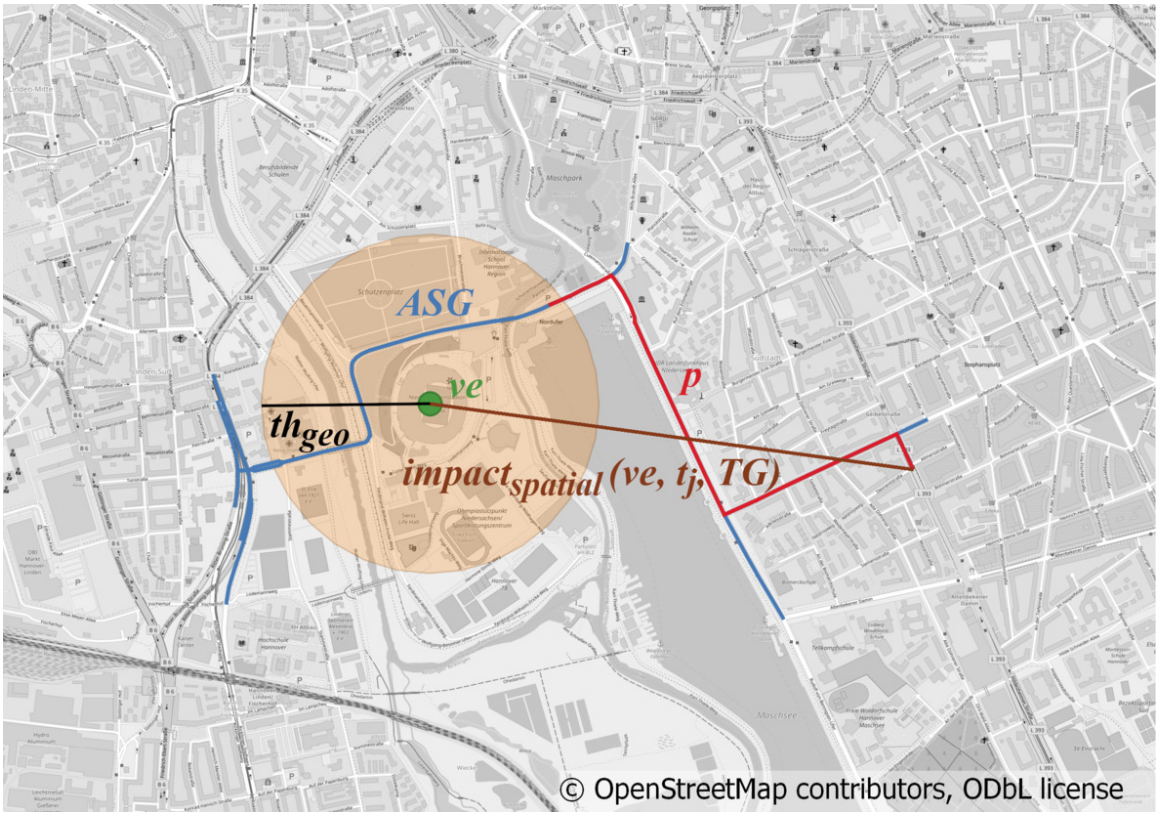


Figure 7.1. Visualization of the spatial dimension of event impact. The green point marks the event venue ve. Blue units form the affected subgraph, while red units form an affected path within this subgraph. The brown line illustrates the distance to ve that denotes the spatial dimension of event impact. The orange circle visualizes the threshold th_{geo}.

tion graph TG at time point t_j as the maximal distance between the event venue and a unit u_d of TG at which load induced by e is observed at t_j.

$$impact_{spatial}(ve, t_j, TG) = \max\{max\text{-}dist_{geo}(ve, u_d) \mid u_d \in ASG \subseteq TG \wedge affected(ASG, ve, t_j)\}, \tag{7.6}$$

where u_d is a unit of the transportation graph with the following properties: (i) u_d is a unit of an affected subgraph $ASG \subseteq TG$; and (ii) u_d is the furthest distant unit from ve among the units in the affected subgraph. Fig. 7.1 illustrates affected subgraphs, affected paths and the spatial dimension of event impact.

7.4.4 Typically Affected Subgraphs

Considering individual event venues, certain units are commonly affected if a planned special event takes place in the respective venue. To this extent, we define *typically affected subgraphs* (TAS_{ve}) to be venue-specific subgraphs of TG that consist of units commonly affected in presence of planed special events at a particular venue ve.

Algorithm 3 formalises the calculation of the typically affected subgraph TAS_{ve} for a single venue ve. The intuition of the algorithm is to identify units that are frequently affected by events that take place in ve. To this extent, a joint event subgraph SG_e is computed. This graph contains all units affected at any time point in temporal proximity of a single event e that took place in ve. SG_e can be viewed as the temporal union of affected subgraphs for a single event. The typically affected subgraph is then constructed from all units that are present in a certain percentage (i.e. at least th_{ta}) of event-specific subgraphs. We define a typically affected threshold $th_{ta} \in [0, 1]$ as the minimum percentage of event-specific subgraphs in which a unit need to be present to be considered as typically affected.

7.5 Temporal Dimension of Event Impact

This section introduces our method to quantify the temporal dimension of event impact by measuring travel time delay in presence of planned special events. In Section 7.4, we defined a methodology to quantify the spatial dimension of event impact, i.e., to measure how far from the event venue affected units can be observed. Complementary, another question of interest is to *which extent the units will be affected*, i.e., to measure the travel time delay on these units introduced by an event. To quantify the temporal dimension of event impact, we rely on the following intuition: For individual event venues, some units are commonly affected when an event takes place, i.e. event-induced load can be observed on these units.

To quantify the temporal dimension of event impact, we propose the following methodology: (1) we compute the typically affected subgraph TAS_{ve} of the event venue ve, and (2) we measure the average delay in the presence of events on the units contained in TAS_{ve}.

Algorithm 3 describes the computation of a typically affected subgraph TAS_{ve} for a single venue ve. For each venue, in which at least one event takes place, we apply the algorithm to obtain an individual TAS_{ve} for this venue. Algorithm 3 is subject to the parameters th_{ul}, th_{temp} and th_{ta}. Section 7.8 discusses the effect of the individual parameters on the extracted subgraphs and suggests useful configurations.

We define the temporal dimension of event impact $impact_{temporal}(ve, t_j, TG)$ of an event taking place in the venue ve at the time point t_j on the transportation graph TG as an average delay in the travel time on the units that are part of the respective typically affected subgraph $TAS_{ve} \subseteq TG$:

Algorithm 3: Calculation of TAS_{ve} for a single venue ve

Input : ve - Event venue
th_{temp} - Threshold for the temporal proximity
th_{ta} - Minimum percentage of events at which units are considered typically affected

Output: TAS_{ve} - The typically affected subgraph of ve

```
1  E ← getAllEvents(ve)          // Determine events that took place in ve
2  counts[] ← ∅                  // Empty mapping between units and counts
3  forall e ∈ E do
4  |  T_e ← {t_j ∈ 𝒯 | dist_temp(t_e, t_j) ≤ th_temp}
5  end
   /* Joint event specific subgraph SG_e across all time points in
      temporal proximity to e                                          */
6  SG_e ← ⋃{ASG ⊆ TG | ∃t_j ∈ T_e : affected(ASG, ve, t_j)}
   /* Count how often each unit appears in event specific subgraphs
      */
7  forall u ∈ SG_e do
8  |  counts[u] ← increment(counts[u])
9  end
   /* Add all units to TAS_ve that appear more often than th_ta        */
10 forall u ∈ counts[] do
11 |  if counts[u]/|E| ≥ th_ta then
12 |  |  TAS_ve ← TAS_ve ∪ u
13 |  end
14 end
15 return TAS_ve
```

$$impact_{temporal}(ve, t_j, TG) = avg(\{ud(u, t_j) | u \in TAS_{ve}\}), \tag{7.7}$$

where u is a unit contained in TAS_{ve} and avg denotes the arithmetic mean. $ud(u, t_j)$ (unit delay) is defined in Equation 7.8 as a function that specifies the additional amount of time it takes to pass the unit u at time point t_j, compared to the ideal situation when travelling at the maximum allowed speed is possible.

$$ud(u, t_j) = max(0, \frac{length(u)}{speed(u, t_j)} - \frac{length(u)}{lim(u)}), \tag{7.8}$$

where $lim(u)$ denotes the speed limit on unit u, $speed(u, t_j)$ denotes the possible speed on u at t_j and $length(u)$ denotes the length of u. The first fraction of Equation 7.8 expresses the required time to pass u at t_j, while the second fraction expresses the required time when travelling with the maximum allowed speed is possible. Note, that we do not consider negative delays (e.g. when people are travelling faster than the speed limit).

The proposed $impact_{temporal}(ve, t_j, TG)$ function computes the average unit delay and summarizes the overall temporal event impact on the respective TAS_{ve}. This function aggregates the individual unit delays and requires only the speed information on the units. This computation weights all units in the typically affected subgraph TAS_{ve} equally and does not require any traffic volume information. Note, that the average can easily be replaced by a weighted average if additional information, e.g. the number of vehicles on a unit, is available and indicates large variations. Furthermore, a weighted average can also be considered to further differentiate the type of the units (e.g. major roads and smaller streets).

7.6 Event Impact Prediction: Features and Models

We treat event impact prediction as a regression problem. In this section we introduce the features designed to enable accurate predictions including characteristics of events, mobility behavior and infrastructure.

7.6.1 Features

Table 7.2 provides an overview of the features we employ to predict both spatial and temporal dimensions of event impact. In the following, we present these features in more detail. To determine the best combination of the features we conduct an exhaustive grid search [BB12]. I.e., we train an individual model for each possible combination of features and finally select the model with the best performance.

Table 7.2. Overview of adopted features. Categorical features are 1-hot encoded.

Feature	Representation	Notation
Event characteristics		
Day of Week	Categorical	e_{dow}
Start Time	Numerical	e_{st}
Venue	Categorical	e_v
Category	Categorical	e_c
Workday	Binary	e_{wd}
Reoccurring	Binary	e_r
No. Participants	Numerical	e_{np}
Entity Popularity	Numerical	e_{ep}
Mobility Behaviour		
Average Venue Impact	Numerical	m_i
Average Nearby Affected Units	Numerical	m_{nau}
Characteristics of the Infrastructure		
Nearby Road Types	Numerical	i_{nrt}

Event Characteristics

The individual characteristics of the particular event may influence its impact on urban transportation. Therefore, we adopt the following event characteristics as features:

Day of Week (e_{dow}). The intuition behind this feature is that traffic may indicate weekday specific patterns, e.g. people might leave work early on Fridays. We apply 1-hot encoding such that each day of the week is represented as an individual dimension. For a particular day, the corresponding dimension is set to 1 while all other dimensions are left as 0.

Start Time (e_{st}). Traffic patterns vary significantly with respect to the time of the day, e.g. during rush hours. We map the scheduled start time of the event to a continuous numerical representation, where each hour is mapped to a number, e.g. 13:15 $\mapsto$ 13.25.

Venue (e_v). Venues located near different parts of the transportation graph might exhibit specific impact patterns. The event venue feature is 1-hot encoded.

Category (e_c). Events in different categories (e.g. *concert*, *fair*, or *sports*) might attract different audiences that exhibit specific mobility behavior. For example, fair visitors might arrive just-in-time or spread across a whole day, whereas concert audience might arrive early to secure a place in the front. We distinguish between 7 event categories that are most typical for the dataset used in our experiments (see Section 7.7.1 for details): {*Comedy, Fair, Convert, Football, Show, Party, Other*}.

Note that the set of categories can be easily adjusted to fit the most frequent events in a particular city. The event category feature is 1-hot encoded.

Workday (e_{wd}). On workdays urban road networks are highly influenced by work-related traffic (e.g. due to commuting between work and residential areas), while on weekends the traffic exhibits other patterns. The workday is a binary feature that indicates whether the event takes place on a working day or on the weekend.

Recurring (e_r). Recurring events are likely to attract a similar audience, and therefore can exhibit common mobility patterns. We consider events of the same category that take place in the same venue to be recurring if they are part of an event series (e.g. football matches in a tournament). Whether an event is recurring is represented as a binary feature. Note that binary encoding in a combination with the venue is sufficient to represent event series in case the venue typically accommodates event series of the same kind. For instance, if recurring events in the football stadium are always football games with similar audience, binary encoding is sufficient. In case different event series take place in the same venue, the binary encoding can be replaced by a 1-hot encoding, where each specific event series is represented by an individual dimension.

No. Participants (e_{np}). The number of participants who attend an event is likely to correlate with the event impact on urban traffic. If the number of participants is not available, we can make use of an approximate value. This value can be estimated using the venue capacity, or the typical number of participants of comparable events in the past (if available). In this chapter, we use the venue capacity as an estimate in cases where the exact number of participants is not known. This feature is represented as an integer. In our experiments, we annotate the number of participants manually, using Web search.

Entity Popularity (e_{ep}). The intuition behind this feature is that popularity of the key actors involved in the event (e.g. popular musicians, celebrities, politicians, etc.) is likely to correlate with the event impact on urban traffic. We approximate entity popularity using the number of search results obtained through a state-of-the-art search engine (Google search engine in our experiments). In particular, we extract surface forms of named entities from the event title and use these as search engine queries. If multiple named entities are mentioned, the most salient is chosen. In our experiments, we annotated the entities manually. In practice, these surface forms can be extracted from the event titles automatically using named entity recognition and disambiguation (NERD) methods. We employ the number of hits returned by the search engine as a feature represented by an integer.

Mobility Behavior

Mobility behavior features reflect the traffic situation at a particular time point relevant to predict the spatial impact of an event.

Average Venue Impact (m_i). Average venue impact is the measure of the

typical traffic situation around the event venue. This feature represents an average impact of the venue ve on traffic at a specific week day and day time.

Given a venue ve, a time point t_j and a transportation graph TG an average venue impact $m_i(ve, t_j) = impact_{avg}(ve, t_j, TG)$ is computed analogously to the event impact computation detailed in Section 7.4 with the following adjustments: (1) instead of the unit load, average unit load is adopted: $\forall u \in TG : ul(u, t_j) \equiv ul_{avg}(u, t_j)$; (2) for the definition of the unit affectedness, only the unit load threshold is considered, since the IQR-rule is not relevant for the average impact, i.e. $affected(u, t_j) \equiv ul_{avg}(u, t_j) \geq th_{ul}$. An average venue impact is represented as a real number.

Average Nearby Affected Units (m_{nau}). The average traffic situation near a venue in the presence of an event can also be characterized by the number of the nearby affected units. We determine an average number of the affected units near the venue as:

$$m_{nau}(ve, t_j) = |\{u \in TG | ul_{avg}(u, t_j) \geq th_{ul}, min\text{-}dist_{geo}(ve, u) < r \}|,$$

where we consider the ranges $r \in \{500m, 1000m, 2000m, 4000m\}$. For each range, m_{nau} is represented as a real number.

Infrastructure characteristics

Mobility patterns are highly dependent on the city infrastructure since the infrastructure is likely to determine the paths people choose to reach their destinations. Thus, we propose the following feature:

Nearby Road Types (i_{nrt}). For each event, we identify the type of the road segments located within the geographic proximity of the venue. Given TG, an event venue ve and a set of road types RT, the count of nearby road types is computed as follows:

$$i_{nrt}(ve, rt) = |\{u \in TG | min\text{-}dist_{geo}(ve, u) \leq th_{geo}, type(u) = rt\}|,$$

where $rt \in RT$ represents a single type, $th_{geo} = 500m$ as discussed in Section 7.4.2 and $type(u)$ is a function that provides the road type of a given unit. For RT we make use of the taxonomy defined by *OpenStreetMap*, which is detailed in Section 7.7.1. The count for each type rt is combined to form a vector of integers.

7.6.2 Regression Models

Finally, we combine the features to form a regression model aimed at the prediction of the spatial dimension of event impact. In particular, we take the following steps: (1) feature normalization; (2) model selection; and (3) hyperparameter optimization.

Feature normalization. First, we normalize the features by removing the mean and by scaling all values to unit variance.

Model selection. The number of events that can be observed in a single city is rather small for machine learning methods. Therefore, we choose the regression models that can work effectively when limited amount of training data is available.

SVR *Support Vector regression* has been applied to a large variety of scenarios. One-hot encoding is well suited for use with SVRs since they are able to create decision boundaries between the dimensions.

KNN The *k-nearest-neighbor* algorithm takes into account the k-nearest-neighbors only. Therefore, this algorithm is not constrained by the limited amount of training data.

RIDGE *Ridge regression* is a linear regression that introduces a penalty to the size of the learned coefficients [HK00]. This leads to more robust coefficients, especially in the context of small training sets.

Note that we do not consider deep learning regression models because of the large amount of training data they require. Since each record in the training dataset corresponds to a planed large-scale special event, the number of training examples in a particular city is typically limited to a range of a few hundred events per year. This number is not sufficient to achieve an optimal performance using current deep learning approaches.

Hyperparameter optimization. We optimize hyperparameters by employing the random search algorithm proposed by Bergstra et al. [BB12]. The following parameters are optimized: SVR: C: the penalty parameter, tol: tolerance for the stopping criterion. RIDGE α: regularization strength. KNN: k: the number of neighbors considered, $leafsize$: size of the leafs used by the algorithms *BallTree*.

7.7 Evaluation Setup

This section describes the evaluation setup for our approach.

7.7.1 Data Sources

The data sources used to evaluate our approach include data about events, mobility infrastructure as well as traffic data.

Figure 7.2. Visualization of the transportation graph TG_H extracted from OSM, which contains all major roads within 20 km distance to the city center of Hanover.

Mobility Infrastructure Dataset.
OpenStreetMap is a provider of publicly available map data. We make use of the OSMs road network to form the transportation graph TG_H. In particular, we extract streets that are located within the 20 km distance to the city center of Hanover. Note, that this distance is sufficient to entirely capture all observed event impacts. Considering the OSM-taxonomy for road types, we restrict the transportation graph to contain only major roads, as reliable traffic information for smaller streets is rarely available. In particular, we extract all roads that belong to one of the following classes: {primary, primary_link, secondary, secondary_link, tertiary, tertiary_link, motorway, motorway_link, trunk, trunk_link}. OSM partitions roads in smaller road segments that correspond to the units of the transportation graph TG_H. The extracted transportation graph contains 23,000 units and 13,000 nodes in total. For each unit $u \in TG_H$ information about the speed limit $lim(u)$ as well as the road type is available from OSM. Fig. 7.2 visualizes the extracted transportation graph by marking all included units with red color.

Traffic Dataset.
The experiments conducted in this chapter employ a proprietary traffic dataset that contains aggregated floating car data. This dataset is available to the authors in the

context of the research project "Data4UrbanMobility"[1]. In particular, the dataset provides traffic speed records for each unit u of the transportation graph TG_H. The dataset is collected by a company that offers routing software distributed under an open license. The dataset contains data contributions obtained from a variety of sources, including the data collected from the users of the routing software as well as traffic data acquired from third party data providers. Although particular statistics of these contributions, such as the number or the types of the monitored cars, are not available to the authors, due to a variety of sources involved we do not expect any particular biases towards certain vehicle types or expense classes. The dataset covers the time span from October 2017 to January 2018 and contains approximately 195 million records in total. The records within the traffic dataset contain the average traffic speed on the individual transportation graph units at discrete time points, i.e. $speed(u, t_j)$, recorded every 15 minutes. The average speed records are computed by the data provider through calculating the average traffic speed from the raw floating car data, averaged over all vehicles for which the data is available for the given unit and time interval. To ensure data quality, in particular with respect to the availability of a sufficient number of speed records per unit, only major roads extracted from OpenStreetMap are considered in this chapter (see the paragraph "OpenStreetMap" above for the description of the corresponding road categories). The data for such major roads is captured on a regular basis within the dataset. On average, 8422.79 records are available per transportation graph unit. We believe that this number is sufficient to capture typical traffic patterns.

Event Dataset.
We extracted an event dataset containing information regarding events that took place in the Hanover region, Germany from various regional event-related websites. Examples are the official website of the city of Hanover, football websites and local magazines. The events included in this dataset took place between October 2017 and January 2018. We selected the venues in the Hanover region that have a capacity for at least 1000 participants, which resulted in 7 different venues such as concert halls ("TUI-Arena", "Capitol", "Swiss Life Hall", "Kuppelsaal", "Aegi-Theater"), a football stadium ("HDI-Arena") and a fairground. We further restricted our event dataset to the events that took place in these venues. In total, 150 major events were obtained which occurred during the time period described above. Since the events were collected from different websites they did not exhibit a shared taxonomy of categories. We analyzed the most frequent categories with respect to each website and harmonized the data by manually defining a shared taxonomy to which the categories were aligned. We obtained 7 events categories such as "party", "comedy", "football", "concert", "fair", "show" and "other".

In addition, contextual information regarding events including event venue, event category, event start time, the number of participants and the most popular entities mentioned in the event title were annotated by the authors manually using infor-

[1] http://data4urbanmobility.l3s.uni-hannover.de/

mation obtained via Web search. The annotated event dataset is made publicly available[2].

7.7.2 Configurations of the Impact Computation

The spatial dimension of the event impact defined in Section 7.4 is subject to parameters such as the unit load threshold th_{ul} and the temporal proximity threshold th_{temp}. In our experimental settings, we vary these parameters to analyze their influence. In particular, the unit load threshold $th_{ul} \in [0, 1]$ corresponds to the degree of unit impairment. For the experimental settings, we choose three different threshold values corresponding to a high, medium and low level of unit impairment, namely $th_{ul} \in \{0.7, 0.5, 0.3\}$.

Regarding temporal proximity, we investigate the spatial impact ahead of the event, i.e. the impact of traffic caused by the event participants arriving at the event venue. In particular, we consider the following time points: $t_j = t_e$, $t_j = t_e - 30\ min$, $t_j = t_e - 60\ min$, $t_j = t_e - 90\ min$, where t_e corresponds to the event start time.

For each event and each combination of the th_{ul} and t_j values, we compute the spatial dimension of the event impact using the metric presented in Section 7.4.3. These impact values serve as numerical labels in the experimental setting.

Temporal event impact is further subject to the threshold th_{ta}, which specifies the minimum percentage of events at which units are considered typically affected. We investigate the effect of thresholds on the extraction of typically affected subgraphs in Section 7.8. Based on these insights, we consider the thresholds $th_{ta} = 0.3$ and $th_{ta} = 0.7$ in our experiments.

7.7.3 Baselines

We considered the following baselines for our experiments:

CBMA: The *Category Based Modelling Approach* was proposed by Kwoczek et al. in [KMN14]. This baseline is based on the intuition that events that belong to the same category result in similar impact on road traffic. To this extent this baseline considers all events within the same event category and uses the average impact as prediction.

CBMA2: The experimental setting in [KMN14] considered only the events that take place at the same venue. To provide a fair comparison following our experimental setup, we extended the CBMA baseline to calculate separate averages for each venue and event category. This should further improve the performance of the CBMA baseline.

[2]The dataset can be found here: http://www.l3s.de/~tempelmeier/crosstown_events.zip

Table 7.3. Overview of considered models.

Model	Prediction Based on
Baselines	
CBMA	Event category [KMN14]
CBMA2	Event category and venue
AVGT	Historic traffic averages
Proposed Regression Models	
SVR	Linear Support Vector regression
KNN	k-nearest-neighbors regression
RIDGE	Ridge regression

AVGT: We consider the impact prediction based on average traffic as another baseline. In particular, the average impact values are computed for each venue ve, each day of the week and each time point of the day where measurements are available. The intuition behind this baseline is that, in the absence of information regarding particular events, predictions can be made based on the average traffic typical for the particular weekday and daytime. The average venue impact was introduced in Section 7.6.1 and is used as a baseline as well as one of the features in our approach.

The CBMA and CBMA2 baselines are used for the prediction of both spatial and temporal dimensions of event impact. The AVGT baseline is a naïve baseline for prediction of the spatial dimension of event impact.

7.7.4 Evaluation Setup and Metrics

Table 7.3 provides an overview of the baselines and the regression models (following the approach proposed in Section 7.6). The listed approaches use the configurations introduced in Section 7.7.2. The events, which represent the instances of the described problem, are split into a test and training set. In particular, we divide them in fractions of 10%/90% and apply 10-fold cross validation. For each fold we ensure that no feature is extracted from traffic data that was collected on any day in which an event contained in the test set took place.

The models are evaluated using the following metrics.

MAE. The *mean absolute error* measures the absolute error of the model prediction. This measure is computed as an average of the absolute errors: $MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$, where n denotes the number of samples in the test set, y_j denotes the prediction and $\hat{y}_j$ denotes the actual observations.

RMSE. *root mean squared error* is computed as follows: $RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$, where the notation is identical. Like MAE, RMSE is a metric for prediction errors. As all errors are squared, RSME is more sensitive

to larger errors than MAE.

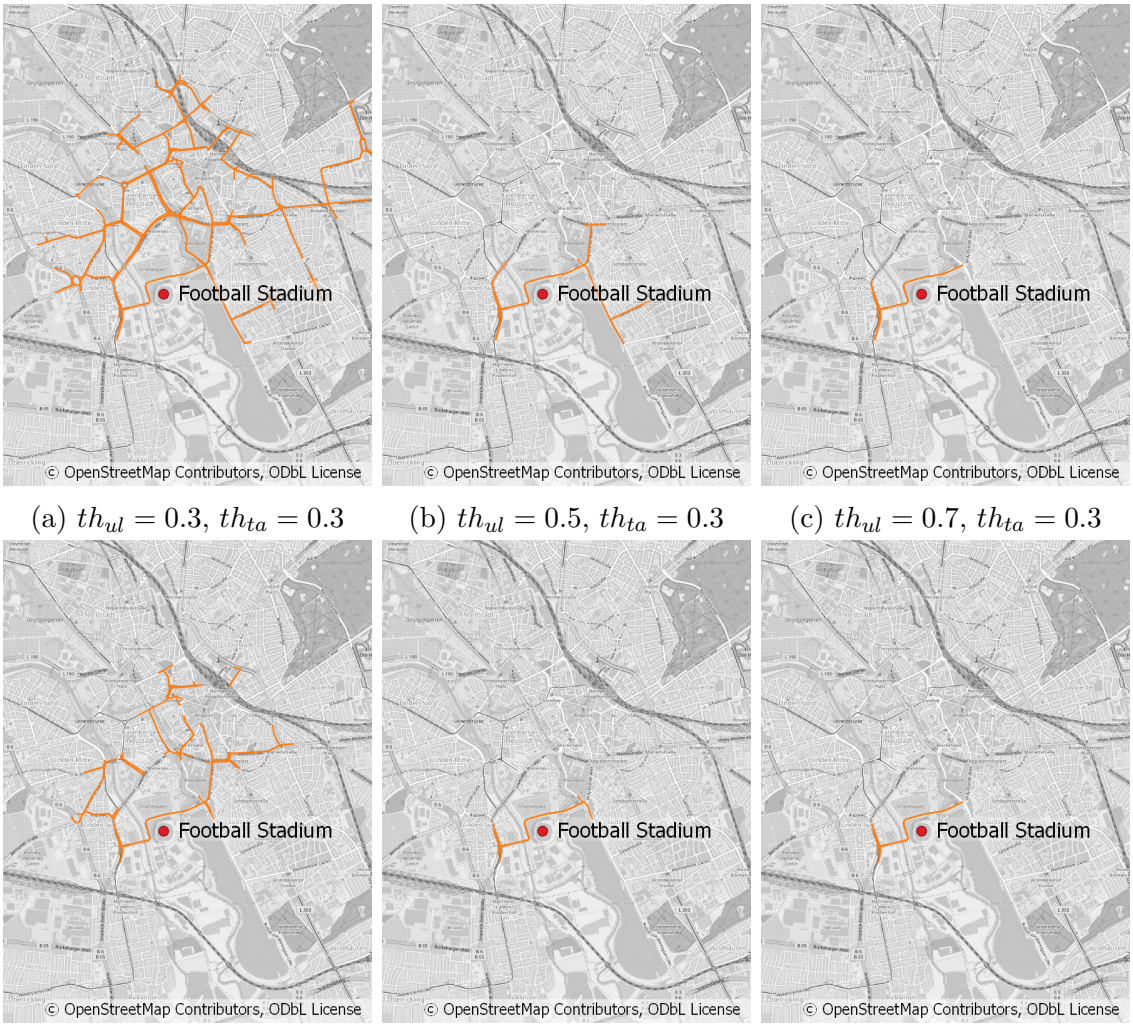
7.8 Identifying Dataset-Specific Parameters for Affected Subgraphs

Optimal values for the parameters introduced in the above sections are strongly dependent on the dataset at hand. In order to illustrate the effect of various parameters on TASs, this section discusses such effects and motivates the parameter values adopted in the experimental evaluation. The extraction of TASs is dependent on the following parameters:

- th_{temp}: The temporal proximity threshold.
- th_{ul}: The unit load threshold.
- th_{ta}: The minimum percentage of events at which units are considered as typically affected.

For the case study, we choose $th_{temp} = 90\ min$ for our experimental setting. As discussed above, we consider only the time ahead of the event. Fig. 7.3 compares different configurations of th_{ul} and th_{ta} for the extraction of typically affected subgraphs of the football stadium in Hanover. The largest graphs are identified with low thresholds as seen in Fig. 7.3a. When th_{ul} increases, only heavily affected units remain and the graph shrinks, which can be seen in Fig. 7.3e and Fig. 7.3c. With high values of th_{ta} and low values of th_{ul}, as observed in Fig. 7.3d, the subgraph is no longer connected. Isolated parts of the subgraph are thus connected to the remainder by the units of the transportation graph that are not frequently affected. This indicates that such a configuration is not suitable to capture typical traffic patterns. For high values of th_{ul}, no major differences between different values of th_{ta} can be observed, as shown in Fig. 7.3c, 7.3f. This implies that the units that are heavily affected are affected on a regular basis as well.

Fig. 7.4 depicts the temporal dimension of event impact for a single football game that took place on January 13th, 2018 in the city of Hanover. The impact was computed on typically affected subgraphs with the above introduced configurations of th_{ul} and th_{ta}. For smaller graphs (e.g. $th_{ul} = 0.5$, $th_{ta} = 0.5$) increased impacts are observed within the temporal proximity of the start time of the football game. Moreover, peaks are present before the start of the game and after an approximate end of the game, indicating the arrival and the departure of the audience. Note that although we can estimate the end time based on the typical duration, precise end time is not available in the dataset and may vary across events. In contrast, for larger graphs, (e.g. $th_{ul} = 0.3$ and $th_{ta} \in \{0.3, 0.7\}$), the peaks can not be clearly identified. For $th_{ta} = 0.3$ even the increased impacts can only be weakly observed. For higher



(a) $th_{ul} = 0.3$, $th_{ta} = 0.3$ (b) $th_{ul} = 0.5$, $th_{ta} = 0.3$ (c) $th_{ul} = 0.7$, $th_{ta} = 0.3$

(d) $th_{ul} = 0.3$, $th_{ta} = 0.7$ (e) $th_{ul} = 0.5$, $th_{ta} = 0.7$ (f) $th_{ul} = 0.7$, $th_{ta} = 0.7$

Figure 7.3. Comparison of the effects of th_{ul} and th_{ta} on the identification and extraction of typically affected subgraphs around the football stadium ("HDI-Arena") in Hanover. Higher thresholds lead to smaller subgraphs of TG that are considered as typically affected.

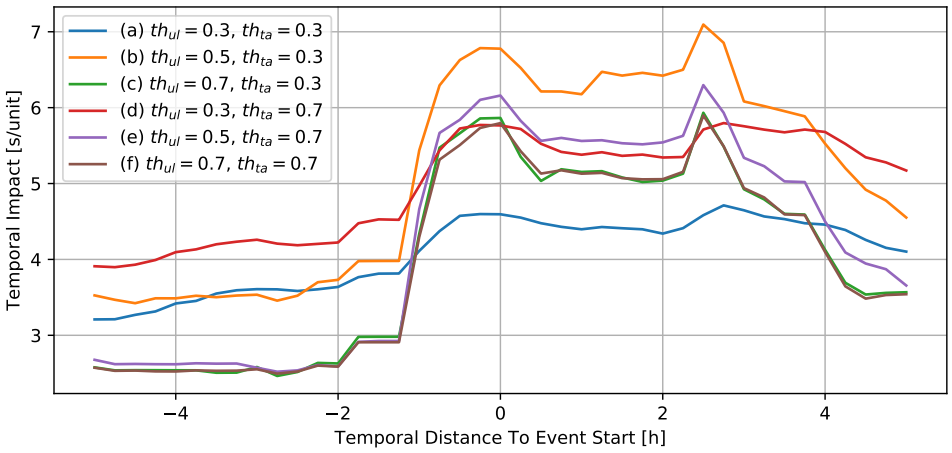


Figure 7.4. Temporal event impact with respect to th_{ul} and th_{ta} for a football game in Hanover on the January 13th, 2018.

threshold values (e.g. $th_{ul} = 0.3$, $th_{ta} = 0.7$) the peaks can still be observed, but the extent of the observed impact is relatively small. In case of restrictive configurations, only few units contribute to the impact and units that are only moderately affected are left out.

We conclude from these observations that the choice of the parameters is an important factor for computing the temporal dimension of event impact. Especially larger graphs extracted with low thresholds are not suited for the temporal impact computation. While smaller graphs obtained with high thresholds are better suited, a balanced configuration yielded the best results.

Therefore, the following configurations are considered in our experiments: $th_{ul} = 0.5$, $th_{ta} = 0.3$; $th_{ul} = 0.5$, $th_{ta} = 0.7$; $th_{ul} = 0.7$, $th_{ta} = 0.7$.

7.9 Evaluation of the Spatial Impact Prediction

This section presents the experimental evaluation of the prediction of the spatial dimension of event impact. In particular, we discuss the overall performance of the proposed approach and provide an analysis of the most indicative features. Furthermore, the performance depending on event venues and event categories is discussed.

7.9.1 Spatial Impact Prediction Performance

First, we present the overall performance of the proposed approach for the task of prediction of the spatial dimension of event impact (in terms of MAE and RMSE) in Tables 7.4a, 7.4b and 7.4c.

Table 7.4. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for the prediction of the spatial dimension of event impact. Scores outperforming the baselines are underlined, best scores are marked bold. t_e is event start time.

(a) $th_{ul} = 0.7$

Model	$t_j = t_e - 90\ min$		$t_j = t_e - 60\ min$		$t_j = t_e - 30\ min$		$t_j = t_e$	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CBMA	0.541	0.833	0.657	1.183	0.753	1.395	0.874	1.799
CBMA2	0.449	0.793	0.526	0.928	0.561	1.022	0.638	1.314
AVGT	0.615	1.019	0.668	1.246	0.751	1.459	0.831	1.760
SVR	**0.409**	0.745	0.469	0.914	**0.469**	1.024	0.522	1.400
KNN	0.435	**0.741**	**0.467**	**0.786**	0.470	**0.873**	**0.477**	**0.990**
RIDGE	0.414	0.748	0.489	0.909	0.530	0.995	0.596	1.282

(b) $th_{ul} = 0.5$

Model	$t_j = t_e - 90\ min$		$t_j = t_e - 60\ min$		$t_j = t_e - 30\ min$		$t_j = t_e$	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CBMA	1.124	3.110	1.332	3.238	1.392	3.287	1.115	1.961
CBMA2	1.144	3.229	1.227	3.223	1.310	3.254	0.889	1.557
AVGT	0.953	**3.011**	1.010	**3.035**	1.043	**3.081**	0.797	1.609
SVR	**0.859**	3.028	**0.935**	3.072	**0.966**	3.118	0.738	1.573
KNN	1.023	3.128	1.069	3.065	1.093	3.132	**0.737**	**1.401**
RIDGE	0.966	3.044	1.071	3.057	1.129	3.102	0.819	1.490

(c) $th_{ul} = 0.3$

Model	$t_j = t_e - 90\ min$		$t_j = t_e - 60\ min$		$t_j = t_e - 30\ min$		$t_j = t_e$	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CBMA	3.433	6.119	2.812	5.188	2.646	4.983	2.009	3.766
CBMA2	3.604	6.553	2.681	5.266	2.874	5.427	2.000	3.823
AVGT	2.694	5.784	2.394	**4.887**	2.382	**4.797**	1.777	**3.543**
SVR	**2.631**	5.913	**2.236**	5.004	**2.233**	5.082	**1.648**	3.659
KNN	3.067	**5.327**	2.612	4.984	2.503	5.065	1.762	3.593
RIDGE	2.983	5.844	2.449	4.972	2.447	4.864	1.737	3.545

In particular, we present the evaluation results of event impact prediction at different points in time ahead of the event start t_e (i.e. $t_j \in \{t_e - 90\ min, t_e - 60\ min, t_e - 30\ min, t_e - 0\ min\}$, as well as for the different values of the unit load threshold th_{ul} presented in Section 7.4.2, namely $th_{ul} \in \{0.7, 0.5, 0.3\}$. Given the unit load of a specific unit u at time point t_j (i.e. $ul(u, t_j)$), the threshold th_{ul} determines if this unit is considered to be affected, i.e. if the unit load is particularly high. Note that smaller values of the MAE and RMSE error metrics correspond to better model performance.

As we can observe, our approach performs best for all considered configurations with respect to MAE and comparable with respect to RMSE. In particular, the proposed SVR model consistently outperforms all the baselines (CBMA, CBMA2, AVGT) for $th_{ul} \in \{0.7, 0.5, 0.3\}$ for all considered time points.

Other considered regression models such as KNN and RIDGE demonstrate performance comparable to SVR for the higher value of the th_{ul} threshold, i.e. $th_{ul} = 0.7$, whereas for the lower load threshold values $th_{ul} \in \{0.3, 0.5\}$ the performance of KNN and RIDGE is less stable compared to SVR. As in the majority of cases SVR achieves best MAE scores, we conclude that SVR is the overall best suited regression model for the spatial impact prediction task on our dataset.

Among the baselines, the CBMA2 baseline that takes into account event and venue information achieves the best performance for high values of load threshold (i.e. $th_{ul} = 0.7$). CBMA2 also outperforms the CBMA baseline that does not take venue information into account for $th_{ul} \in \{0.7, 0.5\}$. This indicates that considering the average traffic situation is not sufficient for predicting the complex strong impact that events might have on urban traffic situations. For the lower load threshold values $th_{ul} \in \{0.5, 0.3\}$, the AVGT baseline shows the best results with respect to MAE and RMSE among the baselines. This indicates that unit loads observed when using lower threshold values can resemble periodic or temporal traffic fluctuations that can be predicted using historical traffic information used by AVGT.

For the smaller values of the unit load threshold, e.g. $th_{ul} = 0.3$, the AVGT baseline results indicate a comparable performance to the proposed SVR model. With respect to the time dimension, for $th_{ul} = 0.3$ AVGT achieves the best improvement over CBMA2 at $t_j = t_e - 90\ min$, indicating that the impact is relatively weak at this amount of time ahead of the event. We conclude that impacts measured using smaller values of the unit load threshold th_{ul} and longer temporal intervals ahead of the event are likely to capture repeated traffic patterns rather than extraordinary traffic peaks and thus, can be approximated by rather straightforward baseline models.

In general, our proposed approach works best for predicting impacts based on high load thresholds th_{ul}. High thresholds represent a high degree of affectedness and therefore are likely to capture uncommon, event-induced traffic pattern. SVR yields the best results in most of the cases and achieves comparable performance in the remaining cases. The error reduction of CBMA2 over CBMA illustrates the influence of the event venue. Since each venue is represented by an own dimension, the SVR benefits from its ability to separate between dimensions.

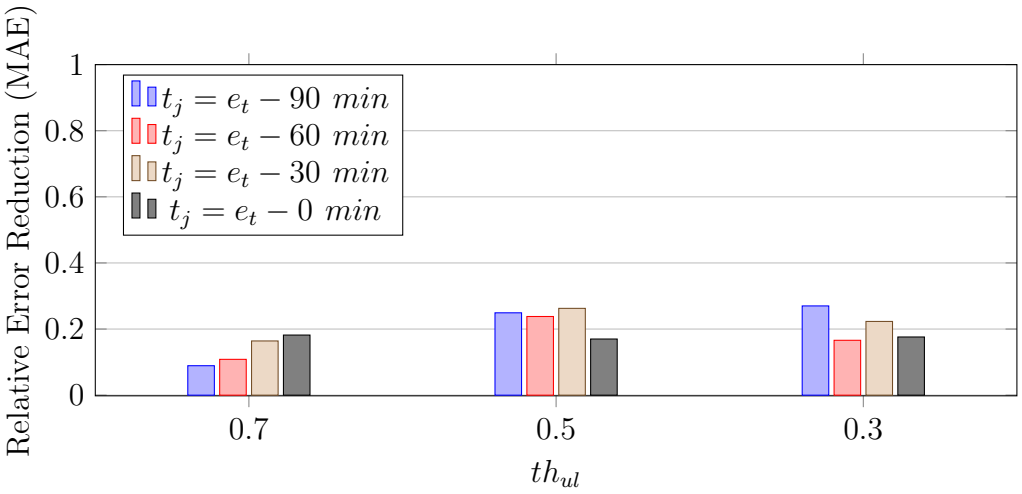


Figure 7.5. Relative error reduction of SVR over CBMA2 for the prediction of the spatial dimension of event impact.

Fig. 7.5 depicts the relative error reduction (with respect to MAE) of SVR over CBMA2 (computed as $\frac{MAE_{CBMA2}-MAE_{SVR}}{MAE_{CBMA2}}$). While SVR is always able to reduce the error (with relative error reduction in the interval [9% - 27%]), SVR improves the most for the lower load threshold values $th_{ul} \in \{0.3, 0.5\}$. Highly likely this is due to CBMA2 baseline lacking information about the average traffic situation, which dominates when using lower load thresholds.

7.9.2 Feature Analysis

Table 7.5 summarizes the best performing feature configurations for all combinations. For $th_{ul} = 0.3$, m_i (the average venue impact) plays an important role. This corresponds to the previous observations that low thresholds can capture periodic traffic patterns. Considering higher threshold values, i.e. $th_{ul} \in \{0.5, 0.7\}$, event characteristics play the most important role for the prediction of the spatial impact. In particular, e_{dow} and e_{wd} are frequently present. This indicates the importance of the day on which an event takes place. The event category e_c is not used in any of the feature sets, i.e. e_c does not add any additional information. A likely explanation is that events of the same category are likely to take place in the same venue, e.g. football matches take place in a football stadium. m_{nau} and i_{nrt} are present in only a few configurations. We conclude that the information about mobility behavior and infrastructure is partly contained in the event characteristics, e.g., in e_v.

Fig. 7.6 presents the feature correlation matrix. Each cell of the matrix corresponds to the value of the suitable correlation metric (or in case of the categorical feature pairs an association metric). The correlation between the pairs of continuous features is measured using Pearson correlation coefficient (PCC). The association

Table 7.5. Best performing feature combinations for all configurations for the prediction of the spatial dimension of event impact. Features included in a configuration are marked with an "x".

th_{ul}	t_j	e_{dow}	e_{st}	e_v	e_c	e_{wd}	e_r	e_{np}	e_{ep}	m_i	m_{nau}	i_{nrt}
0.3	-90		x					x	x	x		
0.3	-60									x		
0.3	-30			x								
0.3	0									x		x
0.5	-90	x	x	x			x					x
0.5	-60	x				x	x		x	x		
0.5	-30	x					x			x		
0.5	0		x				x			x	x	
0.7	-90	x	x					x				
0.7	-60	x				x	x					
0.7	-30	x		x			x					
0.7	0	x	x			x	x	x			x	

between the pairs of categorical features is measured using Cramér's V (CV). The correlation between the pairs including continuous and categorical features is measured using intraclass correlation coefficient (ICC). Features that include multiple dimensions (e.g. i_{nrt} includes one dimension for each road type) are considered separately for each dimension.

As we can observe in Fig. 7.6, some of the event characteristics and some other features in our dataset are highly correlated. In the following, we discuss these correlations in more detail. First, the day of week (e_{dow}) and the working day feature (e_{wd}) are highly correlated in general (CV=0.98). This is expected as e_{wd} can be inferred from e_{dow}. As we can observe from Table 7.5, the best performing feature combinations often include the more fine granular information encoded by the day of week, whereas the combination of the both features can be beneficial for the prediction of the spatial dimension of event impact in some cases. Second, the feature e_r that indicates the reoccurrence of an event is strongly correlated with the event category e_c (CV=0.91) and the event venue e_v (CV=0.71). This is because reoccurring events in our dataset belong to certain categories and typically take place at the venues specialized for this event type. For example, football matches are typically a part of an event series that takes place at the football stadium "HDI-Arena". Consequently, as we can observe in Table 7.5, e_c appears redundant in these settings and is not included in any of the best performing feature combinations, dominated by e_r and e_v. Note that specific correlations between the event venue, the event category and the event reoccurrence observed in our dataset depend on the particular event venue settings in the considered urban region. Therefore, we recommend taking the correlation of these

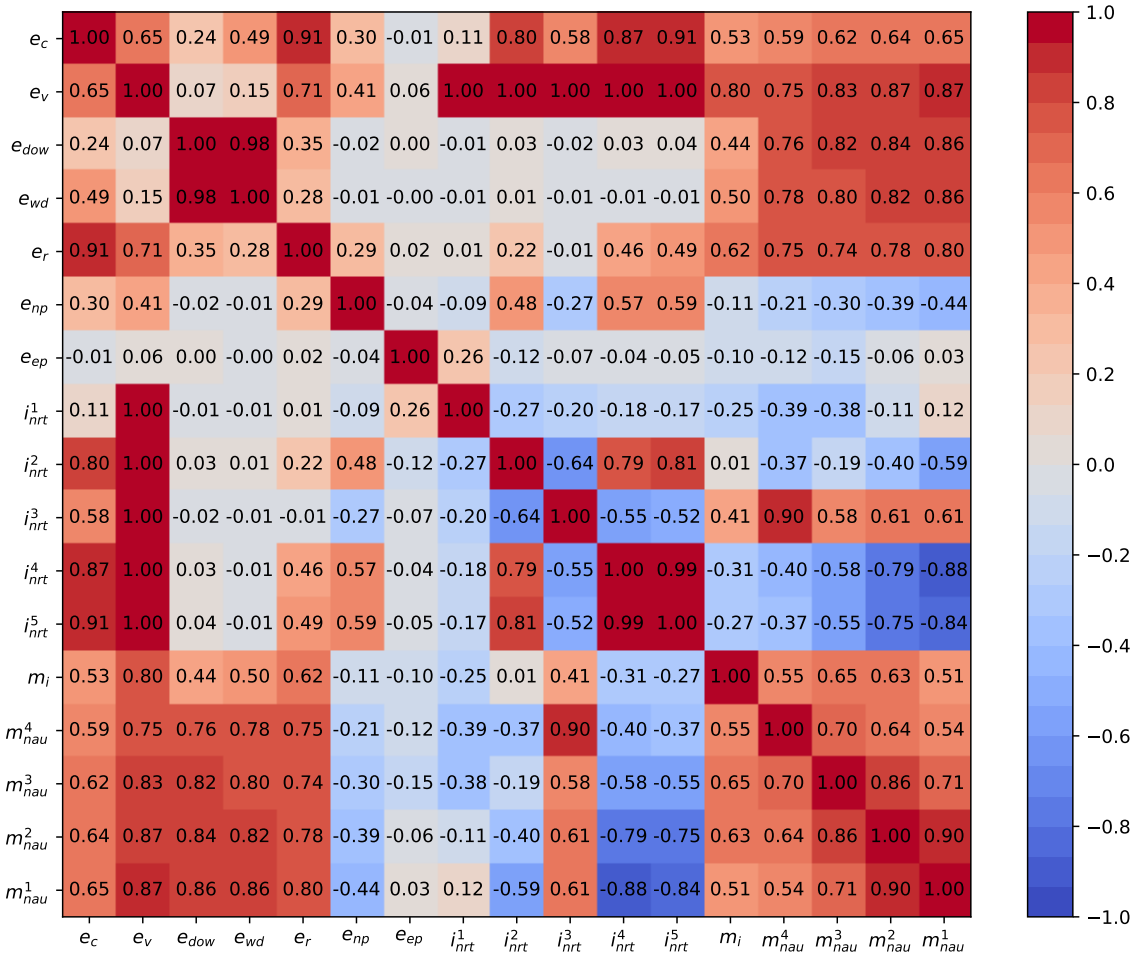


Figure 7.6. The feature correlation matrix. The correlation between the pairs of continuous features was measured as Pearson correlation coefficient (PCC). The association between the pairs of categorical features was measured as Cramér's V (CV). The correlation between the pairs including continuous and categorical features was measured as intraclass correlation coefficient (ICC). The correlation of continuous features with more than one dimension (e.g. i_{nrt}) was determined for each dimension individually. The respective dimension is indicated by the superscript.

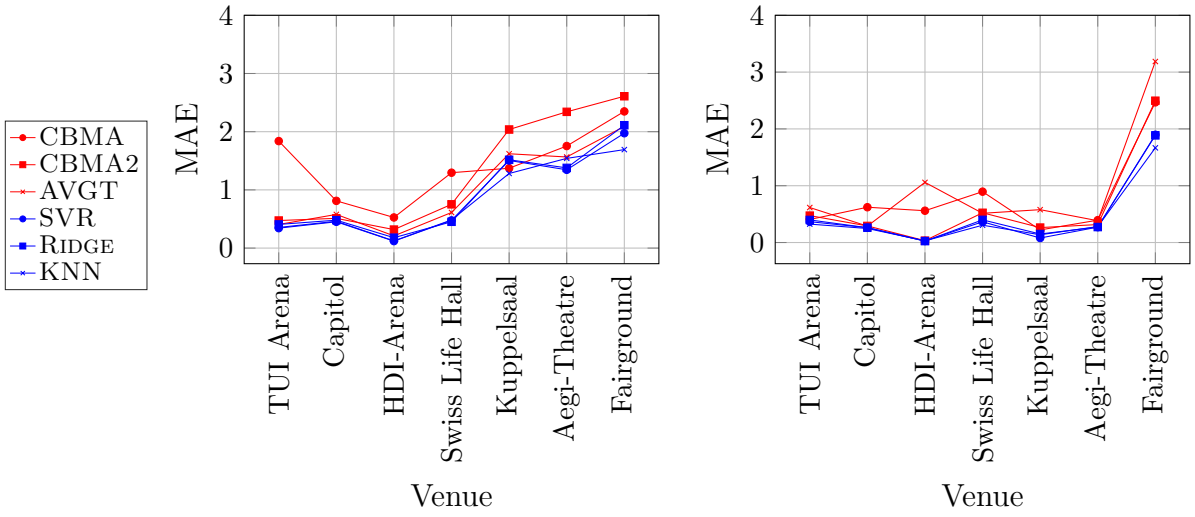
features into consideration while applying the methods proposed in this chapter to further regions. Third, considering the near road types (i_{nrt}), we observe a moderate to strong correlation between the individual road types ($0.52 \leq |(PCC)| \leq 0.99$). Moreover, the road types are perfectly correlated (ICC=1.0) with the event venue e_v, since the road types are static features specific for the particular event venue. Finally, the features encoding the number of nearby affected units (m_{nau}) exhibit mutual correlations. This illustrates the dependency of the number of affected units that are present in different spatial ranges, i.e. the more units are affected nearby the event venue, the more units will be affected at further distances. We further observe a correlation between m_{nau} and e_v in a range of $0.75 \leq \text{ICC} \leq 0.87$. We conclude that the number of units typically affected in presence of special events is venue-specific. In summary, some features describing event characteristics and venues can be highly correlated as we have observed on the dataset considered in this chapter. Whereas highly correlated features do not contribute much to the prediction in general, the particular correlations can depend on the settings in a given urban region. In general, different feature configurations can be required to capture different dimensions of the event impact. I.e., for the coarse grained configurations (i.e. $th_{ul} = 0.3$) historical averages such as m_i play an important role, whereas at finer granularity ($th_{ul} \in \{0.5, 0.7\}$) the consideration of the specific event characteristics is required. The optimal configuration can be determined by performing an exhaustive grid search, i.e. by training individual models for all possible features combination and selecting the best performing one. Note that generally costly process of grid search is feasible in our scenario since the size of the event datasets on which the models are trained is typically small. For larger datasets, ensemble methods such as bagging and boosting can be employed.

7.9.3 Venue Dependence

Fig. 7.7 depicts the MAE scores for $t_j = t_e - \ 30\ min$ and $th_{ul} \in \{0.5, 0.7\}$ with respect to the event venue ve.

In both cases, the MAE is highly dependent on ve. Each venue is located in a different part of the city and therefore is exposed to a different mobility context. Another factor might be the event category in the respective venues, e.g. fairs taking place at the fairground are rather infrequent large-scale events and therefore their impact is harder to predict.

For $th_{ul} = 0.7$ the AVGT performance is worse compared to $th_{ul} = 0.5$. Once more, this illustrates that simple averaging models are only of limited use for the prediction of the spatial dimension of event impact.



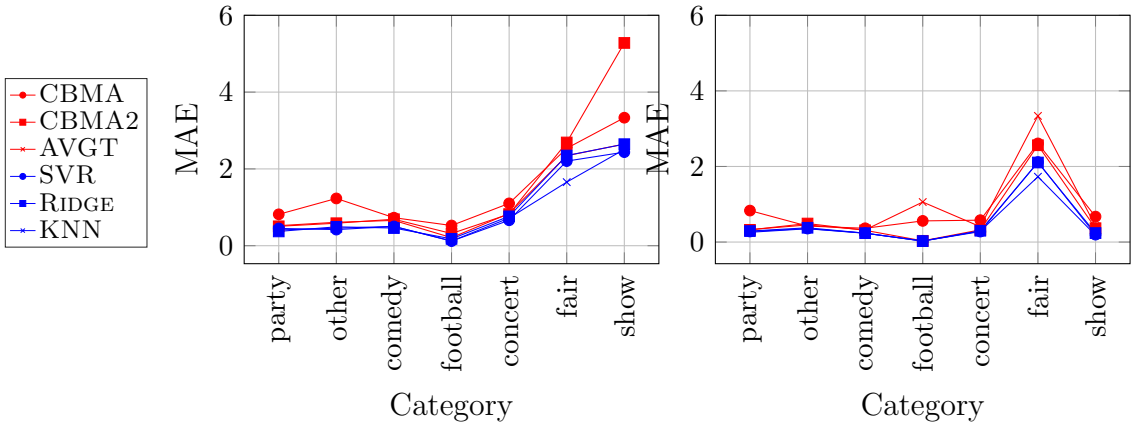
(a) $th_{ul} = 0.5$ (b) $th_{ul} = 0.7$

Figure 7.7. MAE for the prediction of the spatial dimension of event impact for $t_j = t_e - 30\ min$ with respect to event venue ve. Note that the lines between the marks do not correspond to continuous values but are included to improve readability. Red color stands for the baseline approaches, blue color for the proposed regression models.

7.9.4 Event Category Dependence

Fig. 7.8 shows MAE scores with respect to the event category for $t_j = t_e - 30\ min$ and $th_{ul} \in \{0.5, 0.7\}$. Like in Fig. 7.7 red color stands for baseline approaches, blue color for the proposed regression models.

The AVGT baseline shows the same behavior as already observed for the venue analysis in Section 7.9.3. A difference between $th_{ul} = 0.5$ and $th_{ul} = 0.7$ are the scores for the events in the category "show". In our case, shows take place at venues which are especially sensitive to th_{ul}, e.g. the "Aegi-Theatre" which is a central spot within the city of Hanover and surrounded by some hub traffic nodes. For both cases, the impact of fairs results in rather high MAE scores. Here the KNN model achieves the best performance. This might be an indicator of lack of sufficient training data for SVR, especially considering that the task of predicting the impact of a fair on complex urban traffic networks is particularly challenging. Fairs typically are all day events where participants may arrive at nearly any time of the day. Therefore, the impact of fairs is likely to be more diffuse and therefore harder to predict at few fixed time points in the advance of an event. Furthermore, fairs show higher diversity with respect to the overall number of participants compared to other event categories in our dataset.



(a) $th_{ul} = 0.5$ (b) $th_{ul} = 0.7$

Figure 7.8. MAE for the prediction of the spatial dimension of event impact for $t_j = t_e - 30\ min$ with respect to event category. Note that the lines between the marks do not correspond to continuous values but are included to improve readability. Red color stands for the baseline approaches, blue color for the proposed regression models.

7.10 Temporal Impact Evaluation

This section presents the experimental evaluation of the prediction of the temporal dimension of event impact. In particular, we discuss the overall performance as well as the venue-dependent performance.

7.10.1 Temporal Impact Prediction Performance

This section discusses the performance of the prediction of the temporal dimension of event impact.

First of all, the overall performance of the proposed approach is presented. Tables 7.6a, 7.6b, 7.6c present the results of the proposed approach for temporal impact prediction in terms of MAE and RMSE. Analogously to the spatial impact prediction, we present the results at time points ahead of the event start t_e (i.e. $t_j \in \{t_e - 90\ min, t_e - 60\ min, t_e - 30\ min, t_e - 0\ min\}$) as well the configurations discussed in Section 7.8, namely $th_{ul} = 0.5$, $th_{ta} = 0.3$; $th_{ul} = 0.5$, $th_{ta} = 0.7$; and $th_{ul} = 0.7$, $th_{ta} = 0.7$.

As we can observe, our approach performs best for all considered configurations and time points with respect to MAE and RMSE. We can observe a consistent behavior among all configurations which we detail in the following.

Among the considered regression models, the best scores are achieved by SVR and KNN. In most of the cases SVR achieves the lowest errors for

Table 7.6. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for the prediction of the temporal dimension of event impact. Scores outperforming the baselines are underlined, the best scores are marked in bold. t_e is event start time.

(a) $th_{ul} = 0.7$, $th_{ta} = 0.7$

Model	$t_j = t_e - 90\ min$		$t_j = t_e - 60\ min$		$t_j = t_e - 30\ min$		$t_j = t_e$	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CBMA	1.370	1.876	1.280	1.886	1.313	2.045	1.343	2.068
CBMA2	0.808	1.345	0.906	1.584	1.076	1.878	1.150	1.971
SVR	**0.713**	1.350	**0.809**	1.584	0.912	1.734	0.967	1.997
KNN	0.743	**1.262**	0.821	**1.480**	**0.880**	**1.678**	**0.960**	**1.769**
RIDGE	0.750	1.319	0.821	1.558	0.956	1.831	1.073	1.970

(b) $th_{ul} = 0.5$, $th_{ta} = 0.7$

Model	$t_j = t_e - 90\ min$		$t_j = t_e - 60\ min$		$t_j = t_e - 30\ min$		$t_j = t_e$	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CBMA	1.801	2.243	1.688	2.182	1.656	2.235	1.662	2.249
CBMA2	0.762	1.270	0.875	1.508	1.061	1.795	1.131	1.900
SVR	**0.678**	**1.174**	**0.796**	**1.456**	**0.905**	1.694	0.967	1.864
KNN	0.739	1.332	0.822	1.488	0.914	**1.627**	**0.934**	**1.680**
RIDGE	0.727	1.214	0.832	1.480	0.985	1.706	1.062	1.782

(c) $th_{ul} = 0.5$, $th_{ta} = 0.3$

Model	$t_j = t_e - 90\ min$		$t_j = t_e - 60\ min$		$t_j = t_e - 30\ min$		$t_j = t_e$	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
CBMA	1.854	2.281	1.772	2.316	1.800	2.453	1.776	2.432
CBMA2	0.796	1.228	0.928	1.518	1.093	1.795	1.147	1.910
SVR	**0.691**	**1.133**	**0.809**	**1.461**	0.928	1.698	1.017	1.903
KNN	0.748	1.205	0.831	1.518	**0.909**	**1.630**	**0.920**	**1.641**
RIDGE	0.745	1.187	0.873	1.545	1.031	1.704	1.080	1.801

$t_j \in \{t_e - 90\ min, t_e - 60\ min\}$ and KNN for $t_j \in \{t_e - 30\ min, t_e\}$. This might indicate that the complexity of the traffic situation increases in temporal proximity to the event such that not enough training data for complex models like SVR is available. Nevertheless, the simpler KNN-model can still be effectively trained. While RIDGE outperforms the baselines as well, it never achieves better scores than the other considered regression models.

Considering different configurations, we observe that high thresholds lead to the highest error scores e.g. KNNs MAE of 0.96 for $t_j = t_e$, $th_{ul} = 0.7$ and $th_{ta} = 0.7$ in Table 7.6a. We conclude that the difficulty of the regression task increases with the value of the thresholds. This is expected, as higher thresholds result in small typically affected subgraphs that may contain only few units. Thus, the regression task needs to predict delays at a fine granularity, which is more difficult than with lower threshold values.

In comparison to the performance of the prediction of the spatial event impact, our proposed models consistently outperform the baselines with respect to RMSE for the temporal impact prediction. This indicates, that the task of prediction of the temporal dimension of event impact is not as sensitive to the extreme values as the task of spatial impact prediction.

Regarding the baselines, CBMA2 outperforms CBMA in all cases. Since the only difference between the baselines is that CBMA2 also considers the event venue ve, we conclude that the venue plays an important role. Our proposed approach achieves a relative error reduction of up to 19.8% over CBMA2 in this task.

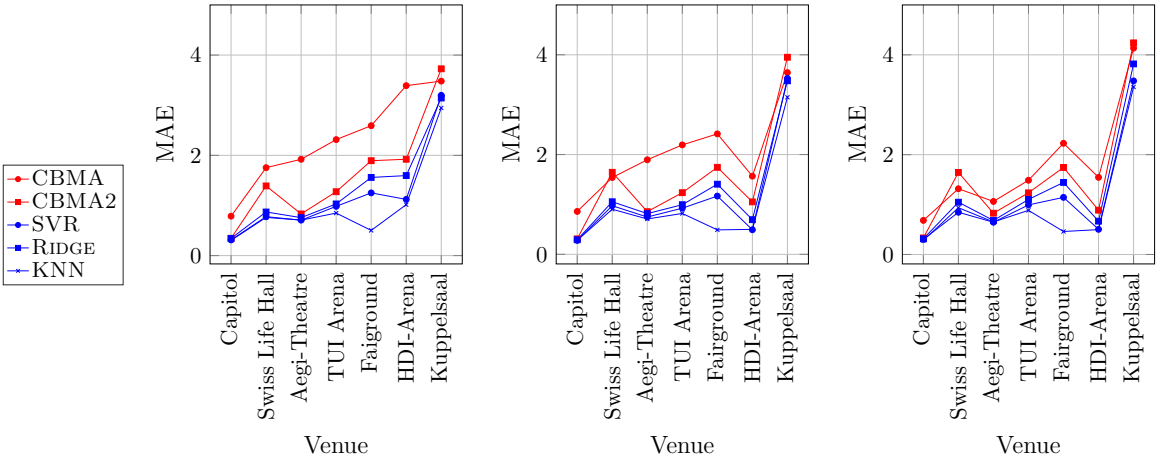
7.10.2 Venue Dependence

Fig. 7.9 depicts MAE scores at time point $t_j = t_e - 30\ min$ with respect to the event venue and the model for the chosen configurations.

Generally, both SVR and KNN exhibit similar performance across all configurations except for the venue "Fairground". The fairground mostly hosts fairs, which seem to be problematic for the temporal impact prediction. For this venue, the KNN achieves by far the best performance for all configurations. This might be an indicator for an insufficient amount of training data for more complex models like SVR. This observation is similar to the respective observation for the task of spatial impact prediction.

The venue "TUI-Arena", which mainly hosts concerts, is located within the same fairground. However, we can not observe a similar problem. We conclude that the difficulty of the problem is not only induced by the geographic location of the venue but also by the event type that takes place.

The venue "Kuppelsaal" clearly exhibits the highest errors. The venue is located in the city center. Therefore, the traffic around the venue is likely to be subject to a large variety of the influence factors, which makes it especially hard to predict the



(a) $th_{ul} = 0.5$, $th_{ta} = 0.3$ (b) $th_{ul} = 0.5$, $th_{ta} = 0.7$ (c) $th_{ul} = 0.7$, $th_{ta} = 0.7$

Figure 7.9. MAE for the prediction of the temporal dimension of event impact for $t_j = t_e - 30min$ with respect to event venue ve. Note that the lines between the marks do not correspond to continuous values but are included to improve readability. Red color stands for the baseline approaches, blue color for the proposed regression models.

temporal dimension of event impact for this particular venue.

7.11 Evaluation Summary

This section summarizes the evaluation of the prediction of both the spatial and temporal dimensions of event impact. In particular, findings, limitations and future work are discussed.

Findings. Our approach outperforms the baselines for the prediction of the spatial and temporal dimensions of event impact. Overall, we observe an error reduction by up to 27% (spatial dimension) and 19.7% (temporal dimension) by our method compared to the best performing baseline (CBMA2), dependent on the configuration.

With higher impact values (th_{ul} threshold of 0.7 compared to 0.5), the differences between the CBMA2 baseline and the proposed regression model increases. Smaller spatial impact values can be well predicted by the AVGT baseline, that captures the average traffic information, whereas specialized event models enable more accurate prediction of more significant event-induced traffic impact. This underlines our observation that small impact thresholds tend to characterize impacts which might be due to a variety of influence factors, most notably temporal fluctuations in traffic, while larger threshold values tend to be directly connected to extra-ordinary incidents, such as planned special events.

Among the adopted regression models, SVR shows overall the best performance, whereas in the individual configurations, especially for higher values of th_{ul}, other regression models can perform slightly better. Therefore, we recommend adopting SVR as a default model. The available data is typically use case specific (e.g. some cities may be dominated by a few events or event venues, other cities may be very dependent on a couple of major roads). When fine-tuning the model for a particular use case, we recommend to additionally investigate the performance of the other regression models considered in this chapter (KNN, RIDGE) to achieve an optimal performance. Among the adopted features, the most important feature group is event characteristics, which reflect diverse information such as the venue, the time and the day of week of the event. There are several possible combinations that indicate similar performance (which is useful if not all features are available or are costly to compute).

The largest errors in prediction of event impact occur at the fairground venue, which is partially due to rather diverse events that happen in this venue, where the number of participants varies from 9000 to 65000 in our dataset.

Limitations & Future Work. Our definition of event impact fits best to specific types of events that have a well-defined starting point where the majority of the participants arrives at the venue. This definition fits well to certain event types, such as football games or concerts. However, it is less suitable for events spread over a longer duration, such as fairs, where the attendance of individuals is spread over a longer period of time.

In this chapter, we provide the definition of the spatial and the temporal dimensions of event impact. To this end, we facilitate prediction of the maximum distance from the event venue where event-induced load can be observed at a particular point in time as well as an average delay that is present on the units contained in the typically affected subgraphs of the transportation graph. Furthermore, we facilitate the identification of the subgraphs typically affected by events.

The sets of features considered in this chapter represent event-, mobility- and infrastructure characteristics. We assume that a wide range of additional factors can influence event impact, such as weather conditions, as well as availability and use of public transportation infrastructure. In the future work, we will consider such features to incrementally enhance models and predictions.

The correlations between planned special events and traffic delays considered in this chapter may also be impacted by further unobserved factors, such as traffic accidents, temporal construction sites or extreme weather conditions in geographic and temporal proximity of the event. Therefore, the observed traffic delays are not necessarily always caused by the event-specific traffic, i.e. the vehicles going to or from the event venue. Nevertheless, regular patterns observed for specific event venues and event types enable an accurate prediction of event impact on urban traffic. Causal analysis is an interesting direction of future research, subject to availability of the corresponding data.

In this chapter, we do not address handling of multiple simultaneous events and focus on the impact prediction for individual large-scale events. Although in general the question of cumulative event impact prediction is important, it appears more relevant for cities, where several large-scale event venues are located in geographic proximity. In the settings of the urban region of Hanover the relevant venues (i.e., the venues with the capacity of at least 1000 participants) are distributed across the city, such that we do not expect to observe cumulative event impacts regularly. Handling of simultaneous events can be an interesting direction for future research, dependent on the event venue settings in the specific urban region.

7.12 Discussion

In this chapter, we used geographic Web information to analyze and predict the spatial and temporal dimensions of event impact. In particular, we use (i) event and event venue information collected from knowledge graphs and the Web, and (ii) road network information collected from OpenStreetMap.

We proposed a method for quantifying the spatial and temporal dimensions of event impact for planned special events in urban areas. We applied this method to create training data for supervised machine learning models. In particular, we identify affected subraphs of the road network that measure the spatial dimension of event impact. We further determine typically affected subgraphs that consist of commonly affected roads across events. We derive the temporal dimension of event impact from the typically affected subgraphs by measuring the average delay during an event.

We presented supervised regression models that accurately predict the distance from the event venue where event-induced traffic can be observed ahead of the event start time. Furthermore, these models facilitate the prediction of the average delay that can be observed on the typically affected subgraphs of the transportation graph. Our evaluation results on a set of real-world events in seven categories demonstrate that the proposed method outperforms existing and naïve baselines in various configurations for both considered dimensions of the event impact. We analyzed the impact of feature sets in different categories, including event-, mobility- and infrastructure characteristics. Potential applications of the proposed model include integrating the proposed models into routing algorithms and next location recommendation.

8 Conclusion and Future Work

The amount of available geographic information on the Web is continuously growing. The quality, coverage, and description types heavily vary across geographic regions and data sources. On the one hand, this information has potentially high value for many location-based applications such as route planning, POI recommendation, and geographic information retrieval. On the other hand, the often incomplete, isolated, and not verified datasets hinder the effective use of geographic Web information. Existing information enrichment and validation approaches fail to address the intrinsic data heterogeneity of, for instance, volunteered geographic information sources such as OpenStreetMap. We tackled these problems by developing specialized machine learning approaches that enrich and validate geographic Web information and enable the effective data use to its full advantage.

8.1 Summary of Contributions

In this thesis, we investigated various sources of geographic information on the Web such as OpenStreetMap, knowledge graphs, and semantic Web markup. In the following, we summarize our contributions in the areas of *validation of geographic Web information*, *enrichment of geographic Web information*, and *applications of geographic Web information*.

8.1.1 Validation of Geographic Web Information

In Chapter 3, we have presented an approach for validating geographic Web information trough automated vandalism detection in OpenStreetMap. Vandalism detection in OpenStreetMap is critical and remarkably challenging due to the large scale of the dataset, the sheer number of contributors, various vandalism forms, and the lack of annotated data to train machine learning algorithms. We presented the Ovid (OpenStreetMap Vandalism Detection) model - a supervised machine model - to

address the aforementioned challenges.

Ovid relies on a neural network architecture that adopts a multi-head attention mechanism to summarize information indicating vandalism from OpenStreetMap changesets effectively. Furthermore, we extract a dataset of real-world vandalism incidents from the OpenStreetMap's edit history for the first time and provide this dataset as open data. Our evaluation results on real-world vandalism data demonstrate that the proposed Ovid method outperforms the baselines by eight percentage points regarding the F1 score on average.

8.1.2 Enrichment of Geographic Web Information

In this thesis, we have considered two problems of enriching geographic Web information, i.e., (i) the enrichment of OpenStreetMap with links to knowledge graphs and (ii) the enrichment of Web markup with missing categorical information.

Enrichment of OpenStreetMap with Links to Knowledge Graphs

In Chapter 4, we tackled the problem of enriching OpenStreetMap with identity links to knowledge graphs, i.e., to identify knowledge graph entities that correspond to the same real-world entity as a given OSM node. The problem of link discovery in these settings is particularly challenging due to the lack of a strict schema and heterogeneity of the user-defined node representations in OSM. We introduced the OSM2KG model for link discovery consisting of the candidate generation, feature extraction, and link classification stages. For candidate generation, OSM2KG employs a geographic blocking approach. For feature extraction, we introduced the *key-value embedding* to capture the semantics of OSM nodes. The key-value embedding addresses the intrinsic inconsistency of OSM annotations by employing an unsupervised embedding model to infer latent representations of OSM nodes. For knowledge graph entities, we introduced selected features to capture the entity semantics. We combined the key-value embeddings and entity features in a supervised classification model that effectively discovers links. Our experiments conducted on several OSM datasets, as well as the Wikidata and DBpedia knowledge graphs, demonstrate that OSM2KG can reliably discover identity links, achieving an F1 score of 92.05% on Wikidata and of 94.17% on DBpedia on average.

Enrichment of Web Markup with Missing Categorical Information

In Chapter 5, we describe the problem of enriching Web markup with missing categorical information. We highlighted the challenges arising from the overall distribution of Web markup data, such as property sparsity, the use of incomplete types, and incorrect annotations. We addressed these challenges by employing domain-level information and vocabulary usage as features for a supervised type classification

model. Furthermore, we exploited the domain-level information to extract more diverse training datasets for the machine learning model. Our experiments, conducted on properties of events and movies, show a performance of 79% and 83% F1 score correspondingly, significantly outperforming existing baselines.

8.1.3 Applications of Geographic Web Information

We highlighted the relevance of rich geographic Web information in the example of two application scenarios. First, we constructed a large-scale corpus of latent representations of OSM entities that enable many downstream geographic machine learning applications. Second, we investigated the prediction of event impact on road traffic by exploiting diverse geographic Web information such as road network information, event information, and event venue information.

The GeoVectors Corpus of Geographic Entity Embeddings

In Chapter 6, we presented the GeoVectors corpus containing latent representations of OSM entities. Using standard OSM entities in machine learning models is challenging due to the large scale of OSM, the extreme heterogeneity of entity annotations, and a lack of a well-defined ontology to describe entity semantics and properties. To address these issues, we introduced the GeoVectors corpus of OSM embeddings. The GeoVectors corpus captures the semantic and geographic dimensions of OSM entities in latent representations and makes these entities directly accessible to machine learning algorithms and semantic applications. GeoVectors is a unique, comprehensive world-scale, linked resource covering the entire OSM dataset and providing latent representations of over 980 million geographic entities in 180 countries. Furthermore, we created a semantic description of GeoVectors, including identity links to the Wikidata and DBpedia knowledge graphs to supply context information. We made this semantic description available as SPARQL endpoint – a semantic interface offering direct access to the GeoVectors metadata.

Application to Event Impact Prediction

In Chapter 7, we tackled the problem of predicting the spatial and temporal impact of special public events on road traffic. First, we introduced two novel formalizations to measure the spatial and the temporal dimension of event impact on road traffic. Then, we presented a supervised machine learning approach to predict the event impact from geographic Web information such as venue information, event information, and road network information. Our evaluation results on real-world event data containing events from several venues in the Hanover region in Germany demonstrate that the proposed combinations of event-, mobility- and infrastructure-related features show the best performance. The proposed spatio-temporal model can accurately predict

the spatial and temporal impact on road traffic in the event context in this region.

8.2 Open Research Directions

In this thesis, we have presented novel approaches for enriching and validating geographic Web information. Furthermore, we have discussed two application scenarios of geographic Web information. The findings presented in these areas pave the way for the following connected research areas.

Data Fusion of Geographic Web Information

In this thesis, we have presented multiple approaches to improve the data quality of individual geographic Web information sources. Another research direction to improve the data quality is to combine entity descriptions from overlapping data sources. *Data fusion* denotes the process of combining multiple incomplete data sources to obtain a single consistent, correct, and more complete data source. Typical data fusion challenges include *data inconsistency*, *data confliction* and *data imperfection* [MJYP20]. While data fusion approaches are often adopted in distributed sensor networks, e.g., for *Internet of Things* applications [AMK+17], there is a lack of models for the fusion of geographic Web data.

The fusion of geographic data requires addressing unique challenges. First, the comparison of geographic entities requires the use of a common coordinate system. Different data sources may use different coordinate systems and may have to be transformed accordingly. Furthermore, the comparison of geographic geometries often requires a planar projection such that additional coordinate system transformations may be necessary. Second, the fusion of inconsistent geographic data types requires specialized algorithms. For instance, it is not trivial to compare or merge a point with, e.g., a polygonal chain. The Web data provenance imposes additional challenges to the fusion process. First, the large scale of available geographic Web data requires scalable fusion architectures. Second, as discussed in Chapter 4 and 5 the heterogeneity in Web data sources is typically high and requires adaptive algorithms. Third, as observed in Chapter 3, the trustworthiness of Web data sources varies and may affect the information correctness.

In Chapter 4 we presented an approach for discovering identity relations between geographic Web entities. First evidence indicates that these links are helpful for, e.g., schema alignment of OSM and knowledge graphs [DTD21]. Therefore, these identity relations are potentially valuable information for future data fusion pipelines.

Exploitation of Tabular and Textual Information

This thesis considers semi-structured geographic Web information sources, such as OpenStreetMap, knowledge graphs, and semantic Web markup. Other widely used Web information sources are tabular (e.g., tables on websites) and textual information (e.g., Wikipedia texts). These sources have proven their utility for various problems, including knowledge base augmentation, question answering, and knowledge graph generation [ZB20, HLS18].

However, the exploitation of these information sources in the geographic information domain is widely unexplored due to the following challenges. First, the detection of geographic references in text or tabular data is not trivial. For instance, the name "Sydney" could refer to the city in Australia or to a person's first name. The natural language description of geographic places such as "near" or "next to" introduces further ambiguity of geographic references. Second, inconsistent descriptions in texts and tables hinder the extraction process. For instance, one table could provide the coordinates of a city as point geometries, while another source could define polygons to describe a city's location. These challenges require specialized algorithms to extract geographic Web information from tabular and textual information effectively.

Application to Geographic Information Retrieval

Geographic information retrieval (GIR) is a field focussing on addressing geographic information needs [PCJ+18]. Popular GIR problems include location-based Web search [AB07] and geographic question answering [CGMS21]. Similar to general information retrieval, modern GIR approaches rely on machine learning models, for instance, to rank the relevance of search results for a particular query. These machine learning models are inherently limited by data quality. In this thesis, we have introduced several approaches to validate (Chapter 3) and enrich (Chapter 4, Chapter 5) geographic Web information and ultimately to increase the data quality. Further, we constructed a corpus of geographic entity embeddings that enable fast adoption of geographic entities in machine learning applications in Chapter 6. As a future research direction, it would be interesting to explore the effects of data quality in downstream GIR applications.

A

Curriculum Vitae

Studies

since 2016 **PhD Studies**
Gottfried Wilhelm Leibniz Universität Hannover
L3S Research Center, Hannover, Germany

2014-2016 **Master of Science** in Computer Science
Gottfried Wilhelm Leibniz Universität Hannover
Master thesis: Temporal Queries on Evolving Graphs

2011-2014 **Bachelor of Science** in Computer Science
Gottfried Wilhelm Leibniz Universität Hannover
Bachelor thesis: Evaluation eines Graph-Datenbanksystems

Professional Experience

2020-2021 **Project lead**. d-E-mand project (funded by the BMWi)
2019-2020 **Project lead**. CampaNeo project (funded by the BMWi)
since 2017 **Reviewer** for several journals including *Geo-spatial Information Science* and *Urban Science*. Subreviewer for several conferences.
2017-2020 **Project member**. Data4UrbanMobility project (funded by the BMBF)
2017-2019 **Teaching Assistant**. Artificial Intelligence I

Publications

Please refer to the publication list in the foreword of this thesis.

Bibliography

[AB07] Dirk Ahlers and Susanne Boll. Location-based Web Search. In *The Geospatial Web, How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society*, pages 55–66. Springer, 2007.

[ADG+14] Muhammad Tayyab Asif, Justin Dauwels, Chong Yang Goh, Ali Oran, Esmail Fathi, Muye Xu, Menoth Mohan Dhanya, Nikola Mitrovic, and Patrick Jaillet. Spatiotemporal Patterns in Large-Scale Traffic Speed Prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):794–804, 2014.

[AHC14] Berk Anbaroglu, Benjamin Heydecker, and Tao Cheng. Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. *Transportation Research Part C: Emerging Technologies*, 48:47–65, 2014.

[aHP15] Robert Meusel andf Heiko Paulheim. Heuristics for Fixing Common Errors in Deployed schema.org Microdata. In *Proceedings of the 12th European Semantic Web Conference*, ESWC'15, pages 152–168, 2015.

[ALVI16] Tarique Anwar, Chengfei Liu, Hai Le Vu, and Md. Saiful Islam. Tracking the Evolution of Congestion in Dynamic Urban Road Networks. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, CIKM'16, pages 2323–2328, 2016.

[AMK+17] Furqan Alam, Rashid Mehmood, Iyad A. Katib, Nasser N. Albogami, and Aiiad Albeshri. Data Fusion and IoT for Smart Ubiquitous Environments: A Survey. *IEEE Access*, 5:9533–9554, 2017.

[ASF+17] Vyron Antoniou, Linda See, Giles Foody, Cidália Costa Fonte, Peter Mooney, Lucy Bastin, Steffen Fritz Hai-Ying Liu, Ana-Maria Olteanu-

Raimond, and Rumiana Vatsev. The Future of VGI. In *Mapping and the Citizen Sensor*, pages 377–390. Ubiquity Press, 2017.

[ASN18] Abdullah Fathi Ahmed, Mohamed Ahmed Sherif, and Axel-Cyrille Ngonga Ngomo. On the Effect of Geometries Simplification on Geo-spatial Link Discovery. In *Proceedings of the 14th International Conference on Semantic Systems*, SEMANTICS 2018, pages 139–150, 2018.

[AYW+16] Shi An, Haiqiang Yang, Jian Wang, Na Cui, and Jianxun Cui. Mining urban recurrent congestion evolution patterns from GPS-equipped vehicle mobility data. *Information Sciences*, 373:515–526, 2016.

[AZMH15] Jamal Jokar Arsanjani, Alexander Zipf, Peter Mooney, and Marco Helbich. An Introduction to OpenStreetMap in Geographic Information Science: Experiences, Research, and Applications. In *OpenStreetMap in GIScience - Experiences, Research, and Applications*, pages 1–15. Springer, 2015.

[Bac14] David Backett. RDF 1.1 N-Triples, 2014. Accessed: July 30th 2021, https://www.w3.org/TR/2014/REC-n-triples-20140225/.

[Bal14] Andrea Ballatore. Defacing the Map: Cartographic Vandalism in the Digital Commons. *The Cartographic Journal*, 51(3):214–224, 2014.

[BB12] James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012.

[BCC+13] Khalid Belhajjame, James Cheney, David Corsar, Daniel Garijo, Stian Soiland-Reyes, Stephan Zednik, and Jun Zhao. PROV-O: The PROV Ontology, 2013. Accessed: August 12th 2021, https://www.w3.org/TR/prov-o/.

[BEM+13] Christian Bizer, Kai Eckert, Robert Meusel, Hannes Mühleisen, Michael Schuhmacher, and Johanna Völker. Deployment of RDFa, Microdata, and Microformats on the Web - A Quantitative Analysis. In *Proceedings of the 12th International Semantic Web Conference*, ISWC'13, pages 17–32, 2013.

[BEP+08] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, SIGMOD'08, pages 1247–1250, 2008.

[BHB09] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[BJ90] George Edward Pelham Box and Gwilym Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Inc., 1990.

[BKH16] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *CoRR*, abs/1607.06450, 2016.

[BL06] Tim Berners-Lee. Linked Data, 2006. Accessed: July 30th 2021, https://www.w3.org/DesignIssues/LinkedData.html.

[BMP21] Christian Bizer, Robert Meusel, and Anna Primpeli. Web Data Commons - Microdata, RDFa, JSON-LD, and Microformat Data Sets, 2021. Accessed: August 12th 2021, http://webdatacommons.org/structureddata/index.html.

[BNZ14] Christopher Barron, Pascal Neis, and Alexander Zipf. A Comprehensive Framework for Intrinsic OpenStreetMap Quality Analysis. *Transactions in GIS*, 18:877–895, 2014.

[BOS09] Azuraliza Abu Bakar, Zulaiha Ali Othman, and Nor Liyana Mohd Shuib. Building a new taxonomy for data discretization techniques. In *Proceedings of the 2nd Conference on Data Mining and Optimization*, DMO'09, pages 132–140, 2009.

[CAS+16] Silvio D. Cardoso, Flor K. Amanqui, Kleberson J. A. Serique, José L. C. dos Santos, and Dilvan A. Moreira. SWI: A Semantic Web Interactive Gazetteer to support Linked Open Data. *Future Generation Computer Systems*, 54:389–398, 2016.

[CBHK02] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[CFC+09] Wei Chen, Zhongqian Fu, Ruizhi Chen, Yuwei Chen, Octavian Andrei, Tuomo Kroger, and Jianyu Wang. An integrated GPS and multi-sensor pedestrian positioning system for 3D urban navigation. In *2009 Joint Urban Remote Sensing Event*, pages 1–6, 2009.

[CGD20] Tarcísio Souza Costa, Simon Gottschalk, and Elena Demidova. Event-QA: A Dataset for Event-Centric Question Answering over Knowledge Graphs. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, CIKM '20, pages 3157–3164, 2020.

[CGMS21] Danish Contractor, Shashank Goel, Mausam, and Parag Singla. Joint Spatio-Textual Reasoning for Answering Tourism Questions. In *Proceedings of The Web Conference*, WWW'21, pages 1978–1989, 2021.

[Chu12] Younshik Chung. Assessment of non-recurrent congestion caused by precipitation using archived weather and traffic flow data. *Transport Policy*, 19(1):167–173, 2012.

[CYH+18] Zhenhua Chen, Yongjian Yang, Liping Huang, En Wang, and Dawei Li. Discovering Urban Traffic Congestion Propagation Patterns With Taxi Trajectory Data. *IEEE Access*, 6:69481–69491, 2018.

[DA16] Andrea Dessi and Maurizio Atzori. A machine-learning approach to ranking RDF properties. *Future Generation Computer Systems*, 54:366–377, 2016.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT'19, 2019.

[DJHM13] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the I-SEMANTICS 2013 - 9th International Conference on Semantic Systems*, ISEM '13, pages 121–124, 2013.

[DON13] Elena Demidova, Irina Oelze, and Wolfgang Nejdl. Aligning freebase with the YAGO ontology. In *22nd ACM International Conference on Information and Knowledge Management*, CIKM'13, pages 579–588, 2013.

[DTD21] Alishiba Dsouza, Nicolas Tempelmeier, and Elena Demidova. Towards Neural Schema Alignment for OpenStreetMap and Knowledge Graphs. In *Proceedings of the 20th International Semantic Web Conference*, ISWC'21, pages 56–73, 2021.

[DTY+17] Stefan Dietze, Davide Taibi, Ran Yu, Phil Barker, and Mathieu d'Aquin. Analysing and Improving Embedded Markup of Learning Resources on the Web. In *Proceedings of the 26th International World Wide Web Conference*, pages 283–292. WWW'17, 2017.

[DTY+21] Alishiba Dsouza, Nicolas Tempelmeier, Ran Yu, Simon Gottschalk, and Elena Demidova. WorldKG: A World-Scale Geographic Knowledge Graph. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, CIKM'21, 10 pages, 2021.

[EBB+18] Mohamed Ben Ellefi, Zohra Bellahsene, John G. Breslin, Elena Demidova, Stefan Dietze, Julian Szymanski, and Konstantin Todorov. RDF dataset profiling - a survey of features, methods, vocabularies and applications. *Semantic Web*, 9(5):677–705, 2018.

[EGT+17] Kemele M. Endris, José M. Giménez-García, Harsh Thakkar, Elena Demidova, Antoine Zimmermann, Christoph Lange, and Elena Simperl. Dataset Reuse: An Analysis of References in Community Discussions, Publications and Data. In *Proceedings of the Knowledge Capture Conference*, K-CAP'17, pages 5:1–5:4, 2017.

[FBMR18] Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web*, 9(1):77–129, 2018.

[FCAC17] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. POI2Vec: Geographical Latent Representation for Predicting Future Visitors. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 102–108, 2017.

[FPEG17] Mohammadhani Fouladgar, Mostafa Parchami, Ramez Elmasri, and Amir Ghaderi. Scalable deep traffic flow neural networks for urban traffic congestion prediction. In *2017 International Joint Conference on Neural Networks*, IJCNN'17, pages 2251–2258, 2017.

[Gan16] Arun Ganesh. Scaling OpenStreetMap with Wikidata knowledge, 2016. Accessed: July 14th 2021, https://blog.mapbox.com/scaling-openstreetmap-with-wikidata-knowledge-675d4495815f.

[GBC16] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. *Deep Learning*. MIT Press, 2016.

[GBG+18] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomás Mikolov. Learning Word Vectors for 157 Languages. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*, LREC'18, 2018.

[GBM16] Ramanathan V. Guha, Dan Brickley, and Steve Macbeth. Schema.org: Evolution of Structured Data on the Web. *Communications of the ACM*, 59(2):44–51, 2016.

[GD19] Simon Gottschalk and Elena Demidova. EventKG - the hub of event knowledge on the web - and biographical timeline generation. *Semantic Web*, 10(6):1039–1070, 2019.

[GHW+19] Jingyue Gao, Yuanduo He, Yasha Wang, Xiting Wang, Jiangtao Wang, Guangju Peng, and Xu Chu. STAR: Spatio-Temporal Taxonomy-Aware Tag Recommendation for Citizen Complaints. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM'19, pages 1903–1912, 2019.

[GKSS13] Thomas Gottron, Malte Knauf, Stefan Scheglmann, and Ansgar Scherp. A Systematic Investigation of Explicit and Implicit Schema Information on the Linked Open Data Cloud. In *Proceedings of the 10th European Semantic Web Conference*, volume 7882 of *ESWC'13*, pages 228–242, 2013.

[GM15] Matt Gardner and Tom M. Mitchell. Efficient and Expressive Knowledge Base Completion Using Subgraph Feature Extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP'15, pages 1488–1498, 2015.

[Goo07] Michael F. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69:221–221, 2007.

[GPLC18] Milan Gritta, Mohammad Taher Pilehvar, Nut Limsopatham, and Nigel Collier. What's missing in geographical parsing? *Language Resources and Evaluation*, 52(2):603–623, 2018.

[GTK+19] Simon Gottschalk, Nicolas Tempelmeier, Günter Kniesel, Vasileios Iosifidis, Besnik Fetahu, and Elena Demidova. Simple-ML: Towards a Framework for Semantic Data Analytics Workflows. In *International Conference on Semantic Systems*, SEMANTiCS'19, pages 359–366, 2019.

[Gut84] Antonin Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching. In Beatrice Yormark, editor, *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, SIGMOD'84, pages 47–57, 1984.

[HB11] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. Morgan & Claypool, 2011.

[HBC+21] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. Knowledge Graphs. *ACM Computing Surveys*, 54(4):71:1–71:37, 2021.

[HK00] Arthur E. Hoerl and Robert W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 42(1):80–86, 2000.

[HLS18] Xu Han, Zhiyuan Liu, and Maosong Sun. Neural Knowledge Acquisition via Mutual Attention Between Knowledge Graph and Text. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, AAAI'18, pages 4832–4839, 2018.

[HMLS20] Liwei Huang, Yutao Ma, Yanbo Liu, and Arun Kumar Sangaiah. Multimodal Bayesian embedding for point-of-interest recommendation on location-based cyber-physical-social networks. *Future Generation Computer Systems*, 108:1119–1128, 2020.

[HPSE15] Stefan Heindorf, Martin Potthast, Benno Stein, and Gregor Engels. Towards Vandalism Detection in Knowledge Bases: Corpus Construction and Analysis. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'15, pages 831–834, 2015.

[HPSE16] Stefan Heindorf, Martin Potthast, Benno Stein, and Gregor Engels. Vandalism Detection in Wikidata. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, CIKM'16, pages 327–336, 2016.

[HR16] Stephan Huber and Christoph Rust. Calculate Travel Time and Distance with OpenStreetMap Data Using the Open Source Routing Machine (OSRM). *The Stata Journal*, 16(2):416–423, 2016.

[HSBW13] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[HSEP19] Stefan Heindorf, Yan Scholten, Gregor Engels, and Martin Potthast. Debiasing Vandalism Detection Models at Wikidata. In *Proceedings of The Web Conference*, WWW'19, pages 670–680, 2019.

[HSJ20] Timo Homburg, Steffen Staab, and Daniel Janke. GeoSPARQL+: Syntax, Semantics and System for Integrated Querying of Graph, Raster and Vector Data. In *Proceedings of the 19th International Semantic Web Conference*, ISWC'20, pages 258–275, 2020.

[HSW19] Daniel Herzog, Sherjeel Sikander, and Wolfgang Wörndl. Integrating Route Attractiveness Attributes into Tourist Trip Recommendations. In *Companion Proceedings of the Web Conference 2021*, WWW'2019, pages 96–101, 2019.

[HSZ20] Lei Han, Juanzhen Sun, and Wei Zhang. Convolutional Neural Network for Convective Storm Nowcasting Using 3-D Doppler Weather Radar Data. *IEEE Transactions on Geoscience and Remote Sensing*, 58:1487–1495, 2020.

[HW08] Mordechai Haklay and Patrick Weber. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008.

[HZY+15] Liang Hong, Yu Zheng, Duncan Yung, Jingbo Shang, and Lei Zou. Detecting urban black holes based on human mobility data. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'15, pages 35:1–35:10, 2015.

[Jas18] Remillard Jason. OpenStreetMap Changest classification, 2018. Accessed: May 1st 2021, https://github.com/jremillard/osm-changeset-classification.

[JFF16] Li Jin, Zhuonan Feng, and Ling Feng. A Context-aware Collaborative Filtering Approach for Urban Black Holes Detection. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, CIKM'16, pages 2137–2142, 2016.

[JGBM17] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Short Papers*, EACL'17, pages 427–431, 2017.

[JLB16] Walter M. Dunn Jr, Steven P. Latoski, and Elizabeth Bedsole. Planned Special Events: Checklists for Practitioners. Technical report, Dunn Engineering Associates,Federal Highway Administration, Washington, DC, USA, 2016.

[JNHQ20] Levente Juhász, Tessio Novack, Hartwig H. Hochmair, and Sen Qiao. Cartographic Vandalism in the Era of Location-Based Games - The Case of OpenStreetMap and Pokémon GO. *ISPRS International Journal of Geo-Information*, 9(4), 2020.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*, ICLR'15, 2015.

[KGG20] Sina Keller, Raoul Gabriel, and Johanna Guth. Machine Learning Framework for the Estimation of Average Speed in Rural Road Networks with OpenStreetMap Data. *ISPRS International Journal of Geo-Information*, 9(11):638, 2020.

[KGSS12] Mathias Konrath, Thomas Gottron, Steffen Staab, and Ansgar Scherp. SchemEX - Efficient construction of a data catalogue by stream-based indexing of linked data. *Journal of Web Semantics*, 16:52–58, 2012.

[KK03] Granino Arthur Korn and Theresa M. Korn. *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*. DOVER PUBN INC, 2003.

[KKL+19] Agustinus Kristiadi, Mohammad Asif Khan, Denis Lukovnikov, Jens Lehmann, and Asja Fischer. Incorporating Literals into Knowledge Graph Embeddings. In *Proceedings of the 18th International Semantic Web Conference*, ISWC'19, pages 347–363, 2019.

[KLS18] Kira Kempinska, Paul A. Longley, and John Shawe-Taylor. Interactional regions in cities: making sense of flows across networked systems. *International Journal of Geographical Information Science*, 32(7):1348–1367, 2018.

[KMK19] Nikolaos Karalis, Georgios M. Mandilaras, and Manolis Koubarakis. Extending the YAGO2 Knowledge Graph with Precise Geospatial Knowledge. In *Proceedings of the 18th International Semantic Web Conference*, ISWC'19, pages 181–197, 2019.

[KMN14] Simon Kwoczek, Sergio Di Martino, and Wolfgang Nejdl. Predicting and visualizing traffic congestion in the presence of planned special events. *Journal of Visual Language and Computing*, 25(6):973–980, 2014.

[KMN15] Simon Kwoczek, Sergio Di Martino, and Wolfgang Nejdl. Stuck Around the Stadium? An Approach to Identify Road Segments Affected by Planned Special Events. In *IEEE 18th International Conference on Intelligent Transportation Systems*, ITSC'15, pages 1255–1260, 2015.

[KNC08] Woon Kim, Suhasini Natarajan, and Gang-Len Chang. Empirical Analysis and Modeling of Freeway Incident Duration. In *11th International IEEE Conference on Intelligent Transportation Systems*, ITSC'08, pages 453–457. IEEE, 2008.

[KS17] Mayank Kejriwal and Pedro A. Szekely. Neural Embeddings for Populated Geonames Locations. In *Proceedingo of the 16th International Semantic Web Conference*, ISWC'2017, pages 139–146, 2017.

[KSS15] Srijan Kumar, Francesca Spezzano, and V. S. Subrahmanian. VEWS: A Wikipedia Vandal Early Warning System. In *Proceedings of the 21th International Conference on Knowledge Discovery and Data Mining*, SIGKDD'15, pages 607–616, 2015.

[KXS+16] Xiangjie Kong, Zhenzhen Xu, Guojiang Shen, Jinzhong Wang, Qiuyuan Yang, and Benshi Zhang. Urban traffic congestion estimation and prediction based on floating car trajectory data. *Future Generation Computer Systems*, 61:97–107, 2016.

[KZ00] Stephen Kokoska and Daniel Zwillinger. *CRC Standard Probability and Statistics Tables and Formulae*. CRC Press, 2000.

[LC10] Ni Lao and William W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67, 2010.

[LGL17] Gang Liu, Peichao Gao, and Yongshu Li. Transport Capacity Limit of Urban Street Networks. *Transactions in GIS*, 21(3):575–590, 2017.

[LIJ+15] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[LJZ17] Yuxuan Liang, Zhongyuan Jiang, and Yu Zheng. Inferring Traffic Cascading Patterns. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'17, pages 2:1–2:10, 2017.

[LLL21] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. STAN: Spatio-Temporal Attention Network for Next Location Recommendation. In *Proceedings of The Web Conference 2021*, WWW '21, pages 2177–2185, 2021.

[LLS+15] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2181–2187, 2015.

[LLWL18] Zhidan Liu, Zhenjiang Li, Kaishun Wu, and Mo Li. Urban Traffic Prediction from Mobility Data Using Deep Learning. *IEEE Network*, 32(4):40–46, 2018.

[LTH+14] Freddy Lécué, Simone Tallevi-Diotallevi, Jer Hayes, Robert Tucker, Veli Bicer, Marco Luca Sbodio, and Pierpaolo Tommasi. STAR-CITY: semantic traffic analytics and reasoning for CITY. In *19th International Conference on Intelligent User Interfaces*, IUI'14, pages 179–188, 2014.

[LWFZ18] Ming Li, René Westerholt, Hongchao Fan, and Alexander Zipf. Assessing spatiotemporal predictability of LBSN: a case study of three Foursquare datasets. *GeoInformatica*, 22(3):541–561, 2018.

[LWS+19] Adam Lerer, Ledell Wu, Jiajun Shen, Timothée Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. Pytorch-BigGraph: A Large Scale Graph Embedding System. In *Proceedings of Machine Learning and Systems 2019*, MLSys'19, 2019.

[LWSG13] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. Understanding variable importances in forests of randomized trees. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 431–439, 2013.

[LWSY18] Wei Liu, Jing Wang, Arun Kumar Sangaiah, and Jian Yin. Dynamic metric embedding model for point-of-interest prediction. *Future Generation Computer Systems*, 83:183–192, 2018.

[LXZ+18] Zhongjian Lv, Jiajie Xu, Kai Zheng, Hongzhi Yin, Pengpeng Zhao, and Xiaofang Zhou. LC-RNN: A Deep Learning Model for Traffic Speed Prediction. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, IJCAI'18, pages 3470–3476, 2018.

[MC12] Peter Mooney and Padraig Corcoran. Characteristics of Heavily Edited Objects in OpenStreetMap. *Future Internet*, 4(1):285–305, 2012.

[MG12] Mahalia Miller and Chetan Gupta. Mining traffic incidents to forecast impact. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing*, UrbComp@KDD'12, pages 33–40, 2012.

[Mik15] Peter Mika. On Schema.org and Why It Matters for the Web. *IEEE Internet Computing*, 19(4):52–55, 2015.

[MJC+20] Gengchen Mai, Krzysztof Janowicz, Ling Cai, Rui Zhu, Blake Regalia, Bo Yan, Meilin Shi, and Ni Lao. SE-KGE : A location-aware Knowledge Graph Embedding model for Geographic Question Answering and Spatial Semantic Lifting. *Transactions in GIS*, 24(3):623–655, 2020.

[MJYP20] Tong Meng, Xuyang Jing, Zheng Yan, and Witold Pedrycz. A survey on machine learning for data fusion. *Information Fusion*, 57:115–129, 2020.

[MMM04] Frank Manola, Eric Miller, and Brian McBride. RDF Primer, 2004. Accessed: July 30th 2021, https://www.w3.org/TR/rdf-primer/.

[MPB14] Robert Meusel, Petar Petrovski, and Christian Bizer. The WebDataCommons Microdata, RDFa and Microformat Dataset Series. In *Proceedings of the 13th International Semantic Web Conference*, ISWC'14, pages 277–292, 2014.

[MRP16] Robert Meusel, Dominique Ritze, and Heiko Paulheim. Towards More Accurate Statistical Profiling of Deployed schema.org Microdata. *ACM Journal of Data and Information Quality*, 8(1):3:1–3:31, 2016.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[MSC+13] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 3111–3119, 2013.

[MYS+17] Chuishi Meng, Xiuwen Yi, Lu Su, Jing Gao, and Yu Zheng. City-wide Traffic Volume Inference with Loop Detector Data and Taxi Trajectories. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'17, pages 1:1–1:10, 2017.

[MYWW15] Xiaolei Ma, Haiyang Yu, Yunpeng Wang, and Yinhai Wang. Large-Scale Transportation Network Congestion Evolution Prediction Using Deep Learning Theory. *PLOS ONE*, 10(3):1–17, 2015.

[MZWL18] Chen Ma, Yingxue Zhang, Qinglong Wang, and Xue Liu. Point-of-Interest Recommendation: Exploiting Self-Attentive Autoencoders with Neighbor-Aware Influence. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM'18, pages 697–706, 2018.

[NA11] Axel-Cyrille Ngonga Ngomo and Sören Auer. LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, IJCAI'11, pages 2312–2317, 2011.

[NGJ+19] Natalya Fridman Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. Industry-scale Knowledge Graphs: Lessons and Challenges. *Communications of the ACM*, 62(8):36–43, 2019.

[NGZ12] Pascal Neis, Marcus Goetz, and Alexander Zipf. Towards Automatic Vandalism Detection in OpenStreetMap. *ISPRS International Journal of Geo-Information*, 1(3):315–332, 2012.

[NHG17] Ming Ni, Qing He, and Jing Gao. Forecasting the Subway Passenger Flow Under Event Occurrences With Social Media. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1623–1632, 2017.

[NHNR17] Markus Nentwig, Michael Hartung, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm. A survey of Current Link Discovery Frameworks. *Semantic Web*, 8(3):419–436, 2017.

[NLC17] Hoang Nguyen, Wei Liu, and Fang Chen. Discovering Congestion Propagation Patterns in Spatio-Temporal Traffic Data. *IEEE Transactions on Big Data*, 3(2):169–180, 2017.

[Ope21a] OpenStreetMap Contributors. History of OpenStreetMap in the OpenStreetMap Wiki, 2021. Accessed: July 20th 2021, https://wiki.openstreetmap.org/wiki/History_of_OpenStreetMap.

[Ope21b] OpenStreetMap Contributors. OpenStreetMap Wiki, 2021. Accessed: July 20th 2021, https://wiki.openstreetmap.org/wiki/Main_Page.

[PAS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery*, KDD'20, pages 701–710, 2014.

[Pau17] Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8:489–508, 2017.

[PB13] Heiko Paulheim and Christian Bizer. Type Inference on Noisy RDF Data. In *Proceedings of the 12th International Semantic Web Conference*, volume 8218 of *ISWC'13*, pages 510–525, 2013.

[PB14] Heiko Paulheim and Christian Bizer. Improving the Quality of Linked Data Using Statistical Distributions. *International Journal on Semantic Web and Information Systems*, 10(2):63–86, 2014.

[PCJ+18] Ross S. Purves, Paul D. Clough, Christopher B. Jones, Mark M. Hall, and Vanessa Murdock. Geographic Information Retrieval: Progress and Challenges in Spatial Search of Text. *Foundations and Trends in Information Retrieval*, 12(2-3):164–318, 2018.

[PDGS15] Bei Pan, Ugur Demiryurek, Chetan Gupta, and Cyrus Shahabi. Forecasting spatiotemporal impact of traffic incidents for next-generation navigation systems. *Knowledge and Information Systems*, 45(1):75–104, 2015.

[PDS12] Bei Pan, Ugur Demiryurek, and Cyrus Shahabi. Utilizing Real-World Transportation Data for Accurate Traffic Prediction. In *12th IEEE International Conference on Data Mining*, ICDM'12, pages 595–604, 2012.

[Per29] Clarence Arthur Perry. The Neighborhood Unit: from the Regional Survey of New York and its Environs. *National Municipal Review*, 18(10):636–637, 1929.

[PNI+18] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT'18, 2018.

[PRPB15] Francisco C. Pereira, Filipe Rodrigues, Evgheni Polisciuc, and Moshe E. Ben-Akiva. Why so many people? Explaining Nonhabitual Transport Overcrowding With Internet Data. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1370–1379, 2015.

[PS17] Nicholas G. Polson and Vadim O. Sokolov. Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, 79:1 – 17, 2017.

[PSB+18] Dharmen Punjani, Kuldeep Singh, Andreas Both, Manolis Koubarakis, Iosif Angelidis, Konstantina Bereta, Themis Beris, Dimitris Bilidas, Theofilos Ioannidis, Nikolaos Karalis, Christoph Lange, Despina-Athanasia Pantazi, Christos Papaloukas, and George Stamoulis. Template-Based Question Answering over Linked Geospatial Data. In *Proceedings of the 12th Workshop on Geographic Information Retrieval*, GIR@SIGSPATIAL'2018, pages 7:1–7:10, 2018.

[PSG08] Martin Potthast, Benno Stein, and Robert Gerling. Automatic Vandalism Detection in Wikipedia. In *Proceedings of the 30th European Conference on IR Research*, ECIR'08, pages 663–668, 2008.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP'14, pages 1532–1543, 2014.

[QB19] Sterling Quinn and Floyd Bull. Understanding Threats to Crowdsourced Geographic Data Quality Through a Study of OpenStreetMap Contributor Bans. In *Geospatial Information System Use in Public Organizations*. Routledge, 2019.

[RBRP17] Filipe Rodrigues, Stanislav Borysov, Bernardete Ribeiro, and Francisco C. Pereira. A Bayesian Additive Model for Understanding Public Transport Usage in Special Events. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2113–2126, 2017.

[RP20] Federica Rollo and Laura Po. Crime Event Localization and Deduplication. In *Proceedings of the 19th International Semantic Web Conference*, ISWC'20, pages 361–377, 2020.

[RR12] Mohan Rao and K Ramachandra Rao. Measuring Urban Traffic Congestion – A Review. *International Journal for Traffic and Transport Engineering*, 2(4):286–305, 2012.

[RT10] Kurt A. Raaflaub and Richard J. A. Talbert. *Geography and Ethnography: Perceptions of the World in Pre-Modern Societies*. Wiley-Blackwell;, 2010.

[SCCC20] Mario Scrocca, Marco Comerio, Alessio Carenini, and Irene Celino. Turning Transport Data to Comply with EU Standards While Enabling a Multimodal Transport Knowledge Graph. In *Proceedings of the 19th International Semantic Web Conference*, ISWC'20, pages 411–429, 2020.

[SCMN13] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'13, pages 926–934, 2013.

[SDSN17] Mohamed Ahmed Sherif, Kevin Dreßler, Panayiotis Smeros, and Axel-Cyrille Ngonga Ngomo. Radon - Rapid Discovery of Topological Relations. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 175–181, 2017.

[SFFN17] Tzanina Saveta, Giorgos Flouris, Irini Fundulaki, and Axel-Cyrille Ngonga Ngomo. Benchmarking Link Discovery Systems for Geo-Spatial Data. In *Joint Proceedings of BLINK2017: 2nd International Workshop on Benchmarking Linked Data and NLIWoD3: Natural Language Interfaces for the Web of Data co-located with 16th International Semantic Web Conference (ISWC 2017)*, 2017.

[SGY+16] Pracheta Sahoo, Ujwal Gadiraju, Ran Yu, Sriparna Saha, and Stefan Dietze. Analysing Structured Scholarly Data Embedded in Web Pages. In *Semantics, Analytics, Visualization. Enhancing Scholarly Data*, pages 90–100, 2016.

[Sin12] Amit Singhal. Introducing the Knowledge Graph: things, not strings. Official Google Blog, 2012. Accessed: May 1st 2021, https://googleblog.blogspot.de/2012/05/introducing-knowledge-graph-things-not.html.

[SK16] Panayiotis Smeros and Manolis Koubarakis. Discovering Spatial and Temporal Links among RDF Data. In *Proceedings of the Workshop on*

Linked Data on the Web, LDOW 2016, co-located with 25th International World Wide Web Conference (WWW 2016), volume 1593, 2016.

[SKD+20] Basel Shbita, Craig A. Knoblock, Weiwei Duan, Yao-Yi Chiang, Johannes H. Uhl, and Stefan Leyk. Building Linked Spatio-Temporal Data from Vectorized Historical Maps. In *Proceedings of the 17th European Semantic Web Conference*, volume 12123 of *ESWC'20*, pages 409–426, 2020.

[SKK16] Ridha Soua, Arief Koesdwiady, and Fakhri Karray. Big-data-generated traffic flow prediction using deep learning and dempster-shafer theory. In *2016 International Joint Conference on Neural Networks*, IJCNN'16, pages 3195–3202, 2016.

[SKW07] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th International World Wide Web Conference*, WWW'07, pages 697–706, 2007.

[SLHA12] Claus Stadler, Jens Lehmann, Konrad Höffner, and Sören Auer. LinkedGeoData: A core for a web of spatial open data. *Semantic Web*, 3(4):333–354, 2012.

[SNL17] Mohamed Ahmed Sherif, Axel-Cyrille Ngonga Ngomo, and Jens Lehmann. Wombat - A Generalization Approach for Automatic Link Discovery. In *The Semantic Web - 14th International Conference*, ESWC 2017, pages 103–119, 2017.

[SRM+14] Guus Schreiber, Yves Raimond, Frank Manola, Eric Miller, and Brian McBride. RDF 1.1 Primer, 2014. Accessed: July 30th 2021, https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/.

[STD21] Ashutosh Sao, Nicolas Tempelmeier, and Elena Demidova. Deep Information Fusion for Electric Vehicle Charging Station Occupancy Forecasting. In *IEEE 24th International Conference on Intelligent Transportation Systems*, ITSC'21, 6 pages, 2021.

[SVA+17] Michael Schultz, Janek Voss, Michael Auer, Sarah Carter, and Alexander Zipf. Open land cover from OpenStreetMap and remote sensing. *International Journal of Applied Earth Observation and Geoinformation*, 63:206–213, 2017.

[SWH15] Wei Shen, Jianyong Wang, and Jiawei Han. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2015.

[TD16] Davide Taibi and Stefan Dietze. Towards Embedded Markup of Learning Resources on the Web: An Initial Quantitative Analysis of LRMI Terms Usage. In *Proceedings of the 25th International World Wide Web Conference*, WWW'16, 2016.

[TD21a] Nicolas Tempelmeier and Elena Demidova. Linking OpenStreetMap with Knowledge Graphs - Link Discovery for Schema-Agnostic Volunteered Geographic Information. *Future Generation Computer Systems*, 116:349–364, 2021.

[TD21b] Nicolas Tempelmeier and Elena Demidova. OVID: A Machine Learning Approach for Automated Vandalism Detection in OpenStreetMap. In *Proceedings of the 29th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'21, 4 pages, 2021.

[TDD18] Nicolas Tempelmeier, Elena Demidova, and Stefan Dietze. Inferring Missing Categorical Information in Noisy and Sparse Web Markup. In *Proceedings of the Web Conference*, WWW'18, pages 1297–1306, 2018.

[TDD20] Nicolas Tempelmeier, Stefan Dietze, and Elena Demidova. Crosstown traffic - supervised prediction of impact of planned special events on urban traffic. *GeoInformatica*, 24(2):339–370, 2020.

[TDEP18] Jean-Noel Thepaut, Dick Dee, Richard Engelen, and Bernard Pinty. The Copernicus Programme and its Climate Change Service. In *2018 IEEE International Geoscience and Remote Sensing Symposium*, IGARSS'18, pages 1591–1593. IEEE, 2018.

[TFWD19] Nicolas Tempelmeier, Udo Feuerhake, Oskar Wage, and Elena Demidova. ST-Discovery: Data-Driven Discovery of Structural Dependencies in Urban Road Networks. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'19, pages 488–491, 2019.

[TFWD21] Nicolas Tempelmeier, Udo Feuerhake, Oskar Wage, and Elena Demidova. Mining Topological Dependencies of Recurrent Congestion in Road Networks. *ISPRS International Journal of Geo-Information*, 10(4):248, 2021.

[TGD21] Nicolas Tempelmeier, Simon Gottschalk, and Elena Demidova. GeoVectors: a Linked Open Corpus of OpenStreetMap Embeddings on World Scale. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, CIKM'21, 9 pages, 2021.

[TKS20] Kota Tsubouchi, Hayato Kobayashi, and Toru Shimizu. POI Atmosphere Categorization Using Web Search Session Behavior. In *Proceedings of the 28th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'20, pages 630–639, 2020.

[TR15] Guillaume Touya and Andreas Reimer. Inferring the Scale of OpenStreetMap Features. In *OpenStreetMap in GIScience - Experiences, Research, and Applications*, pages 81–99. 2015.

[TRL+19] Nicolas Tempelmeier, Yannick Rietz, Iryna V. Lishchuk, Tina Kruegel, Olaf Mumm, Vanessa Miriam Carlow, Stefan Dietze, and Elena Demidova. Data4UrbanMobility: Towards Holistic Data Analytics for Mobility Applications in Urban Regions. In *Companion of The 2019 World Wide Web Conference*, WWW'19, pages 137–145, 2019.

[TSF+20] Nicolas Tempelmeier, Anzumana Sander, Udo Feuerhake, Martin Löhdefink, and Elena Demidova. TA-Dash: An Interactive Dashboard for Spatial-Temporal Traffic Analytics. In *Proceedings of the 28th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'20, pages 409–412, 2020.

[TTdR20] Quy Thy Truong, Guillaume Touya, and Cyril de Runz. OSMWatchman: Learning How to Detect Vandalized Contributions in OSM Using a Random Forest Classifier. *ISPRS International Journal of Geo-Information*, 9(9):504, 2020.

[VBGK09] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. Silk - A Link Discovery Framework for the Web of Data. In *Proceedings of the WWW2009 Workshop on Linked Data on the Web*, volume 538 of *LDOW'09*, 2009.

[VK14] Denny Vrandecic and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[VKG14] Eleni I. Vlahogianni, Matthew G. Karlaftis, and John C. Golias. Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43:3 – 19, 2014.

[VMSTF21] John E. Vargas-Munoz, Shivangi Srivastava, Devis Tuia, and Alexandre X. Falcão. OpenStreetMap: Challenges and Opportunities in Machine Learning and Remote Sensing. *IEEE Geoscience and Remote Sensing Magazine*, 9(1):184–199, 2021.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention

is All you Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 5998–6008, 2017.

[WCY20] Senzhang Wang, Jiannong Cao, and Philip Yu. Deep Learning for Spatio-Temporal Data Mining: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 14:1–20, 2020.

[wgs14] World Geodetic System 1984. Its Definition and Relationships with Local Geodetic Systems. Technical report, National Geospatial-Intelligence Agency, U.S. Department of Defense, 2014.

[Win99] William E. Winkler. The State of Record Linkage and Current Research Problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, 1999.

[WL17] Hongjian Wang and Zhenhui Li. Region Representation Learning via Mobility Flow. In *Proceedings of the 26th ACM International Conference on Information and Knowledge Management*, CIKM'17, pages 237–246, 2017.

[WLFY21] Meng-xiang Wang, Wang-Chien Lee, Tao-Yang Fu, and Ge Yu. On Representation Learning for Road Networks. *ACM Transactions on Intelligent Systems and Technology*, 12(1):11:1–11:27, 2021.

[WLL+16] Quan Wang, Jing Liu, Yuanfei Luo, Bin Wang, and Chin-Yew Lin. Knowledge Base Completion via Coupled Path Ranking. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL'16, 2016.

[WMWG17] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[Wor99] World Wide Web Consortium. Resource Description Framework (RDF) Model and Syntax Specification W3C Recommendation, 1999. Accessed: July 30th 2021, https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.

[Wor08] World Wide Web Consortium. RDFa in XHTML: Syntax and Processing, 2008. Accessed: July 30th 2021, https://www.w3.org/TR/rdfa-syntax/.

[Wor14] World Wide Web Consortium. RDF 1.1 N-Quads, 2014. Accessed: July 30th 2021, https://www.w3.org/TR/n-quads/.

[Wor20] World Wide Web Consortium. JSON-LD 1.1 A JSON-based Serialization for Linked Data W3C Recommendation, 2020. Accessed: July 30th 2021, https://www.w3.org/TR/json-ld11/.

[WPC+16] Xiaomeng Wang, Ling Peng, Tianhe Chi, Mengzhu Li, Xiaojing Yao, and Jing Shao. A Hidden Markov Model for Urban-Scale Traffic Estimation Using Floating Car Data. *PLOS ONE*, 10(12):1–20, 2016.

[WWG15] Quan Wang, Bin Wang, and Li Guo. Knowledge Base Completion Using Embeddings and Rules. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI'15, pages 1859–1866, 2015.

[WWL16] Fei Wu, Hongjian Wang, and Zhenhui Li. Interpreting traffic dynamics using ubiquitous urban data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'16, pages 69:1–69:4, 2016.

[WYSG17] Minjie Wang, Su Yang, Yi Sun, and Jun Gao. Human mobility prediction from region functions with taxi trajectories. *PLOS ONE*, 12(11):1–23, 2017.

[WZJ20] Shirui Wang, Wen'an Zhou, and Chao Jiang. A survey of word embeddings based on deep learning. *Computing*, 102(3):717–740, 2020.

[WZX14] Yilun Wang, Yu Zheng, and Yexiang Xue. Travel time estimation of a path using sparse trajectories. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'14, pages 25–34, 2014.

[XYW+16] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. Learning Graph-based POI Embedding for Location-based Recommendation. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, CIKM'16, pages 15–24, 2016.

[XZX+19] Xuejing Xie, Yi Zhou, Yongyang Xu, Yunbing Hu, and Chunling Wu. OpenStreetMap Data Quality Assessment via Deep Learning and Remote Sensing Imagery. *IEEE Access*, 7:176884–176895, 2019.

[YFGD16] Ran Yu, Besnik Fetahu, Ujwal Gadiraju, and Stefan Dietze. A Survey on Challenges in Web Markup Data for Entity Retrieval. In *Proceedings of the ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference*, (ISWC'16), 2016.

[YG18] Can Yang and Gyözö Gidófalvi. Fast map matching, an algorithm integrating hidden Markov model with precomputation. *International Journal of Geographical Information Science*, 32:547–570, 2018.

[YGF+19] Ran Yu, Ujwal Gadiraju, Besnik Fetahu, Oliver Lehmberg, Dominique Ritze, and Stefan Dietze. KnowMore - knowledge base augmentation with structured web markup. *Semantic Web*, 10(1):159–180, 2019.

[YGFD17] Ran Yu, Ujwal Gadiraju, Besnik Fetahu, and Stefan Dietze. FuseM: Query-Centric Data Fusion on Structured Web Markup. In *33rd IEEE International Conference on Data Engineering*, ICDE'17, pages 179–182, 2017.

[YGZ+16] Ran Yu, Ujwal Gadiraju, Xiaofei Zhu, Besnik Fetahu, and Stefan Dietze. Towards Entity Summarisation on Structured Web Markup. In *The Semantic Web - Satellite Events*, ESWC'16, pages 69–73, 2016.

[YHMH19] Carl Yang, Do Huy Hoang, Tomás Mikolov, and Jiawei Han. Place Deduplication with Embeddings. In *Proceedings of The Web Conference*, WWW'2019,, pages 3420–3426, 2019.

[ZB20] Shuo Zhang and Krisztian Balog. Web Table Extraction, Retrieval, and Augmentation: A Survey. *ACM Transactions on Intelligent Systems and Technology*, 11(2):13:1–13:35, 2020.

[ZCM+20] Qi Zhang, Jianlong Chang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Spatio-Temporal Graph Structure Learning for Traffic Forecasting. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'20, pages 1177–1185, 2020.