Thèse n° 7633

Dynamic pattern recognition in large-scale graphs with applications to social networks

Présentée le 21 décembre 2020

à la Faculté des sciences et techniques de l'ingénieur
Laboratoire de traitement des signaux 2
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Volodymyr MIZ

Acceptée sur proposition du jury

Prof. J.-Ph. Thiran, président du jury
Prof. P. Vandergheynst, directeur de thèse
Dr M. Redi, rapporteuse
Prof. M. Luczak-Roesch, rapporteur
Prof. R. West, rapporteur

École
polytechnique
fédérale
de Lausanne

2020

To my parents…

Acknowledgements

I would like to express my gratitude to a number of people who supported me during my PhD studies. This thesis would not have been possible without you all.

First, I would like to thank my supervisor Pierre Vandergheynst. He has been very supportive and inspiring during my PhD journey, shaping my research agenda according to my (not always conventional and practical) scientific interests. This required a lot of creativity and patience from his side. I appreciate the freedom he gave me, which allowed me to grow and to develop an independent research mindset. It has been a great pleasure to be his student over the past four years.

I would also like to thank Markus Luczak-Roesch, Miriam Redi, Jean-Philippe Thiran, and Robert West for the time they invested as jury members during my PhD defense. Their invaluable advice helped me greatly to enhance this thesis and to broaden my understanding of the field. Despite their generous input in this work, I take full responsibility for all errors and misconceptions remaining in this thesis.

I gratefully acknowledge the financial support from the European Union's H2020 Framework Programme (H2020-MSCAITN-2014), also known as MacSeNet and SpaRTaN innovating training networks. In addition to funding a substantial part of my research, the program has also provided a very supportive environment, where I have met a lot of great people.

During my PhD studies, Benjamin Ricaud and Nicolas Aspert have been great mentors. I feel very privileged to have been working with them in one team. Benjamin has been there with me from day one. His encouraging, supportive, and positive attitude helped me to push through the toughest moments of my PhD life and to get important things done. I would like to thank him for involving me in various exciting research projects that allowed me to broaden my expertise. Nicolas helped me to grow in many other ways. His professional perseverance showed me how to bring reproducible research to the next level and that perfection has no limits. I learned many technical skills and best practices from him while collaborating on challenging applied research problems. I am grateful to Nicolas and Benjamin for creating a friendly atmosphere and a healthy collaborative environment for me.

I would like to acknowledge Mário Figueiredo for inspiring my interest in the theoretical side of the research during my internship in his lab at IST Lisbon. I am very grateful to him for being creative in choosing the right project for me to work on. It kept me motivated and interested during my stay in his lab and beyond. His optimism and enthusiasm kept me going not only during the internship but also long after the end of it.

I would like to thank all people I collaborated with in Firmenich D-lab for teaching me how to be innovative in industry. During our collaboration, I was fortunate to work with colleagues who taught me lessons I could not have learned in academia.

I would like to thank my colleagues Andreas, Helena, Konstantinos, Kirell, Michaël, Rodrigo, and Youngjoo for inspiring feedback and discussions on numerous occasions during my stay in the LTS2.

I feel very fortunate to have had Axel, Joëlle, Ayman, and Etienne as my students. It was a great pleasure working with them and helping them with their Master's projects. These collaborations allowed me to learn a lot and to start looking at research problems from a different angle.

Special thanks go to my friends who have been by my side during all my ups and downs. Thanks for winning and losing basketball games with me, organizing spontaneous trips to amusement parks, pulling through freezing nights in a tent, teaching me headstands, crossing rivers in Iceland, cycling, hiking, and skiing in the breathtaking Swiss mountains. You all have made this journey vivid and colorful.

Lastly, and most importantly, I am deeply grateful to my parents for always believing in me and being there for me no matter what. I dedicate this thesis to you.

Lausanne, 3 December 2020

Abstract

A graph is a versatile data structure facilitating representation of interactions among objects in various complex systems. Very often these objects have attributes whose measurements change over time, reflecting the dynamics of the system. This general data framework can be used in many fields to represent complex data structures: brain networks and neuronal spikes, web networks and clickstreams, social networks and activity of the users, among others. In all of these examples, the structural and dynamic components of the data are inseparable, which significantly complicates the detection, analysis, and interpretation of patterns that emerge in the networks. The increasing size and complexity of graph-structured data require scalable and interpretable algorithms for dynamic pattern detection in such systems.

In this dissertation, we present an unsupervised approach for dynamic pattern detection in large-scale graphs. In this approach, we combine intuitions derived from attention mechanisms, Hopfield networks, and memory networks to build scalable, efficient, and interpretable algorithms. We then demonstrate multiple applications of this approach in recommendation systems, information recovery algorithms, and collective behavior studies. Additionally, we use our algorithm to detect dynamic activity patterns in social and communication networks. We conduct extensive experiments on Wikipedia data, detecting and analyzing patterns in the viewership activity in its web network. To study the collective behavior of Wikipedia readers, we develop an automated pattern interpretation model, which allows for comparison of trending topics across multiple language editions of Wikipedia. The results of the experiments reveal provocative insights into how people interact and search for information in online social networking environments, opening new avenues for future research on collective behavior analysis at a large scale.

Finally, we present a distributed data processing framework for Wikipedia server logs that allows others to reproduce all pattern detection experiments presented in this thesis and to conduct similar collective behavior studies on the latest data.

Key words: Graph, Network, Pattern Detection, Dynamic Network, Spatio-temporal Pattern, Graph-structured Data, Time-series, Memory, Attention, Neural Network, Hopfield Network, Social Network, Wikipedia, Collective Behavior

Résumé

Les graphes sont des structures de données polyvalentes permettant de représenter des interactions entre des objets dans des systèmes complexes. Souvent, ces objets possèdent des attributs qui varient au cours du temps, reflétant la dynamique du système. Ce cadre général peut servir à représenter des systèmes complexes dans de nombreux domaines : réseaux cérébraux et activité neuronale, réseaux de pages web et flux de clics sur les hyperliens, ou encore réseaux sociaux et activités des utilisateurs. Dans tous ces exemples, la structure et la dynamique des données ne sont pas séparables, ce qui complique considérablement la détection, l'analyse et l'interprétation des activités qui émergent de ces réseaux. Il est nécessaire d'avoir des algorithmes évolutifs et interprétables pour détecter ces activités dans des réseaux dont la taille et la complexité croissent.

Dans cette thèse, nous présentons une approche non supervisée pour la détection de motifs dynamiques dans des graphes de grande taille. Dans notre approche, nous combinons des intuitions dérivées des mécanismes d'attention, des réseaux de Hopfield et des réseaux à mémoire pour construire un algorithme évolutif, efficace et interprétable. Nous proposons ensuite de multiples applications de notre approche dans les systèmes de recommandation, les algorithmes de récupération d'informations et les études de comportement collectif. Nous utilisons notre algorithme pour détecter les motifs d'activité dynamiques dans les réseaux sociaux et de communication. Nous menons des expériences approfondies sur les données Wikipédia, détectant et analysant les tendances de l'activité d'audience sur son réseau Web. Pour étudier le comportement collectif des lecteurs de Wikipédia, nous développons un modèle automatisé d'interprétation permettant de comparer des tendances d'activité pour plusieurs éditions linguistiques de Wikipédia. Les résultats de nos expériences révèlent des informations intéressantes sur la manière dont les gens interagissent et recherchent des informations dans les environnements de réseaux sociaux en ligne, ouvrant de nouvelles voies pour de futures recherches sur l'analyse du comportement collectif à grande échelle.

Enfin, nous présentons les outils spécifiques de traitement distribué pour les données de Wikipédia. Ceux-ci permettent de reproduire toutes les expériences de détection présentées dans cette thèse et de mener des études de comportement collectif similaires sur les données les plus récentes.

Mots clefs : Graphe, Réseau, Détection de Motifs, Réseau Dynamique, Modèle Spatio-temporel, Données Structurées par Graphes, Séries Temporelles, Mémoire, Attention, Réseau neuronal, Réseau de Neurones d'Hopfield, Réseau social, Wikipédia, Comportement collectif

Contents

Curriculum Vitae

List of Figures

List of Tables

1 Introduction

It is now evident that where one discipline ends and the other begins no longer matters, for it is in the nature of the case that the boundaries are ill-defined.
— Patricia Smith Churchland [1]

1.1 Motivation

The amount of available graph-structured data has dramatically increased in the past few decades. Researchers working in different fields now have access to biological, social, transportation, and information networks, among others. However, the full potential of graph-structured data remains undiscovered because of the large quantity and high complexity of the data [2–4].

In the traditional data analytic framework, it is assumed that all measurements are independent. This assumption allows for the powerful machinery of statistical analysis to be applied to a wide range of research questions. However, theories that incorporate the structural aspect of the data argue that measurements are not always independent. The fundamental difference between the traditional and graph-based frameworks is the inclusion of information on *relationships* among measurements in a study. This perspective introduces a different range of constraints on data analysis and model building. Therefore, when working with graph-structured data, it is necessary to consider the influence of the underlying structure of the data on the performance of machine learning and data mining models.

Most of the theoretical tools for graph-structured data analysis originally come from graph theory [5]. Graph theory is a branch of mathematics that provides an elegant framework allowing us to formalize, describe, and study complex data structures. Its principal object is a *graph*, where *nodes* represent data items and *edges* encode pairwise relationships between them. Due to such universal and intuitive definition, complex systems in various fields can be represented as a graph[I]. Any graph can be represented as a matrix (the most widely used are adjacency and Laplacian matrices). This aspect allows using a general mathematical formalism to solve various problems of a diverse nature, which has influenced a wide adoption of graph-based techniques in physical, social, and life sciences.

Since the first applications of graph theory to network analysis, traditional approaches have been very successful in advancing our understanding of different processes in networks. Nevertheless, the increasing complexity and scale of the graph-structured data require more advanced methods in order for us to discover hidden patterns and develop a better understanding of the complex nature of real-world networks [6]. The emerging field of machine learning on graphs has introduced approaches that have proven to be efficient in solving this problem for various types of networks [7].

In recent years, many scientific fields have been transformed by the adoption of graph-based machine learning and data mining techniques. Scientists working in different research areas realized that the underlying structure of the data plays an essential role in the analysis of complex systems. However, hardly any other field has benefited from studying structural properties of the data as much as sociology and, in particular, social network analysis.

Researchers started studying structural aspects of social networks long before they developed into what we call a *social network* today [8]. In 1853, developing sociology as a science, Auguste Comte described society as a set of interconnections among social actors [9]. He defined two main facets of the field, *statics* and *dynamics*. According to Comte, statics reflect "laws of social interconnections," while dynamics describe "the laws of action and reaction of the different parts of the social system."

Early collective behavior studies were largely influenced by Gustave LeBon [10]. In 1897, he analyzed the phenomenon of crowd behavior and suggested that when individuals join crowds, they imitate the behavior of other members and lose their personalities. He compared the diffusion of ideas in a crowd to a process of contagion. Another fundamental idea of patterns in human interactions was introduced by Georg Simmel in 1908. According to Simmel, "Society exists where a number of individuals enter into interaction" [11].

These seminal contributions introduced principle concepts that anticipated the emergence of social network analysis as a field. By contrast, quantitative research was infeasible because the early works lacked mathematical formalism. This changed in 1934 when Moreno and Jennings presented the notion sociometry [12]. The central object of sociometry is a sociogram – a graph-based representation of social relations between people [13]. Later, to give sociograms a more objective and formal representation, Forsyth and Katz proposed to use matrices to study social networks, bringing the elegance of mathematical data analysis to the field [14].

[I] A graph is an abstract data structure that represents various real-world networks. Even though *graph* is a more abstract while *network* is a more concrete concept, the terms *graph* and *network* will be used interchangeably in this thesis.

Throughout the second half of the twentieth century, researchers used these foundations to create quantitative analytical approaches and models connecting mathematical social network analysis with graph theory [15–21]. Graph theory introduced potent machinery to social network analysis, providing an appropriate representation of the data and introducing a set of clear definitions of social concepts that allowed researchers to formalize important properties of social networks.

Since the creation of the first online social media services in 1995[II], the size of analyzed social networks has significantly increased. The availability of large-scale datasets with digital traces of billions of people led to the emergence of computational social science that, contrary to traditional quantitative social science that typically assumes independence of observations, focuses on the combination of spatial data, social networks, online content, and human interactions [22].

The fast growth of online social media platforms introduced new challenges to the analysis of the data generated by these platforms. First, the size and complexity of the data made it more difficult to build scalable and explainable models from the data mining perspective [23, 24]. Second, the data coming from online platforms may be incomplete and biased, which may lead to erroneous conclusions of sociological experiments [25]. As a result, comprehensive collective behavior analysis has also become more challenging.

These challenges raise interesting research questions both in data mining and in computational social science. Some of these questions were formulated at the very beginning of the history of social network analysis. Others developed as a result of the effects caused by the rapid expansion of social networking platforms. Is it possible to model the spread of ideas in networks similar to contagion processes, as Gustav LeBon suggested? What are the patterns of information propagation through the network? How can we detect patterns of misinformation spread? What kind of biases are created by social media? What are the collective interests of users based on their dynamic activity patterns? Looking at these problems from the data mining perspective, we can summarize them in one general question: *How can dynamic patterns be reliably detected and interpreted in graph-structured data?*

To address these problems and answer these questions we need new scalable and interpretable data mining approaches in order to analyze patterns that arise in the large-scale dynamics of graph-structured data. To do that, in this thesis we

- Develop a new spatio-temporal pattern detection algorithm for large-scale graph-structured datasets

- Use the algorithm to gain key insights into the dynamics of social networks and to interpret patterns of collective behavior in online networking environments

[II] First online social networking services – The Globe and Classmates – were created in 1995.

We present an unsupervised algorithm for dynamic pattern detection in large-scale graphs. In this approach, we combine intuitions derived from attention mechanisms, Hopfield networks, and memory networks to build a scalable, efficient, and interpretable algorithm. Briefly, the algorithm processes a graph structure with dynamic node attributes and outputs a reduced subgraph with detected patterns.

Interpretability is an essential, though largely ignored aspect of machine learning and data mining algorithms. We design an interpretable pattern detection approach and demonstrate its performance in multiple scenarios. We show how the interpretation of the detected patterns transforms them into collective behavioral insights that unravel patterns of online interactions and shed the light on cultural peculiarities of internet users.

To study collective behavior using the proposed approach, we use server logs of social networks. Social media platforms collect various data logs that represent the activity of billions of users on the internet. These logs constitute the *dynamic* component of the data. Depending on the nature of a particular platform, this data can reflect various aspects of user activity, such as evaluations of third-party content, hashtags, clicked links, watched videos, sent emails, etc. This data can be used to analyze different patterns of user activity, to identify their interests, to provide personalized recommendations, and to segment users based on their preferences.

Apart from these data logs, we also have access to the underlying structure of the data. For instance, it can be represented as a network of users or a network of webpages that users click on. This graph structure represents the *static* or structural component of the data. As we learned from earlier studies in social network analysis, people influence each other when they interact, so we use this information to improve the quality of pattern detection and to better interpret the detected patterns. Therefore, we shall focus on the combination of *structural* and *dynamic* components.

Most of the platforms keep their data private, which complicates the analysis. However, there are a few small subsets of data that were made available for research purposes. One of the most widely used datasets is the Enron email communication network. We use this dataset to compare the performance of our algorithm to previous works.

Due to the scarcity of the available data, pattern detection approaches are usually tested on relatively small and outdated datasets. To tackle this issue, we go further in our evaluation and develop a new large-scale graph dataset with dynamic attributes and use it to demonstrate the scalability and efficiency of our approach.

To create our dataset we use Wikipedia, the world's most visited online encyclopedia and collaborative knowledge sharing platform, where some server logs were made available for research purposes. Wikipedia data represents an interesting test case for our pattern detection approach. Along with viewership statistics and underlying web network structure, we have access to rich semantic information about Wikipedia articles, which allows for a detailed interpretation of the detected patterns. Though other studies also researched the collective behavior of Wikipedia readers [26–30], they mostly focused on preselected subsets of Wikipedia articles. Contrary to the previous works, we develop a scalable approach and run our experiments on the entire web network of Wikipedia. Furthermore, we present a distributed framework allowing other researchers to reproduce all the experiments presented in this thesis and to apply the proposed algorithm to the latest data.

In this thesis, we introduce a new approach for dynamic pattern detection in graph-structured data. We apply it to multiple datasets of different scale and detect dynamic patterns of user activity. Since very few datasets come with labeled data, we employ an unsupervised approach for pattern detection. Then, we focus more closely on Wikipedia and use its web network and viewership statistics to detect and interpret dynamic patterns that reflect the collective interests of Wikipedia readers.

1.2 Thesis structure and contributions

The leading theme of this thesis is dynamic pattern detection in graph-structured data. We propose an algorithm allowing us to detect such patterns at a large scale and describe multiple applications where our approach can be used. Among other applications, we extensively explore dynamic patterns in the collective behavior of Wikipedia readers, focusing on the evolution of collective interests and the impact of real-world events on the dynamics of trending topics.

The overall structure of this thesis is illustrated in Fig. 1.1. Chapter 3 is the core of this thesis, where we propose a new algorithm for dynamic pattern detection in large-scale graphs. The following chapters cover the detailed analysis of the proposed algorithm, its evaluation, and its applications. In Chapter 4, we analyze our approach, focusing on the interpretability of the detected patterns, scalability of the proposed algorithm, and potential applications. Next, in Chapter 5, we extend the proposed approach with an automated pattern labeling module and use the detected patterns to study the collective behavior of Wikipedia readers across multiple language editions. Finally, in Chapter 6, we focus on the reproducibility of all the experiments presented in this thesis and introduce a large-scale data processing framework for Wikipedia data allowing to study the dynamics and evolution of its web network.

A more detailed summary of the contributions is presented in Sections 1.2.1-1.2.4.

1.2.1 Dynamic pattern detection in large-scale graphs (Chapter 3)

The main results presented in this chapter were published in [31] (The Web Conference 2019).

Chapter 3 introduces a new algorithm for spatio-temporal pattern detection in the dynamics of graph-structured data. We use this algorithm for all experiments presented in this thesis.

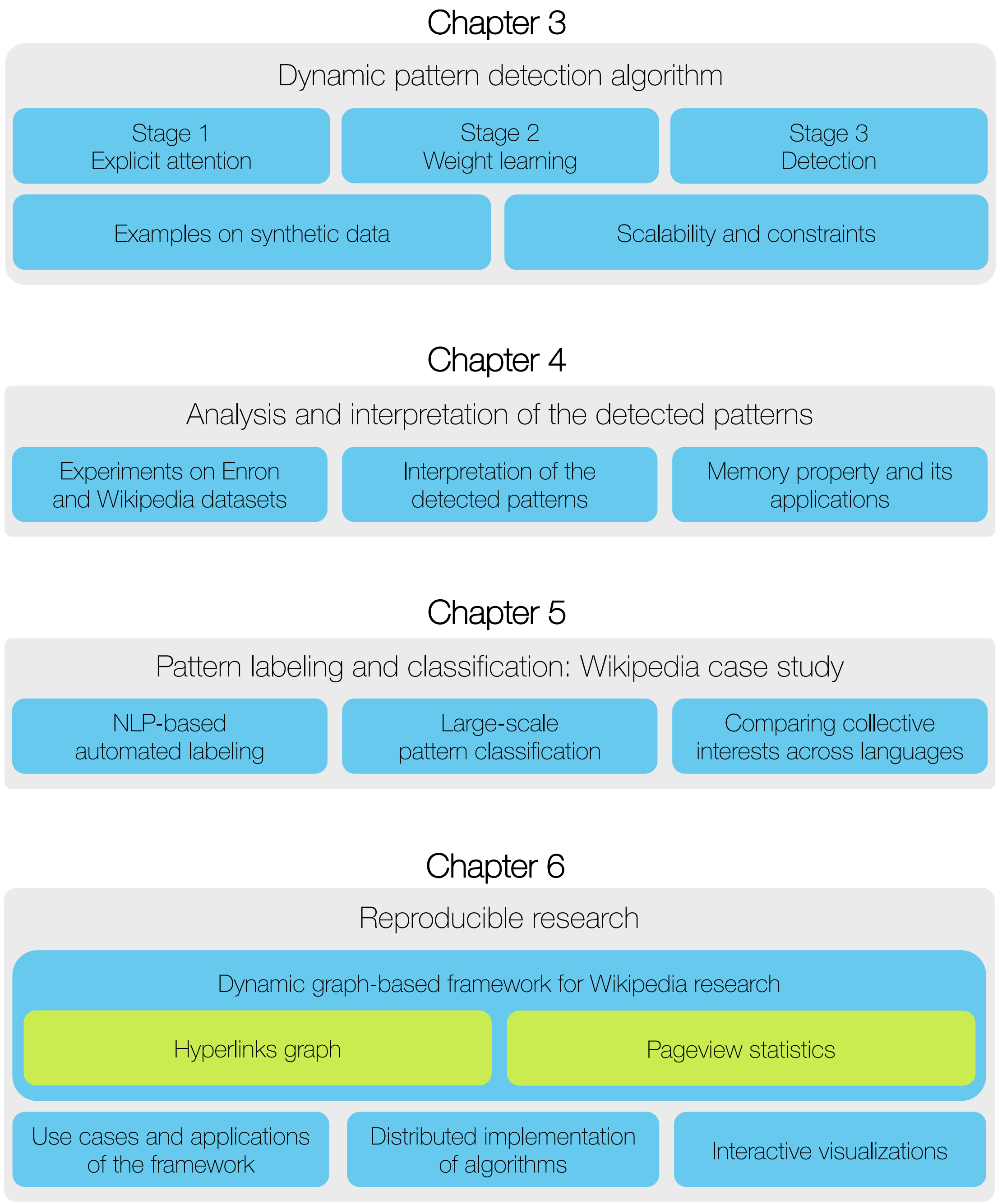

Chapter 3
Dynamic pattern detection algorithm
Stage 1
Explicit attention
Stage 2
Weight learning
Stage 3
Detection
Examples on synthetic data
Scalability and constraints
Chapter 4
Analysis and interpretation of the detected patterns
Experiments on Enron and Wikipedia datasets
Interpretation of the detected patterns
Memory property and its applications
Chapter 5
Pattern labeling and classification: Wikipedia case study
NLP-based automated labeling
Large-scale pattern classification
Comparing collective interests across languages
Chapter 6
Reproducible research
Dynamic graph-based framework for Wikipedia research
Hyperlinks graph
Pageview statistics
Use cases and applications of the framework
Distributed implementation of algorithms
Interactive visualizations

Figure 1.1 – Outline of the thesis. In Chapter 3, we focus on the proposed algorithm. In Chapter 4, we present the detailed analysis of detected patterns and describe potential applications. Chapter 5 focuses on the interpretation of the patterns. We use Wikipedia server logs as a case study to understand the dynamic patterns in the collective behavior of Wikipedia readers. Finally, Chapter 6 describes our efforts towards reproducible research, including a large-scale framework for Wikipedia data processing and interactive visualizations.

In this chapter, we describe the general pattern detection framework and formalize our model. We run experiments on synthetic datasets to provide a better understanding of its features.

In our approach, we demonstrate the intrinsic memory properties of graph-structured data that enable us to use our algorithm in multiple applications. We formalize the connection between attention mechanisms and associative memory models using the Hopfield network model. Additionally, we implement the aggregation process in Hopfield networks similarly to attention mechanisms that are used in graph neural networks. To learn the Hopfield network, we use the Hebbian learning rule for feature aggregation, which enables us to learn localized patterns.

Hebbian learning rule has multiple benefits. First, it is unsupervised, which allows us to perform learning on the dataset without labeled patterns. Second, since we know that the edge weight between nodes depends only on the one-hop neighborhood of these nodes, the rule allows for an intuitive interpretation of the results. Third, due to the locality of computations, we can implement learning using an efficient message-passing approach. Finally, Hebbian rule is easy to adapt to different types of patterns by using application-specific similarity functions.

In addition to pattern detection, we discuss applications of the proposed algorithm in recommendation systems. Dynamic recommendation systems that target data domains with underlying graph structure can benefit from the memory properties of the detected patterns. We show that the task of recommendation is very similar to the problem of incomplete pattern recovery and use the recall mechanism of Hopfield networks to complete recommendation profiles.

1.2.2 Analysis and evaluation of the detected patterns (Chapter 4)

The main results presented in this chapter were published in [31] (The Web Conference 2019) and in [32] (presented at Wikimania 2019)[III].

In this chapter, we proceed with a more detailed analysis of our approach. We focus on the attention mechanism and show how it affects the main properties of the processed graph, including degree distribution, modularity, and clustering coefficient. We also show how the attention mechanism can effectively reduce the amount of processed data, while preserving the desirable quality of pattern detection.

We evaluate our approach using the Enron email dataset, a classic benchmark for spatio-temporal pattern detection algorithms, and compare our detection performance to the results reported in other works. We show that our algorithm detects all the anomalies presented in state-of-the-art works.

[III] https://wikimania.wikimedia.org/wiki/2019:Research/Wikipedia_graph_mining_dynamic_structure_of_collective_memory

Then, we continue the analysis of our approach and scale it to the entire Wikipedia web network (>7M pages and >500M links), comparing the performance of pattern detection to Google Trends. We demonstrate the scalability of our algorithm and run large-scale dynamic pattern detection experiments on the Wikipedia web network. To represent the dynamics of the data, we use server logs that contain information about viewership statistics of each article.

Contrary to the Enron dataset, ground truth labels are not available in Wikipedia data. However, in the experiment on the Wikipedia data, we verify the performance of our approach using an alternative source of information that allows for a thorough qualitative analysis of the results.

Wikipedia dataset has a rich set of semantic attributes that carry additional information about the network, such as article titles and their categories. We observe that viewership patterns detected in Wikipedia activity correspond to clusters of densely connected articles on similar topics that correlate with real-world events. This observation allows us to verify the performance of our approach and interpret the detected patterns using Google Trends – a service that tracks trending topics based on the most popular queries in the search engine.

Finally, we describe two applications beyond pattern detection. We show that the detected patterns possess memory properties, similar to the ones observed in Hopfield networks. The memory is represented as an adjacency matrix of a weighted graph that serves as a content addressable memory system, which can be used to recover learned or "remembered" patterns from incomplete inputs. We provide an example where these properties can be used in recommendation systems as well as information recovery applications. Furthermore, we demonstrate an application related to sociology and study the phenomenon of collective memory – the way social groups remember the past. We interpret patterns as real-world events and investigate which associations with past events they trigger.

1.2.3 Pattern labeling and classification (Chapter 5)

The main results presented in this chapter were published in [33] (The Web Conference 2020).

In this chapter, we extend our pattern detection algorithm with an automated interpretation module that uses node attributes to generate a summary of detected patterns. Such a summary can be used by an expert for further analysis of the results. In the applications, we focus on textual attributes and develop a model that identifies topics of the detected patterns based on these attributes. We focus on a large-scale application of our approach and run experiments on Wikipedia data. We use textual attributes of Wikipedia articles to extract topics and train a classification model to assign labels to the detected patterns. Then, we apply the developed approach to study the collective behavior of Wikipedia readers and research the evolution of trending topics across multiple language editions.

To assess our automated pattern interpretation module, we conduct a case study focusing on the first months of the COVID-19 pandemic. We study the evolution of interests of Wikipedia readers and align them with the unfolding of the pandemic. We analyze changing trends across seven languages, including English, French, Russian, Spanish, German, Chinese, and Italian, highlighting the ways in which the global lockdown affected the interests of Wikipedia readers during the pandemic.

Overall, our findings on collective behavior indicate four main reasons for differences in readers' interests across languages:

Media coverage. The majority of the patterns we have detected are triggered by real-world events, which means that readers' interests are mainly driven by media coverage of these events in different languages.

Geographic proximity. Some patterns appear only in one language edition. That is especially apparent when we consider natural disasters. Such events are most interesting to local (w.r.t. disaster) Wikipedia readers. These patterns emerge only in locally spoken languages.

Cultural differences. Despite globalization, the geography of a spoken language also affects the cultural interests of Wikipedia readers. The preferences of readers related to sports, music, art, literature, movies, and other categories of interest that define culture, vary across different regions of the world. We found that such cultural interests of the readers affect trends in every language edition of Wikipedia.

Pandemic effect. Being a rather special case, the COVID-19 pandemic affected readers' interests globally. It influenced the readers to shift their focus from sports to topics related to healthcare and more suitable forms of home entertainment that conform to social distancing measures.

1.2.4 Towards reproducible research (Chapter 6)

The main results presented in this chapter were published in [34] (The Web Conference 2019).

In this chapter, we focus on the reproducibility of the results presented in this thesis. We start with a general overview of the reproducibility in computer science and engineering, highlighting its importance and necessity.

Then, we describe our efforts towards the reproducibility of the results presented in this thesis. We introduce a large-scale data processing framework for Wikipedia server logs that we used in the majority of the experiments. We focus on two aspects of the spatio-temporal dataset – *space*, represented as a graph of Wikipedia articles, and *time*, represented by the number of views per article per hour.

We also provide multiple use cases where our Wikipedia data processing framework can be useful. Aside from spatio-temporal datasets that we use to evaluate our dynamic pattern detection approach, we can use the framework to generate datasets for GNN benchmarks, to build knowledge graphs, and to select small subsets of Wikipedia that focus on selected topics.

We conclude this chapter with a brief description of our dissemination efforts and present several interactive visualization tools. These visualizations allow lay audiences to engage with the results of our research. We also highlight the importance of nonacademic forms of science communication, which helps researchers to reach more people by making the findings accessible for a general audience.

2 Related work and terminology

The main theme of this thesis is large-scale spatio-temporal pattern detection in graph-structured data. Along with being related to the previous works on the main topic, the proposed approach has diverse connections to the research on attention mechanisms, graph neural networks (GNNs), and associative memory models.

In this chapter, we put our work into the context of prior research on these topics. To start with, we review existing approaches for dynamic pattern detection in graph-structured data. Then, we describe applications and evaluation strategies that are used to assess performance of pattern detection algorithms. After that, we overview attention mechanisms in GNNs, discussing their applications in graph data mining and pattern detection. Lastly, we describe a link between associative memory models and attention mechanisms in GNNs, connecting these topics to pattern detection in dynamic graphs, which brings us back to the main theme of this thesis.

2.1 Dynamic pattern detection in networks

The increasing availability of graph-structured data sparked interest among data mining researchers, leading to the development of graph mining algorithms [35]. Initially, researchers mainly focused on static graphs and only in the past few years, growing volumes of spatio-temporal data influenced new developments in the field. In particular, anomaly detection in dynamic graphs gained popularity a short time ago [36–38]. A more recent survey of the emerging field of spatio-temporal data mining emphasized the importance of specialized data mining techniques for the dynamic networks domain [39]. The authors highlighted the inevitable emergence of new complex applications that inherently deal with dynamic graph-structured data. They provided an overview of the related research covering six major problems, such as visualization, clustering, predictive learning, frequent pattern mining, anomaly detection, change detection, and relationship mining.

Since the field of spatio-temporal data mining emerged only recently, terminology and notation vary from one work to another. There were a few comprehensive attempts to unify the field in multiple surveys, providing a thorough categorization of dynamic pattern detection approaches. We will use these taxonomies to put our work into the context of the field.

Aggarwal and Subbian [36] categorized all dynamic graph-based algorithms into two broad categories. First, *slowly evolving networks*, where significant changes in the graph structure or in its attributes occur over the long-term periods of weeks or months, and second, *streaming networks*, where the data is continuously updated and comes in the form of graph streams. The slowly evolving networks category mainly covers snapshot-based approaches that compare graph snapshots across different time steps. This category includes link prediction, community detection, and tensor factorization methods. Streaming networks imply a continuous graph data stream arriving at the input of the model for further processing. This category focuses on dynamic clustering and dense pattern mining. According to this categorization, our approach belongs to the *streaming networks* category since we process continuous streams of time-series graph attributes.

Akoglu et al. [37] proposed a similar categorization, however, they focused on large-scale algorithms for anomaly detection. The authors also organized graph-based anomaly detection methods into two big categories, *static* and *dynamic*, distinguishing approaches that work with attributed and non-attributed graphs. Conceptually, these categories are similar to the ones that we discussed in the previous paragraph. Also, the authors highlighted the importance of interpretable methods in a separate category and distinguished several application-specific approaches. We attribute our approach to the *dynamic* category and also focus on the interpretability of the detected patterns.

Another taxonomy of dynamic anomaly detection algorithms was proposed by Ranshous et al. [38]. They identified five general categories of anomaly detection problems in temporal networks, namely, (1) anomalous nodes, (2) anomalous edges, (3) anomalous subgraphs, (4) event detection, and (5) change detection. Same as in [37, Def. 4], the category *event detection* in [38, Type 4] covers the case where all nodes of a subgraph contribute to the creation of an event at the same time. In our work, we focus on a similar problem, where given a graph G, the task is to detect a pattern in the attributes that causes a significant structural transformation in the graph.

These surveys cover a substantial body of research and propose convenient taxonomies of pattern detection methods that allow us to position our work in the context of prior art. In addition to that, we would like to mention another area of research that does not appear in the aforementioned surveys despite being closely related to the work presented in this thesis. Pattern detection in dynamic networks has also been extensively studied in the area of *temporal network mining*, where researchers focus on collective dynamics, synchronization, and self-organisation phenomena in complex networks [40, Ch. 5], [41–43]. Kovanen et al. studied *temporal motifs* that define frequently occurring contact patterns in dynamic networks. In addition to the topological structure of the motifs, they also incorporated the temporal order of similar events occurring in node attributes [44]. Mirtello et al. introduced a measure of *dynamical strength of social ties* and proposed a variety of metrics that take into account both temporal and structural components of human interactions in mobile networks to detect communication patterns [45]. Mitra, Tabourier, and Roth defined patterns as dynamic network communities that emerge as a result of interactions among a set of nodes over time [46]. Inspired by these works, Pfitzner et al. introduced a notion of *betweenness preference* in time-aggregated networks that highlighted a spatio-temporal dimension of dynamic networks and its influence on the dynamical processes evolving in temporal networks [47]. Later, Weng et al. demonstrated a connection between memory and betweenness preference, constructing networks from temporal data observations [48]. Several works developed approaches for network inference from co-occurrence observations capturing *Markovian* [49] and *non-Markovian* [50, 51] characteristics of temporal networks.

The formalism we propose in this thesis shares some features with these temporal network mining approaches, such as the combination of temporal and structural information about the data, burstiness of the dynamic attributes, and co-occurrence patterns. However, we only rely on the given network structure and do not attempt to infer connections between nodes.

Many modern applications require detecting patterns in *data streams*. One of the major problems of algorithms for spatio-temporal data streams is their high computational complexity. To cope with this issue, a few works treated the spatial and temporal components of the data independently [52–54]. Nonetheless, a few recent works focused on developing scalable and parallelized approaches for large-scale anomaly detection in temporal networks. Gao et al. [55] followed the reductionist approach and proposed a sampling-based method. Similar to our algorithm, their model first reduces the amount of data by sampling only task-relevant nodes, and then embeds every pattern into a separate community. Another effective solution is to use a parallel message-passing approach in the implementation. Chaudhry et al. [56] created a distributed framework, FlowGraph, which detects anomalies in spatio-temporal data streams using a distributed message-passing approach. In our algorithm, we also adopt a similar strategy to scale and parallelize computations.

In this thesis, we propose a *graph-based dynamic pattern detection algorithm* that contributes to the collection of spatio-temporal approaches in graph mining. As we can see in the literature, patterns in dynamic networks are often called different names causing confusion. They are referred to as events, drifts, outliers, change points, dissimilarities, anomalies, or dynamic concepts. A common feature is a causal effect that results in a change of the network structure or its attributes, which represents a spatio-temporal pattern. In further, to avoid confusion, here and in the following chapters, we will refer to all the aforementioned types of spatio-temporal dynamic transformations as *patterns*.

2.2 Applications and evaluation of pattern detection algorithms

Enron email dataset. Email exchange pattern detection. In our evaluation, we use *Enron email dataset* [57]. The main reason why we chose the Enron email dataset is that it is often used as a benchmark for pattern detection in dynamic graphs. We use Enron dataset for evaluation of our algorithm and demonstrate its performance using the detected anomalies presented in other works as ground truth.

Multiple studies used the dataset to detect various events and anomalies in communication patterns [58–63]. Two features are common among these works. First, in all studies, the authors created an email communication graph, connecting employees who exchanged emails. Second, it is common to create time slices of the dynamic graphs and to compute the proposed measures for each time slice. Nonetheless, we can distinguish two different families of anomaly detection approaches that were applied to the Enron email dataset – structure-based and feature-based.

Structure-based approaches measure the connectivity of the graph and focus on significant changes in the structure of the network. Priebe et al. [59] proposed a structure-based anomaly detection approach that uses a moving window to calculate statistics over time-sliced graphs. As a parameter defining an anomaly, the authors used unusually high local connectivity of the graph in a k-hop neighborhood. To detect spikes of the connectivity-based measure, they used running mean and standard deviation as a threshold for each time window. A similar structure-based approach was proposed by Wang et al. [58]. The authors performed a rigorous theoretical analysis of various locality-based measures proposed in [59, 64, 65], providing insights into the behavior of the model in cases when the time series of graphs are stationary before the anomaly. Moriano et al. [62] also used structural properties of the graphs, however, in contrast to the previous works, they focused on community-based patterns. First, they defined an initial community partition. Then, they made an assumption that anomalies occur when the number of cross-community communications increases.

Feature-based approaches focus on node attributes and track feature dissimilarities over time. Wan et al. [61] developed a hybrid anomaly detection approach by tracking two types of deviations of node features derived from the communication behavior of users (e.g., in and out degrees, number of sent/received/replied emails, etc.). Individual deviation compares the current feature vector of a node at time t to its historical mean, while cluster deviation compares the same vector to the historical median of the features of all nodes in the cluster. Then, those nodes whose deviations exceed predefined thresholds of normal behavior are considered anomalous. Koutra et al. [60] proposed another approach based on the feature similarity between connected nodes. They compute pairwise affinity scores using a random-walk-based measure and compare consecutive time snapshots of the graph using a variant of Euclidean distance. Then, the similarities are merged into a series and anomalies are detected when the similarity values exceed a median-based threshold. Rayana and Akoglu [63] proposed an ensemble of multiple anomaly detection approaches with different scoring functions to extract anomalies in graph snapshots. Each detector in the ensemble uses graph-based features of the nodes (e.g. degree) over time to detect events in multivariate time series. Once the detection is done by each detector in the ensemble, the algorithm performs score- and rank-based aggregation to find a consensus among all detectors and to verify the accuracy of the detection.

Wikipedia graph dataset. User activity pattern detection. As a part of our evaluation, we also apply the developed pattern detection approach to Wikipedia data. We detect viewership patterns of the readers that allow us to study their collective interests across multiple languages.

Early studies on the interests of Wikipedia readers [66, 67] used page view counts to identify the most popular articles. They found that entertainment (movies, music, sports, etc.) and people's biographies are the most popular topics among Wikipedia readers. The analysis of the results of the experiments, described in Section 4.2, confirms this general tendency. Moreover, we can draw previously unseen relationships between events that appear in the news and the popularity of clusters of connected Wikipedia articles.

Another topic of recent investigations is the motivation of Wikipedia readers [68]. The authors reported that the fact that a topic was referenced in the media (30%), in a conversation (22%), or it is a current event (13%) is among the main motivations for reading about a particular topic. This indicates a strong influence of trends and news on the readers' consumption of articles. Trends are an essential part of the search for information and that is what we want to extract and analyse. Trends are highly influenced by the readers' environment and, therefore, they should reflect similarities and differences across languages and cultures. However, the previously cited works mainly focused on the English version of Wikipedia.

Concerning language biases and differences across versions of Wikipedia, there are two distinct groups of studies related to them. The first group studies editorial activity and page content, while the second focuses on the viewership and readers' behavior.

Editorial activity and content across languages. Editorial differences across languages have been investigated for particular topics such as medicine-related articles [69], aircraft crashes [70], or edit wars on popular pages [71]. All these works point out differences due to political and cultural influences associated with a particular language. Depending on the language, some parts within an article are more detailed, biased, or more debated. As a consequence of this editorial behavior, page contents in different languages contain noticeable variations when comparing articles about famous people[72], food-related pages [73], or history of states [74]. All that results in discrepancies in the textual content and in the hyperlink structure [75].

Viewership. In a large-scale poll, Lemmerich et al. [76] analyzed visitors' motivations across 14 languages. The authors demonstrated that Wikipedia viewership patterns and use cases vary in different language editions and connected their findings to the socio-economic characteristics in certain countries, such as the Human Development Index.

We focus on the automatic detection of viewership trends and biases in multiple language editions of Wikipedia. In our experiments, the topics are not predefined but extracted automatically according to their trend score. The topics are defined based on the summaries of Wikipedia articles involved in the detected spatio-temporal patterns. We also present the results based on the model with predefined topics trained on Wikidata properties. We illustrate the dynamical evolution of the most popular topics among the readers of English, French, and Russian language editions of Wikipedia and reveal differences and commonalities across them.

2.3 Attention in graph mining

Being effective at edge prediction tasks, Graph Convolutional Neural Networks (GCN) [77, 78] and their variants have recently gained popularity in anomalous edge detection. Since GCNs are not suitable for temporal pattern detection, it is generally extended with an attention-based Gated Recurrent Unit (GRU) to capture long- and short-term dynamic patterns. In [79], Zheng et al. created AddGraph, a semisupervised model capturing anomalous edges. To detect anomalous edges, they learn a joint representation for each node, combining the structural properties of a node with its attributes. Then, they use the representations of two neighboring nodes to compute the anomaly score based on short- and long-term patterns captured by the GRU. A model with a similar architecture, SrtGNN [80], allows detecting anomalous edges in more complex settings where the set of nodes is changing over time. Zhang et al. [81] solved change-point detection problem in multivariate time-series data, which can also be seen as anomalous edge detection problem. The architecture also contains GCN and GRU modules. They build a network based on correlations between time-series data points and use GCN to detect changes in the network structure. As a result, their model detects anomalous deletion and creation of edges between time-series, which is interpreted as change points. In our method, we use aggregation, message passing, and attention approaches to learn dynamic patterns in a similar way as it is done in graph neural networks.

Over the past few years, attention-based models demonstrated outstanding performance solving various machine learning problems, such as automated translation [82], object recognition [83, 84], and image captioning [85]. The abstract idea behind the attention mechanism is to focus on the specific parts of the signal to learn the model only on the task-related or de-noised signal and reduce the amount of processed data. Adding an attention layer to machine learning models has been shown to decrease the computational cost of training and to improve the interpretability of the results.

More recently, the success of the attention mechanism in computer vision and natural language processing influenced its adoption in models that work with graph-structured data [86]. In the pursuit of computational efficiency and scalability of graph neural networks, graph attention networks (GAT) were introduced [87, 88][I].

[I] Due to the conceptual similarity across the models, let us refer to all graph attention models as GAT

GATs extend graph convolutional networks (GCN) [77, 78] with an attention layer, which essentially computes weights between nodes in an input graph. To learn node embeddings, GCNs aggregate information by sending "messages" between the nodes in the graph. Typically, when we learn an embedding of a node, we compute a weighted sum of the attributes of its neighbours, which is normalized by their degrees. While GCNs treat all messages from all neighbors equally, attention weights in GATs prioritize these messages based on their relevance to solving the task at hand. Attention weights determine how much influence we want to give to a neighboring node when we propagate the information over an input graph structure. In other words, attention weights define the importance of neighbors of a given node based on their attributes.

Attention weights can be learned using either a trainable function with softmax [87] or a similarity-based approach [88], where the priority is given to neighbors with similar attributes. A similar approach was used in inductive graph representation learning [3], were the priority during aggregation was given to the nearby nodes (however, regardless of the similarity of the attributes).

GATs demonstrated several advantages compared to other graph-based models without the attention mechanism. Incorporating attention improves the predictive performance of the models by learning dynamic and adaptive representations of the neighborhood [88]. The computational cost of training has also decreased due to the ability of the attention-based models to focus only on the task-relevant parts of the input graphs avoiding their noisy parts [87, 89]. Lastly, in healthcare applications, where explainability of machine learning models is of utmost importance, GATs have been shown to make the predictions more interpretable [90].

The ability of GATs to benefit from the underlying graph data structure to improve the performance of machine learning algorithms inspired the use of GATs in multiple graph-based data mining applications, where the data is naturally structured as a graph. Multiple recommendation systems adopted the approach by modelling the context-depending preferences of the users using an attributed graph [91, 92]. The authors used graph-based attention to learn representations of the user and recommended item graphs and generated recommendations as a dot product between the learned representations. Xiao et al. [93] used GATs to model the behavior of individuals in content-production web-based environments. Based on the history of the users' activity and their social interactions, the authors modelled the content production strategies of authors in academic social networks. Another interesting application is emotion classification in collections of documents. In topic modelling [94, 95], the attention mechanism was used to learn topics and dependencies between words and documents. The context-based semantics of documents was computed as a dot product between the learned attention weights and the weighted adjacency matrix between words and documents. Finally, GATs were used to forecast demand in bike-sharing networks [96]. The authors represented bike stations as nodes and interstation commutes as edges and used attention to incorporate correlations in the demand on neighboring stations into the forecasting model.

In this thesis, we develop an unsupervised similarity-based attention module for pattern detection in dynamic graphs. This module is inspired by the attention mechanisms in graph neural networks, which we described in this section. We build our attention mechanism upon the existing works. It inherits similar beneficial characteristics of the existing models such as computational efficiency, scalability, and interpretability. Furthermore, we extend our attention mechanism with memory properties that can be used for various applications, including recommendation systems and information recovery. To benefit from the memory properties of our attention module, we use the intuition gained from memory networks [97, 98], another graph-based approach similar to GATs. We are going to discuss the connection between GATs and memory networks in the following section.

2.4 Relationship between memory and attention models

Another effective graph-based approach, which is similar to GATs with their attention mechanism, is memory networks [97, 98]. We can see a conceptual similarity if we interpret the neighborhood of a node as the associative memory. Similar to the graph attention mechanism, memory networks compute features of a node based on the attributes of the neighbors and store the updated features in the same position. Once the full memory representation is computed, the memory model uses either an *argmax* [97] or *softmax* [98] attention to map patterns and labels into space or *to memorise*. Finally, the memory model retrieves these patterns based on the score computed as a dot product between the memorized representation and the input sample.

Recently, Ramsauer et al. [99] formalized the relatedness between the attention mechanism and the memory-based models using a variation of Hopfield networks[100]. They showed that attention mechanism can be effectively replaced by a new type of Hopfield network, achieving superior performance on multiple-instance learning tasks [101].

The learning part of the proposed approach is based on a Hopfield network with the Hebbian update rule. In the proposed pattern detection algorithm, we use the discussed relatedness of memory networks and attention-based models for the interpretation of the detected patterns. Also, in Section 4.3, we show how the memory properties of our model can be used in various applications, including recommendation systems, information retrieval, and collective behavior analysis.

3 Dynamic pattern detection in large-scale graphs

Dynamic networks change over time. The rate at which the nodes and edges are added or removed defines the *spatial* dynamics. The situation complicates when nodes in the network have attributes that also change dynamically, adding *temporal* dynamics to the problem. This combination of spatial and temporal dynamics creates a new complex data domain, introducing new challenges and creating the need for new pattern detection techniques. There are numerous use cases where we need these algorithms. Web-pages (nodes) and hyperlinks (edges) connecting them form web networks that change over time. Every page has a click-stream or viewership activity (dynamic attribute). Sensor networks are composed of interconnected sensors producing time-series of measurements. Social networks, where friends are connected and each user generates a massive amount of data over time, such as likes, views, and messages. Those are just a few well-known examples. We can also find examples of spatio-temporal data in other fields, such as finance (transaction networks), medicine and neuroscience (brain networks), and transportation (road networks). In all those examples, networks are large and growing extremely fast.

Increasing complexity of the data requires scalable and interpretable algorithms for dynamic or spatio-temporal pattern detection in graph-structured data. In this chapter, we introduce our pattern detection algorithm that is inspired by several approaches, including attention mechanisms, associative memory networks, and graph neural networks. Due to the combination of multiple properties derived from these approaches, our method tackles the problem of spatio-temporal pattern detection from a different angle, providing an efficient solution that can be scaled to large-scale graphs with time-series attributes. Finally, we test our algorithm on a synthetic dataset, discussing its limitations and constraints.

3.1 Proposed pattern detection framework

In this section, we focus on the proposed approach. The main goal of the approach is to detect spatio-temporal patterns in graphs with dynamic (time-series) node attributes.

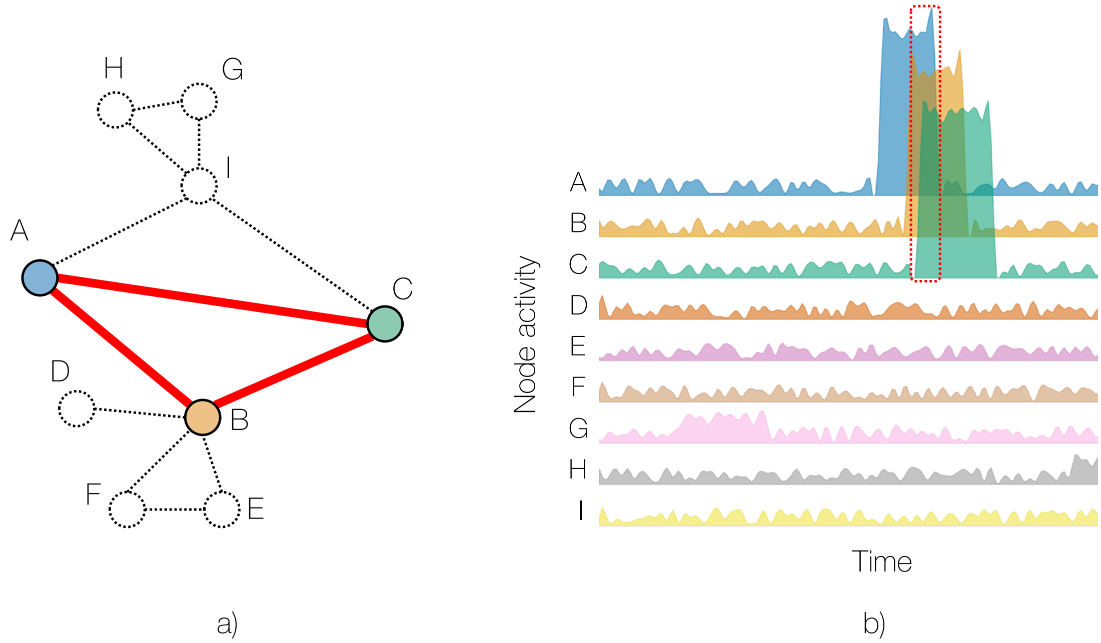


Figure 3.1 – Spatio-temporal data structure combining a graph topology and time series. a) Graph topology. Edges highlighted in red depict the spatial component of a spatio-temporal pattern. Dashed nodes have uniform activity. Colored nodes undergo a spike of the dynamic activity. b) Time-series signals residing on the vertices of the graph. Signals associated to nodes A, B, and C are correlated: a pattern propagates from node A to C through B. This is an abstract illustration of a dynamic pattern detected by our method.

Our method is unsupervised. There are two main concepts behind the proposed approach. First, it is inspired by an idea reminiscent of similarity-based attention graph neural networks [88]. We developed an attention mechanism that focuses on task-specific graph attributes to reduce the computational cost. Second, we implemented an unsupervised learning process inspired by the update rules used in Hopfield memory networks and the message-passing approach used in graph neural networks. To the best of our knowledge, these concepts were not used in the spatio-temporal pattern detection literature. We describe each step in more detail in Subsections 3.1.2 and 3.1.3. Now, let us focus on the overall aspects of the approach.

The general graph-based data domain can be described as follows. We are given a weighted graph G with a set of vertices V and a set of edges E, connecting the vertices. Each node in the network has a time-series attribute $x(t)$, which governs the dynamics of the network. Each edge between vertices v_i and v_j has a weight w_{ij} that changes over time. The edge weight update depends on the attributes of the nodes it connects and the function applied to those attributes. All in all, the developed pattern detection algorithm can be applied to the data that has a structure depicted in Fig. 3.1. It consists of an attributed graph with time-series attributes on the nodes.

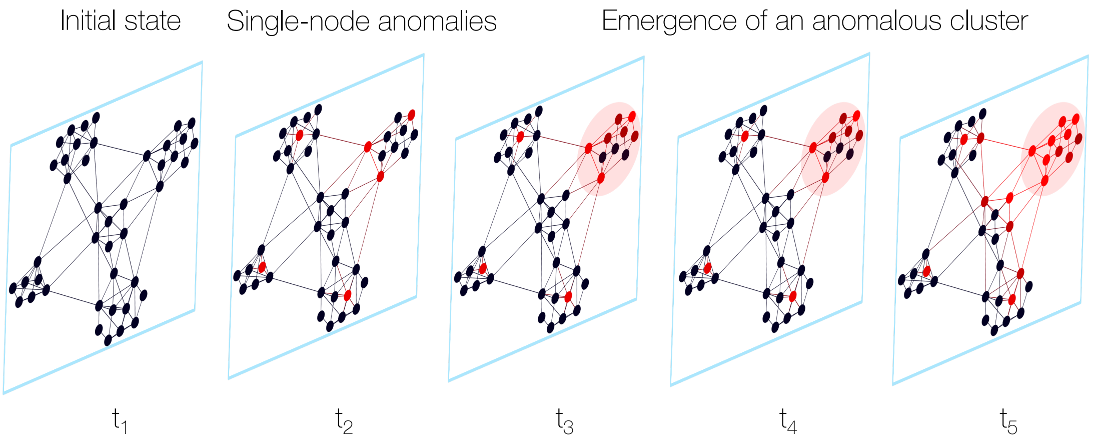


Figure 3.2 – Illustration of the emergence of a spatial pattern. The algorithm ignores single-node anomalies during t_1 and t_2. The detection of the anomalous cluster occurs only when multiple nodes in close neighborhood exhibit anomalous behavior in temporal domain, which happens between t_3 and t_5.

Compared to the definitions given in the pattern detection literature (see Section 2.1), our definition of a pattern is more general since we track a heterogeneous spatio-temporal pattern that can emerge in situations when patterns in nodes' attributes spread or propagate over the network, as it is illustrated in Fig. 3.2. As we can see, a spatio-temporal pattern can not be captured by a single sub-graph or a static snapshot.

Within our framework, a pattern is characterized by two components. First, *a graph pattern* that involves multiple nodes, possibly anomalous at different time steps. Second, *a temporal pattern*, a correlated change in time-series attributes of nodes in the same neighborhood.

Tracking these types of patterns has a number of applications in spatio-temporal data mining [39, 102, 103]. Our approach introduces the following novelties that distinguish it from the related works described in Section 2.1:

1) For each pattern detected using our method, the model produces a comprehensive set of spatio-temporal indicators that facilitate the interpretation of the patterns and reasons for their emergence. In that sense, it is closer to the spatio-temporal data mining [39], where the purpose is to extract the anomalous events and keep as much information about them as possible for the sake of interpretability. Describing detected patterns to domain experts is a powerful feature of our data mining process. We will illustrate that in more detail using real-world examples in Section 4.2.2. We will see how our approach can be used to provide insights into the collective behavior of users on the web and social networks. We will also demonstrate real-world examples of how the visitor activity evolves and propagates over the network.

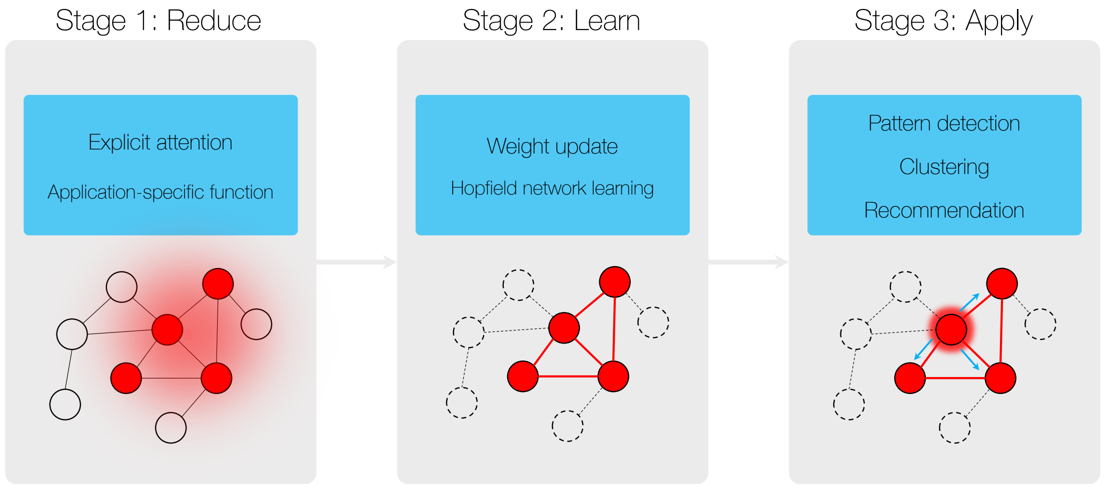


Figure 3.3 – General framework of the proposed pattern detection algorithm. Stage 1 selects nodes that are potentially related to an emerging pattern. The selection is done using an application-specific function applied to attributes of the nodes. Stage 2 is responsible for learning weights between selected nodes. The dashed nodes and edges do not take part in learning. The weight update function is also application-specific. Once Stages 1 and 2 are completed, the learned representation can be used for pattern detection, clustering, or recommendation.

2) We define the concept of *potential anomaly* that introduces a prior on the presence of a pattern, implementing an attention mechanism that enhances the scalability and interpretability of the method. Indeed, in many pattern detection applications, domain experts can separate the data into two parts: one part contains potential anomalies, while the other contains non-anomalous samples. Our concept rigorously defines the separation and allows discarding non-anomalous samples. This step significantly reduces the amount of data to process. The concept of *potential anomaly* is general and can be used for other methods and applications.

3.1.1 General framework

Pattern detection in such temporal networks generally contains two stages [38]. The first stage is usually responsible for pattern-related feature extraction from domain-specific data. The second stage applies the pattern detection algorithm to the extracted features and reports the detected patterns.

We build our method upon this scheme in the following way (see an illustration of the general framework in Fig. 3.3). First, we add to the Stage 1 an explicit attention step that keeps only potentially relevant signals and focuses attention of the model on those parts of the network that are most likely to have an anomaly. It reduces the amount of data that is processed in Stage 2 by discarding task-irrelevant nodes. Second, in Stage 2, our model contains an unsupervised learning step that computes dynamic associative attention weights. This step enables us to interpret the detected patterns providing their detailed spatio-temporal descriptions. Such descriptions represent a group of interconnected nodes (spatial information), where every node has a time-series attribute (temporal information) that indicates the time when the pattern occurred.

The learning step, Stage 2, is inspired by the model of a memory neural network, the Hopfield network with the Hebbian learning rule. In our model, we adapt the learning rule to fit our spatio-temporal data structure in the following way. Edge weight is reinforced when two neighbors have a correlated pattern of activity during the same time slice. This particular network design shares similarities with the hotspot anomaly detection for graph streams [104], however, the authors use it for feature engineering, while in our approach, this update is a part of learning.

The Hopfield network approach learns an associative memory network, where the nodes correspond to the ones of the initial graph. During the learning process, edges between the nodes are either strengthened or removed depending on the temporal behavior of each node. As a result of learning, nodes with similar behavior are connected by stronger links and clustered together in the memory network. These clusters contain groups of nodes with similar temporal features.

Finally, in Stage 3, we use the learned representation to solve the task at hand. It could be pattern interpretation, clustering, or recommendation. We will demonstrate multiple applications in Sections 4.3, 4.4, and 5.3.

Now, let us discuss each step in more detail in the following sections.

3.1.2 Stage 1: Explicit attention and potential anomalies

The goal of this stage is to focus attention of the model on the nodes that are most likely to be a part of an emerging spatio-temporal pattern and reduce the amount of data to be processed, while preserving task-related information. At this stage, for each node in the original graph, we extract features from the raw time-series attributes. We design an explicit attention mechanism and introduce a notion of *potential anomaly*. The mechanism aims to keep only those nodes that are most likely to be anomalous. Let V be the set of nodes of graph G and $x_i[t] \in \mathbb{R}$ be the value associated to node $v_i \in V$ at the time $t \in [T-1, 0]$. The historical time-series have a length of T samples.

Definition 1 (Single node potential anomaly)**.** *Given a score function $f_i : \mathbb{R} \to \mathbb{R}$ for each node v_i and a threshold value $c_0 \geq 0$, a node $v_i \in V$ is said to have a potential anomaly at time t when its time-series value is such that $|f_i(x_i[t])| > c_0$.*

Note that the potential anomaly is local on the graph, i.e., it depends only on the time-series attribute of a given node. If the score function is such that $f_i = f - \hat{f}(v_i)$, for some function f and $\hat{f}(v_i)$ a summary statistics of the scores $f(v_i[t])$, our definition corresponds to the definition of the anomaly in [38]. A basic example of a potential anomaly is the time-series values exceeding a fixed threshold c_0. In that case, f_i is the identity.

Applying the score function f_i to the time-series gives us 1) features reflecting a pattern and 2) an initial indication of anomalous behavior. In practice, removing nodes that do not have pattern-defining features reduces the amount of data by an order of magnitude without losing relevant information that is required for pattern detection.

The general definition of the potential anomaly allows for various attention score functions, such as a moving average or ARMA filter prior to the thresholding, a short-time Fourier transform, a wavelet transform, or a user-defined function. The choice of the scoring function should be based on the dataset and the prior knowledge on the nature of the time-series data to make an effective compromise between efficiency and scalability.

To extract potential anomalies in the attributes, we use a user-defined scoring function f_i^b, which is applied to time-series attributes x_i of all nodes in the graph. We select values that are above the activity rate parameter c_0^b. The *burstiness* b_i of a signal x_i of a node i is

$$b_i = \sum_{t=0}^{T-1} k_i[t], \qquad k_i[t] = \begin{cases} 1, & \text{if } |f_i^b(x_i[t])| > c_0^b, \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

Potential anomalies have to satisfy the following requirement. A potentially anomalous node must have a sufficient number of bursts b_i in their time-series attribute. Unless this requirement holds, we discard the node. The minimal number of bursts (potential anomalies) per node depends on the dataset and should be chosen accordingly. Empirically, high values lead to more aggressive thresholding. Hence, when dealing with sufficiently large graphs with a high level of noise, it is recommended to start with higher values and decrease it gradually until the desired result is achieved. Smaller graphs require lower thresholds. The time window for which the statistics is computed affects the accuracy of the detection. The larger the historical data, the more accurate the results of the anomaly detection algorithm.

To identify spikes k_i in the time-series attributes, we use an algorithm that detects bursts based on the recent history of time-series attributes. Here, we use an approach based on Z-score or standard score. Z-score was first introduced in 1968 in financial literature to predict the bankruptcy of a business [105]. Now, it is being widely used in statistics. The main advantage of Z-score is that it needs very little historical data. The algorithm gives feasible results when we have the same amount of historical data as the period for which the anomaly is being detected.

In our pattern detection approach, we adapt an implementation of a robust peak detection algorithm based on z-score [106] and use it as our score function f_i^b to detect bursts in time-series attributes. For a node v_i with time-series signal x_i, Z-score z_i is the number of standard deviations σ_i by which $x_i[t]$ (current value at time t) is above or below of the mean μ_i of historical observations $x_i[t-\Delta t, t-1]$, where Δt is the size of the historical time-window:

$$z_i = \frac{x_i[t] - \mu_i}{\sigma_i} \tag{3.2}$$

Prior knowledge about the nature of the time-series data is important when using Z-score as a scoring function. We should take into account the nature of the time-series attributes to define the size of the time-window. The size of the time window Δt defines the sensitivity of the algorithm to the new streaming data. Higher values of Δt lead to a larger time-window and therefore make it less adaptive to changes in streaming time-series data. When choosing Δt we should take into account prior knowledge about the stationary periods of the data. If we know that there are weekly trends that are stationary, we should set Δt to the value that corresponds to stationary periods in our time-series. If Δt is lower than that stationary period, it will lead to a false-positive detection of a pattern. Although, if we set Δt too high, the sensitivity of the algorithm will drop and the false-negative rate will increase.

3.1.3 Stage 2: Hopfield network. Learning dynamic associative attention weights

Once we focused the attention of the model on potentially anomalous nodes and partially solved the temporal part of the problem, we can proceed with solving its spatial remainder. To do that, we learn dynamic connections between the nodes based on the prior connectivity and the correlation among time-series attributes of the nodes. Our approach is based on the Hopfield model of artificial memory [100]. It is an unsupervised learning method.

The presented pattern detection approach is aimed at detecting groups of vertices that have correlated abnormal behavior. The learning stage is intended to make this coherent behavior apparent in the memory network by learning weights between connected nodes in the graph.

To implement it, we use a synaptic-plasticity-inspired computational model, the Hebbian learning rule [107]. The main idea of this model of brain memory is that a co-activation of two neurons results in the reinforcement of a connection (synapse) between them. Although, contrary to the original learning rule, in our model, we do not take causality of activations into account.

In our case, the Hopfield network has N nodes. These nodes correspond to the ones given in the dataset. However, we do not consider all the nodes. Instead, we learn the weights only between those nodes that remain after the attention-based reduction performed at Stage 1 (only the nodes containing pattern-related features).

The learning process is as follows. We use the initial structure of the given network. For two initially connected nodes i and j of the Hopfield network, at time t, we update the weight of an edge e_{ij} between them according to the similarity measure $\text{Sim}\{i, j, t\}$. This process is illustrated in Fig. 3.4. Note that we only perform this step for the nodes that are initially connected and do not compare every possible pair of nodes. That is crucial for the tractability of the method in cases when we deal with large and dense networks.

For each time step t, the edge weight w_{ij} between i and j is updated as follows:

$$\Delta w_{ij} = \begin{cases} +\mathrm{Sim}\{i,j\}, & \text{if } \mathrm{Sim}\{i,j\} > \lambda, \\ -\alpha \mathrm{Sim}\{i,j\}, & \text{otherwise,} \end{cases} \tag{3.3}$$

where $\lambda \geq 0$ is the *sparsity parameter* and corresponds to the edge weight threshold. Similarly to firing neurons, nodes expressing similar behavior have their connection weight increased. When $\alpha > 0$, the weight decreases allowing older patterns to be "forgotten" to keep only the latest patterns. If we compute patterns over longer time windows and set $\alpha = 0$, our approach detects patterns that represent densely connected components that are hard to interpret. Therefore, if we compute patterns over longer periods, we advise to increase α, and decrease it otherwise. The value of λ influences the sparsity of the final network. Increasing λ reduces the number of edges in the resulting Hopfield memory network. One should increase the value of λ, when looking for the most outstanding patterns and decrease it when higher sensitivity is required.

Before describing the similarity function, let us introduce the activity function y_i at node v_i:

$$y_i[t] = x_i[t] \times k_i[t]. \tag{3.4}$$

Note that the activity function can also be defined from the features calculated at Stage 1, $y_i[t] = f_i(x_i[t]) \times k_i[t]$, where k_i (Eq. (3.1)) is a binary vector that encodes bursts of activity in the attributes (non-zero values indicate bursts). Here, we use the definition formalized in Eq.(3.4).

Alternatively, we could define various similarity measures. For example, we could do the following:

$$\mathrm{Sim}\{i,j,t\} = y_i[t]y_j[t]. \tag{3.5}$$

When y_i and y_j are normalized, this measure gives the Pearson correlation between the nodes (if $\lambda = 0$ and $\alpha = 1$). The L^2 distance with a Gaussian kernel $\mathrm{Sim}\{i,j,t\} = \exp(-|y_i[t] - y_j[t]|^2)$ is also a plausible candidate.

During the learning process, the connections between nodes with similar activity are reinforced (see Fig. 3.4). To prune low weight edges, we introduce a threshold w_{min} and remove edges whose weight is below the threshold.

After removing low-weight edges, each Hopfield network transforms into a modular graph structure with strongly connected clusters of nodes having similar activity. These groups can be either isolated connected components or communities within the largest connected component.

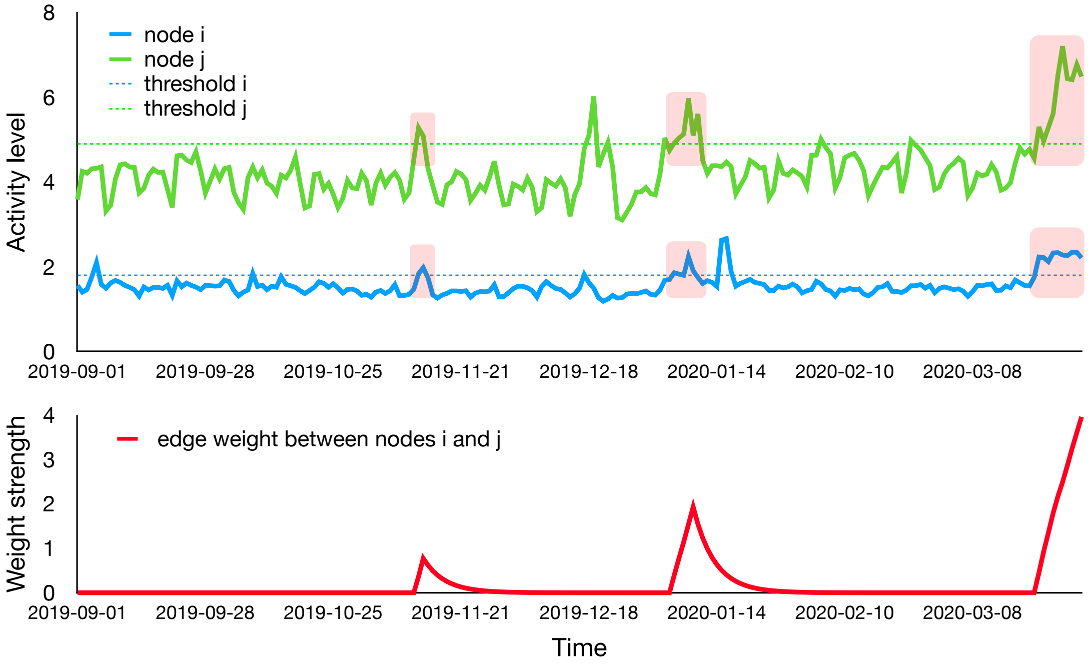


Figure 3.4 – Illustration of learning of edge weight w_{ij} between two connected nodes i and j when $\alpha > 0$. The dynamic attributes of the nodes are represented as two arbitrary time series. Correlation of the activity spikes leads to strengthening of the edge weight between two nodes (red line) as described in (3.3). The spikes (top plot, highlighted in red) are registered when the activity surpasses a threshold, which is computed separately for every node (see Eq. 3.1). The bursts are encoded in the binary vector k_i. Once the activity becomes decorrelated and $\alpha > 0$, the edge weight between the nodes gradually fades out. If $\alpha = 0$, the weight continuously increases.

3.1.4 Stage 3: Application. Pattern detection, information recovery, recommendation.

Once we learn the Hopfield networks, we can use the resulting data structures for multiple applications. Let us consider a few examples.

Pattern detection. The main application of our approach is pattern detection. In this application, clusters in Hopfield networks correspond to spatio-temporal patterns that reflect dynamic changes in the network over time. The analysis of the Hopfield networks and their communities provides a good way to detect, analyze, and interpret the dynamic patterns in graphs with time-series attributes.

To detect communities, we can use modularity-based methods [108, 109]. The structure of the communities allows for the detailed interpretation of the detected patterns. We will further focus on this application in the following chapters of this thesis. In particular, in Chapter 4, we will see how the learned data structure can be used to detect dynamic patterns of user activity in Wikipedia web network. The pattern interpretation can also be extended with application-specific modules. In Section 5.2, we will see an example of such module. In that application, we will use a natural language processing module for topic detection to get further insights into the detected patterns and their interpretation.

Information recovery. Another possible application is the data recovery from incomplete samples. This can be useful when we want to infer some node attributes in cases where they were not initially available or got destroyed as a result of an attack on the network. This application is illustrated in Section 3.2 using a synthetic dataset.

The recovery can be done using the memory properties of our approach that are reminiscent of the memory properties observed in Hopfield networks. Once the pattern is detected as a by-product, we have a memory matrix at our disposal, which can be used for pattern retrieval from partial samples. Starting from an initial partial memory pattern $P_0 \in \mathbb{R}^{N\times T}$, the recall of a learned pattern is done by the following iterative computation:

$$P_{j+1} = h_\theta(WP_j), \tag{3.6}$$

where $W \in \mathbb{R}^{N\times N}$ is the weight matrix of the Hopfield network. The function $h_\theta : \mathbb{R}^{N\times T} \to \mathbb{R}^{N\times T}$ is a nonlinear thresholding function (step function giving values $\{-1, 1\}$) that binarize the vector entries. The value θ is the threshold (same for all entries). For each $j \geq 0$, P_j is a matrix of binarized time series attributes of the network, where each row is associated to a node of the network and each column corresponds to a time step of the time series attribute. We stop the iteration when the iterative process has converged to a stable solution ($\|P_{j+1} - P_j\| \leq \varepsilon$, where ε is small, the norm is the Frobenius norm). The initial pattern P_0 is a binary matrix, where the rows have all values set to -1 (inactivity) except the ones associated to the partial memory pattern obtained from the time-series using the expression of k_i defined in Eq. (3.1). The computation of the iterative process is efficient as the matrices are sparse and in practice, it converges after a feasible number of steps.

The pattern recovery or "recall" equation of Hopfield networks (Eq. 3.6) is central to our method and its applications related to information recovery and recommendation systems. We used the same intuition as in Hopfield networks, where the model is trained to reconstruct data from incomplete inputs. In our case, each pattern is a Hopfield network with a weight matrix W and a set of binary activations P that encode time series attributes. Each network is a set of nodes connected by weighted edges. Weights represent the strength of the connection between nodes. Same as in classical Hopfield networks, we used binarized activations. Each node can have one state at a time, and these states can be -1 or +1. In the original paper, J. J. Hopfield used binary units to compute the Hamming distance between two binary states of the network, which is defined as the number of entries, where the binary values are different. However, binarization is not a required prerequisite, and the recovery can also be done with nonbinary states, as it was shown in [110].

Recommendation. Dynamic recommendation systems based on the explicit attention mechanism became popular in the past few years [91, 92]. Such recommendation systems target dynamic data domains with underlying graph structure. A common feature of dynamic recommendation systems is the way they learn representations of items. All recommended items are connected in a graph. The weights in the graph are learned based on the dynamic attributes of the nodes. Then, the recommendations are inferred using a dot product between the learned attention weight matrix and the incomplete set of attributes (e.g., users with incomplete profiles of their interests). Similarly to these works, the weighted network, which we learn with our approach, can also be used as a basis of a dynamic recommendation system. Moreover, the recommendation task is similar to the previously described task of information recovery. This task can be formulated as follows. Given a weighted network of connected entities W and partial information about their dynamic attributes $P_0 \in \mathbb{R}^{N \times T}$ infer missing attributes. The recommendation is defined as a dot product between the weight matrix W and the matrix of the dynamic attributes P_0. Therefore, the inference step is the same as in the information recovery problem and can be solved using Eq. (3.6).

For instance, the network W could represent web-pages that have viewership activity as a dynamic attribute. When readers click on a certain web-page, the task would be to recommend another web-page to visit based on the partial historical activity of the previous users, represented as P_0. The practical demonstration of this feature will be presented in Section 4.3 of the following chapter. We will illustrate it on the dynamics of Wikipedia web network, where given multiple articles on the topic, we will infer other articles on the same topic based on the learned Hopfield memory network.

3.2 Example on synthetic data

In the previous section, we formalized the connection between the attention mechanism and the associative memory model based on the Hopfield network. We showed that graph-structured data inherently possesses memory properties that enable us to use our approach for multiple applications. To make this connection more intuitive, in this section, we provide a visual example using simplified synthetic data, which is illustrated in Fig. 3.5 a).

Let us consider a sparse random Erdős–Rényi graph with 500 vertices and 12171 edges (5% of a fully connected graph). Each node in the graph has a time-series attribute of length $T = 5000$, which is stored in matrix P. We simulate an anomalous activity pattern on a random subset of nodes between 500 and 600 time steps generating a consistent spike of activity. The goal of our approach is to detect a subgraph, where this anomaly has occurred. Using Eq. 3.1, Stage 1 of the algorithm focuses attention of the model on potentially anomalous nodes that are highlighted in red in Fig. 3.5 (b). Once we have selected those nodes, we are not considering the rest, which are colored in green. Then, in Stage 2, we use Eq. 3.3 to learn the connections between the nodes in red, which we can see on the same figure. Once the learning is done, we can proceed to the final stage of the algorithm and use the representation in an application.

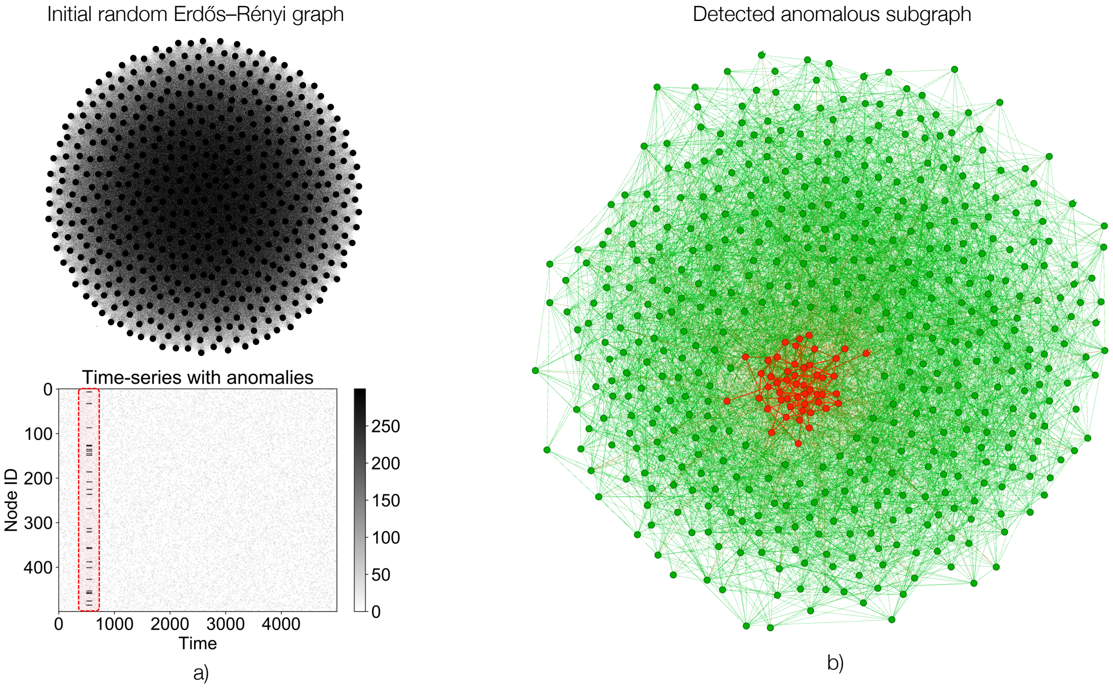


Figure 3.5 – Attention focus and anomaly detection example. Synthetic data. a) Initially, we have a sparse random Erdős–Rényi graph (top) with 500 nodes, each of which has a time-series attribute (bottom) of length $T = 5000$. We simulate anomalies in a random subset of time-series attributes between 500 and 600 time steps and apply our algorithm. b) Results of the anomaly detection algorithm. During Step 1, the algorithm focuses attention on a subset of potentially anomalous nodes (in red). Computing the Step 2, the algorithm does not consider the rest of the network (in green) and learns weights only among the nodes whose features are potentially related to the anomalous pattern. The resulting anomalous cluster is highlighted in red.

Let us demonstrate an application of information recovery. We illustrate the recovery process in Fig. 3.6 and zoom in to highlight the anomalous part of the time-series between 500 and 600 time steps. First, we create our partial memory pattern P_0 erasing time-series attributes from 50% of the nodes. As we can see on Fig. 3.5 b, computing a dot-product adjacency matrix W of the learned graph and the partial memory pattern, as shown in Eq. 3.6, we can successfully recall or recover missing time-series data.

Besides, this experiment also demonstrates the effect of the explicit attention mechanism. When the model recalls the memorised pattern, it focuses only on the important parts of the signal that contribute to the detection of the pattern, ignoring the uniform activity of the nodes. Fig. 3.6 (panel 3) illustrates the explicit attention effect.

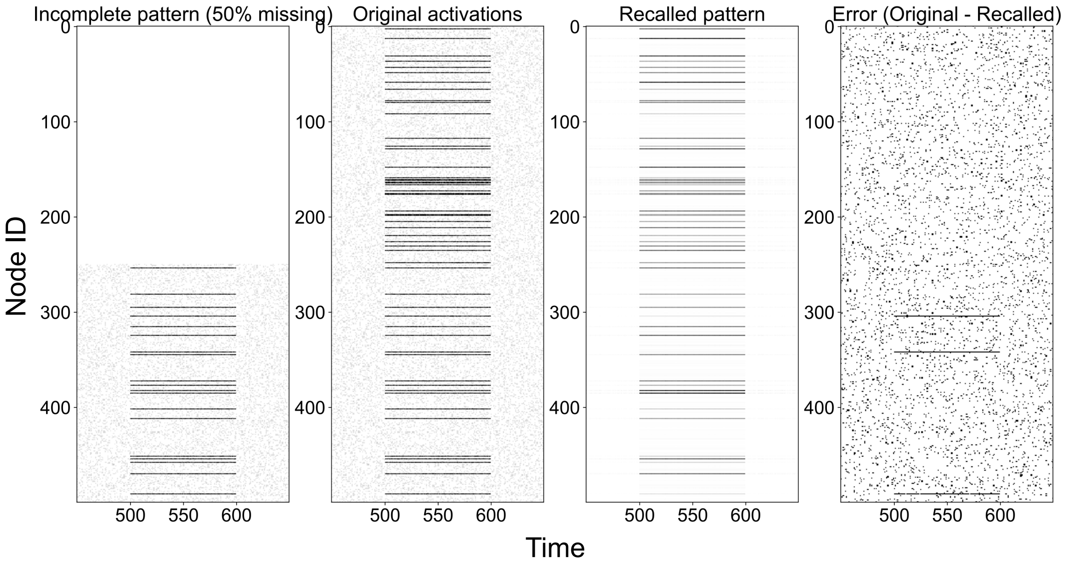


Figure 3.6 – Recall experiment on the synthetic data. The recall process is done by computing a dot product between the partial pattern P_0(panel 1) and the learned weight matrix W as shown in Eq. 3.6. We create a partial activity pattern removing time-series attributes for 50% of the nodes in the graph (panel 1) and use the pattern detection algorithm to recover the missing part. The restored pattern is shown on the panel 3. The error (panel 4) is computed by subtracting original and recalled patterns. We can see that the pattern (horizontal lines) is recovered and the uniform activity is ignored by the attention mechanism (appears as noise in the panel 4).

We tested the recall mechanism in settings with different fractions of missing data. Since the random graph is well-connected and the random signal is relatively uniform, the recovery of the missing fraction of the pattern shows good results even with only 10% of available information. However, when we ran experiments on real data with a sparser nonrandom network, we have observed that the recovery performance gradually deteriorates when we remove more data from incomplete patterns. All in all, the sparser the network, the harder it is to reconstruct the missing fractions of incomplete patterns. We will discuss this observation in more detail in the experiments described in Section 4.3.

3.3 Scalability and constraints

The proposed algorithm can be scaled to large dynamic graphs. In Chapter 4 and Chapter 5, we scale it to tens of millions of nodes and billions of edges. The computations are tractable because:

- they are local on the graph, i.e., weight updates depend on a node and its one-hop neighbors and can be implemented using parallel message-passing approaches

- weight updates are iterative
- a weight update occurs only between initially connected nodes and not between all possible combinations of nodes

These three facts enable us to build a distributed implementation based on a message-passing approach to speed up computations. For this purpose, we use a graph-parallel Pregel-like abstraction, implemented in the Apache Spark GraphX framework [111, 112].

Ranshous et al. [38] provided a detailed analysis of complexity of dynamic pattern detection algorithms using Big O notation. The most scalable algorithms presented in the survey have complexity $\mathcal{O}(NT)$, $\mathcal{O}(ET)$, or $\mathcal{O}((N+E)T)$, where N is the number of nodes, T is the number of time steps, and E is the number of edges[I]. We use their survey to compare our algorithm to other works.

Let us analyze the complexity of every stage of our algorithm. Stage 1 (Sec. 3.1.2) of our algorithm has complexity $\mathcal{O}(NT)$. Stage 2 (Sec. 3.1.3) has complexity $\mathcal{O}(ET)$. The recall process involves multiplication by a sparse matrix with $2E$ nonzero entries, hence the complexity is $\mathcal{O}(ET)$. We can see that, in terms of complexity, our algorithm is among the most scalable algorithms presented in the survey.

Additionally, it is important to point out that the number of nodes and edges considered in the computations is not necessarily the number found in the input graph. The attention mechanism, which we implemented in Stage 1, discards a large number of nodes N that are unrelated to patterns. Moreover, Stage 2 sparsifies time series attributes as Eq. (3.4) sets a large number of values to zero reducing the number of time steps T that are considered in computations. All that allows us to reduce the amount of the processed information by an order of magnitude, which positively affects performance of our algorithm.

One of the constraints of our algorithm is that the number of patterns that can be memorized by a single network is limited [113]. Indeed, without the forgetting parameter α in Eq. 3.3, the clusters of nodes will accumulate inside the graph, eventually overlapping and forming larger clusters of unrelated anomalies. To address this issue and to keep track of older events, we create snapshots of memory by slicing the time series into time windows of finite duration and by creating multiple networks for each slice. The time window size depends on the application.

3.4 Discussion

The algorithm presented in this chapter should be regarded as a system with two main components that can be changed based on the application and the data domain. First component is based on the attention score function that can also be seen as *feature extraction module*,

[I] Here, we modify the notation used in the survey. To denote the number of edges in the graph, we use E instead of m.

which we have defined as a burst detection function (Eq. 3.1). Instead, practitioners can use ARMA filter prior to thresholding, short-time Fourier transform, a wavelet transform, or any other user-defined function. Second component is responsible for *weight update* (Eq. 3.3) that is controlled by a *similarity function* (Eq. 3.5). These functions can also be changed in case one needs to take into account, for instance, negative co-activations or other patterns of temporal dynamics.

In this chapter, we also demonstrated the performance of the proposed algorithm on synthetic and real datasets, highlighting its advantages in terms of interpretability and scalability. However, before we go on with other experiments and applications, we would like to point out a few important nuances that should be taken into account when applying it. These nuances are related to the Hebbian learning rule and its constraints.

In Stage 2 (Sec. 3.1.3), to learn the memory network, we adapted the Hebbian Learning rule, as shown in Eq. 3.3. We chose this rule because of multiple reasons. First, it is unsupervised. Second, it is intuitive and computationally efficient. Finally, the rule is easy to implement and to adapt for different purposes. As we are going to see in the following chapter, due to these properties, the proposed anomaly detection algorithm shows good performance on datasets of different sizes and complexities.

However, we would like to discuss some aspects of the rule that can affect the performance and effectiveness of our algorithm. These aspects were first described by Miller and MacKay in [114]. They pointed out that correlation-based rules without constraints are unstable and result in either exploding or vanishing connection weights. To mitigate this effect, they considered two types of weight decay, subtractive and multiplicative. Multiplicative decay takes into account the current strength of the connection between two neurons and adapts the weight proportionally. Subtractive decay updates the weights at a fixed rate and does not consider the current strength of the connection. Both types of constraints represent fundamentally different ways to control the weight update. Under a subtractive constraint, the structure of the model converges to a set of the most correlated neurons. The weights of the connections between these neurons reach the maximum allowed strength, while the rest of the connections die out. Under a multiplicative constraint, we can represent a wider range of connection weights and avoid tuning the minimal and maximal allowed weight limits.

In the algorithm presented here, we adapt the subtractive constraint, adjusting the weight at a fixed rate (Eq. 3.3). Then, we introduce a limit w_{min}, which is a minimum strength of a connection between two neurons. Connections that have lower weights are discarded. The drawback of such solution is that we need to take care of the time window for which we compute the anomalies. This time window depends on the data and requires prior knowledge about it. For instance, as we will see in the applications, knowing the periodicity of the data gives us a feasible starting point in choosing that time window. This works well in most cases, however, it would be useful to introduce a multiplicative constraint option to the model to cover more potential use cases and to make our approach more flexible and versatile. A multiplicative constraint can be useful when the periodicity of the data and the duration of potential anomalies are unknown. Moreover, a multiplicative constraint would allow us to avoid getting disconnected components in the resulting graph. Disconnected components are common when we use subtractive constraints.

Lastly, in some applications, large hubs may introduce a bias into the data. For instance, when attributes of the hubs are irrelevant to solving the problem or when the attributes are noisy, such nodes propagate their attributes to the neighbors during the aggregation and learning stage, significantly affecting attributes of the rest of the network. To solve the problem of large hubs that are highly connected with the rest of the network, we can introduce a structural constraint, which was not described by Miller and MacKay. A similar constraint is widely used in GNNs during the feature aggregation stage. When changing weights, such constraint takes into account the structural properties of the network and adapts weights with respect to the connectivity patterns, such as a node's degree and centrality measures. This allows mitigating the influence of the attributes located in the hubs.

4 Analysis and evaluation of the detected patterns

In the previous chapter, we have introduced our pattern detection algorithm for dynamic graphs. We demonstrated an example of pattern detection on a synthetic dataset. Also, we briefly discussed the main aspects and potential applications of our approach.

In this chapter, we continue exploring different aspects of the proposed algorithm in more detail. We conduct large-scale experiments and perform an in-depth analysis and evaluation of the results. We focus on the effect of the attention mechanism on the scalability, discuss applications of the memory properties, and provide a detailed interpretation of detected patterns.

In Section 4.1, we compare the performance of our approach with other works using the Enron email dataset that is commonly used for the evaluation of such algorithms.

In Section 4.2, we scale the algorithm up to the size of the entire English Wikipedia web network, detecting anomalous patterns of viewership activity on webpages. In Section 4.2.1, we provide a detailed analysis of the algorithm giving concrete examples on real-world data, illustrating the explicit attention effects and memory features of our model.

To give a better understanding of why the model detects certain patterns, we demonstrate one of the most important features of the proposed algorithm – its interpretability. We illustrate and interpret multiple concrete examples in Section 4.2.2. Also, based on the interpretation of the detected patterns, we uncover insights about the collective behavior and interests of billions of Wikipedia readers.

After the detailed analysis and interpretation of the detected patterns, we focus on the memory properties of our algorithm and related applications. In Section 4.3, we show how we can use the memory properties in other applications, such as recommendation systems and information recovery. Finally, in Section 4.4 we present another memory-based application in the digital humanities. We study the sociological phenomenon of collective memory, the way social groups remember the past, at a much larger scale than it had been done before.

4.1 Detecting patterns in the activity of Enron email network

In this section, we compare the quality of our pattern detection approach to the results produced by other methods. To do that, we use the Enron email dataset [57]. This dataset is a widespread test case for a diverse set of algorithms, including anomalous pattern detection in dynamic graphs.

To test the performance of the proposed algorithm, we compare our results with the body of work presented in the related works (see Sec. 2.1) and use the events presented in these works as ground truth to validate the accuracy of our pattern detection. We follow the same data preparation approach as in the related works. We create an email communication graph and use time slices of the dynamic graph to detect dynamic patterns. Contrary to the methods presented in the related works, we use a hybrid approach, combining node features and graph structure to detect patterns in the dynamic graph of Enron email communications. We define a pattern as a sudden anomalous increase in email communication among a group of employees of the corporation.

Let us focus on the dynamic data structure in more detail. To apply our pattern detection algorithm to the Enron dataset, we represent the data as a dynamic graph data structure, which is depicted in Fig. 3.1. The underlying graph is the network of email communications. Nodes in the graph correspond to the email addresses of employees. Two nodes are connected if they have exchanged at least one email over 5 years. It is an undirected, unweighted graph. Time series associated to the nodes are captured from the email activity; each temporal value is the number of emails sent from the associated address during one day.

First, we use Stage 1 (Sec. 3.1.2) to select employees that have spikes in their communications. Then, we use feature similarity to compute strongly connected employees. Here, during learning performed at Stage 2 of our approach (Sec. 3.1.3), the connection between email addresses is reinforced if a similar number of emails was sent by both of them over the same hour, indicating an active email exchange.

After learning a Hopfield network for every monthly snapshot, we investigate the structural component. To do that, we take four monthly time slices corresponding to the periods that were discussed in the literature. In the previously discussed related works, the authors observe four anomalous periods and relate them to the specific news reports involving Enron. We use these events as a ground truth. These are the following periods:

- **December 1999.** A sham deal between an Enron entity and Merrill Lynch, an investment department of Bank of America, to boost the stock price.

- **April 2001.** A public scandal, involving Wall Street analyst R. Grubman and Enron's CEO J. Skilling. Mr. Slilling insulted Mr. Grubman during an interview after a question on the refusal of releasing the balance sheet of Enron.

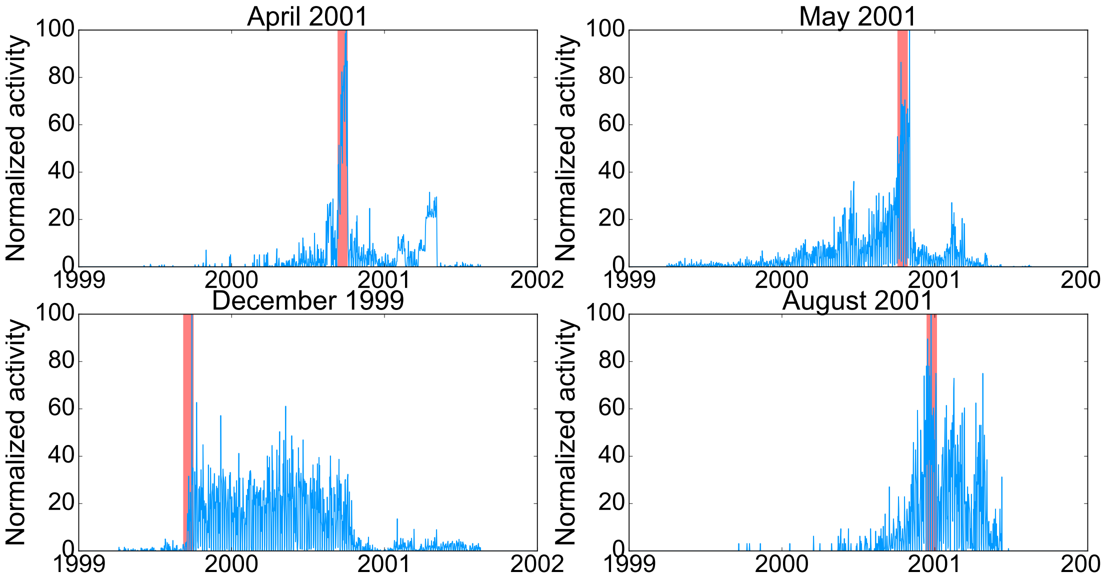


Figure 4.1 – Anomaly detection in Enron email network. Red areas highlight the month periods previously reported as anomalies (ground truth supported by real world events). Blue lines reflect the normalized (scale 0-100) overall activity level in the network computed by the proposed algorithm. We can see that the algorithm detects anomalies in all reported cases.

- **May 2001.** Closure of Enron's largest foreign investment, the Dabhol Power Company in India, due to another scandal leading to Enron's bankruptcy.
- **August 2001.** Resignation of Enron's CEO J. Skilling, followed by the bankruptcy of the company in November 2001.

For each month, we select the largest connected component of the learned graph. We sum up the activity of its nodes to get a single time-series representative of the group activity, which is illustrated in Fig. 4.1. We define a pattern as a spike of overall activity in a cluster of email addresses that we detect after learning. The spike is detected using a standard-deviation-based threshold. As we can see, all four curves have a larger activity during the chosen month than in the rest of the period. For April and May 2001, when two major scandals happened, it is more than twice the maximal activity for the rest of the month, showing the evidence of an anomalous pattern in the email communication network. The increases in activity that happened in December 1999 and August 2001 have a longer trace after the main anomalous pattern was detected. However, we can see that the high activity does not spread for more than one year.

Concerning the monthly components of active nodes for the 4 chosen months, it involves 29, 25, 126, 28 nodes, respectively, for December 1999, April, May, and August 2001. Almost all of them correspond to the addresses of Enron employees (some emails in the dataset have external domains). Except for the event that happened in May, the activity involves less than 30 employees. The closure of the largest foreign investment led to the creation of a large connected component in the graph. It contains 100 employees out of 158, indicating that the major pre-bankruptcy event had severely impacted the whole company.

To conclude, our method detected all anomalous patterns presented in the state-of-the-art literature. Our approach reveals the days of the peak activities, the duration of the events, and the involved employees, facilitating investigation of the patterns and interpretation of our method. Along with interpretability, scalability of the proposed pattern detection model is one of its main advantages. We are going to focus on that in more detail in the following section, where we scale up our algorithm to the entire Wikipedia web network.

4.2 Large-scale pattern detection in Wikipedia web network

Over recent years, the Web has significantly affected the way people learn, interact in social groups, and store and share information. Apart from being an essential part of modern life, social networks, online services, and knowledge bases generate a massive amount of logs containing traces of global online activity on the Web. Most of this data is related to the standard activity of the users. However, the larger these logs become, the harder it is to detect deviations from normal behavior in the network. Localization of these anomalies becomes even more difficult because of the continuous expansion and dynamic nature of these networks. All in all, web and social networks represent an interesting, complex, and continuously evolving data domain, which requires new pattern detection techniques.

Being one of the most visited websites in the world, Wikipedia is an excellent example of a large-scale and constantly expanding dynamic network. The scale of the web network and the openness of its viewership logs make it a perfect test dataset, which enables us to demonstrate all aspects of dynamic pattern detection algorithms. In this section, we use Wikipedia server logs to analyze and to give a better understanding of multiple aspects of our approach, which we have presented in the previous chapter (Sec. 3.1).

To demonstrate the efficiency and scalability of the proposed pattern detection algorithm, we test our approach on the entire English Wikipedia web network and its viewership dynamics. We build a network of Wikipedia articles and use the visitor activity of each article, i.e., the number of visits of an article per hour, as a node attribute. The static underlying network is the Wikipedia hyperlink network. Two pages are connected if there is at least one hyperlink reference between them.

The results of our experiments demonstrate that we can use our algorithm to detect anomalous patterns in the activity of Wikipedia web network and interpret them to analyze the collective behavior of Wikipedia readers. Each pattern is a densely connected subgraph of Wikipedia articles whose behavior deviates from the norm. The emergence of each pattern is triggered by a sudden increase in viewership activity in a small, local part of the web network of Wikipedia articles. We observe that for the Wikipedia data, the subgraphs contain linked pages closely related to an event that triggered a sudden increase of visits during a short period. These clusters of anomalous nodes can then be used for more detailed investigation and interpretation as it is shown in Section 4.2.2.

Interpretability of the results is one of the main advantages of our approach. It provides a comprehensive description of the detected patterns. As a result, we are able to perform a thorough qualitative evaluation of our results. On the other hand, the quantitative analysis of the results of the experiments on Wikipedia data is complicated. In the experiments on the Enron email dataset (Sec. 4.1), we had the ground truth and we used it to validate the detected anomalies quantitatively. However, in the case of Wikipedia, a quantitative evaluation of the results turns into a challenge because we do not have ground truth labels in the Wikipedia dataset.

Nonetheless, in the experiments on Wikipedia data, we use alternative methods to validate the results of pattern detection. We evaluate the quality of detected anomalous patterns using trending events extracted from Google Trends. During the experiments, we noticed that detected anomalous patterns in Wikipedia's viewership dynamics can be associated to real-world events. This observation inspired us to validate our results using Google Trends and to use it as a reference that indicates anomalous search activity of internet users.

To give a more detailed qualitative analysis of the detected patterns, we also interpret them from the collective behavior point of view. There are several studies on mining patterns in visitor or editor activities on Wikipedia that are aimed at getting better insights on collective behavior and social interactions [26, 27, 115, 116]. To mitigate the high computational cost inflicted by the large amounts of data, most of the studies focused on particular topics of interest and subsets of selected Wikipedia articles. For instance, only traumatic events, such as attacks and bombings, have been investigated in [28], [29] based on the Wikipedia edit activity data. Tinati et al. [115, 116] proposed a Transcendental Information Cascade model and applied it to Wikipedia editorial activities to extract patterns of information propagation over the article network. Analyzing Wikipedia daily page views, Kanhabua et al. [27] investigated 5500 events from 11 categories such as aviation accidents, earthquakes, hurricanes, or terrorist attacks. Wikipedia hourly visits on the pages of celebrities were used to investigate the fame levels of tennis players [30]. Agarwal et al. [117] analyzed spatio-temporal patterns during the election period in the UK. The study focused on editors' and readers' engagement with Wikipedia's political content related to UK Members of Parliament. These studies point out the high interest in Wikipedia data and the increasing need for more systematic spatio-temporal pattern detection methods.

The first investigation from an anomalous pattern detection point of view was presented by Mongiovi et al. [118, 119], where Wikipedia pagecounts data are combined with the graph of hyperlinks. However, they applied their method to a preselected subset of Wikipedia. Due to the introduced concept of explicit attention and potential anomaly, our distributed algorithm (Sec. 3.1) allows us to handle the full Wikipedia network and long-term visitor activity records.

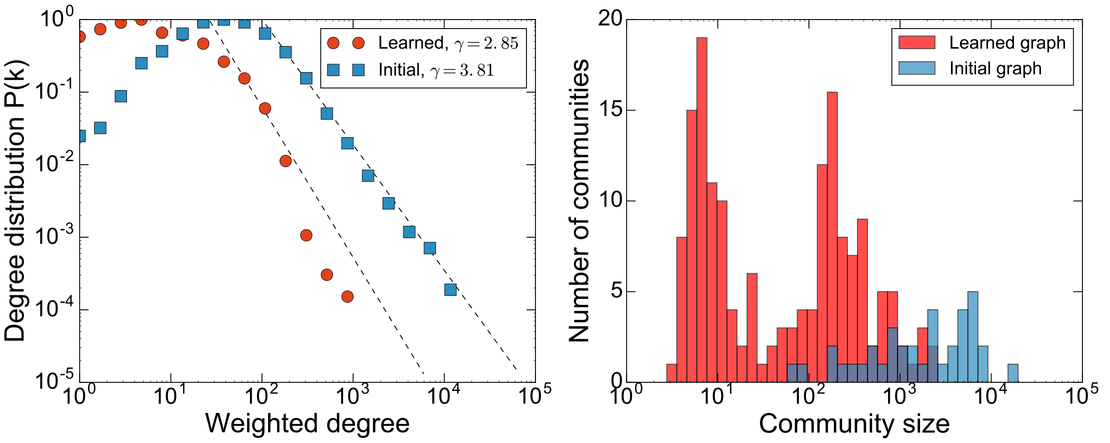


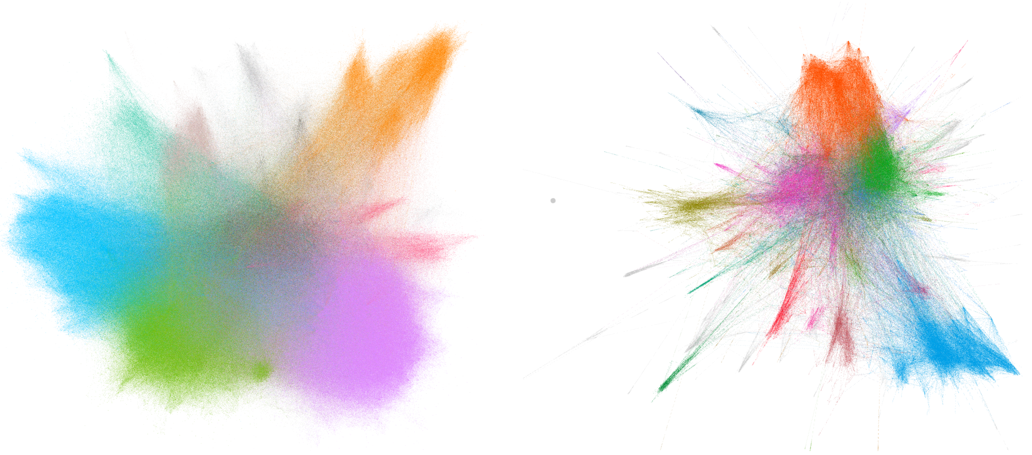
Figure 4.2 – *Left.* Weighted degree distribution in log-log scale for the Wikipedia graph and Hopfield network learned over the entire 7 months time span. Linearity in log-log scale corresponds to power-law behavior $P(k) \sim k^{-\gamma}$. The learned graph preserves a similar scale-free behavior, but is less connected and has fewer hubs than the initial graph. *Right.* Community size distribution of the initial Wikipedia graph of hyperlinks (blue) and the learned Hopfield network (red). The total number of communities: 32 for the initial graph, 172 for the learned one.

4.2.1 Analysis of the pattern detection stages

We start by analyzing the initial graph of Wikipedia webpages connected with hyperlinks. In this experiment, the time-series attributes of the nodes correspond to the viewership statistics of the associated web pages. To analyze the reductionist effect of the explicit attention mechanism of Stage 1 of our algorithm (Sec. 3.1.2), we detect potential anomalous patterns that emerged as a result of the long-term dynamics of the Wikipedia web network. Finally, to extract fine-grained patterns, we learn the Hopfield network (Sec. 3.1.3) using the viewership statistics from October 2014 to April 2015.

This experiment highlights the effectiveness of the explicit attention mechanism, which allows reducing the amount of processed data by an order of magnitude. After the learning stage, only 275’498 edges have strictly positive weights (4.2% of the initial graph). We remove the disconnected nodes and preserve only the largest connected component of the graph. The number of remaining nodes is 35’839 (31% of the initial number).

The analysis of the static underlying graphs shows that both Wikipedia graphs, initial and learned, have statistically heterogeneous connectivity and similar structure (Fig. 4.2). This highlights the effectiveness of the explicit attention mechanism and shows that we preserve the essential structural information after significant data reduction performed at Stage 1 of the algorithm.



(a) Initial (0.1M nodes and 5M edges) (b) Learned (0.02M nodes and 0.1M edges)

Figure 4.3 – Illustration of the data reduction effect achieved due to the explicit attention mechanism. Wikipedia graph of hyperlinks (left) and learned Hopfield network (right). The attention mechanism allowed us to reduce the number of nodes and edges by an order of magnitude. Colors correspond to the detected communities. We can see that the learned graph is much more modular than the initial one, with a larger number of smaller communities that potentially correspond to spatio-temporal patterns.

However, the initial Wikipedia graph is dominated by large hubs that attract most of the connections to numerous low-degree neighbors. These hubs correspond to general topics in the Wikipedia network that often link broad topics. For instance, the article “International Standard Book Number” that has a large number of hyperlinks pointing to it from pages covering very diverse subjects. If we look at the viewership statistics, the activity of the visitors in these large clusters is uniform and does not expose any patterns over time. We aim at extracting smaller communities that correspond to localized patterns in the dynamics of the network. This is the reason why we need Stage 2 of our approach (Sec. 3.1.3) to get the learned graph.

Visualizations of the initial and learned graphs using a force layout algorithm show striking differences. Looking at Fig. 4.3, we can visually assess the reduction effect of the explicit attention mechanism. The initial Wikipedia graph is dense and cluttered with a significant number of unused references, while the learned graph reveals smaller, refined, and more separated communities. This is also confirmed by the numerical measures such as the community size and degree distributions of the graphs (Fig. 4.2, right). The number of communities and their size change after learning. Initially, a small number of large communities dominate the graph (blue), while after the learning (red) we see a five times increase in the number of communities. Moreover, as a result of the learning, the size of the communities decreases by one order of magnitude. The modularity of the learned graph is 25% higher, strengthening the evidence of the creation of associative densely-connected structures.

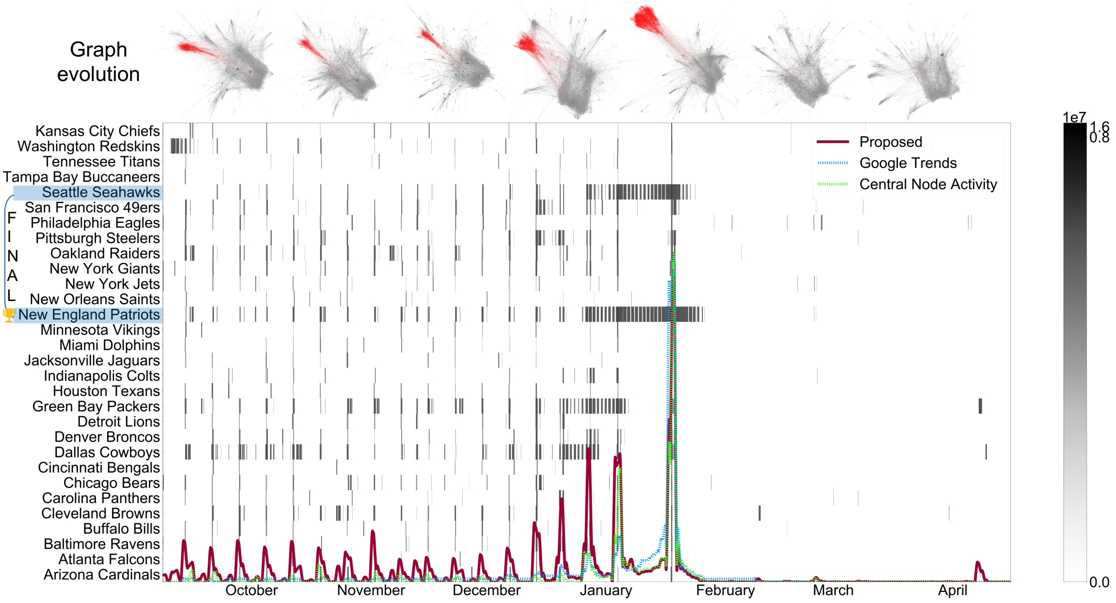


Figure 4.4 – Evolution of the National Football League 2014-2015 championship cluster and visits on its articles. We show 30 NFL teams from the main cluster. Top: the 7 monthly learned graphs in gray, with an explicit attention focus on the NFL cluster highlighted in red. This sequence illustrates the attention flow of the model. Middle table: visitor activity per hour on the NFL teams' Wikipedia pages in grey scale (the more visits, the darker). Bottom: the total number of visits (normalized) of the articles of the cluster over time (red), the Google Trends curve for the keyword "Super Bowl" (dashed blue), and the activity of the central node of the pattern-related cluster (dashed green).

These measurements indicate that, as a result of learning performed at Stage 2, we obtain a graph structure with refined strongly connected clusters that correspond to an interpretable summary of patterns in the network dynamics. We provide more concrete examples of the detected patterns and their interpretation in Section 4.2.2. We show that the analysis of each cluster of nodes in the learned graph gives us an overview of the events that occurred during the 7-month period and caused the anomalous behavior of Wikipedia readers during that period. Each cluster is a group of pages related to a common topic such as a championship, a tournament, an awards ceremony, a world-level contest, an attack, an incident, or popular festive events such as Halloween or Christmas.

4.2.2 Interpretation of the detected patterns

Interpretation is one of the most important features of the proposed algorithm. Due to a rich set of attributes, Wikipedia dataset is a perfect demonstration example for this feature. In this section, we focus on the interpretability of the detected patterns in more detail. We pick multiple patterns with different properties and explain the reasoning behind the detection.

In this experiment, we detect short-term patterns. To do that, we split the dataset into monthly snapshots. An average event attracting the attention of Wikipedia users usually lasts no longer than two weeks. Therefore, we are going to detect multiple anomalous patterns at once. Monthly graph snapshots are smaller compared to the 7-months snapshot and contain 10’000 nodes on average after learning. However, the properties and distributions of monthly graph snapshots are similar to the 7-months one, described in the previous section.

Before going deeper into the analysis and interpretation of the anomalous patterns in the network dynamics, we investigate the evolution of the graph structure with an emblematic example of the Super Bowl, the finals of the National Football League 2014-2015 championship. We track this pattern for several months between 2014 and 2015 and use it to illustrate the effect of the explicit attention mechanism (Fig. 4.4).

NFL is one of the most popular sports leagues in the USA and it triggers a lot of interest in the related articles on Wikipedia. Due to the high popularity of Wikipedia articles on this topic, we localized a cluster related to the NFL in multiple graph snapshots. Fig. 4.4 shows the detailed information about the NFL clusters. The top part of the figure illustrates the learned attention graphs for each monthly snapshot, where the attention cluster related to NFL is highlighted in red.

The same figure shows the evolution of the final stages of the championship. The final game of the 2014 season, Super Bowl XLIX, had been played on February 1, 2015. This explains the continuous expansion of the attention cluster until February where its size reaches the maximum. The activity drops right after this event, the cluster disappears, and the attention vanishes.

For the sake of readability of the figure, we extracted 30 NFL team pages from the original attention cluster (485 pages) to show the details of the evolution in time as a table in Fig. 4.4. This fraction of nodes reflects the overall dynamics of the entire cluster. Each row describes the hourly activity of a page, while the columns split the plot into months. The sum of visits for the selected pages is plotted as a red line at the bottom.

The dynamics of the detected cluster reflects the real timeline of the NFL championship. The spiking nature of the overall activity corresponds to weekends when most of the games were played. Closer to the end of the championship, the peaks become stronger, following the increasing interest of fans and expanding the attention area of the model. We see the highest splash of activity on 1 February, when the final game was played.

Note that this and other detected clusters were obtained in an unsupervised manner. The football team pages were automatically connected in a cluster having “Super Bowl” as a common topic. Moreover, the cluster is not formed by one Wikipedia page and its direct neighbors. It involves many related articles in the network that are several hops away from each other.

The NFL championship case is an example of a periodic (yearly) event. The interest to the championship increases over the months until the expected final event, which causes the emergence of the anomalous pattern in the network dynamics.

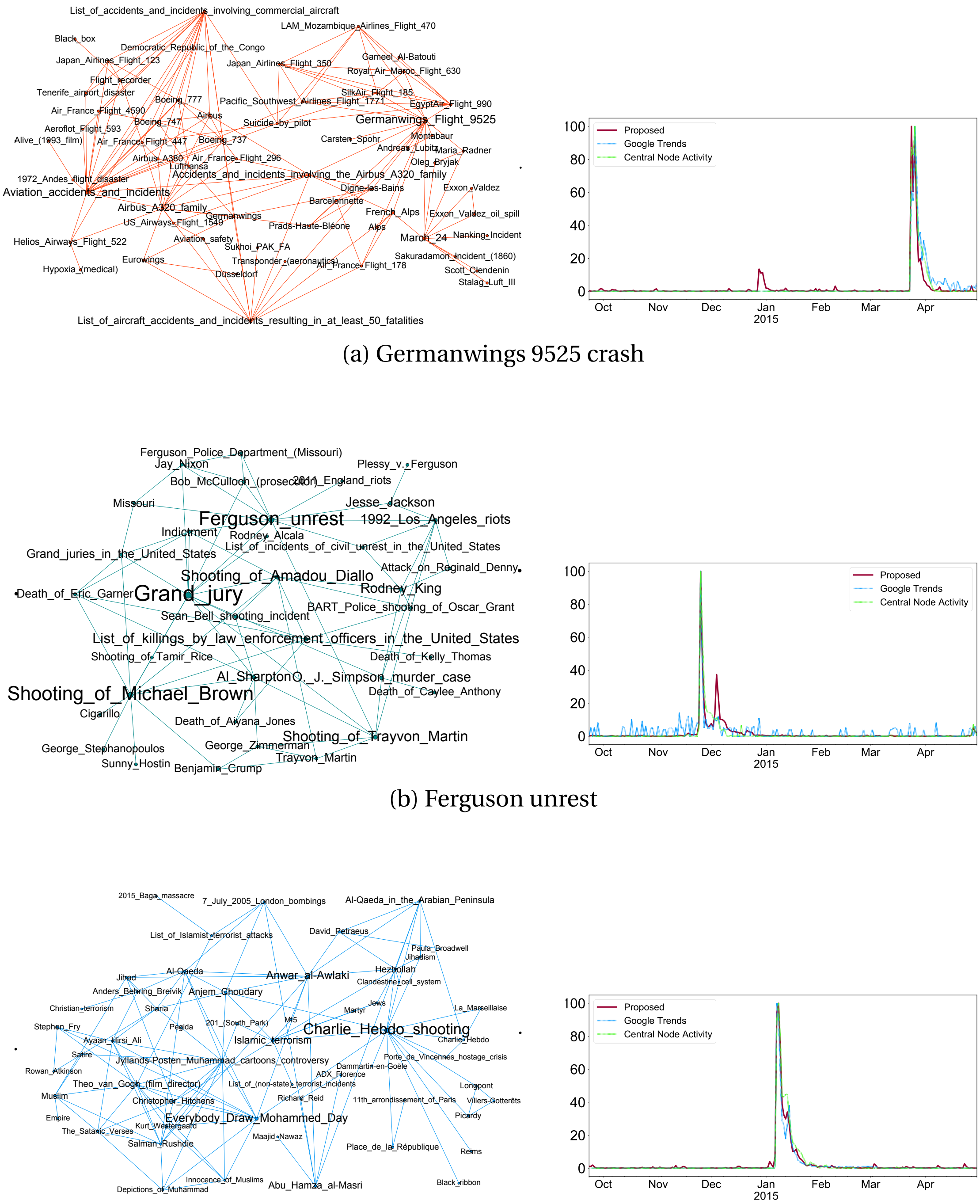
Accidents and incidents are events of a different nature as they appear suddenly, without prior activity. Despite this fundamental difference in the dynamics, the proposed method allows detecting such events and related patterns as well. We provide examples of three accidents to demonstrate the ability of our method to detect patterns that emerge due to the dynamics influenced by unexpected events.

We pick three events among the 172 detected and interpret them to show the details of our anomaly detection approach. Fig. 4.5 shows the extracted clusters from the learned graph (left) and the overall timeline of the clusters' activity (right). Same as in the NFL example, we evaluate the quality of the pattern detection using Google Trends as a reference.

Ferguson unrest. Second wave. November 24, 2014 – December 2, 2014. The second wave of the Ferguson unrest is characterized by dynamics that has beginning and end dates. A sharp increase in the activity at the beginning of protests highlights the start of the main event, which caused a spike of activity on the related Wikipedia pages. The start of the unrest triggers the emergence of the core attention cluster. We also see that the cluster becomes active once again at the end of the unrest, allowing us to record the two related anomalous patterns in the visitor activity.

Charlie Hebdo shooting. 7 January 2015. The terrorist attack on the headquarters of the French satire magazine is another example of an unexpected event. The attention cluster emerged over a period of 72 hours following the attack. All pages in the cluster are related to the core event and experienced a considerable increase in activity during the 72-hours period. We can see that just by looking at the title of the pages one can get an overall summary of what the event is about. There is a sharp peak of activity on the first day of the attack, slowly decaying over the following week.

Germanwings flight 9525 crash. 24 March 2015. This cluster not only involves pages describing the crash or providing more information about it but also captures pages of similar events that happened in the past. It includes, for example, a page with a list of airplane crashes and an article about another accident that happened in December 2014, the Indonesia AirAsia Flight 8501 crash. As a result, the attention cluster related to one event is connected to the attention cluster of the other event, causing a residual memory flashback and an attention spike in December. This is an example in which our approach captures two relevant events and groups them together in one cluster, allowing us to detect a secondary pattern that is only implicitly related to the main one.



(a) Germanwings 9525 crash

(b) Ferguson unrest

(c) Charlie Hebdo attack

Figure 4.5 – Graphs and activity timelines of the 3 events that triggered the emergence of anomalous patterns in the network dynamics. Left: clusters of pages grouped after learning in the Hopfield network. Right: A normalized sum of all visits of the articles of each cluster over time (in red). The Google Trends curves for the keywords "Germanwings 9525 crash" (a), "Ferguson unrest" (b) and "Charlie Hebdo attack" (c) are displayed in blue. Activity of the central page in the pattern is shown in green.

Table 4.1 – Examples of Wikipedia article titles found in the clusters associated to the presented events

Charlie Hebdo attack	Germanwings 9525 crash	Ferguson unrest
Porte de Vincennes hostage	Inex-Adria Flgt. 1308	Shooting of Tamir Rice
Al-Qaeda	Pacific S-W Flgt. 1771	Shooting of Amadou Diallo
Islamic terrorism	SilkAir Flight 185	Sean Bell Shooting Incident
Hezbollah	Suicide by pilot	Shooting of Oscar Grant
2005 London bombings	Aviation safety	1992 Los Angeles riots
Anders Behring Breivik	Air France Flgt. 296	O.J. Simpson murder case
Jihadism	Air France Flgt. 447	Shooting of Trayvon Martin
2015 Baga massacre	Airbus	Attack on Reginald Denny

Finally, in Table 4.1, we summarize our exploration of the clusters of learned graphs by providing a list of handpicked page titles that appear inside each cluster. These Wikipedia articles refer to previous events and related subjects, covering events that occurred outside of the analyzed 7-months period.

These examples illustrate the associative properties of the patterns that are detected by our algorithm. An anomalous pattern is characterized by a group of connected nodes with similar features. Nodes are grouped by the attention mechanism based on the attributes at Stage 1 and the learning processes at Stage 2. These stages also connect related patterns based on the shared cross-pattern attributes. All in all, these properties of the dynamic patterns allow us to interpret the cause of the detection and make the algorithm more intuitive.

4.2.3 Evaluation

To evaluate the quality of the detected patterns, for each of the events presented before, we compare the total number of visits in the clusters with Google Trends curves reflecting anomalous search activity of internet users. Here, we use Google Trends as a reference for qualitative evaluation. There is a nearly identical correspondence between the detected anomalous patterns and Google Trends, as we can see in Fig. 4.4 and 4.5. In all 4 examples, the anomalous activity and Google Trends curves reach their maximum at the same time and have a very similar shape. The differences that appear during the months prior to the Super Bowl date are explained by the fact that our "Super Bowl" cluster contains articles about football teams and other topics related to the Super Bowl. Due to the associative nature of the detected anomalies, this example has a better quality of detection than Google Trends.

As discussed previously, periodic spikes in visitor activity occur during the weekends when football matches are played. We observe the same phenomenon in the case of the Germanwings crash, where we observe a small peak of activity in December. This peak is the result of the prior activity on the page related to another airplane crash that happened in December

2014. This example demonstrates the richness of the detected patterns. They describe a group of events that influenced anomalous dynamics in visitor activity, as confirmed by Google Trends.

In addition, contrary to Google Trends, our patterns represent more than a single keyword since they emerge as a result of the anomalous dynamics on multiple pages describing different interrelated concepts.

4.3 Using detected patterns for information recovery and recommendations

As we saw in Section 2.4, there is a relationship between memory models and the attention mechanism. Inspired by this insight, we design another experiment to confirm this hypothesis. In this experiment, we analyze the memory properties of patterns detected by our method. We test our hypothesis that the proposed method, as a memory, allows recalling or recovering events from partial information. Note that we do not design a complete experiment to compare to other recommendation systems. The goal of this experiment is to present an example of the recall mechanism, demonstrating the recommendation ability of the network and to illustrate its complexity.

Let us consider a potential application of the memory property in recommendation systems. In recommendation systems, given partial information about a user's activity on a set of Wikipedia pages, the goal is to recommend new related articles based on the activity history and preferences of other users. In other words, the goal is to complete a partial pattern of a user's viewership activity that would reflect articles related to a potential topic of interest for that user.

The recommendation can be formalized as a recall process of Hopfield networks. We perform the recall using the Stage 2 (Sec. 3.1.3) and recover the pattern using a dot product between the the learned weight matrix and the incomplete pattern, similarly to the way it is done in other attention-based recommendation systems [91, 92] (see Sec. 2.3 for a more detailed overview of the related works). We show that the learned graph structure can recover a full pattern of anomalous user activity from an incomplete input, represented as a cluster of pages and its activations.

To emulate incomplete viewership patterns, we remove the activity information from a few pages in one of the detected patterns. We build the input matrix P_0 setting to (-1) (inactive state) all the time series except for the selected pages. Then, we iteratively compute a dot product as shown in Eq. (3.6).

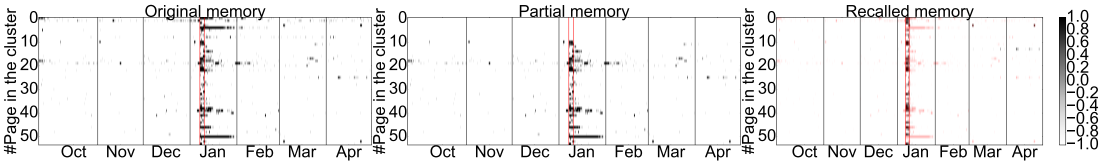


Figure 4.6 – Recall of an event from a partial pattern (Charlie Hebdo attack). The red vertical lines define the start of the event and its most active part, ending 72 hours from the start. Left: full activity over time of the pages in the cluster. Middle: pattern with 20% of nodes set inactive (top lines). Right: the result of the recall using the Hopfield network model. In light red are shown the difference with the original pattern (the forgotten activity).

In the cluster associated to the Charlie Hebdo Attack, we preserve the viewership data for a subset of articles (here, 80%) and remove that data for the rest of the articles (see Fig. 4.6). We apply the learned graph for the January snapshot when the anomalous pattern that emerged as a result of the attack was detected. As a result of the recall, for instance, if a user read the article about Charlie Hebdo Shooting, she will be recommended to read Everybody Draw Mohammed Day article, which was not present in the incomplete pattern.

After the recall, we can remark on a few important facts. First, if we focus on the short period when the event occurs (within the red vertical lines in Fig. 4.6), most of the time-series attributes are recovered. Second, the model forgets a part of the activity, plotted in light red, outside of the event bounds. This missing part comprises pages that are active outside of the timeline of the event, giving evidence that they are not directly related (or weakly related) to the pattern. This also demonstrates the effect of the attention mechanism. The model focuses only on those parts of the time-series attributes that correspond to the anomalous pattern while ignoring the rest. Third, the sparsity of the network also affects performance. The sparser the pattern-related cluster, the harder it is to reconstruct the missing parts of the activity pattern. Finally, we noticed that it is harder to reconstruct the activity from weakly connected pages or from pages that are further away from the core nodes in the graph. We also tested the reconstruction feature on incomplete patterns with different fractions of missing data and observed a linear correlation between the fraction of missing information and the reconstruction error.

The result of this experiment shows that our method can be used to recover the signals related to the detected anomalous patterns given a noisy or incomplete input. This feature can be used for recommendation systems, where we need to complete a user's interest profile based on incomplete information about her activity and the historical activity of other users. On the other hand, this feature can be helpful when investigating attacks on web networks. Our approach can be used when the data required for anomalous pattern interpretation is destroyed by some intruders that are interested in hiding traces after an attack on the network.

4.4 From Hopfield to Collective memory

The Wikipedia graph is very rich in terms of node-related attributes. In addition to the viewership statistics of millions of readers, we have access to the content and summaries of articles and the categories they belong to. This collection of additional sources opens new avenues to applications in domains outside of computer science. Incorporating additional data into our model led us to another interesting application in the field of digital humanities – collective memory of Wikipedia readers.

In 1925, Maurice Halbwachs introduced the term *Collective memory* [120]. He defined it as a set of memories that exists beyond the memory of an individual and affects a common understanding of the past by social groups.

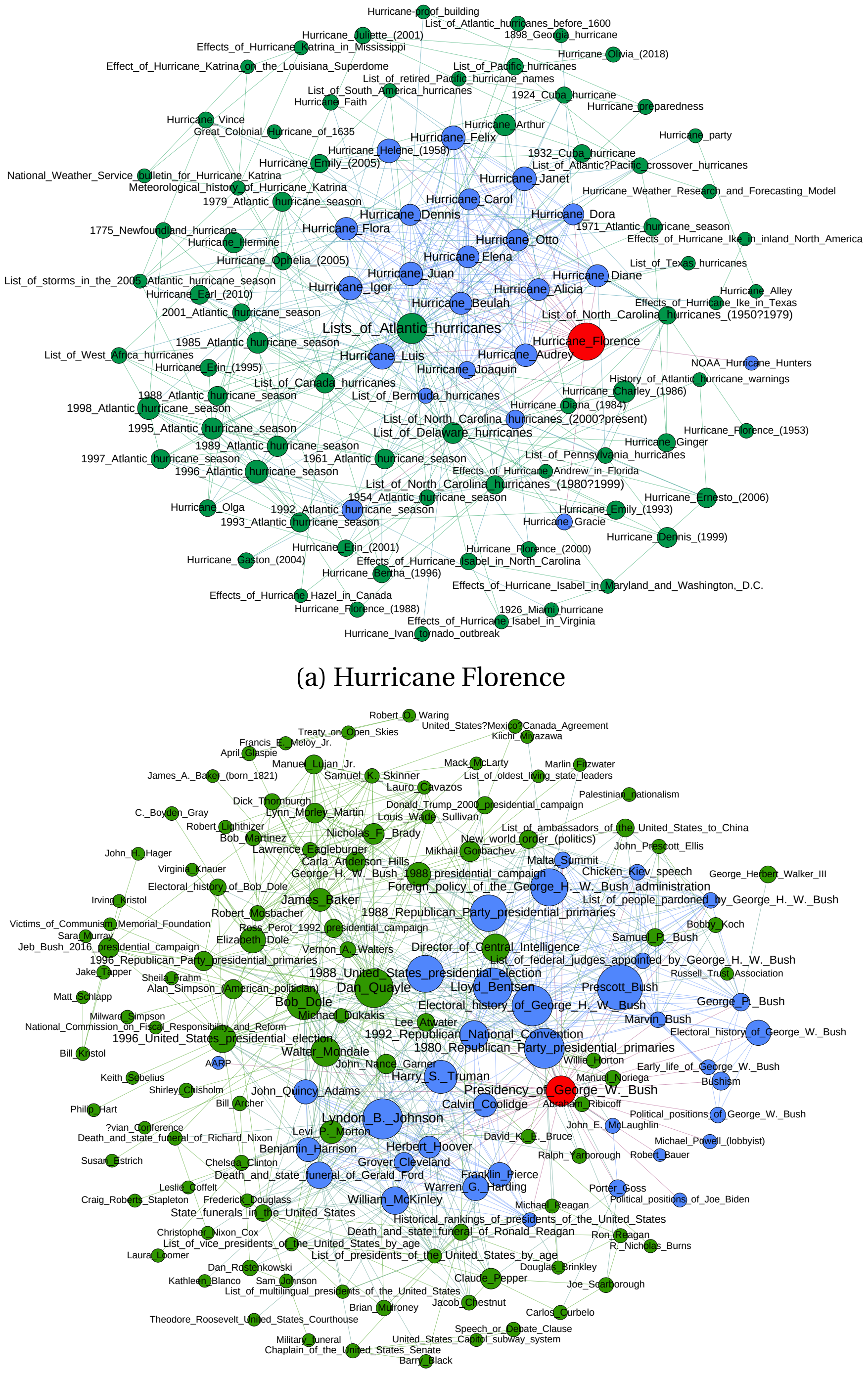
Collective memory is an interesting social phenomenon of human behavior. Studying this concept deepens our understanding of a common vision of events in communities. It shows how present events influence remembering of the past. Halbwachs's hypothesis initiated a range of studies in sociology [121], [122], psychology [123], [124], cognitive sciences [29], and only recently in computer science [115], [26], [27].

The experiments we have discussed in the previous sections show how we can use the Hebbian rule and Hopfield's associative memory model to identify dynamic patterns in the collective activity of internet users based on Wikipedia web network structure and its viewership attributes. We saw that the structures, which we extracted from the Wikipedia Web network, comprise clusters of pages related to certain events. When we look at the structures of the learned graphs, we can see that they also inherit the associative nature of Wikipedia and comply with the definition of collective memory.

This observation leads to an interesting question. Can these structures be similar to artificial models of human memory?

If we think of webpages as neurons, web networks resemble biological neural networks. Indeed, interconnections have a complicated structure, while nodes produce time-series of activations represented as visits of webpages and action potentials or "spikes" of neurons. In biological neural networks, neurons self-organize during learning and form strongly connected groups called neural assemblies [125]. These groups express similar activation patterns in response to specific stimuli. When learning is completed, and the stimuli are applied once again, reactions of the assemblies correspond to consistent dynamic activity patterns or memories. Synaptic plasticity mechanisms, formalized by Donald O. Hebb and then applied in the associative memory model by John Hopfield, govern this self-organization process.

This resemblance leads us to another question. Can we use the same mechanism to model individual memory and collective memory described by Halbwachs?


Hurricane-proof_building
List_of_Atlantic_hurricanes_before_1600
Hurricane_Juliette_(2001)
1898_Georgia_hurricane
Effects_of_Hurricane_Katrina_in_Mississippi
Hurricane_Olivia_(2018)
Effect_of_Hurricane_Katrina_on_the_Louisiana_Superdome
List_of_Pacific_hurricanes
List_of_retired_Pacific_hurricane_names
List_of_South_America_hurricanes
1924_Cuba_hurricane
Hurricane_Faith
Hurricane_preparedness
Hurricane_Vince
Hurricane_Arthur
Great_Colonial_Hurricane_of_1635
Hurricane_Felix
Hurricane_party
Hurricane_Helene_(1958)
1932_Cuba_hurricane
Hurricane_Emily_(2005)
List_of_Atlantic?Pacific_crossover_hurricanes
National_Weather_Service_bulletin_for_Hurricane_Katrina
Hurricane_Janet
Meteorological_history_of_Hurricane_Katrina
Hurricane_Weather_Research_and_Forecasting_Model
1979_Atlantic_hurricane_season
Hurricane_Carol
Hurricane_Dennis
Hurricane_Dora
1775_Newfoundland_hurricane
Hurricane_Flora
1971_Atlantic_hurricane_season
Hurricane_Hermine
Hurricane_Otto
Effects_of_Hurricane_Ike_in_inland_North_America
Hurricane_Ophelia_(2005)
Hurricane_Elena
List_of_Texas_hurricanes
Hurricane_Juan
Hurricane_Diane
List_of_storms_in_the_2005_Atlantic_hurricane_season
Hurricane_Igor
Hurricane_Alicia
Hurricane_Alley
Hurricane_Earl_(2010)
Effects_of_Hurricane_Ike_in_Texas
Hurricane_Beulah
2001_Atlantic_hurricane_season
List_of_North_Carolina_hurricanes_(1950?1979)
Lists_of_Atlantic_hurricanes
1985_Atlantic_hurricane_season
Hurricane_Florence
List_of_West_Africa_hurricanes
Hurricane_Luis
Hurricane_Audrey
NOAA_Hurricane_Hunters
Hurricane_Erin_(1995)
Hurricane_Joaquin
History_of_Atlantic_hurricane_warnings
List_of_Canada_hurricanes
Hurricane_Charley_(1986)
1988_Atlantic_hurricane_season
List_of_Bermuda_hurricanes
1998_Atlantic_hurricane_season
Hurricane_Diana_(1984)
List_of_North_Carolina_hurricanes_(2000?present)
1995_Atlantic_hurricane_season
Hurricane_Florence_(1953)
List_of_Delaware_hurricanes
Hurricane_Ginger
1989_Atlantic_hurricane_season
1997_Atlantic_hurricane_season
1961_Atlantic_hurricane_season
List_of_Pennsylvania_hurricanes
1996_Atlantic_hurricane_season
Effects_of_Hurricane_Andrew_in_Florida
List_of_North_Carolina_hurricanes_(1980?1999)
1954_Atlantic_hurricane_season
Hurricane_Ernesto_(2006)
Hurricane_Olga
1992_Atlantic_hurricane_season
Hurricane_Emily_(1993)
1993_Atlantic_hurricane_season
Hurricane_Gracie
Hurricane_Dennis_(1999)
Hurricane_Erin_(2001)
Hurricane_Florence_(2000)
Hurricane_Gaston_(2004)
Effects_of_Hurricane_Isabel_in_North_Carolina
Hurricane_Bertha_(1996)
Effects_of_Hurricane_Isabel_in_Maryland_and_Washington,_D.C.
Effects_of_Hurricane_Hazel_in_Canada
Hurricane_Florence_(1988)
1926_Miami_hurricane
Effects_of_Hurricane_Isabel_in_Virginia
Hurricane_Ivan_tornado_outbreak

(a) Hurricane Florence

Robert_O._Waring
United_States?Mexico?Canada_Agreement
Treaty_on_Open_Skies
Kiichi_Miyazawa
Francis_E._Meloy_Jr.
April_Glaspie
Manuel_Lujan_Jr.
Mack_McLarty
Marlin_Fitzwater
James_A._Baker_(born_1821)
Samuel_K._Skinner
List_of_oldest_living_state_leaders
Lauro_Cavazos
Palestinian_nationalism
Dick_Thornburgh
Donald_Trump_2000_presidential_campaign
C._Boyden_Gray
Lynn_Morley_Martin
Louis_Wade_Sullivan
Robert_Lighthizer
List_of_ambassadors_of_the_United_States_to_China
Bob_Martinez
Nicholas_F._Brady
New_world_order_(politics)
John_H._Hager
Lawrence_Eagleburger
Mikhail_Gorbachev
John_Prescott_Ellis
Carla_Anderson_Hills
Malta_Summit
Virginia_Knauer
George_H._W._Bush_1988_presidential_campaign
Chicken_Kiev_speech
George_Herbert_Walker_III
Electoral_history_of_Bob_Dole
Foreign_policy_of_the_George_H._W._Bush_administration
Irving_Kristol
James_Baker
List_of_people_pardoned_by_George_H._W._Bush
Robert_Mosbacher
1988_Republican_Party_presidential_primaries
Victims_of_Communism_Memorial_Foundation
Bobby_Koch
Ross_Perot_1992_presidential_campaign
Sara_Murray
Elizabeth_Dole
Samuel_P._Bush
Jeb_Bush_2016_presidential_campaign
Vernon_A._Walters
Director_of_Central_Intelligence
1996_Republican_Party_presidential_primaries
List_of_federal_judges_appointed_by_George_H._W._Bush
Jake_Tapper
1988_United_States_presidential_election
Russell_Trust_Association
Sheila_Graham
Alan_Simpson_(American_politician)
Dan_Quayle
Lloyd_Bentsen
Prescott_Bush
Matt_Schlapp
Bob_Dole
George_P._Bush
Electoral_history_of_George_H._W._Bush
Milward_Simpson
Michael_Dukakis
Marvin_Bush
National_Commission_on_Fiscal_Responsibility_and_Reform
Lee_Atwater
1992_Republican_National_Convention
Electoral_history_of_George_W._Bush
1996_United_States_presidential_election
1980_Republican_Party_presidential_primaries
Bill_Kristol
Walter_Mondale
Willie_Horton
AARP
John_Nance_Garner
Early_life_of_George_W._Bush
Keith_Sebelius
Harry_S._Truman
Manuel_Noriega
Bushism
Shirley_Chisholm
John_Quincy_Adams
Presidency_of_George_W._Bush
Bill_Archer
Calvin_Coolidge
Abraham_Ribicoff
Philip_Hart
Political_positions_of_George_W._Bush
Lyndon_B._Johnson
?vian_Conference
John_E._McLaughlin
Levi_P._Morton
Death_and_state_funeral_of_Richard_Nixon
David_K._E._Bruce
Michael_Powell_(lobbyist)
Benjamin_Harrison
Herbert_Hoover
Ralph_Yarborough
Robert_Bauer
Susan_Estrich
Chelsea_Clinton
Grover_Cleveland
Death_and_state_funeral_of_Gerald_Ford
Franklin_Pierce
Leslie_Coffelt
Warren_G._Harding
Porter_Goss
Craig_Roberts_Stapleton
Frederick_Douglass
William_McKinley
Michael_Reagan
Political_positions_of_Joe_Biden
State_funerals_in_the_United_States
Historical_rankings_of_presidents_of_the_United_States
Christopher_Nixon_Cox
Death_and_state_funeral_of_Ronald_Reagan
Ron_Reagan
List_of_vice_presidents_of_the_United_States_by_age
R._Nicholas_Burns
Laura_Loomer
List_of_presidents_of_the_United_States_by_age
Dan_Rostenkowski
Douglas_Brinkley
Kathleen_Blanco
Sam_Johnson
Claude_Pepper
Joe_Scarborough
List_of_multilingual_presidents_of_the_United_States
Brian_Mulroney
Jacob_Chestnut
Carlos_Curbelo
Theodore_Roosevelt_United_States_Courthouse
Speech_or_Debate_Clause
Military_funeral
United_States_Capitol_subway_system
Chaplain_of_the_United_States_Senate
Barry_Black

(b) George H. W. Bush death

Figure 4.7 – Collective memories extracted using the proposed pattern detection model. Top: Hurricane Florence (September 10-19, 2018). Bottom: death of George H.W. Bush (November 30, 2018). The nodes highlighted in red correspond to the core pages of the events. The 1-hop neighborhood of the core pages (articles that are directly linked on Wikipedia) is colored in blue. Collective memories related to the core events comprise all pages in the clusters.

When we apply our dynamic pattern detection algorithm to Wikipedia data, we can see that the web network self-organizes similar to a Hopfield network. Our approach learns collective attention memory patterns under the influence of visitors' activity similar to neurons in the brain. We can see the patterns that were detected during the experiments on Wikipedia data resemble collective memory, containing clusters of linked pages that have a semantically related meaning. The topic of a cluster corresponds to a real-world trending event that triggers the interest of Wikipedia visitors during a finite period.

These similarities enable us to interpret the detected patterns as clusters of collective memory concepts. Let us take a closer look at a few examples. Collective memories are usually triggered by traumatic events that involve human death. Multiple studies investigated the connection of collective attention to aviation accidents [26], natural disasters [27], terrorist attacks [28], [29], and armed conflicts [126]. In these studies, the authors researched various aspects of the online social media ecosystem that result in an increased interest in traumatic events among internet users.

Previously, we saw examples of such collective memories in Fig. 4.5 and Table 4.1. These memories were extracted by the proposed pattern detection approach. Charlie Hebdo attack reminded people about terrorism and other attacks involving mass shootings. The Germanwings crash triggered memories about similar airplane accidents caused by suicide by the pilot. Collective memories related to the Ferguson unrest were focused on previous victims of discrimination by the police.

Fig. 4.7 illustrates two more recent examples of collective memory. First, a natural disaster, Hurricane Florence, occurred in September 2018. It triggered memories about the previous hurricanes and their consequences. Another set of collective memories was triggered by the death of George H. W. Bush. The detected cluster mostly contains Wikipedia articles describing the history of his political career and biographies of the related political figures.

In both cases, we observe a core page that starts a pattern and its multihop neighborhood that specifies the pattern's context. The availability of this information opens interesting opportunities for interpretation of the detected collective memories. We can analyze the collective associations of the readers and get a better understanding of their perception of occurring events. These insights could be used to improve the quality of controversial content and to avoid polarization of the readers. We could use the results to suggest editors what links should be added and where to reduce polarization by giving a more balanced overview of the topic and by presenting a topic matter from different points of view.

These and many other similar examples[I] demonstrate an interesting connection of our experiments on Wikipedia data to humanities. Using the proposed anomaly detection model, sociologists and historians could study the perception of web content by a large and diverse population of readers. The model can be used to study the collective behavior of readers,

[I] Interactive version of the results is available on https://wiki-insights.epfl.ch/wikitrends/

including their motivation, historical references associated to current events, and other factors affecting the attention and opinion of the public.

4.5 Discussion

Results of the experiments discussed in this section demonstrate the effectiveness and scalability of our pattern detection model. Particularly, experiments on Wikipedia web network show that we can scale our model to tens of millions of nodes and an order of a billion edges on a fairly small server with 32GB of RAM and a 12-core CPU. We will provide a distributed implementation and technical details of the model in Chapter 6.

We verified the quality of detection using Google Trends as a reference for qualitative analysis. In the case of the Germanwings 9525 airplane crash, our model provided additional information detecting another airplane crash that happened earlier. We extracted that additional information from the attention flow of the model, which was illustrated in more detail in the Super Bowl example (Fig. 4.4). The dynamics of the network attributes forms an attention cluster, which reflects the increasing anomalous dynamics in a localized part of the network. The attention clusters can be further used for a better and more detailed interpretation of the model.

It should be noted that despite being very popular, Google Trends is not an exact reference in terms of trend detection quality. It has recently been shown that, in some cases, Google Trends may have inconsistencies in the results [127]. In this qualitative study, the authors demonstrate that the same queries may return different results when submitted at different times, especially for short-term trends (less than 8 months). However, the same study shows that queries requesting trends over longer periods of time (5 years with monthly resolution) provide qualitatively accurate and consistent results. Nonetheless, possible inconsistencies should be taken into account when evaluating the results of our approach and Google Trends should not be considered as an exact standard for evaluation despite being a good overall reference.

In Section 4.4, the memory properties of the model allowed us to see one more application of the proposed anomaly detection approach. We saw how it can be used in the digital humanities to study the phenomenon of collective memory at a larger scale. Furthermore, the memory properties of the model are useful when we need to recover missing information or provide recommendations based on incomplete viewership patterns. This property can be used in various applications related to information retrieval and recommendation systems.

Nonetheless, the interpretability of the model can be improved. Depending on the type of node attributes, we can introduce automated interpretation modules tailored to specific tasks, such as natural language processing (NLP) or image processing. For instance, there are a lot of applications in which nodes, apart from time-series, also have textual attributes. Such applications require an NLP-powered interpretation model in order to identify topics in the pattern-related subnetworks. If nodes contain images (e.g., Instagram network of followers), the classification of image attributes of the nodes would give us additional insights about the types of detected patterns.

In the examples we have presented in this section, Wikipedia articles also contain article text and summaries. Looking back at the previous section, the Enron dataset has a corpus of emails sent from each email address. Considering this additional textual attributes, to automate the interpretability of our method, we need a topic detection module that would give a summary or a topic describing the detected pattern. We develop such a module and apply it to the Wikipedia dataset to study the viewership interests of the encyclopedia's readers across multiple languages. We will focus on this topic detection module in the next chapter.

5 Pattern labeling and classification

As we saw in the previous chapter, dynamic patterns emerge as a result of the characteristic dynamics in the attributes of the graph, which is defined by the explicit attention function. In Section 4.2, even though we managed to detect viewership patterns, we did not have a fully automated way to interpret and explain them. Moreover, we noticed that the subjective aspect of pattern interpretation may lead to biases introduced by a practitioner performing interpretation. In this chapter, to make sure that the interpretation of the results is objective and unbiased, we develop an extension module for *automated interpretation* of the detected patterns.

Here, by an *interpretation module,* we understand a machine learning model that allows us to generate a short summary describing the detected patterns based on the node attributes. Such a model can be used by experts and practitioners for further analysis of detected patterns. The interpretation module is an abstract concept and can be designed based on the available types of attributes. In the applications presented in this chapter, we focus on the textual attributes of the nodes and develop our interpretation module accordingly. This extension module uses textual data extracted from the patterns for topic modeling, labeling, and classification, which allows us to assign topics to detected patterns and classify them into general categories.

In the experiments on Wikipedia data that we saw in the previous chapter, the patterns reflected the increased interest of Wikipedia readers in particular topics. In combination with our pattern detection algorithm, the model presented in this chapter allows identifying trending topics on Wikipedia and obtaining a global overview of the detected trends.

As of September 2020, Wikipedia is available in 313 languages. This feature of the encyclopedia makes this experiment more insightful and to demonstrate an application of our algorithm in social sciences. The combination of our pattern detection approach and the interpretation module presented in this chapter allows us to study collective behavior of Wikipedia readers at a large scale.

The experiments presented in this chapter are not only demonstrative from the dynamic pattern detection side. Running experiments on Wikipedia data is also insightful from the sociological point of view because of its scale and the diversity of its readers. It covers a large multicultural population of readers with diverse interests. What makes this dataset even more interesting is that Wikipedia readers have different motivations [68]. Some readers look for an up-to-date source of information related to an event that appeared in the news, while others are interested in solving work or school-related tasks. Some readers are satisfied with a quick overview of the topic, while others strive for an in-depth understanding of all related facts.

Wikipedia gathered a lot of different people in a global and unified information medium. Along with different motivations, Wikipedia readers also have various backgrounds, hobbies, religious affiliations, political views, and speak different, often multiple, languages. Such diversity makes Wikipedia an open window to cultural differences across different languages and populations. Analysing Wikipedia's viewership statistics could help identify the collective interests of socially diverse communities of people speaking different languages and study them over chosen periods of time. In this section, we use the proposed pattern interpretation approach to accomplish this task.

One of the challenges with Wikipedia is that the structure and content of the online encyclopedia differ across languages. Some editions are more developed than others. Regarding the coverage of certain topics, the level of detail and content vary largely from one language to another. This difference is especially apparent when we read controversial articles related to culture, politics, or history, as it is shown in [128] for the web in general and in [129, 130] for Wikipedia in particular. For instance, the Italian version of the article about Leonardo da Vinci is much more detailed than its English counterpart. The content difference is also noticeable in fairly noncontroversial topics. For example, the article Cat in Spanish focuses on the animal's diseases and covers this aspect in greater detail than the same article written in English. In addition to content discrepancies, the hyperlink structure of the same article also varies across languages [129]. Such differences may trigger diverging associations depending on the language in which one reads the very same article. Since our pattern detection approach takes into account both the temporal and structural aspects of Wikipedia, it is effective in capturing such differences despite the structural discrepancy in the organization of the encyclopedia across languages.

We use our algorithms to detect viewership patterns in multiple language editions of Wikipedia and transform them into trending topics using the pattern labeling model presented in this chapter. Furthermore, the model allows us to analyze and compare trends across different language editions of Wikipedia. We run the labeling experiments on the patterns detected in the web network of Wikipedia and the viewership activity of the readers across English, Russian, and French editions.

This chapter is structured as follows. In Section 5.1, we present the architecture of the automated pattern classification model that defines topics based on textual attributes of the nodes in the detected patterns. To develop the model, we use multiple approaches for topic modeling and classification. Then, we run experiments on Wikipedia data and evaluate our approach. To extract patterns from each language edition, we perform an experiment similar to the one described in Section 4.2, but this time, we focus on a different period. We use our pattern detection algorithm to extract patterns from Wikipedia viewership over the period from September to December 2018. After that, we label the detected viewership patterns with topics and compare the trending topics distribution across English, Russian, and French language editions. Finally, we use the developed interpretation model in another case study presented in Section 5.3. We analyze changes in the interests of Wikipedia readers during the COVID-19 pandemic, extending the study to seven language editions, including English, French, Russian, Spanish, German, Chinese, and Italian. We conclude this chapter with a discussion, analyzing the performance of the topic detection and classification model and comparing our results to the collective behavior insights reported in other related studies.

5.1 Proposed pattern labeling and classification approach

To automate the interpretation of the detected dynamic patterns in networks with textual attributes, we develop a topic detection and classification model that extends our pattern detection approach. Our extension model consists of the following steps:

1. Dynamic pattern detection (using the algorithm presented in Chapter 3).
2. Topic modeling based on the textual node attributes.
3. Training classification model and automated labeling.

Step (1): Dynamic pattern detection. Before building a topic model and training a classifier, we need to extract dynamic patterns from a given graph. To extract patterns represented as subnetworks of nodes with textual attributes, we use the pattern detection algorithm presented in Chapter 3. The algorithm keeps only the nodes whose dynamic attributes conform to the explicit attention function. Besides, the algorithm sets a weight on each edge, reflecting the dynamic correlation of dynamic attributes of linked nodes. The weight of the edge between two nodes increases when both attributes encounter a correlated anomalous activity spike, reflecting the strength of the correlation. As described in the previous chapter (Sec. 4.2.2), this way, the algorithm creates attention clusters in an unsupervised way. Once the subnetwork of patterns is detected, we extract clusters of densely connected nodes using a community detection algorithm [108].

Step (2): Topic modeling. In our case, topic modeling is nontrivial since we may have a set of semantically incoherent textual attributes in one pattern. For example, in the experiments on Wikipedia data (Chapter 4), a cluster related to a terrorist attack can comprise pages related to geographic landmarks, politicians, ideology, religion, or other attacks that people read to complement information. To mitigate this issue, in addition to the text, we incorporate graph attributes into the topic model to improve its performance. Then, we use topic modeling and semisupervised learning approaches to define and assign high-level topics to every node in the extracted pattern.

We tested 2 methods for topic modeling and keyword extraction, Latent Dirichlet Allocation (LDA) [131] and Term Frequency – Inverse Document Frequency (TF-IDF) [132]. We compared both methods qualitatively using multiple test cases and found that they yield similar results.

For both methods, to improve the performance of the topic model and to solve the problem of semantically incoherent textual attributes, we propose a graph-based alternative where we incorporate features extracted from the graph structure into the model. Since we deal with semantically incoherent textual attributes, multiple experiments showed that plain LDA and TF-IDF models demonstrate poorer performance and worse quality of topics compared to our graph-based alternative.

To train the graph-based LDA model, we preprocess a group of textual node attributes and create a document term matrix. Then we feed it into the LDA model; one document is a concatenation of all textual attributes in a given pattern. After that, we perform the training in three steps. First, with the entire text, second, with nouns only, and third, with nouns and adjectives. To look at the terms that belong to one part of speech (nouns or adjectives), we use part-of-speech tags from the Penn Treebank Project [133].

Concerning the graph-based TF-IDF model, we extract the top k descriptive words from each pattern using their TF-IDF scores. Then, we modify the original TF-IDF method as follows. When computing TF-IDF coefficients, we use the graph structure of each pattern to extract the degrees of the nodes. This gives us more context about the pattern. Since the degree of a node is the number of edges connected to it, high degree nodes are more important in the semantic sense and give more context related to the topic of the pattern. We compute the degree n of every node within the subnetwork and multiply the counts of every word in this document by n. That way, we give more importance to the words in the textual attributes of those nodes that have a higher degree by increasing the TF value while keeping the same IDF value.

Step (3): Labeling and training. The goal of this step is to label patterns using the textual attributes of the nodes. To do that, we represent every pattern as a concatenation of textual attributes of the nodes that belong to the pattern. Hence, every data item is a text document that represents detected patterns. Unless we have labels to train the classification model, we select a small set of keywords, that we have extracted from each pattern in Step 2, for labeling. From the keywords, we select the most general and descriptive ones and based on these keywords, we define the number of labels for classes. The number of classes should be chosen based on the dataset. We then proceed in a semisupervised fashion. First, we manually label a small subset (50 samples per class) of patterns based on the extracted keywords from each pattern. We use these labels as the ground truth. Once this is done, we train a model to classify the documents that represent our patterns and use this model to label the rest of the unlabeled documents. We use a transformer model, although in this case, other classifiers also perform well (for instance, we also tried SVM, which demonstrated good performance).

Note that pattern labeling process is language dependent. When we need to classify patterns that have textual attributes written in multiple languages, the process described in the previous paragraph should be performed for all languages independently. However, to do this efficiently, we use regular expressions constructed for multiple languages, similarly to the unsupervised labeling approach presented in [126].

To classify unlabeled nodes, we use a pretrained uncased BERT model [134], with 12 hidden layers and a hidden size of 768 (i.e., the last hidden layer generates a 768-dimensional vector for every word). BERT also stacks multiple attention layers. Attention heads enable the model to capture the relationships between words, giving more weight to some words compared to others. This model is trained on two main tasks. First, *Masked Language Modeling* (MLM), where 15% of the word tokens are masked, and BERT is trained to predict the correct word. Second, *Next Sentence Prediction* (NSP), where the model is fed two sequences A and B and is asked to predict whether B is the next sentence after A. We use the pretrained model as a feature extractor for the textual attributes of the nodes. To complete the model, on top of it, we add a classifier that consists of five fully connected layers, taking the features extracted by BERT as input. We performed this process for multiple languages independently.

5.2 Wikipedia viewership pattern labeling and classification

In this section, we use our automated pattern labeling and classification model to:

- extract the most popular topics during a chosen period (trends)
- label each trend according to the summary of Wikipedia pages related to it.

To extract the clusters of the most trending pages, we rely on the method described in Section 3.1. Using this method, we obtain well-separated clusters of Wikipedia pages and their viewership activity over time. The developed pattern labeling model presented in the previous section is language-independent and relies solely on Wikipedia pageview statistics and the graph of hyperlinks. To test the model, we identify and compare the most popular topics among readers of three Wikipedia language editions, English, French, and Russian, over the last 4 months of 2018. We provide the analysis of the detected trends, reporting similarities and differences across three language editions of Wikipedia.

5.2.1 Experimental setup

To extract patterns represented as subnetworks of trending Wikipedia articles, we use the same pipeline as in Chapter 4 and apply our pattern detection approach. The algorithm keeps only the pages that encounter surges of user interest over time and their connections. A viewership pattern corresponds to a spike in the activity of a page. The desired magnitude of a spike is controlled by a sensitivity parameter in the explicit attention function.

Wikipedia categories are not sufficient to define a high-level topic for every cluster because they are too specific. We use the summaries of Wikipedia articles as textual attributes of the patterns to build a topic model and to train our classification model. We collect summaries of every article using the Wikipedia API. We also compare our results to an alternative approach, where instead of summaries, the model is trained on Wikidata items associated to Wikipedia pages.

To compare multiple language editions, we select a small set of keywords and their translations for labeling. In the beginning, we started with 27 specific topic labels. However, the distribution of the number of samples per class was skewed, which made the classification problem more complicated. As a workaround, we define more general topics and limit the number of high-level topics to eight. The labels are *football, sports (other than football), politics, movies, music, conflicts, religion, science, and videogames.* Note that the label "conflicts" is more general than military conflicts. We use it to describe a broad range of traumatic events that result in numerous fatalities. Therefore, we also included natural disasters, mass shootings, terrorist attacks, and airplane crashes into this category.

We then proceed in a semisupervised manner. Some of the pages are easier to label and we design a few rules to label them automatically. We take advantage of the fact that some articles are homonyms and, to prevent ambiguity, their title contains some useful information in parentheses such as "album", "actor", or "footballer". For example, all articles that have a title with this pattern "xxx (album)" will be classified as "music". For the remaining unlabeled articles, we use the keywords that we have extracted during the previous step. We define multiple sets of words, where each set corresponds to a label. As an example, the keywords "political", "party", and "republican", represent the topic "politics". The labeling loops through all the summaries of Wikipedia articles and checks if all these keywords are present in an article's extract. If the condition holds, it labels the article as "politics". It does the same for all other sets of keywords that represent different labels.

We have also tested Wikidata as an alternative data source for automated labeling instead of Wikipedia article summaries. Wikidata is a structured knowledge base focused on items that represent high-level topics, concepts, and categories. It demonstrated comparable performance and produced relevant keywords allowing us to infer high-level topics for the clusters by collecting properties for each page and extracting the topic from the resulting high-level summary with LDA. Due to the structure of the knowledge base, there is a major positive aspect of using Wikidata for labeling and topic detection purposes. Wikidata properties and concepts are identified by a unique code, with direct relationships to their equivalents in all Wikipedia languages. Therefore, there is no need to deal with different languages when running the topic detection pipeline. In addition, this reduces the need for experts in every studied language to assess the quality of the topic detection, hence scaling the study to more languages would become much easier. The only drawback is that querying the Wikidata API is relatively slow, and when it comes to collecting data for thousands of pages, this step quickly becomes a bottleneck.[I] Wikidata word extraction gave promising results with relevant keywords for the clusters.

Recently, scientists from Wikimedia Research released a Wikidata-based model for topic classification [II]. The authors trained the FastText document classification model [135] using Wikidata items as labels. The advantage of using Wikidata items is that there is no need for feature extraction and language-specific variants of the classification model. This eliminates the LDA stage from the topic detection pipeline. Each Wikidata item has properties, which are used to create a bag-of-words representation of each Wikidata item. The model predicts topics for Wikidata items, not for Wikipedia articles as in our BERT-based model, which we trained on summaries of Wikipedia articles. Most of the Wikipedia articles in the subnetwork of the extracted patterns have a Wikidata Qid, which enables us to match Wikidata items with corresponding Wikipedia articles. Therefore, the two models are compatible and can be used interchangeably. We use the FastText model for evaluation and to compare the performance of both approaches.

In the following subsections, we use both models, BERT-based and FastText-based, to identify the topics of articles in the extracted clusters of trending pages. First, in Section 5.2.2, we analyze the results of the BERT-based model, trained on Wikipedia article summaries using the proposed pattern interpretation approach. And second, in Section 5.2.3, we compare these results to the FastText-based model trained on Wikidata items.

5.2.2 BERT-based pattern labeling. Trend distribution analysis

The BERT-based approach allows labeling entire clusters of Wikipedia articles with a single topic. It uses the structure of pattern-clusters and topological properties of the graph, such as degree, betweenness centrality, and PageRank of nodes in the clusters. To train the classification model, we use summaries of Wikipedia articles that appear in the detected patterns.

To illustrate all aspects of the method, we detect viewership patterns on Wikipedia over a 4 month period (16 August to 31 December 2018). We first review multiple examples of trends and analyse the structure of the patterns. Then, we present general trending topic distributions that highlight similarities and differences across English, French, and Russian language editions of Wikipedia.

[I] To test this data source, we deployed the entire Wikidata inside a local MongoDB instance. As of January 2020, the size of the deployed Wikidata knowledge base is around 220 GB.

[II] https://meta.wikimedia.org/wiki/Research:Language-Agnostic_Topic_Classification

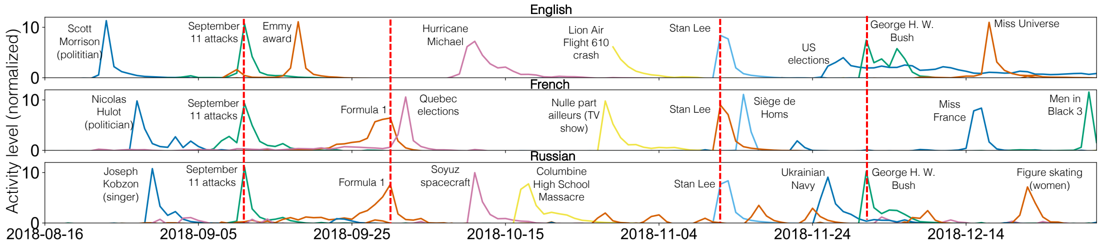


Figure 5.1 – Trends over time and across languages. Each colored curve, together with a short description, is associated to a trend. They represent the number of visits on the pages belonging to the trend over time. The associated keyword is the title of the most central page of the trend. Red dashed lines highlight the moments when readers' interests in multiple language editions coincide.

Overview of the detected trends. In Fig. 5.1, we show the most popular trends for the three languages over the last four months of 2018. Note that we extracted more trends during step (1), but we limited them to the most popular ones to get an uncluttered picture. Each colored curve represents the evolution of the popularity of a trend over time. We annotate each popularity peak with the title of the most central page of the trend. More precisely, for each cluster (obtained at step (1) of the processing), we select a page having the largest page rank [136]. Vertical red dashed lines highlight particular dates when some trends emerge synchronously across languages, for example, the 09/11 commemoration or the death of Stan Lee in November.

We can see some trends in 2 of the 3 languages, such as Formula 1 in French and Russian or the death of George H. W. Bush, in November, in Russia and the US. Some trends are only seen in one of the languages and are related to cultural or local events such as the death of a popular Russian singer Joseph Kobzon in September, French activist and politician Nicolas Hulot publishing a book, the death of the former presenter of a popular TV show in France "Nulle Part Ailleurs" or the Miss France contest.

The structure of the detected patterns provides insights into users' curiosity and their eager willingness to explore topics. For Hurricane Michael, for example, readers did not only visit the page of the hurricane that was written and edited live, they also followed hyperlinks to the pages of the affected cities, previous hurricanes, or similar natural disasters. Readers were also interested in pages defining a cyclone or providing a scientific explanation of this phenomenon. More than a hundred different pages underwent a burst of visits during this event. We have discussed and illustrated this feature in the previous chapter in Sections 4.2.2 and 4.4, where we showed similar effects in the clusters related to Hurricane Florence, George H. W. Bush, Charlie Hebdo shooting, Ferguson unrest, and Germanwings airplain crash.

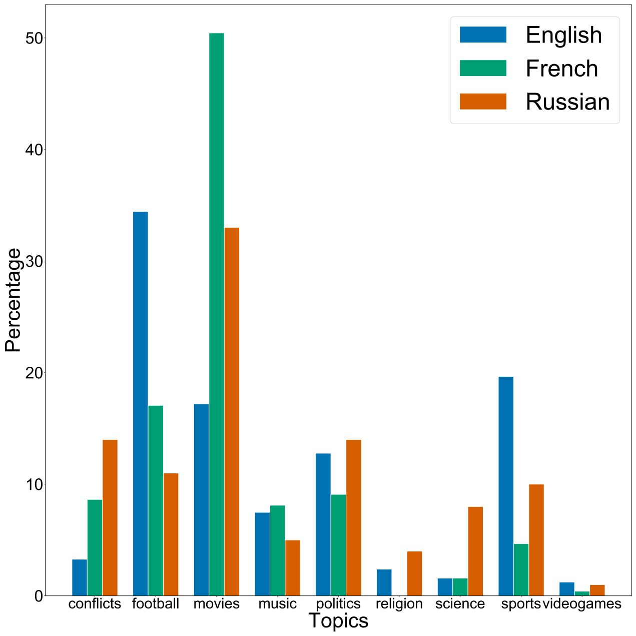


Figure 5.2 – Distribution of attention to the most popular topics across languages based on the model trained on Wikipedia summaries (16 August – 31 December 2018)

In addition to the structure, the size of the pattern-clusters can also reveal cultural particularities that characterize different language editions. For instance, the spike of activity on the page dedicated to Stan Lee was very high in the three languages shortly after his death. However, the cluster size was much smaller in the French and Russian editions. English speakers extensively explored his work through hyperlinks referring to pages about comics, movies based on them, and starring movie actors. Most of the francophone readers focused only on his biography, despite the existence of many hyperlinks pointing towards the most popular heroes of his comics or adapted movies mentioned on his French Wikipedia page. On the French Wikipedia cluster, apart from Stan Lee, only the pages of Magneto, Jack Kirby, Steve Ditko, Larry Lieber, POW! entertainment, CNN, and the expression "Excelsior" are present in the cluster.

Lastly, all trending topics have been covered by the media, showing their massive influence on Wikipedia readers. Most of the viewership anomalies have a similar duration of a few days, with a similar peak shape across all languages. This observation indicates that the lifespan of a trend is independent of the culture or language.

Global statistics over 4 months. Figure 5.2 shows the distribution of the most popular topics across the different editions of Wikipedia over the period of the last four months of 2018. We can see that while English-speaking readers prefer topics related to football and general sports, the most popular content among French and Russian-speaking visitors is mostly related to the movie and TV industry. The other topics related to entertainment, music, and video games, get almost equal attention in all languages.

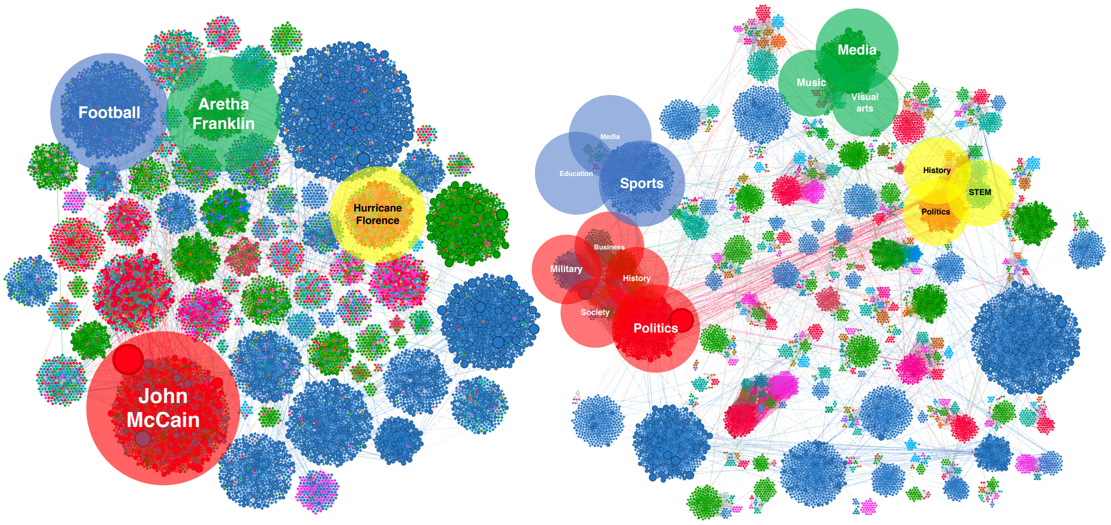


Figure 5.3 – Clusters of trending Wikipedia articles with topics assigned by the classification model trained on Wikidata items (16-31 August 2018). Node colors correspond to different topics. Each cluster contains pages belonging to various topics. Left: a graph with clusters of pages related to trending subjects. Right: the same graph with articles grouped by topic in each trending cluster.

Politics is equally popular in the three editions of Wikipedia, while the content about geopolitical issues (natural disasters, foreign affairs, or conflicts) appears to draw more interest in French and even more in Russian editions. Science is among the most popular topics among Russian-speaking readers while being equally low in French and English. French-speaking readers have the lowest interest in religion, in contrast to English and Russian ones. English-speaking readers appear to be more interested in religion than in science.

We focus on a more general overview of the trending topics and compare our findings to the results reported by Lemmerich et al. [76] in the Discussion section (Sec. 5.4).

5.2.3 FastText-based pattern labeling. Trend distribution analysis

Contrary to the experiments described in the previous section where we trained a BERT-based classifier on the summaries of Wikipedia articles, in this section, we use the FastText model that was trained on Wikidata items. Another difference is that we predict the topics for each page separately and not for the entire clusters as we did it in the previous section. As a result, after labeling, we obtain multitopic patterns (see Fig. 5.3), which provide a more detailed picture of the events. The benefit of this approach is that we do not need to handpick topics.

We can use all topics extracted from taxonomies that are available for all Wikipedia projects. The full list of topics with precision and recall indicators is available on the project's website [III].

Another advantage of the Wikidata-based model is that it works with all languages available on Wikipedia, which makes the model almost fully automated. We do not need to predefine classes and to fine-tune keyword extraction models for multiple languages.

Comparing the results to the summary-based model. Since we run the experiments on the same subset of Wikipedia data, the events are the same as in the summary-based model. Therefore, we further focus only on the general statistics of the topic distribution over four months. Figure 5.4 illustrates the 11 most popular topics: *STEM, Sports, Politics and Government, Media, Music, History, Military and Warfare, Films, Visual Arts, Philosophy and Religion, and Society.* As we can see, the set of topics is slightly different from the one we had in the summary-based model. However, most of the topics are semantically similar, which enables us to compare the results.

To start with, let us focus on the similarities of the detection between two models. The results of the topic detection by the two classification models are very similar. We can see that the result of the topic detection is the same for the following topics: Sports, Conflicts (Military and Warfare), and Science (STEM). Nonetheless, there are a few topics where the trend distributions are different. Religion (Philosophy and Religion) is almost the same with only one disparity. We did not detect this topic in the French language edition using the BERT-based model. Furthermore, note that the results of topic detection are divergent for Movies (Films), Music, Politics (Politics and Government). We can explain it by the presence of more general categories such as Media, which can be related to both Music and Movies, and Society, which is often involved in clusters created by political events. If we take into account these general topics, the results of both models are nearly the same.

As we can see in Fig. 5.3, when we use the Wikidata-based approach for topic classification, we obtain heterogeneous clusters. Hence, each cluster corresponds to one trend, which covers multiple topics. These topics are implicitly related to the trending events that triggered the creation of each cluster.

Let us look at several examples in more detail. The majority of pages in Sports-related clusters (highlighted in blue) are labeled as Sports. However, depending on the nature of sports, we can also see smaller subclusters of pages related to other topics, such as Media, Education, Healthcare, and Engineering inside of the main cluster. Needless to say that media and education are inalienable parts of many popular sports these days. We can also see that trends related to politics (highlighted in red) are also very diverse. If we look at the cluster that emerged after the death of John McCain, we can see that the majority of the pages, indeed, belong to politics. What is interesting is that we can also see fairly big subclusters that comprise articles on topics such as Society, Military and Warfare, History, and Business.

[III] https://meta.wikimedia.org/wiki/Research:Language-Agnostic_Topic_Classification/Wikidata_model_performance

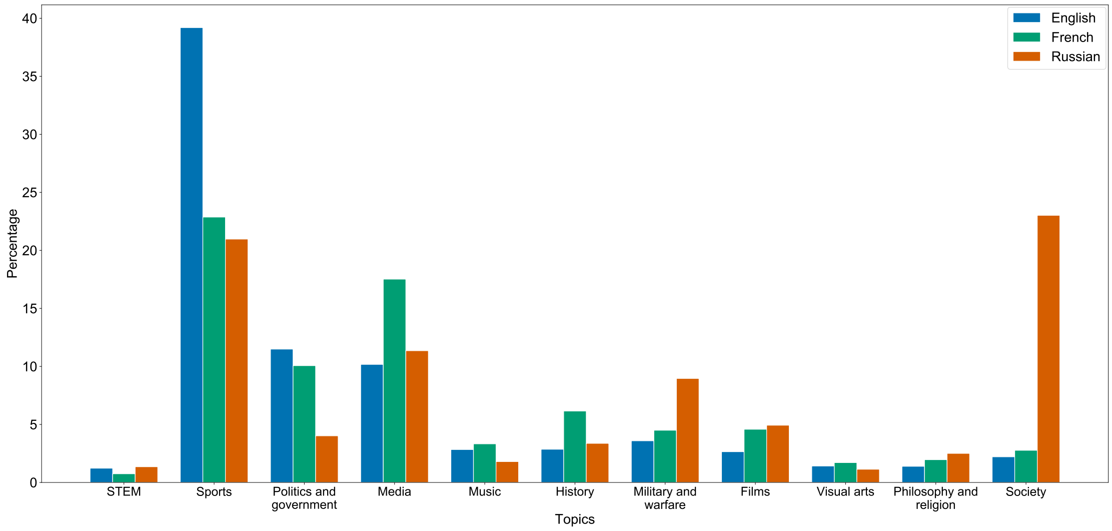


Figure 5.4 – Distribution of attention to the most popular topics across languages based on the classification model trained on Wikidata items (16 August – 31 December 2018)

All these areas are inevitably involved in political careers and processes. Clusters created as a result of the readers' interest in natural disasters (yellow) are also heterogeneous. In the cluster that emerged after Hurricane Florence, we can see that it is composed of pages that cover a wide range of topics including STEM (mostly articles on Earth&Environment), Politics (articles about politicians involved in solving the crisis), and History (articles related to previous disasters and their causalities). Finally, art-related topics (green) are also diverse. We can see that the cluster that emerged due to the anniversary of Aretha Franklin's death includes articles on different topics associated with various artistries, such as Media, Music, and Visual Arts.

To understand better the heterogeneity of the clusters classified with the Wikidata-based approach, let us look at Fig. 5.5, which illustrates more concrete examples. We can see that clusters related to sports, such as NCAA (American football), US Open (Tennis), and Belgian Grand Prix (Auto racing, Formula 1), have the least diverse set of topics among others. Naturally, most of the pages are classified as Sports. Media, Business, and Politics are also among the common topics in the sports-related clusters. However, it is interesting to see that the secondary topics in these clusters reflect specific features of different sports. Engineering advancements in the automotive industry play a significant role in Formula 1, which is reflected by the presence of topics Transportation and Engineering in its cluster. NCAA is an organization that regulates student athletes from North American institutions, so we can see Education among the most represented secondary topics in that cluster. We can also notice a similar effect in the clusters related to politics and show business where secondary topics give us a wider perspective on the specific nature of the occurred events.

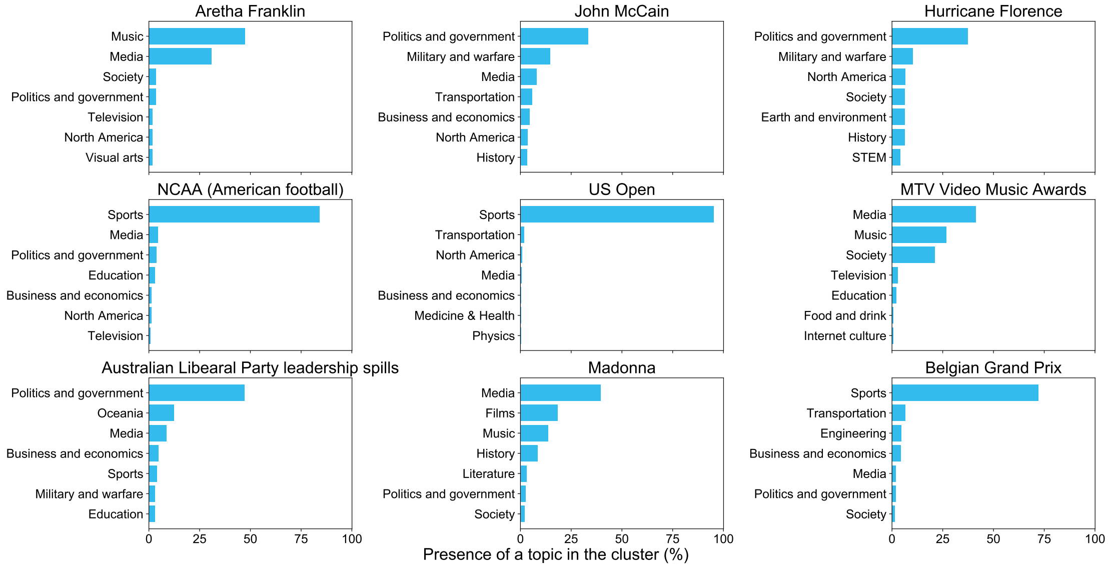


Figure 5.5 – Distribution of topics in the most popular trends of August 2018 (English Wikipedia). The topics were classified using the Wikidata-based model. All trends comprise a heterogeneous set of topics, providing a diverse perspective on the specific nature of the emerged trends.

5.3 Changing interests of Wikipedia readers. COVID-19 case study

In this section, we apply the developed topic classification pipeline to analyze the evolution of the changing interests of Wikipedia readers during an outstanding global event, the COVID-19 pandemic.

A few months before I started writing this thesis, the COVID-19 pandemic had unfolded. The pandemic started in China at the end of 2019 and very rapidly spread around the world. As of 24 November 2020, more than 59 million cases of COVID-19 have been reported in more than 188 countries[IV]. At the beginning of the pandemic, strict confinement measures were introduced, first in China, then in several other countries in Asia, and finally, across Europe and in other countries around the world. These measures globally affected the world, leading to dramatic changes in mobility patterns among many others. Following these restrictions, the interests of Wikipedia readers have also changed [137]. That change inspired us to run another experiment to investigate the evolution of trending topics throughout the pandemics in different language editions of Wikipedia. We focus on the data over the period from December 2019 until May 2020.

[IV] https://en.wikipedia.org/wiki/COVID-19_pandemic

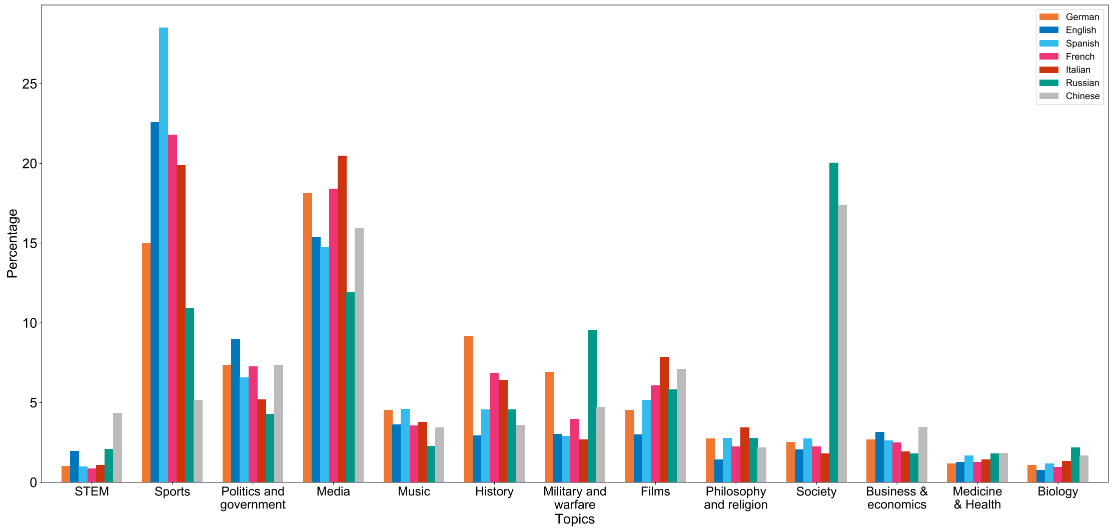


Figure 5.6 – Distribution of attention to the most popular topics across languages during the first 4 months of the COVID-19 pandemic (17 December 2019 – 30 April 2020).

Wikidata-based topic classification model facilitates extension of our study to more languages. We study 7 language editions of Wikipedia, English, German, French, Italian, Chinese[V], Russian, and Spanish. We use the same approach as in the previous experiments using the FastText-based model trained on Wikidata topics.

To start with, let us focus on the general trends over the period (see Fig. 5.6). We can see that the distribution of trending topics looks similar to what we saw in the previous experiments. Sports and Media are the leading trends, followed by Music, Films, and Politics&Government. We also noticed the emergence of two new clusters that were not captured before, namely, Medicine&Health, and Biology, which was triggered by the increased interest of the readers in the articles related to viruses, influenza, and the COVID-19 pandemic itself. The popularity of Society is 20-25% higher in Chinese and Russian editions compared to other languages that we analyzed. After a qualitative analysis of the classification results, we have discovered that there is a significant overlap between the topics Politics and Society in the Chinese and Russian editions. We found that Wikidata items, related to local political figures and elections in regions where the majority of people speak Russian and Chinese, often belong to the topic Society. All in all, analyzing the Wikidata-based classifier, we came to the conclusion that the training data should be refined for items related to Russian and Chinese History and Society to avoid confusion in the classification results[VI].

[V] Note that Wikipedia has been banned in China since 23 April 2019

[VI] Communicated with one of the authors of the model, Isaac Johnson

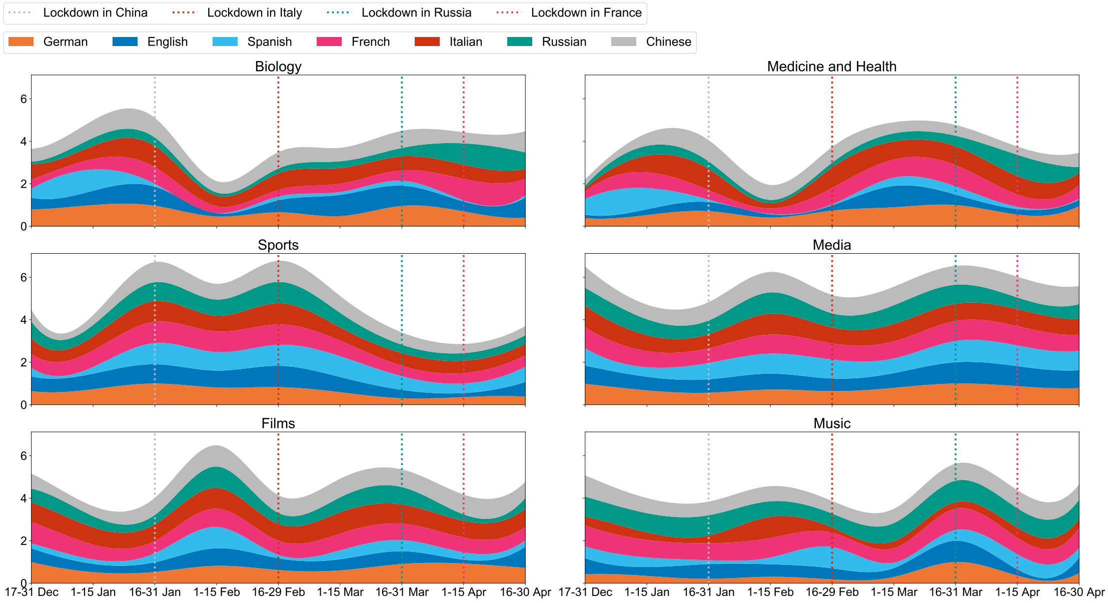


Figure 5.7 – Evolution of trends across languages during the first 4 months of the COVID-19 pandemic (17 December 2019 – 30 April 2020). Each data point is an aggregation of trends over a two-week period. Trends are normalized between 0 and 1 for each language. Thicker lines correspond to a rising popularity of the topic, while thinner lines reflect diminishing trends. The vertical lines indicate the beginning of lockdowns in different countries; right to left: China (grey), Italy(red), Russia(green), France(pink).

Fig. 5.7 provides a dynamic picture of changing trends. To capture the dynamics, we aggregated trending topics bi-weekly; each data point represents the popularity of a topic during a selected two-week period. In this study, we focus on the short-term dynamics, which reflects change points in the trends in a moving time window. This allows us to get a live picture of how users shift their attention from one topic to another. In the stacked chart below, you can see a dynamic view of the changing popularity of some of the most popular topics. We normalized the popularity of each topic in each language between 0 and 1. The more drastic the attention shift, the thicker the line on the plot. Four vertical lines correspond to the beginning of lockdowns in different countries.

We can see that the COVID-19-related topics, such as *Biology* and *Medicine&Health*, have an attention spike in January. Then, after a short-term drop, these topics develop a steady momentum starting from February. In Chinese Wikipedia, we observe the most significant increase in attention to these topics in January. The interest in the topics remains consistent throughout the entire period and only slightly diminishes in April. Looking at other language editions, we observe that at the beginning of February, the interest of the readers to Biology and Medicine&Health drops. However, soon after that, the topic Biology regains popularity among Italian-speaking readers, followed by English- and German-speaking audiences. Attention to Medicine&Health also bounces back, first in Italian and French editions, and then in German and English. Russian-speaking readers develop an interest in both topics closer to the end of March. All in all, most of these observations reflect the COVID-19 development timeline in the locations where these languages are spoken primarily, however, it is still hard to align geographically the results for English, French, and Spanish language editions because of their global adoption in different regions of the world.

Sports is the most popular topic in all languages at the beginning of the pandemic. However, we can notice an abrupt change of attention levels across all languages. The readers become indifferent to this topic starting from March. One of the possible explanations is that the pandemic resulted in the cancellation of the majority of sports events around the world.

Attention to *Media, Films,* and *Music* is mostly uniform across all languages during the pandemic. The spike in the topic Films can be explained by the worldwide popularity 92^{nd} Academy Awards ceremony, which occurred on February 9. When we look at the topic of Music, we can see slight shifts towards indifference among Italian-speaking readers, which happens in the second half of February. This can be attributed to the strict lock-down measures that were introduced on March 9, however, this is just an observational hypothesis.

Finally, let us compare our results to the ones reported in [137]. The main difference between the two approaches is the comparison strategy. In our approach, we focused on live short-term attention shifts or change points, while the authors of the study compared attention levels to the previous year, reporting long-term changes. Even though we used a different approach, we can see that our findings confirm some of the observations reported in [137]. During the first months of the pandemic, articles on Biology and Medicine&Health gained a lot of interest from readers across all language editions, while Sports-related articles lost a significant share of their audience. Nonetheless, there are a few discrepancies that can be attributed to the differences of the comparison strategies. For instance, we did not notice similar short-term changes in the attention to the topics Media, Films, and Music. These topics retain the same level of short-term attention throughout the pandemic period.

5.4 Discussion

Why do we need a complex pattern detection algorithm to detect trends? The simplest trend detection approach that first comes to mind would be to label Wikipedia articles with topics and then count articles that have spikes of viewership activity. Such an approach would give us a global overview of the topics of all pages that have spikes of activity. However, it would not necessarily detect trending topics. This simple approach would give us a lot of disconnected pages that may have had a spike due to various reasons unrelated to trending topics. For instance, it could be articles that describe days of the year or cataloging articles, such as ISBN or DOI. To avoid such one-page false trends, we use the underlying graph structure of the Wikipedia web network. Our approach allows us to detect clusters (dynamic patterns) of densely connected pages undergoing a similar increase in readership activity. The algorithm ensures that trends are represented by multiple pages on the same topic and that Wikipedia readers are actively exploring the extracted subnetwork of those pages. This way, our approach allows detecting global trends and provides a higher quality of trend detection.

Why are some topics different across languages? The difference could be attributed to multiple reasons. First, *media coverage* and Wikipedia's featured articles that appear on the main page of Wikipedia. The trends are a mass phenomenon that is related to important topics and, as such, are well covered by the media. The question is whether the media increases or inflates the interest of people in the trends. The authors of [68, 76] give a positive answer: on average around 25% of readers are motivated by media coverage, it even reaches 30% for the English and Russian versions. Some of the trends we captured also confirm this observation (Fig. 5.2, Fig. 5.4). For example, when a famous person dies, particularly in the show business, the media often broadcasts some of her works, be it movies (death of Stan Lee), TV shows (death of P. Gildas, "Nulle part Ailleurs" event on the French Wikipedia in October) or music (death of J. Kobzon event on the Russian Wikipedia in September). This would, in turn, increase the curiosity of people about the person and her work. Besides events getting more attention from their media coverage, we may assume some trends to appear exclusively due to media providers that produce and advertise their TV shows, such as Miss Universe, Miss France, or even Emmy awards that are presented in Fig. 5.1.

Some trends may be less influenced by the media than others. For example, sports events do not mainly rely on the media to attract the interest of fans. In this case, the reader's motivation could fall into the category of "conversation" or "event" as described in [68, 76]. In that study, these types of motivation have also high scores on average: 24% for "conversation" and 17% for "event" (motivation triggered by the event itself) and similar values for the English and Russian editions. In our results (Fig. 5.2, Fig. 5.4), sports (including football) is the second most popular topic after movies and even the first one in the English Wikipedia.

It is important to remark that if the media alone were the only driving force of the readers' interest to some Wikipedia topics, the trends would have had a different shape. The clusters of articles related to trending topics would have been made of a single or a few pages, as people would go to the page covering the event and left Wikipedia after the first read. Indeed, this phenomenon is common and can be observed on Wikipedia pages that are highlighted by some popular websites such as Wikipedia's or Google's front pages. Although, in our study, we exclude pages that have a single spike and only select clusters of connected pages with a correlated increase of visits. Again, referring to [68, 76], the main source of motivation of Wikipedia readers is "intrinsic learning" (except for the English version where it comes second) with 37% on average. Readers motivated by the media may visit a page, but they stay and follow hyperlinks on Wikipedia because they are motivated by *intrinsic learning*. This is demonstrated by the existence of the clusters of pages that we capture in our study. Moreover, many pages in each cluster bring complementary information that is not covered by the media or is not directly related to the event itself. For example, in the Hurricane Florence cluster, we can see some pages that list past hurricanes and pages related to meteorology, showing the importance of hyperlinks for intrinsic learning.

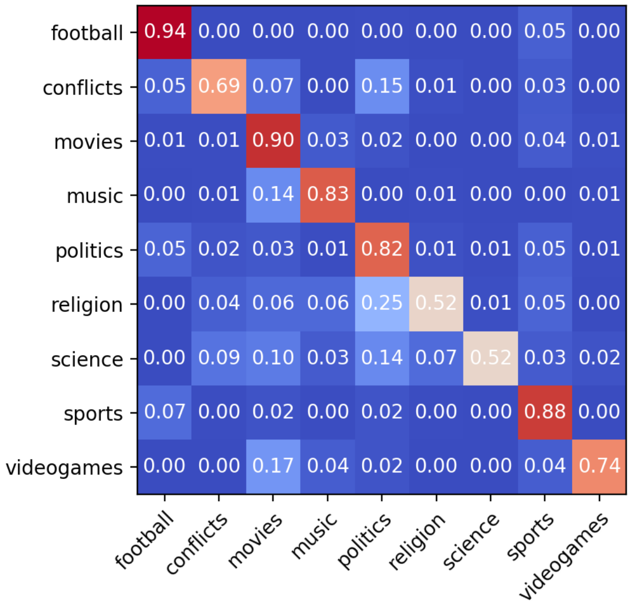


Figure 5.8 – Confusion matrix of BERT classification model. All topics, except Science and Religion, are classified correctly with a high accuracy. Most of the classification errors are caused by the similarity of the keywords and the cross-topic mix of pages in the clusters. Politics is often confused with Conflicts, Religion, and Science. Movies are often misclassified as Video Games or Music.

Wikipedia's structure plays a significant role in intrinsic learning since readers rely on cross-article links that lead to related content. Indeed, readers follow hyperlinks and are in search of more information than just basic facts about the trends. This behavior is equally shared across languages, although, for instance, clusters are smaller in the French, Russian, German, Chinese, Italian, and Spanish versions than in the English one. We assume this is due to the smaller size of the Wikipedia network and the smaller number of people reading Wikipedia in other languages. The English version of Wikipedia contains around 6 million pages, while the French and German ones contain around 2 million; the Russian, Italian, Spanish, and Chinese editions have around 1.5 million articles each.

Second, *geographic proximity*. This is especially apparent when we consider natural disasters. For example, the hurricane in North Carolina did not spark interest among French- and Russian-speaking readers and appeared only in the English edition of Wikipedia. Although, we can see that some outrageous traumatic events trigger memories among the readers in all languages. We can see the topic related to the 9/11 attacks as a supporting example. Besides, the COVID-19 case study supports the hypothesis of the influence of geographic proximity. Changes in the trends across languages are highly correlated with the development of the pandemic in the regions of the world where these languages are spoken by the majority of the population.

Finally, *cultural differences*. This difference is especially vivid when we look at sports. Readers of the English Wikipedia (mostly dominated by readers from the USA) tend to be interested in NFL championships, while the French or Russian-speaking readers do not express as much excitement about this topic and prefer European football championships (soccer) (Fig. 5.2). However, some sports such as golf or tennis are equally interesting to all groups of readers. This cultural influence can also be seen in music- and movie-related trends that are different across languages. Moreover, [76] also reports cultural differences in the motivation of the reader. For example, the "intrinsic learning" motivation between western and eastern cultures is different, with higher values for eastern people such as Chinese- and Russian-speaking readers. This is confirmed in our study where topics related to science have a higher number of clusters in the Russian and Chinese editions of Wikipedia (Fig. 5.2, Fig. 5.6). An example can be seen in Fig. 5.1 with the "Soyuz spacecraft" peak in the Russian version in October. It is the only science-related trend among the top trends in this timeline.

BERT vs FastText for classification. BERT model was trained on Wikipedia summaries, while the FastText one was trained on Wikidata items. If we compare Fig. 5.2 and Fig. 5.4, we can see that, semantically, the results of topic detection are similar in both models. The main difference between the two approaches is that the approach based on Wikipedia summaries labels entire clusters with one topic, while the approach using Wikidata items labels single pages. The first approach involves more manual intervention, requires time-consuming fine-tuning of the parameters, and basic knowledge of the language of a Wikipedia edition in which we detect trends. The main advantage of the model is that we use graph-based attributes for topic modeling, which makes the topic model more precise than the Wikidata-based one (see Fig. 5.8).

The second approach, Wikidata-based, is much less manual than the one described previously. There is no need for building a topic model, which significantly reduces the amount of preprocessing work. This approach gives more detailed results, providing a fine-grained overview of topics per cluster. However, we noticed that due to the imbalanced nature of the data and the higher number of topics in the classification model, when unsure, the Wikidata-based model tends to choose topics that have a high number of training samples (e.g. STEM or Biography). Nonetheless, both models serve well for topic labeling and the overall statistics of the distribution of attention to topics across languages is not affected by the models' inaccuracies.

Limitations of automated labeling. Automated labeling is not 100% accurate. For the BERT-based model (trained on Wikipedia summaries), we analyze the errors made by the classification model and show the results in Fig. 5.8 and in Table 5.1. FastText (trained on Wikidata items) classification metrics for selected topics are shown in Table 5.2. The detailed performance overview of the FastText model is available on the project's website [VII].

[VII] https://meta.wikimedia.org/wiki/Research:Language-Agnostic_Topic_Classification/Wikidata_model_performance

Table 5.1 – Classification metrics. BERT-based model trained on Wikipedia article summaries

Topic	Precision	Recall	F1	Support
Football	0.93	0.95	0.94	1359
Conflicts	0.75	0.69	0.72	112
Movies	0.86	0.89	0.88	646
Music	0.84	0.85	0.84	288
Politics	0.82	0.81	0.82	520
Religion	0.75	0.56	0.64	90
Science	0.79	0.60	0.68	70
Sports	0.84	0.85	0.85	751
Videogames	0.79	0.70	0.74	47
Accuracy			0.87	3883
Macro AVG	0.82	0.77	0.79	3883
Weighted AVG	0.87	0.87	0.87	3883

First, let us focus on the BERT-based classification model and look at the category "Football". In most erroneous cases, the model confuses "Football" with "Sports". Similar behavior is noticed for the label "Sports". Indeed, "Football" as "Sports" classes have similar meaning, so it is natural even for humans to misclassify the two. Similarly, Music and Movies are mixed as well as Politics&conflicts or Religion as they may involve the same figures. This indicates that the classification would improve if we chose more detailed topics that are more specific and had a more balanced dataset rather than training with more data.

However, if we look at the performance of the FastText model, which provides a more detailed classification and has more labels (64 in FastText vs 27 in BERT) and much more training samples than the BERT-based one (4M in FastText vs 10K in BERT), we can see that the performance also fluctuates across topics. Indeed, we can see that the model performs well when classifying articles related to Sports, Music, and Films. However, when it comes to more difficult topics, such as Politics, Religion, Science, and Conflicts (Military&Warfare), we can see that the performance of both models degrades. We also noticed that the training data for the FastText model needs to be refined because of the overlap in History, Society, and Politics in Russian and Chinese editions.

Qualitatively, the main difference between the two models is the type of topic labels that are used to train the classification models. As shown in Table 5.2, the FastText classification model was trained on a mix of general and specific topics, while in the BERT-based model, we have only general topics. For instance, we have pages labelled as *Culture.Media.Films* and *Culture.Media.Music* that are semantically very similar to a more general topic *Culture.Media.Media**. Another example is STEM. We can see a general topic *STEM.STEM**, which has many training samples resulting in higher accuracy and therefore stronger prediction confidence, and more specific topics, such as *STEM.Medicine&Health* and *STEM.Biology*.

Table 5.2 – Classification metrics. FastText model trained on Wikidata items

Topic	**Precision**	**Recall**	**F1**	**Support**
Culture.Sports	0.941	0.924	0.932	39322
History and Society.Military and warfare	0.858	0.607	0.711	6068
Culture.Media.Films	0.949	0.908	0.928	7409
Culture.Media.Music	0.890	0.842	0.865	11143
History and Society.Politics and government	0.844	0.552	0.668	6158
Culture.Philosophy and religion	0.729	0.438	0.547	3128
STEM.STEM*	0.948	0.799	0.867	33119
Culture.Media.Media*	0.903	0.833	0.866	29419
History and Society.History	0.736	0.427	0.541	3247
Culture.Visual arts.Visual arts*	0.824	0.665	0.736	8219
History and Society.Society	0.833	0.514	0.636	4801
History and Society.Business&economics	0.671	0.424	0.520	2908
STEM.Medicine&Health	0.838	0.486	0.615	1745
STEM.Biology	0.985	0.919	0.951	21686
Accuracy			0.746	
Macro AVG	0.837	0.629	0.712	

Overall, qualitative analysis of the errors shows that they are not significant in terms of semantics or meaning of the misclassified topics. However, when interpreting the results and making conclusions, the accuracy of the classifiers should be carefully considered.

6 Towards reproducible research

Reproducibility of experiments is an important component of the scientific method. Even though designing a reproducible experiment is harder and more time-consuming, it comes with a number of benefits. Along with establishing the trustworthiness and credibility of the research results, it enables faster evaluation and potential improvement of a proposed hypothesis. Scientific papers that have data, code, and comprehensive documentation available, give reviewers and readers a clearer picture of the ideas behind the text and formulas written in the paper. All that results in wider adoption of the proposed methods and ensures high quality of the presented results.

Despite the obvious benefits, the culture of reproducible research started developing fairly recently. According to a survey of 1576 researchers on reproducibility crisis [138], which was conducted in 2016, about 70% of respondents working in the field of physics and engineering failed to reproduce results from someone else's work. Chemists and biologists reported even higher failure rates. The respondents reported that replicating results often turns into a challenge because of multiple reasons. One of the most common obstacles is the unavailability of the code and the raw data that were used to run the original experiments. However, the problem cannot be solved by simply opening the data and the code because it covers only one aspect of the problem [139]. Once we get everything we need to replicate an experiment, we still need to deploy an appropriate environment, preprocess the raw data, and solve numerous compatibility issues, especially when we have to deal with the code that is three or more years old. All in all, there is a need for tools and frameworks that make the reproducibility process simpler and more accessible not only to professional researchers working in a specific field but also to non-experts. These tools should minimize the time spent replicating experimental setups and motivate more people to verify, reuse, and improve upon existing methods.

Dissemination of research results is as important as their reproducibility. Conveying complex ideas to lay audiences simply and entertainingly is a crucial part of research projects. One way to implement that is to encourage researchers to create interactive visualizations and informal blog posts that illustrate research results in a less formal and potentially more creative manner than research papers. Not only it allows reviewers and other researchers to get a better understanding of the results, but also it allows lay audiences to get a feeling of being a part of the research process, which ensures trust and a wider adoption of scientific findings in real life.

While designing the experiments presented in this thesis, we were motivated by the benefits and inspired by the potential impact of reproducible research. As a result, one of the most important features of the experiments in this thesis is that anyone can reproduce them and run the experiments on the latest available data. In this chapter, we provide thorough instructions on how to do that.

As we saw in the previous chapters, the datasets provided by Wikimedia Foundation have become the core benchmark for our pattern detection algorithm due to their scale and openness. Since Wikipedia is the most visited encyclopedia in the world, these datasets are massive and require considerable effort to load, clean, and transform into the desired data structure. To facilitate the process of working with Wikipedia, the graph of articles, and viewership statistics, we created a toolbox for practitioners and researchers interested in various aspects of the spatio-temporal dynamics of Wikipedia.

In this chapter, we describe our efforts towards reproducibility of the research presented in this thesis. We start with the graph-based processing framework for Wikipedia data (Sec. 6.1). After that, in Section 6.2, we discuss the dataset's use cases and show how to use it beyond the applications that we have already discussed in the previous chapters. Then, in Section 6.3, we give a brief overview of the distributed implementation of the anomaly detection algorithm presented in Chapter 3. This implementation uses the presented graph-based data framework as a back-end. Finally, we introduce our initiative Wikipedia Insights[I], a local hub for research blog posts and interactive visualizations. The main goal of this hub is to inspire other researchers and undergraduate students to work on Wikipedia data and to share the results of Wikipedia-related projects with a wider audience. We conclude this chapter with a discussion of potential extensions and improvements of the proposed data processing framework.

6.1 Graph-structured dataset for Wikipedia research

In this section, we present a convenient and versatile graph-based toolbox for researchers and practitioners. This toolbox is the core milestone in the roadmap towards reproducibility of the experiments presented in this thesis. The main goal of this toolbox is to simplify the access to Wikipedia data and its further analysis. All experiments on Wikipedia data that we have discussed rely on this distributed graph-based framework for data processing, which greatly facilitates the reproducibility process.

Like any other website on the Web, Wikipedia stores weblogs that contain viewership statistics of every page. Wikimedia Foundation, Wikipedia's parent non-profit organization, makes the web activity records and the hyperlink structure of Wikipedia publicly available, so anyone

[I] https://wiki-insights.epfl.ch

can access the records either through an API or through the database dump files. Even though the data is well structured, efficient preprocessing and wrangling requires professional data engineering skills. First, the dumps are very large and it takes a long time for researchers to load and filter them to get what they need to study a particular question. Second, although the API is well documented and easy to use, the number of queries and the response size are very limited.

Even though the API is quite convenient, it can cause reproducibility issues. The network of hyperlinks evolves with time, so the API can only provide the latest network configuration. To solve this problem, as a workaround, researchers use static preprocessed datasets. Two of the most popular datasets for Wikipedia network research are available on the SNAP archive, Wikipedia Network of Hyperlinks [140] and Wikipedia Network of Top Categories [141–143]. The initial publications referring to these datasets have been cited more than 1000 times, showing a high interest in these datasets among researchers. These archives were created from Wikipedia dumps in 2011 and 2013, respectively. However, Wikipedia has evolved since then and the research community would benefit from being able to access more recent data.

The popularity of Wikipedia data among researchers continues to rise, leading to the development of new datasets. To facilitate studies focusing on Wikipedia revision history, Mitrevski, Piccardi, and West [144] developed a parser that produces Wikipedia revision history in HTML format based on wikitext, Wikipedia's markup language. They provided a more complete and accurate data source for a range of applications related to link prediction, popularity, and navigational importance. The increasing popularity of GNNs inspired Mernyei and Cangea [145] to create a benchmark based on Wikipedia web network and textual features of Wikipedia articles on computer science. The goal of the dataset is to provide a benchmark for semisupervised node classification and single-relation link prediction models. Consonni, Laniado, and Montresor [146] proposed a graph-based dataset of Wikipedia hyperlinks, covering the 9 largest language editions. They proposed a few potential use cases, including link recommendation and prediction, anomaly detection, and controversy studies.

The need for graph-based datasets is highlighted in numerous research works. Multiple studies analyzed Wikipedia from a network science perspective and used its network structure to improve Wikipedia itself or to gain insights into the collective behavior of its users. In [147], Zesch and Gurevych used Wikipedia category graph as a natural language processing resource. Buriol et al. [148] studied the temporal evolution of the hyperlink graph of Wikipedia. Bellomi and Bonato conducted a study [149] of the macrostructure of the English Wikipedia network and cultural biases related to specific topics. West et al. proposed an approach enabling the identification of missing hyperlinks in Wikipedia to improve the navigation experience [150]. Multiple applications of Wikipedia citation network dataset were proposed by Singh et al. [151], including citation recommendation and knowledge graph construction.

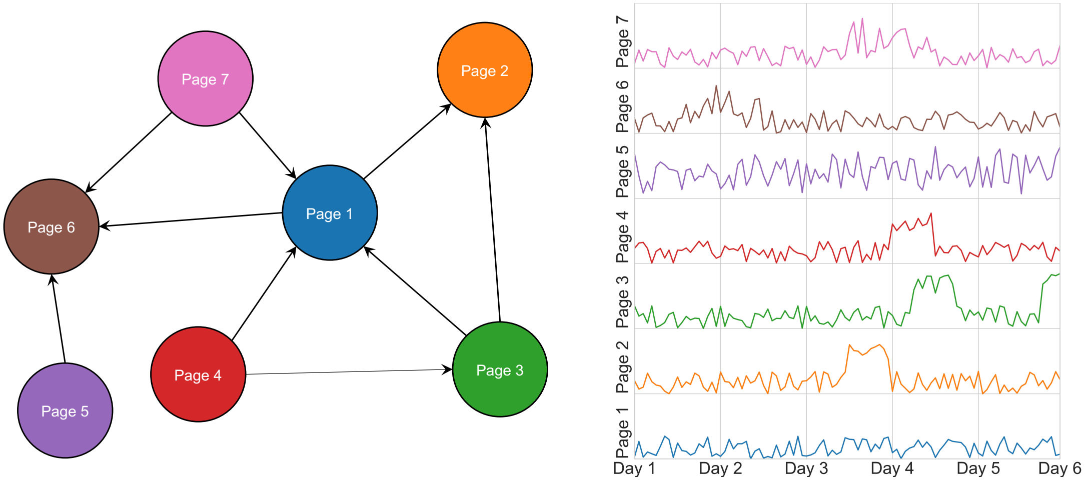


Figure 6.1 – An illustration of a subset of Wikipedia web pages with viewership activity (pagecounts). *Left:* Wikipedia hyperlinks network, where nodes correspond to Wikipedia articles and edges represent hyperlinks between the articles. *Right:* hourly page-view statistics of Wikipedia articles.

Another direction of Wikipedia research focuses on the pagecounts analysis. Moat et al. [152] used Wikipedia viewership statistics to gain insights into stock markets. Yasseri et al. [153] studied editorial wars in Wikipedia, analyzing activity patterns in the viewership dynamics of articles that describe controversial topics. Mestyán et al. [154] demonstrated that Wikipedia pagecounts can be used to predict the popularity of a movie. As we saw in Section 4.4, the collective memory phenomenon was studied in [26], where the authors analyzed visitor activity to evaluate the reaction of Wikipedia users on aircraft incidents.

The hyperlink network structure, on the one hand, and the viewership statistics (pagecounts) of Wikipedia articles, on the other hand, have attracted significant attention from the research community. Recent studies open new directions where these two datasets are combined. The emerging field of spatio-temporal data mining [39] highlights an increasing interest and a need for reproducible network datasets that contain dynamically changing components.

Following the recent advances of scientific research on Wikipedia, our framework focuses on two components: the *spatial* component (Wikipedia hyperlinks network) and the *temporal* component (pagecounts). We design a database that allows querying this hybrid data structure conveniently (see Fig. 6.1). Since Wikipedia weblogs are continuously updating, we designed this database in a way that will make its maintenance as easy and fast as possible. Finally, the framework enables us to work with *any language edition* of Wikipedia, opening new avenues for multilingual studies.

There are multiple ways to access Wikipedia data, but none of them provide native support of a graph data structure. Therefore, normally, if researchers want to study Wikipedia from the network science perspective, they have to create the graph themselves, which is usually very time-consuming. To do that, they need to preprocess large dumps of data or to use the limited API.

In spatio-temporal data mining [39], researchers are most interested in the dynamics of the networks. Hence, when it comes to Wikipedia analysis, one needs to merge the hyperlink network with page-view statistics of the web pages. This is another large chunk of data, which requires another round of time-consuming preprocessing.

After the preprocessing and merge are completed, researchers usually realize that they do not need the full network and the entire history of visitor activity. However, there is no easy workaround: to get a certain subset of pages for a specified period, everyone has to perform the aforementioned steps.

Here, we will present a graph-based solution that eliminates the preprocessing steps described above, a graph database that simplifies access to the hyperlink structure and its viewership statistics of Wikipedia. With a set of intuitive queries, we provide the following features:

- Process and load web networks and viewership activity(pagecounts) for any language edition of Wikipedia.
- Process and load links connecting multiple language editions (langlinks).[II]
- Query relatively large subgraphs of Wikipedia pages (1K–100K nodes) without redirects.
- Use filters by the number of page views, category/sub-category, graph measures (n-hop neighborhood of a node, node degree, page rank, centrality measures, and others).
- Query viewership statistics for a subset/subgraph of Wikipedia pages.
- Query a subgraph of pages with a number of visits higher than a threshold, in a predefined range of dates.

The database enables its users to query subgraphs with millions of links. However, requesting a large subgraph from the database may take several hours. Besides, it may require a large amount of memory on the hosting server. Such queries may cause an overload of the database server that has to process queries from multiple users at the same time. Therefore, instead of setting up a remote database server, we have decided to provide the code to deploy a local or a cloud-based one from Wikipedia dumps. This should allow researchers to explore the dataset on their server, design new queries, and possibly contribute to the project.

Lastly, our framework allows working with any version and language edition of Wikipedia dumps. This gives researchers the ability to reproduce previous studies on Wikipedia data and to conduct new experiments on the latest data.

[II] The feature was contributed by Carlos Badillo

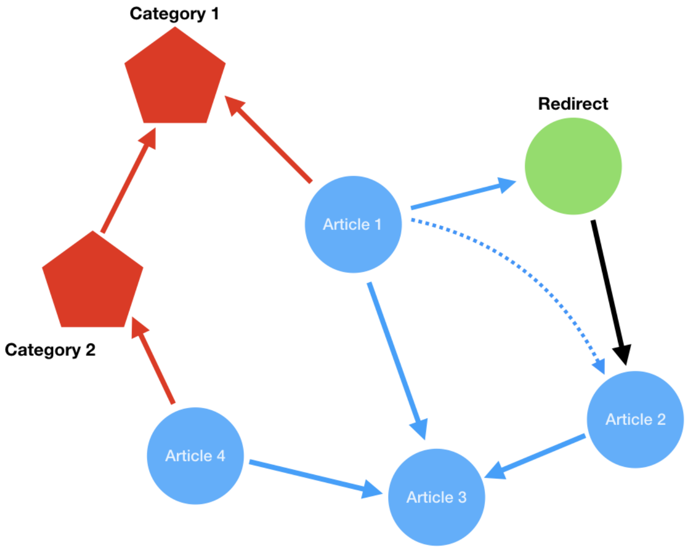


Figure 6.2 – Wikipedia graph structure. In blue: articles and hyperlinks referring to them. In red: category pages and hyperlinks connecting the pages or subcategories to parent categories. In green: a redirected article, i.e., Article 1 refers to Article 2 via the redirected page. In black: a redirection link. The blue dashed line is the new link created from the redirection.

Furthermore, Wikipedia graph and pagecounts can be used separately, which creates more use cases and applications of the dataset. The latest deployment instructions are available online [155].

In the following sections, we present the spatial and temporal components of the dataset (Sec. 6.1.1 and 6.1.2), describe the data processing stages (Sec. 6.1.3), and provide performance analysis of the queries (Sec. 6.1.4).

6.1.1 Network of Wikipedia articles

We represent the Wikipedia graph as a multigraph with different kinds of nodes and links. The objects are described in Table 6.1; the relationships between the objects are illustrated in Fig. 6.2. We store Wikipedia network of articles in a property graph database Neo4J and connect it to the rest of the data framework using Scala connectors. The overall architecture of the data processing framework is depicted in Fig. 6.3).

Nodes in the graph represent Wikipedia webpages of two types, *articles* and *categories*. Hyperlinks between webpages are stored as directed edges between the nodes. Both articles and their categories are stored in the same graph. In Wikipedia, categories refer to their elements (articles or subcategories) with hyperlinks. Each article has hyperlinks that point to the categories they belong to. Inside the graph database, articles and category pages differentiate into nodes of distinct types with different labels.

Table 6.1 – Entities in the graph-based dataset of Wikipedia pages

Name	Nature	Description
article	node	Wikipedia article
category	node	Wikipedia category article
links_to	link	hyperlink between 2 articles
belongs_to	link	hyperlink between an article or subcategory and a category page

To provide a convenient interface to access articles and categories, we introduce two types of links. The ”links_to“ relations are hyperlinks between articles (excluding categories), and the ”belongs_to“ relations are linking articles to their categories or subcategories to their parent categories. Edges of the latter type represent hyperlinks inside articles, which point to category pages.

Graph structure. The internal structure of Wikipedia web network justifies our choice in favor of graph databases. The category structure in Wikipedia is shaped as a tree. The advantage of such structure is that it is easy to handle it when a user creates articles and wants to classify them into subcategories. However, it complicates the retrieval of a set of all articles belonging to a given category (or subcategory). One has to explore the entire hierarchy of subcategories inside that category and collect all encountered articles. The inherent features of graph databases allow simplifying this task. Traversing and performing the breadth-first search in the graph is one of the basic functions of graph databases, which makes this solution a more efficient alternative to relational databases.

Redirects. To handle renamed or merged pages, Wikipedia relies on redirecting pages. When renaming a page, moderators create a new page with a new title. However, they do not remove the initial page because it would break hyperlinks from articles that point to the renamed page. To avoid broken links, the initial page becomes a "redirect", a page that automatically redirects a visitor to a new page. Redirect pages are invisible to users. We remove these pages from our dataset and create a skip-connection to the correct article (the blue dashed arrow of Fig. 6.2). First, it simplifies the queries when exploring the graph. Second, it makes the structure cleaner and easier to understand. Lastly, it halves the number of nodes in the graph; at the time of writing, the number of articles in the English Wikipedia is close to 6 million, while the number of redirects is around 8 million.

Langlinks. Wikipedia is available in multiple languages (309 as of January 2020). Pretty much every article on Wikipedia has a counterpart in another language. Each multilingual article has a link, referring to its twin in another language. The toolbox facilitates preprocessing, storing, and querying multilingual articles and links connecting them.

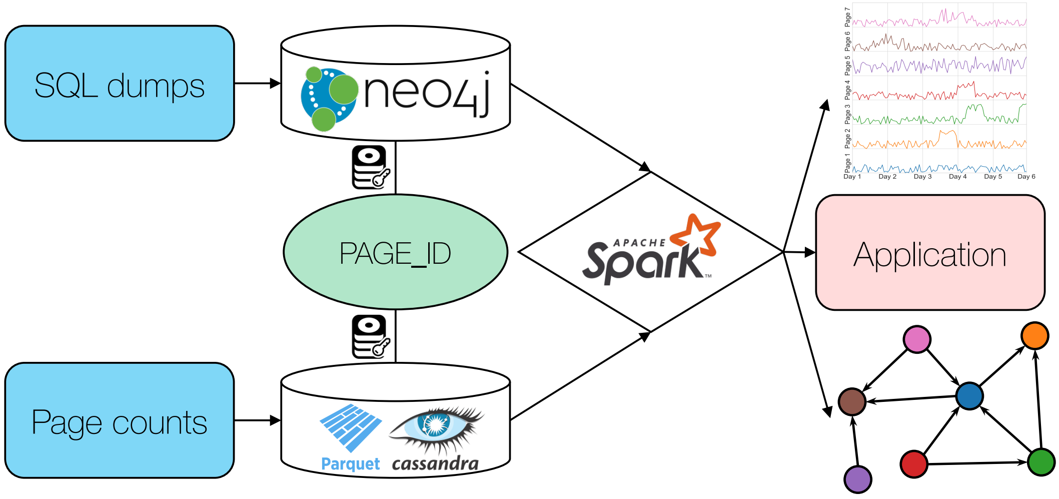


Figure 6.3 – Schematic structure of the framework. SQL dumps with Wikipedia's hyperlink structure are preprocessed and stored in a Neo4J database instance. There are two options to use preprocessed page view statistics (pagecounts). First, directly from Parquet files. Second, from a Cassandra database instance. The two databases can be used separately. If needed, the two data sources can be connected by *PAGE_ID* field as a key and used together as a spatio-temporal data structure. Both databases have connectors to Apache Spark allowing for distributed processing of the data and seamless integration with other distributed frameworks.

6.1.2 Wikipedia viewership data

Viewership statistics is an independent data source. It does not depend on the graph dataset described previously, however, both data sources can be used together in one application (see Fig. 6.3). The time series of visits are stored separately in Parquet files in the form of a collection of indexed *key:value* pairs. Each key is a triplet *(language code, page id, time-stamp)* and the value is the *number of visits* during the hour given by the time-stamp for the page associated to the page id in a particular language edition of Wikipedia. Note that different language editions may have the same page id for different pages, therefore it is important to keep the language code as a part of the key.

There are two ways to use the preprocessed pagecounts. They can be used as raw Parquet files[III] or stored in a Cassandra database (see Fig. 6.3). Parquet is a convenient format to work with when we need to deal with distributed frameworks and environments. This option gives practitioners the freedom to choose frameworks and tools. The other option is storing preprocessed entries in a Cassandra database. Cassandra is well integrated with open-source frameworks for distributed data processing, which motivated our decision to use this database as a default option. Its structure provides a flexible way of recording new entries following the evolution of time, page creation and deletion that occur in the encyclopedia. Querying a

[III] The feature was contributed by Anthony Miyaguchi specific period is very convenient and efficient as well. It is done by submitting a request with a specific range of key values (a range applied to the time-stamp key of the key couple).

To reduce the amount of data to be stored, we introduce a threshold for the number of visits per page per day. We store the number of hourly visits for an article if the daily total of its visits is above this threshold (100 by default). This reduces the number of entries by an order of magnitude without losing relevant information. The framework handles missing records automatically, which is also very convenient.

6.1.3 Data extraction and preprocessing

Before creating the graph database, we perform the following preprocessing steps. After having downloaded Wikipedia dumps [156], we parse the SQL files to extract the titles of articles and categories, page and category ids, and hyperlinks. Before storing the data in the graph database, we remove the redirects and modify the hyperlinks pointing to them to link to the correct articles. After these steps are completed, we load the data into the graph database.

If we decide to work with the time series data representing pageviews, we download the pagecounts dumps [157] (number of visits per page per hour), and extract the hourly visits. As described previously, we remove entries with a low number of visits. If a page has less than 100 daily visits, we do not store visit records for that page and that day. The data has a resolution of 1-hour. We store the values above this daily threshold in Parquet files that can be transferred into a NoSQL database. As an option, we deploy the data in an instance of Cassandra database.

Wikipedia dumps and pagecount statistics have different release lifecycles. The separation of the two data sources, the graph and time-series data, simplifies the update process and the maintenance. Wikipedia dumps are released monthly. Every month, we can compare the new and previous version dumps of pages and links and update only a part of the graph database. We can add new nodes and links to the database and delete the removed ones. Pagecount statistics is released daily. We can perform daily updates of the time series database. To do that, every day, we add 24 new entries (one per hour) for each article.

6.1.4 Performance of graph queries

To test the performance of the graph database and compare it with the relational counterpart, we constructed the graph of English Wikipedia pages based on the August 1st 2018 SQL dumps. After resolving the redirects, the graph consists of about ca. 7.4 million articles, comprising both regular pages (ca. 5.7 million) and category pages (ca. 1.7 million), and ca. 511 million edges. Once the data was imported into the graph database, we ran queries to extract various subgraphs, e.g., retrieve all pages and subcategories belonging to a given category and all the links between these pages. Table 6.2 demonstrates the query results and the time required to process them.

Table 6.2 – Size and performance for different subgraph requests

Category	Articles	Hyperlinks	Subcategories	Search depth	Proc. time
Philosophy	571	5’165	202	2	0.4 s
	5’370	177’754	1’144	3	29.7 s
	26’480	1’094’550	4’084	4	574 s
Physics	2’263	27’911	207	2	3.3 s
	10’128	223’870	971	3	55 s
	33’917	972’206	3’712	4	501 s
Science	1’762	19’189	455	2	3 s
	18’751	260’043	2’842	3	292 s
Actors	1’107	3’313	654	2	1.6 s
	10’805	47’196	2’922	3	90 s
Global conflicts	859	6’598	223	2	1 s
	6’179	152’517	1’208	3	48.5 s
	22’663	706’357	3’905	4	541 s
Exoplanets	989	18’926	69	unlimited	0.8 s

The presented results have been computed on a 24-core Intel Xeon E5 system, equipped with SSD drives and using the Neo4j open-source database. Given the highly connected structure of Wikipedia, we had to restrict the depth of certain queries as the returned set expands dramatically.

While it is possible to use relational databases to store pages and link information, retrieving a subgraph using a tabular structure would require an increasing number of subqueries or table joins when increasing the depth of the queried subgraph. This results in longer processing times and complex query syntax. Extracting a subgraph requires complex queries to find all the nodes belonging to the subgraph. Then we need to perform an additional search to find all edges connecting nodes in the set.

To compare the performance, we conducted an experiment where we query the same data from raw files, a relational database, and a graph database. We used Neo4J as a graph database and PostgreSQL as a relational database. To query the raw files, we used Apache Spark. We used a truncated version of the Wikipedia SQL dumps to perform the comparison. Before running the experiment, to simplify the queries and create more efficient indexes, we have done basic preprocessing of the data. We replaced the textual ids of each page, represented as a combination of a page title and its namespace, by unique page ids. Additionally, we removed redirects.

Queries performed on the graph database are often at least an order of magnitude faster than on the raw files. For instance, querying the subcategory graph of the ”Physics“ category with depth 2 requires approximately 5 minutes to retrieve all the nodes belonging to the subgraph. Moreover, it takes several additional minutes to retrieve its edges, whereas the same data is completely extracted in less than 5 seconds from the graph database.

Table 6.3 – Size and performance for article neighbor subgraph requests

Page	Articles	Hyperlinks	Subcategories	Search depth	Proc. time
Switzerland	1’400	144’911	24	1	4.5 s
United States	2’215	258’939	28	1	17.5 s
Charlie Chaplin	1’289	147’203	23	1	4 s
Albert Einstein	1’025	114’518	30	1	2.3 s
Computer science	684	47’067	13	1	1 s
	68’756	7’883’471	1’450	2	3’600 s

Using a relational database improves the situation, as the nodes of the subgraph are returned in less than a second. Returning the edges from the subgraph remains, however, time-consuming (ca. 10 to 40 seconds in our experiments), in addition to requiring multiple nested queries whose complexity increases with the search depth. In that particular example, given a relatively small size of the result, the timing can be heavily impacted by the cache of each application. In addition, the type of storage they run on also affects the performance, hence we advise using SSD storage for better performance. For instance, using a query (on both databases) to retrieve a subgraph of depth 3, then retrieving the same subgraph of depth 2 will most likely only use the cache and yield much faster results. When the search depth increases sufficiently, the relational database can lead to faster processing than the graph database, at the expense of query complexity.

Similarly, we queried subgraphs consisting of page neighbors (i.e., connected via a ”links_to“ relation), up to a certain depth. We also restricted the queries by the number of outgoing links from the top page since some of them have a substantial number of direct connections. We provide the results of these queries in Table 6.3. Increasing the depth of such queries (e.g., for a depth greater than one) leads to large responses, resulting in a longer processing time (cf. the ”Computer science“ entry in Table 6.3).

6.2 Dataset applications and use cases

6.2.1 Spatio-temporal datasets

To test the performance of spatio-temporal algorithms, we need to use both the web network and its pageview statistics. A combination of the two data sources transforms the dataset into a spatio-temporal data structure, which can be used to test algorithms created for dynamic graphs. We saw such examples in Chapter 4 and Chapter 5, where we used our spatio-temporal anomaly detection algorithm to detect dynamic patterns in Wikipedia user activity and tracked the development and evolution of real-world events.

The problem of suggesting missing hyperlinks between related pages has been investigated in multiple studies. Some of them are based on the text of articles, some of them on the visit patterns. For the latter, the possibility to query a group of articles and their visits over time from the database could facilitate the search for missing links.

6.2.2 Generating datasets for graph neural networks

Research on graph neural networks (GNNs) has become very popular in recent years. This drives the need for graph datasets of different sizes and properties. Due to a reach set of additional attributes, a Wikipedia-based dataset can provide interesting test cases for GNNs. We can use our framework to generate different graph-based datasets with various attributes, such as the text of articles, viewership statistics, and categories or general topics of Wikipedia articles. Moreover, the availability of multiple language editions opens new avenues for research on transfer learning using GNNs, where a network trained on one language learns to classify graphs from another language edition.

6.2.3 Knowledge graphs and graph embeddings

Apart from the examples that we have already discussed, there are a few other research projects that have already used the dataset for knowledge graphs and graph embeddings. The proposed data processing framework can be used to build knowledge graphs. London et al. constructed a Wikipedia-based knowledge graph using a taxonomy generator [158]. Another use case is time-series forecasting using graph embeddings. Miyaguchi et al. used this toolkit to forecast Wikipedia page views using embeddings of its web network [159]. In this example, the authors used both the graph and pagecounts databases.

6.2.4 Working with selected subgraphs of Wikipedia

In some cases, we need to work with a selection of pages. Even though we provide convenient tools to select subgraphs of Wikipedia, the structure of the encyclopedia is rather chaotic, which makes this task more difficult. Wikipedia articles are classified according to the category hierarchy established by the editors of Wikipedia. Wikipedia categories are cumbersome. The absence of strict guidelines or strong authority on category labeling resulted in a complex category schema. Gathering all pages belonging to a category is a difficult task at the moment. It requires visiting all subcategories belonging to the initial category and collecting the articles they refer to. Furthermore, we often notice that there are collections of several subcategories (and hence articles) that are only remotely related to the original category or subcategory. Those subcategories can be very generic and encompass a large number of articles, e.g., one of the subcategories of ”Physics“ is ”Writing systems“ (linked via ”Physical systems“). Indeed, a deep category hierarchy, its complexity, and the lack of tools for accessing the network of categories makes it impossible to have a global view on the structure and efficient maintenance.

Table 6.4 – Number of nodes in the subgraph of the ”Physics“ category

Depth	Articles	Subcategories
1	69	27
2	2’263	206
3	10’128	970
4	33’917	3’711
5	80’349	16’917
6	232’818	74’004
7	2’041’232	251’551

To illustrate the complexity of the category structure, we run multiple different queries. Each query defines a category and asks for all the articles belonging to this category and its subcategories. The results are shown in Table 6.2. In the case of broad categories, the number of articles grows rapidly as we go deeper in the subcategory hierarchy. Each subcategory may have subcategories of its own. We define the depth to be the distance in hops from the initial category to the furthest subcategory in the subcategory tree. For instance, the category *Physics* already contains 33’917 articles and 972’206 hyperlinks at depth 4. This number grows to more than 2 million pages when articles are collected up to subcategory depth 7 as shown in Table 6.4. In fact, this is one-third of all articles in the English Wikipedia. This result is surprisingly large and additional investigation is required to understand the structure and check its correctness. The web network of Wikipedia articles and categories is highly connected, so the number of links and the time to retrieve the data grows very quickly. For general categories, after 4 hops in the category tree, the result of the query reaches tens of thousands of pages and more than a million links.

This complexity in the category hierarchy makes it memory-expensive to query subgraphs of articles in the same category. Even though the performance of the database queries drops when the network expands, it is possible to query these subgraphs. Hence, our proposed database opens new avenues to the popularization of research on large subnetworks of categories. This may give a better understanding of the category and article structures. The results may lead to a better organization of categories and a more efficient process of verification of their consistency.

6.3 Reproducibility of the experiments presented in this thesis

We provide two implementations of the algorithm presented in Chapter 3. The Python version[IV] is a demo version of the algorithm, which allows testing it on toy and random datasets. The distributed version is written in Scala[V] and can be used to detect anomalies in large-

[IV] https://github.com/mizvol/anomaly-detection

[V] https://github.com/epfl-lts2/sparkwiki/blob/master/src/main/scala/ch/epfl/lts2/wikipedia/PeakFinder.scala scale dynamic graphs. In our main test case, we use Wikipedia data to demonstrate the performance of the algorithm and its scalability. The Scala implementation is fully integrated with the graph-based data framework, presented in Section 6.1.1. It allows reproducing the experiments presented in this thesis and running experiments on the new data. Detailed documentation and instructions are available in the corresponding repositories.

To reproduce the experiments presented in Chapter 5, we recommend using the distributed version of the algorithm implemented in Scala, which is fully integrated with the data processing framework presented in this chapter. When working with Wikipedia data, we suggest starting with a high threshold for daily pageview counts and decreasing it gradually to avoid overly long processing times. Another practical suggestion is to start with a short time window of 3 days maximum because of the same reason. It is important to point out that one needs the same amount of historical pageview data stored in the database as the size of the analyzed time window. Therefore, if we need to detect patterns over 3-day period, we need to have 3 previous days of historical pageview data available in the database. For more details and a complete step-by-step roadmap to the reproducibility of the experiments, we ask the reader to consult the documentation available in the code repository.

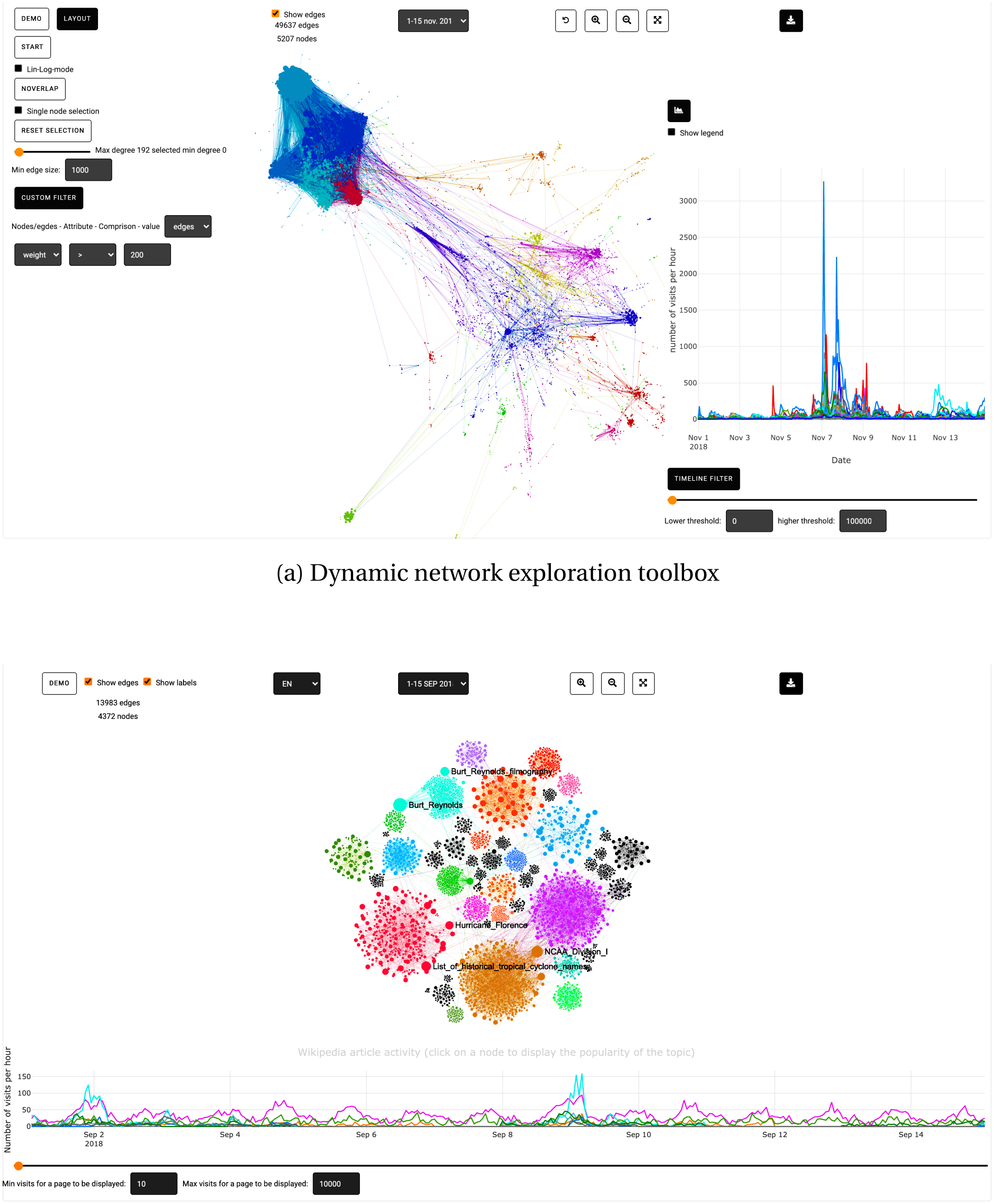
The implementations of the algorithm and the graph-based data processing framework sparked interest in Wikipedia research among students pursuing their master's and bachelor's degrees. Not only they reproduced the experiments presented in this thesis using the provided tools and implementations, but they also developed their ideas bringing insights into Wikipedia ecosystem. During their internships in the lab, they created various projects ranging from natural language processing to interactive visualizations with applications in collective behavior analysis, topic detection, and graph-based document classification. All that led to the creation of a local hub, Wikipedia Insights, where we showcase their projects[VI]. There, among informal blog posts, you can find interactive visualizations and graph exploration tools that represent another dimension of the results presented in this thesis.

We can see two examples of such visualizations in Fig. 6.4. First (Fig. 6.4, left), a web-based dynamic graph exploration tool[VII]. It provides users with multiple features that allow for filtering, changing the layout of the graph in real time, and animating the spatio-temporal dimension of the network. The tool is extendable and it is possible to build custom spatio-temporal visualizations upon it. Second (Fig. 6.4, right), an interactive version of the results that we have presented in Section 5.2[VIII]. It allows exploring the detected patterns and visualizing activity of the network over time to get a better understanding of how our pattern detection approach actually works.

[VI] https://wiki-insights.epfl.ch/

[VII] https://wiki-insights.epfl.ch/dynamic-graphs/

[VIII] https://wiki-insights.epfl.ch/wikitrends/



☑ Show edges
49637 edges
5207 nodes
1-15 nov. 201
Max degree 192 selected min degree 0
Min edge size: 1000
Nodes/egdes - Attribute - Comprison - value: edges
weight > 200
Lower threshold: 0 higher threshold: 100000

(a) Dynamic network exploration toolbox

☑ Show edges ☑ Show labels
13983 edges
4372 nodes
EN
1-15 SEP 201
Min visits for a page to be displayed: 10 Max visits for a page to be displayed: 10000

(b) Interactive demonstration of Wikitrends project

Figure 6.4 – Top: web-based dynamic graph exploration tool providing a range of features for filtering, changing the layout of temporal graphs, and animating spatio-temporal changes. Bottom: evolution of Wikipedia trends over time. Interactive version of the results presented in Section 5.2).

Even though we provided a thorough analysis of the results in the original papers, sometimes, proactive readers, who decide to dive into the results with the help of the visualizations, find new interesting insights that we did not notice initially. This makes us confident about the high value of reproducible research to the general public and the importance of dissemination of the results of scientific findings to different audiences.

6.4 Discussion

In this chapter, we have presented a toolbox allowing researchers to preprocess, store, and access Wikipedia web network and viewership activity of the pages. The main goal of this toolbox is to provide a convenient tool for researchers working on Wikipedia and analyzing the dynamic properties of this network. We designed the database with the idea of reproducible research in mind. We would like this project to become an important building block in Wikipedia research community that should speed up the research process and facilitate its reproducibility.

Reproducible research. In science, it should be mandatory for any published results to be reproducible. This implies unlimited access to the data used for the experiments. However, when the dataset evolves with time, as it is in the case with Wikipedia articles and viewership statistics, it may be difficult to recover the exact data used in a given study. Some articles may have been removed or some links may have appeared after the publication of scientific work. A workflow designed for scientists must include a simple mechanism that facilitates the reproduction of the experiments. The goal of the toolbox described in this section is to provide such solution.

Potential benefits for Wikipedia. A better understanding of Wikipedia structure, both from an article and a category point of view, is an important matter for the encyclopedia and the organization of its knowledge. Finding missing hyperlinks, suggesting links during the creation of pages, monitoring Wikipedia visitor activity, or structuring the category tree, are among the numerous possible applications of the proposed toolbox.

Toolbox development and future steps. Wikimedia's archives are a treasure for open science and open research. There are multiple ways for improving and enlarging the toolbox. For instance, we can add the information about Wikipedia edits and editors to the nodes in the graph database. They could be structured as a graph of articles or a graph of users with time series of edit activity, opening new avenues for various applications and studies.

7 Conclusion

There can be no explanation which is not in need of a further explanation.
— Karl Popper [160]

We can observe graph-structured data with dynamic attributes in many fields. The main goal of this thesis was to develop a scalable and interpretable algorithm for dynamic pattern detection in such datasets. To demonstrate the efficiency and core features of the proposed approach, we focused on the applications to web and social networks. In these applications, we strove to deepen our understanding of collective behavior patterns in the online activity of internet users, detecting trends, collective interests, and common navigation patterns.

In the proposed pattern detection approach, we used properties derived from the connection between GNNs with attention mechanism (GAT) and memory networks (Chapter 3). Rather than learning the attention function, we have defined it based on prior knowledge about the data. That allowed us to combine the structure of the data and the domain knowledge to solve the problem of dynamic pattern detection at a large scale. One of the main benefits of our approach is that it is fully unsupervised, which allowed us to detect patterns in unlabelled data. Moreover, we demonstrated that we could use the learned patterns in applications related to recommendation systems and information recovery (Sec. 4.3).

A core feature of our pattern detection method is the interpretability of the results. In Chapter 4, we analyzed the detected patterns and showed multiple applications where this feature could be useful. In particular, the interpretation of the patterns detected in Wikipedia viewership activity allowed us to gain insights into the collective behavior of Wikipedia readers. We have observed that some patterns of readers' activity have associative properties, i.e., current events trigger related memories of the past. To understand the associative nature of users' preferences, we studied the collective memories of Wikipedia readers (Sec. 4.4). The detected collective memories reflect the way people perceive real-world events and what associations those events trigger when people read related Wikipedia articles. In Chapter 5, we extended our approach with an automated pattern interpretation module, which allowed us to compare the global interests of Wikipedia readers across multiple languages.

Even though our approach can be used to find dynamic patterns in any online social networking platform, we used Wikipedia server logs in the majority of our experiments (Chapters 4, 5). Openness, scale, and the diverse audience of readers are among the driving forces that led us to choose Wikipedia for our case studies. There are also other positive aspects, which define Wikipedia as an unique social networking environment. First, due to its non-profit nature, contrary to other social platforms, it prioritizes user privacy over profit and does not create profiles of users to increase engagement. Second, Wikipedia does not use user information to train personalized recommendation models that are prone to create online filter bubbles, enforce polarization, and distort readers' view of reality. As a result, the navigation patterns in Wikipedia appear more natural, revealing truly spontaneous intentions of the readers that are not biased by personalized recommendation algorithms.

Collective behavior analysis of users' activity data collected from for-profit social networks is a completely different story. Business incentives push major online social networks toward using personalized recommendation algorithms to increase users' engagement with the content, which results in higher exposure of users to personalized ads. Such algorithms are designed to direct the content towards specific audiences based on the interests derived from their prior activity history. As a result, the user activity data is often prone to algorithmic bias, which is inflicted by personalized recommendation systems.

Personalized content targeting influences and drives the interests of the users, thereby strengthening the online filter bubble effect. The algorithmic bias makes the patterns of collective behavior less natural since the data logs stop reflecting the spontaneous interests of the users. First, that raises ethical concerns, approaching the point where we cannot distinguish whether users made a conscious decision or blindly followed a recommendation produced by the algorithm. Second, we face problems with pattern detection algorithms because they end up capturing patterns that mostly reflect the interests imposed by the recommendation algorithms. In other words, we cannot guarantee the full consciousness of users' decision-making process when we perform collective behavior studies. At some stage, we cannot identify whether a user read what she read because she was genuinely curious to read an article or because she was influenced by a personalized recommendation system. All in all, contrary to Wikipedia data, it is hard to guarantee the naturalness of the collective behavior patterns in for-profit social networks.

Another important aspect of Wikipedia is its decentralized content curation, which is done by a diverse group of editors who ensure that the shared point of view is neutral. Neutral point of view (NPOV) is one of the fundamental concepts of Wikipedia content creation[I]. NPOV minimizes the polarization of content and ensures that the opinions that appear in the articles are not affected by editorial bias, represented fairly, and supported by reliable sources.

Nonetheless, when we compared different language editions of Wikipedia, we noticed that some aspects affect readers' perception of particular topics. We found that some interests are driven by the media, readers' geographical proximity to real-world events, and cultural differences. In addition, we can see that the web network structure of Wikipedia influences people to read only the content that was explicitly linked by the editors, forming a particular image of the topic of interest. As we have discussed in Chapter 5, the same pages written in different languages can reflect diverging points of view, affecting the perception of the facts by the readers and causing the emergence of different opinions on the topic. This phenomenon is especially apparent in subnetworks related to controversial subjects that cover politics, history, and culture.

To provide different perspectives on the topic, we can use collective memories from multiple languages that we have extracted with our approach. Comparing collective memory patterns across languages could help to describe discrepancies in perception of various events by readers that read Wikipedia in different languages. Explaining these differences explicitly to the readers could enhance their understanding of controversial topics and reduce polarization. For instance, we could do that by using visual illustrations of collective memory patterns, as we have shown in Chapter 4 (Sec. 4.4).

Lastly, we have presented a distributed graph-based framework for Wikipedia data processing. The framework allows for reproduction of all the experiments that were discussed in this thesis as well as running similar studies on the latest Wikipedia data (Chapter 6). In addition, the general purpose of the framework is to facilitate studies that use Wikipedia data, particularly its web network and viewership statistics. Recent applications of our framework and numerous contributions to the code showed that other researchers can use in their research projects related to knowledge graphs, graph embeddings, and time series forecasting (Sec. 6.2).

7.1 Future work

Limitations of the proposed approach. In this thesis, we have presented multiple applications and demonstrated various capabilities of the proposed pattern detection approach. Nonetheless, there are limitations that should be taken into account when working with the approach. These limitations can also serve as a basis for future work that could be build upon the research presented in this thesis.

[I] https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view

First, the initial structure of the graph is crucial for our pattern detection approach. Our approach can only detect communities (dynamic patterns) in the graph that is explicitly defined by the existing edges. However, co-activation patterns can also occur across parts of the graph that are not connected via existing edges. Such patterns would indicate shortcomings of the existing link infrastructure and might suggest new links that should be added to the network. This could potentially improve the quality of the pattern detection. A viable solution would be to add a link prediction layer on top of our pattern detection algorithm to infer edges between communities that have similar co-activation signatures. Another solution would be to use temporal network mining approaches that allow inferring networks from time-series observations (see Section 2.1 for a more detailed overview).

Second, in all the experiments presented in this thesis, we used Pearson correlation to compute the edge weights between the nodes (Sec. 3.1.3, Eq. 3.5). These weights represent the similarity of time-series attributes. This similarity measure does not take into account the causality of time series attributes of the nodes in the graph. Also, it does not consider negative co-activation of the attributes, which might be desirable in some applications. One can overcome this limitation by using a different similarity function (Eq. 3.5), which should be chosen based on the desired type and nature of the detected patterns.

Other applications of the proposed pattern detection approach. Going beyond collective behavior analysis in web and social networks, we can think of other types of dynamic patterns in different graph-structured datasets. Weather patterns can be detected in the measurements collected by a network of meteorological stations. Sensor networks exhibit patterns of activity in the Internet of Things that can be related to an anomalous event or an accident in the network. Detection of patterns that appear in transportation networks can be used to improve navigation systems. These are just a few examples of applications where our graph-based pattern detection algorithm can be potentially useful.

In this thesis, we focused on a similarity-based attention mechanism (Sec. 3.1.2). However, we can use other types of attention function based on application requirements, domain expertise, and prior knowledge of the data. Though our approach has not yet been applied to these scenarios, the universality of the system makes it an excellent candidate for future applications.

Graph-based forecasting of activity patterns. In Section 4.3, we have demonstrated how activity patterns propagate from one node to another in the network. It would be interesting to see how accurately we can predict the emergence of the pattern based on the activity attributes from just a few nodes. For instance, such a forecasting model would be useful for automated web application scaling, when there is a need to seamlessly scale up resources in response to an increase in user activity. Aside from web applications, such a forecasting model could also be used in memory controllers and storage devices that are normally partitioned into storage areas or bins. We can represent these partitions as a graph structure. Each data partition is accessed with a certain frequency, which represents the dynamic attributes of each node in the graph. Detecting partition access patterns with further forecasting of access frequency could help to speed up memory operations in such devices.

Other applications of the data processing framework. Initially, the creation of the framework was influenced by the absence of a large-scale dataset that would allow us to compare the performance of dynamic pattern detection algorithms. Eventually, we developed the framework into a more versatile set of tools, speeding up the research pipeline when working on Wikipedia data (Sec. 6.1). In addition to the use cases presented in this thesis (Chapters 4 and 5), the framework can also be used in other applications.

We can extend the dataset with other attributes that characterize Wikipedia articles. For example, we could add Wikidata properties to the nodes, providing more categorical and structured information about Wikipedia articles. That would allow us to query more fine-grained subgraphs of Wikipedia articles, opening new avenues for research in other areas such as sociology and digital humanities.

Another application is to use our framework to create GNN benchmarks for graph or node classification tasks. In Chapter 5, we have extracted subgraphs of Wikipedia articles that describe particular events. Every node in the subgraphs has a topic label, such as sports, movies, music, STEM, politics, and so on. These labels can be used in node classification tasks to predict missing node topics based on the structure of their neighborhood. This also allows us to assign general topics to each subgraph based on the topics of the articles it is composed of. Then, we collect these subgraphs into a labeled dataset and use them for graph classification tasks.

Cross-lingual trending topic detection on Wikipedia. In Chapter 5, we used our approach to detect and compare general trends across multiple language editions of Wikipedia. In that study, we identified trending topics in each language edition separately. In addition to giving access to separate language editions, our data processing framework (Sec. 6.1) allows working with the web network of Wikipedia composed of pages that are connected by langlinks, the links that connect articles with their counterparts in other languages. Studying the patterns in the langlinks network allows focusing on general multilingual trends and specifically on the topics that Wikipedia readers intentionally explore and compare across multiple languages.

Detection of filter bubbles created by personalized recommendation systems. Even though social networks were created to connect us, when used maliciously, they can provide a set of powerful instruments that divide, radicalize, and polarize society [161–163]. Personalized recommendation algorithms play a crucial role as they are responsible for the creation of filter bubbles that often create the illusion of a single correct point of view. There is an increasing need for algorithms that uncover and visually demonstrate opinion biases created by social networks. We can use our approach to study how personalized recommendation systems affect dynamic patterns of user activity in social networks. To do that, we need to compare the detected activity patterns before and after the deployment of a recommendation system. Such studies would help to quantify the filter bubble effect and to emphasize the need for more responsible applications of personalized recommendation algorithms to make the internet a more neutral and inclusive communication environment.

Even though the principal goal of this thesis was to advance the field of *graph machine learning*, we crossed several scientific boundaries towards the end of our journey. To create our pattern detection approach, we borrowed intuitions about learning and memory from *neuroscience*. We were inspired by *social science* when applying our methods to digital traces of humans and analyzing dynamic patterns of their collective behavior. Combinations of intuitions and methodologies derived from these sciences allowed us to look at graph machine learning problems from an unconventional perspective with unique insights. This helped us to develop original approaches for graph-structured data analysis, bringing us back to Patricia Churchland's statement: "It is now evident that where one discipline ends and the other begins no longer matters" [1].

8 Science communication, outreach, and contributions

8.1 Conference papers and journal publications

- **MDPI Algorithms (30 October 2020).** Spikyball sampling: exploring large networks via an inhomogeneous filtered diffusion [164].
- **TheWebConf'20.** What is trending on Wikipedia? capturing trends and language biases across Wikipedia editions [33].
- **TheWebConf'19.** Anomaly detection in the dynamics of web and social networks [31].
- **TheWebConf'19.** A graph-based dataset for Wikipedia research [34].
- **ArXiV'17.** Wikipedia graph mining: dynamic structure of collective memory [32].

8.2 Datasets

- Wikipedia graph dataset and pagecounts preprocessing toolkit [165].
- Enron email time-series network [166].
- Wikipedia. Events and collective memory detection dataset [167].

8.3 Posters and exhibitions

- **SPARS 2017.** Graph-based echo-state networks.
- **Applied Machine Learning Days (AMLD) 2017.** Graph-based echo-state networks with applications to NLP and image classification.
- **AMLD 2018.** Wikipedia graph mining: dynamic structure of collective memory.
- **EPFL Workshop on Graph ML 2019.** Anomaly detection in the dynamics of web and social networks.
- **EPFL Digital Humanities Center inauguration.** Interactive data visualisation of Wikipedia user activity trends[I].

[I] Wikipedia user activity visualization: https://wiki-insights.epfl.ch/wikitrends/

8.4 Talks

• **AMLD. January 2018 (Lightning talks session).** Wikipedia graph mining: dynamic structure of collective memory.
• **Priberam. May 2018 (Invited Talk).** Collective memory and anomaly detection.
• **ENS Lyon. November 2018 (Invited Talk).** Anomaly detection in the dynamics of web and social networks.
• **Wikimania. August 2019 (Research Track).** Wikipedia graph mining: dynamic structure of collective memory.
• **L3S Research Seminar. November 2020 (Invited talk).** Dynamic pattern recognition in large-scale graphs.

8.5 Media coverage

• **HackerNews. Top 10.** Wikipedia graph mining: dynamic structure of collective memory.
• **EPFL MediaCom.** What can Wikipedia tell us about human interaction?[II].

[II] EPFL MediaCom interview: https://actu.epfl.ch/news/what-can-wikipedia-tell-us-about-human-interacti-3/

Bibliography

[1] P. S. Churchland, *Neurophilosophy: Toward a unified science of the mind-brain*. MIT press, 1989.

[2] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data", *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[3] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs", in *Advances in neural information processing systems*, 2017, pp. 1024–1034.

[4] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, *et al.*, "Relational inductive biases, deep learning, and graph networks", *arXiv preprint arXiv:1806.01261*, 2018.

[5] M. Newman, *Networks*. Oxford university press, 2018.

[6] I. Chami, S. Abu-El-Haija, B. Perozzi, C. Ré, and K. Murphy, "Machine learning on graphs: a model and comprehensive taxonomy", *arXiv preprint arXiv:2005.03675*, 2020.

[7] W. L. Hamilton, *Graph Representation Learning*. Morgan & Claypool, 2020.

[8] L. Freeman, "The development of social network analysis", *A Study in the Sociology of Science*, vol. 1, p. 687, 2004.

[9] A. Comte, *The positive philosophy of Auguste Comte*. Calvin Blanchard, 1855.

[10] G. LeBon, *The crowd. new brunswick*, 1995.

[11] G. Simmel, *On individuality and social forms: Selected writings*. University of Chicago Press, 1971.

[12] S. Wasserman, K. Faust, *et al.*, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

[13] J. L. Moreno, "Who shall survive? foundations of sociometry, group psychotherapy and socio-drama", 1953.

[14] E. Forsyth and L. Katz, "A matrix approach to the analysis of sociometric data: preliminary report", *Sociometry*, vol. 9, no. 4, pp. 340–347, 1946.

[15] F. Harary and R. Z. Norman, *Graph theory as a mathematical model in social science*, 2. University of Michigan, Institute for Social Research Ann Arbor, 1953.

[16] S. Milgram, "The small world problem", *Psychology today*, vol. 2, no. 1, pp. 60–67, 1967.

[17] W. W. Zachary, "An information flow model for conflict and fission in small groups", *Journal of anthropological research*, vol. 33, no. 4, pp. 452–473, 1977.

[18] J. A. Davis, "The davis/holland/leinhardt studies: an overview", in *Perspectives on social network research*, Elsevier, 1979, pp. 51–62.

[19] J. P. Boyd, *Social semigroups. a unified theory of scaling and blockmodelling as applied to social networks. fairfax*, 1991.

[20] P. Pattison and P. Philippa, *Algebraic models for social networks.* Cambridge University Press, 1993, vol. 7.

[21] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks", *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[22] D. M. Lazer, A. Pentland, D. J. Watts, S. Aral, S. Athey, N. Contractor, D. Freelon, S. Gonzalez-Bailon, G. King, H. Margetts, *et al.*, "Computational social science: obstacles and opportunities", *Science*, vol. 369, no. 6507, pp. 1060–1062, 2020.

[23] L. Manovich, "Trending: the promises and the challenges of big social data", *Debates in the digital humanities*, vol. 2, no. 1, pp. 460–475, 2011.

[24] R. Tinati, S. Halford, L. Carr, and C. Pope, "Big data: methodological challenges and approaches for sociological analysis", *Sociology*, vol. 48, no. 4, pp. 663–681, 2014.

[25] I. Sen, F. Floeck, K. Weller, B. Weiss, and C. Wagner, "A total error framework for digital traces of humans", *arXiv preprint arXiv:1907.08228*, 2019.

[26] R. Garcıéa-Gavilanes, A. Mollgaard, M. Tsvetkova, and T. Yasseri, "The memory remains: understanding collective memory in the digital age", *Science Advances*, vol. 3, no. 4, e1602368, 2017.

[27] N. Kanhabua, T. N. Nguyen, and C. Niederée, "What triggers human remembering of events? a large-scale analysis of catalysts for collective memory in wikipedia", in *Digital Libraries (JCDL), 2014 IEEE/ACM Joint Conference on*, IEEE, 2014, pp. 341–350.

[28] M. Ferron and P. Massa, "Studying collective memories in wikipedia", *Journal of Social Theory*, vol. 3, no. 4, pp. 449–466, 2011.

[29] M. Ferron, "Collective memories in wikipedia", PhD thesis, University of Trento, 2012.

[30] B. Yucesoy and A.-L. Barabási, "Untangling performance from success", *EPJ Data Science*, vol. 5, no. 1, p. 17, 2016.

[31] V. Miz, B. Ricaud, K. Benzi, and P. Vandergheynst, "Anomaly detection in the dynamics of web and social networks using associative memory", in *The World Wide Web Conference*, 2019, pp. 1290–1299.

[32] V. Miz, K. Benzi, B. Ricaud, and P. Vandergheynst, "Wikipedia graph mining: dynamic structure of collective memory", *arXiv preprint arXiv:1710.00398*, 2017.

[33] V. Miz, J. Hanna, N. Aspert, B. Ricaud, and P. Vandergheynst, “What is trending on wikipedia? capturing trends and language biases across wikipedia editions”, in *Companion Proceedings of the Web Conference 2020*, 2020, pp. 794–801.

[34] N. Aspert, V. Miz, B. Ricaud, and P. Vandergheynst, “A graph-structured dataset for wikipedia research”, in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 1188–1193.

[35] D. Chakrabarti and C. Faloutsos, “Graph mining: laws, generators, and algorithms”, *ACM computing surveys (CSUR)*, vol. 38, no. 1, 2–es, 2006.

[36] C. Aggarwal and K. Subbian, “Evolutionary network analysis: a survey”, *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, p. 10, 2014.

[37] L. Akoglu, H. Tong, and D. Koutra, “Graph based anomaly detection and description: a survey”, *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[38] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. F. Samatova, “Anomaly detection in dynamic networks: a survey”, *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 3, pp. 223–247, 2015.

[39] G. Atluri, A. Karpatne, and V. Kumar, “Spatio-temporal data mining: a survey of problems and methods”, *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, p. 83, 2018.

[40] I. Scholtes, “Harnessing complex structures and collective dynamics in large networked computing systems”, 2012.

[41] I. Scholtes and M. Esch, “Complex structures and collective dynamics in networked systems: foundations for self-adaptation and self-organization”, in *2014 IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems Workshops*, IEEE, 2014, pp. 1–2.

[42] R. Lambiotte and N. Masuda, *A guide to temporal networks.* World Scientific, 2016, vol. 4.

[43] R. Lambiotte, M. Rosvall, M. Schaub, I. Scholtes, and J. Xu, “Beyond graph mining: higher-order data analytics for temporal network data”, in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &*, vol. 38.

[44] L. Kovanen, M. Karsai, K. Kaski, J. Kertész, and J. Saramäki, “Temporal motifs in time-dependent networks”, *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2011, no. 11, P11005, 2011.

[45] G. Miritello, E. Moro, and R. Lara, “Dynamical strength of social ties in information spreading”, *Physical Review E*, vol. 83, no. 4, p. 045 102, 2011.

[46] B. Mitra, L. Tabourier, and C. Roth, “Intrinsically dynamic network communities”, *Computer Networks*, vol. 56, no. 3, pp. 1041–1053, 2012.

[47] R. Pfitzner, I. Scholtes, A. Garas, C. J. Tessone, and F. Schweitzer, “Betweenness preference: quantifying correlations in the topological dynamics of temporal networks”, *Physical review letters*, vol. 110, no. 19, p. 198 701, 2013.

[48] T. Weng, J. Zhang, M. Small, R. Zheng, and P. Hui, “Memory and betweenness preference in temporal networks induced from time series”, *Scientific reports*, vol. 7, p. 41 951, 2017.

[49] M. G. Rabbat, M. A. Figueiredo, and R. D. Nowak, “Network inference from co-occurrences”, *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 4053–4068, 2008.

[50] I. Scholtes, N. Wider, R. Pfitzner, A. Garas, C. J. Tessone, and F. Schweitzer, “Causality-driven slow-down and speed-up of diffusion in non-markovian temporal networks”, *Nature communications*, vol. 5, no. 1, pp. 1–9, 2014.

[51] I. Scholtes, N. Wider, and A. Garas, “Higher-order aggregate networks in the analysis of temporal networks: path structures and centralities”, *The European Physical Journal B*, vol. 89, no. 3, pp. 1–15, 2016.

[52] E. Wu, W. Liu, and S. Chawla, “Spatio-temporal outlier detection in precipitation data”, in *Knowledge discovery from sensor data*, Springer, 2010, pp. 115–133.

[53] C.-T. Lu, Y. Kou, J. Zhao, and L. Chen, “Detecting and tracking regional outliers in meteorological data”, *Information Sciences*, vol. 177, no. 7, pp. 1609–1632, 2007.

[54] J. H. Faghmous, M. Uluyol, L. Styles, M. Le, V. Mithal, S. Boriah, and V. Kumar, “Multiple hypothesis object tracking for unsupervised self-learning: an ocean eddy tracking application.”, in *AAAI*, 2013.

[55] X. Gao, Q. Zheng, D. A. Vega-Oliveros, L. Anghinoni, and L. Zhao, “Temporal network pattern identification by community modelling”, *Scientific Reports*, vol. 10, no. 1, pp. 1–12, 2020.

[56] H. N. Chaudhry, A. Margara, and M. Rossi, “Temporal pattern recognition in large scale graphs”, in *Proceedings of the 13th ACM International Conference on Distributed and Event-based Systems*, 2019, pp. 250–251.

[57] B. Klimt and Y. Yang, “The enron corpus: a new dataset for email classification research”, in *European Conference on Machine Learning*, Springer, 2004, pp. 217–226.

[58] H. Wang, M. Tang, Y. Park, and C. E. Priebe, “Locality statistics for anomaly detection in time series of graphs”, *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 703–717, 2014.

[59] C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park, “Scan statistics on enron graphs”, *Computational & Mathematical Organization Theory*, vol. 11, no. 3, pp. 229–247, 2005.

[60] D. Koutra, J. T. Vogelstein, and C. Faloutsos, “Deltacon: a principled massive-graph similarity function”, in *Proceedings of the 2013 SIAM International Conference on Data Mining*, SIAM, 2013, pp. 162–170.

[61] X. Wan, E. Milios, N. Kalyaniwalla, and J. Janssen, “Link-based event detection in email communication networks”, in *Proceedings of the 2009 ACM symposium on Applied Computing*, 2009, pp. 1506–1510.

[62] P. Moriano, J. Finke, and Y.-Y. Ahn, “Community-based event detection in temporal networks”, *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019.

[63] S. Rayana and L. Akoglu, “Less is more: building selective anomaly ensembles”, *Acm transactions on knowledge discovery from data (tkdd)*, vol. 10, no. 4, pp. 1–33, 2016.

[64] Y. Park, C. E. Priebe, and A. Youssef, “Anomaly detection in time series of graphs using fusion of graph invariants”, *IEEE journal of selected topics in signal processing*, vol. 7, no. 1, pp. 67–75, 2012.

[65] C. E. Priebe, Y. Park, D. J. Marchette, J. M. Conroy, J. Grothendieck, and A. L. Gorin, “Statistical inference on attributed random graphs: fusion of graph features and content: an experiment on time series of enron graphs”, *Computational statistics & data analysis*, vol. 54, no. 7, pp. 1766–1776, 2010.

[66] A. Spoerri, “What is popular on wikipedia and why?”, *First Monday*, vol. 12, no. 4, 2007.

[67] J. Lehmann, C. Müller-Birn, D. Laniado, M. Lalmas, and A. Kaltenbrunner, “Reader preferences and behavior on wikipedia”, in *Proceedings of the 25th ACM conference on Hypertext and social media*, ACM, 2014, pp. 88–97.

[68] P. Singer, F. Lemmerich, R. West, L. Zia, E. Wulczyn, M. Strohmaier, and J. Leskovec, “Why we read wikipedia”, in *Proceedings of the 26th International Conference on World Wide Web*, International World Wide Web Conferences Steering Committee, 2017, pp. 1591–1600.

[69] G. Domingues and C. Teixeira Lopes, “Characterizing and comparing portuguese and english wikipedia medicine-related articles”, in *Companion Proceedings of The 2019 World Wide Web Conference*, ACM, 2019, pp. 1203–1207.

[70] R. Garcıéa-Gavilanes, M. Tsvetkova, and T. Yasseri, “Dynamics and biases of online attention: the case of aircraft crashes”, *Royal Society open science*, vol. 3, no. 10, p. 160 460, 2016.

[71] T. Yasseri, A. Spoerri, M. Graham, and J. Kertész, “The most controversial topics in wikipedia”, *Global Wikipedia: International and cross-cultural issues in online collaboration*, vol. 25, 2014.

[72] E. S. Callahan and S. C. Herring, “Cultural bias in wikipedia content on famous persons”, *Journal of the American society for information science and technology*, vol. 62, no. 10, pp. 1899–1915, 2011.

[73] P. Laufer, C. Wagner, F. Flöck, and M. Strohmaier, “Mining cross-cultural relations from wikipedia: a study of 31 european food cultures”, in *Proceedings of the ACM Web Science Conference*, ACM, 2015, p. 3.

[74] A. Samoilenko, F. Lemmerich, K. Weller, M. Zens, and M. Strohmaier, “Analysing timelines of national histories across wikipedia editions: a comparative computational approach”, in *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[75] M. Gabella, “Cultural structures of knowledge from wikipedia networks of first links”, *IEEE Transactions on Network Science and Engineering*, 2018.

[76] F. Lemmerich, D. Sáez-Trumper, R. West, and L. Zia, “Why the world reads wikipedia: beyond english speakers”, in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, ACM, 2019, pp. 618–626.

[77] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering”, in *Advances in neural information processing systems*, 2016, pp. 3844–3852.

[78] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks”, *arXiv preprint arXiv:1609.02907*, 2016.

[79] L. Zheng, Z. Li, J. Li, Z. Li, and J. Gao, “Addgraph: anomaly detection in dynamic graph using attention-based temporal gcn.”, in *IJCAI*, 2019, pp. 4419–4425.

[80] L. Cai, Z. Chen, C. Luo, J. Gui, J. Ni, D. Li, and H. Chen, “Structural temporal graph neural networks for anomaly detection in dynamic graphs”, *arXiv preprint arXiv:2005.07427*, 2020.

[81] R. Zhang, Y. Hao, D. Yu, W.-C. Chang, G. Lai, and Y. Yang, “Explainable unsupervised change-point detection via graph neural networks”, *arXiv preprint arXiv:2004.11934*, 2020.

[82] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate”, *arXiv preprint arXiv:1409.0473*, 2014.

[83] V. Mnih, N. Heess, A. Graves, *et al.*, “Recurrent models of visual attention”, in *Advances in neural information processing systems*, 2014, pp. 2204–2212.

[84] J. Ba, V. Mnih, and K. Kavukcuoglu, “Multiple object recognition with visual attention”, *arXiv preprint arXiv:1412.7755*, 2014.

[85] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: neural image caption generation with visual attention”, in *International conference on machine learning*, 2015, pp. 2048–2057.

[86] J. B. Lee, R. A. Rossi, S. Kim, N. K. Ahmed, and E. Koh, “Attention models in graphs: a survey”, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 6, pp. 1–25, 2019.

[87] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks”, *arXiv preprint arXiv:1710.10903*, 2017.

[88] K. K. Thekumparampil, C. Wang, S. Oh, and L.-J. Li, “Attention-based graph neural network for semi-supervised learning”, *arXiv preprint arXiv:1803.03735*, 2018.

[89] J. B. Lee, R. Rossi, and X. Kong, “Graph classification using structural attention”, in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1666–1674.

[90] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, “Gram: graph-based attention model for healthcare representation learning”, in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 787–795.

[91] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, and J. Tang, “Session-based social recommendation via dynamic graph attention networks”, in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, pp. 555–563.

[92] S. Ge, C. Wu, F. Wu, T. Qi, and Y. Huang, “Graph enhanced representation learning for news recommendation”, in *Proceedings of The Web Conference 2020*, 2020, pp. 2863–2869.

[93] Y. Xiao, A. Krishnan, and H. Sundaram, “Discovering strategic behaviors for collaborative content-production in social networks”, in *Proceedings of The Web Conference 2020*, 2020, pp. 2078–2088.

[94] C. Wang and B. Wang, “An end-to-end topic-enhanced self-attention network for social emotion classification”, in *Proceedings of The Web Conference 2020*, 2020, pp. 2210–2219.

[95] L. Yang, F. Wu, J. Gu, C. Wang, X. Cao, D. Jin, and Y. Guo, “Graph attention topic modeling network”, in *Proceedings of The Web Conference 2020*, 2020, pp. 144–154.

[96] S. He and K. G. Shin, “Towards fine-grained flow forecasting: a graph attention approach for bike sharing systems”, in *Proceedings of The Web Conference 2020*, 2020, pp. 88–98.

[97] J. Weston, S. Chopra, and A. Bordes, “Memory networks”, *arXiv preprint arXiv:1410.3916*, 2014.

[98] S. Sukhbaatar, J. Weston, R. Fergus, *et al.*, “End-to-end memory networks”, in *Advances in neural information processing systems*, 2015, pp. 2440–2448.

[99] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, L. Gruber, M. Holzleitner, M. Pavlović, G. K. Sandve, V. Greiff, *et al.*, “Hopfield networks is all you need”, *arXiv preprint arXiv:2008.02217*, 2020.

[100] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities”, *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[101] M. Widrich, B. Schäfl, H. Ramsauer, M. Pavlović, L. Gruber, M. Holzleitner, J. Brandstetter, G. K. Sandve, V. Greiff, S. Hochreiter, *et al.*, “Modern hopfield networks and attention for immune repertoire classification”, *arXiv preprint arXiv:2007.13505*, 2020.

[102] K. Benzi, B. Ricaud, and P. Vandergheynst, “Principal patterns on graphs: discovering coherent structures in datasets.”, *IEEE Trans. Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 160–173, 2016.

[103] A. Griffa, B. Ricaud, K. Benzi, X. Bresson, A. Daducci, P. Vandergheynst, J.-P. Thiran, and P. Hagmann, “Transient networks of spatio-temporal connectivity map communication pathways in brain functional systems”, *NeuroImage*, vol. 155, pp. 490–502, 2017.

[104] W. Yu, C. C. Aggarwal, S. Ma, and H. Wang, “On anomalous hotspot discovery in graph streams”, in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, IEEE, 2013, pp. 1271–1276.

[105] E. I. Altman, “Financial ratios, discriminant analysis and the prediction of corporate bankruptcy”, *The journal of finance*, vol. 23, no. 4, pp. 589–609, 1968.

[106] J.-P. v. Brakel. (2014). Robust peak detection algorithm (using z-scores), [Online]. Available: https://stackoverflow.com/a/22640362 (visited on 06/22/2020).

[107] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[108] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks”, *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, P10008, 2008.

[109] V. A. Traag, L. Waltman, and N. J. van Eck, “From louvain to leiden: guaranteeing well-connected communities”, *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[110] A. H. Salavati, K. R. Kumar, and A. Shokrollahi, “Nonbinary associative memory with exponential pattern retrieval capacity and iterative learning”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 3, pp. 557–570, 2013.

[111] J. E. Gonzalez, R. S. Xin, A. Dave, D. Crankshaw, M. J. Franklin, and I. Stoica, “Graphx: graph processing in a distributed dataflow framework.”, in *OSDI*, vol. 14, 2014, pp. 599–613.

[112] R. S. Xin, J. E. Gonzalez, M. J. Franklin, and I. Stoica, “Graphx: a resilient distributed graph system on spark”, in *First International Workshop on Graph Data Management Experiences and Systems*, ACM, 2013, p. 2.

[113] R. McEliece, E. Posner, E. Rodemich, and S. Venkatesh, “The capacity of the hopfield associative memory”, *IEEE transactions on Information Theory*, vol. 33, no. 4, pp. 461–482, 1987.

[114] K. D. Miller and D. J. MacKay, “The role of constraints in hebbian learning”, *Neural computation*, vol. 6, no. 1, pp. 100–126, 1994.

[115] R. Tinati, M. Luczak-Roesch, and W. Hall, “Finding structure in wikipedia edit activity: an information cascade approach”, in *Proceedings of the 25th International Conference Companion on World Wide Web*, International World Wide Web Conferences Steering Committee, 2016, pp. 1007–1012.

[116] R. Tinati, M. Luczak-Roesch, W. Hall, and N. Shadbolt, “More than an edit: using transcendental information cascades to capture hidden structure in wikipedia”, in *Proceedings of the 25th International Conference Companion on World Wide Web*, 2016, pp. 115–116.

[117] P. Agarwal, M. Redi, N. Sastry, E. Wood, and A. Blick, “Wikipedia and westminster: quality and dynamics of wikipedia pages about uk politicians”, in *Proceedings of the 31st ACM Conference on Hypertext and Social Media*, 2020, pp. 161–166.

[118] M. Mongiovi, P. Bogdanov, R. Ranca, E. E. Papalexakis, C. Faloutsos, and A. K. Singh, "Netspot: spotting significant anomalous regions on dynamic networks", in *Proceedings of the 2013 SIAM International Conference on Data Mining*, SIAM, 2013, pp. 28–36.

[119] M. Mongiovi, P. Bogdanov, and A. K. Singh, "Mining evolving network processes", in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, IEEE, 2013, pp. 537–546.

[120] M. Halbwachs, *Les cadres sociaux de la mémoire*. Albin Michel, 2013.

[121] J. Assmann and J. Czaplicka, "Collective memory and cultural identity", *New German Critique*, no. 65, pp. 125–133, 1995.

[122] J. A. Barash, *Collective Memory and the Historical Past*. University of Chicago Press, 2016.

[123] A. Coman, A. D. Brown, J. Koppel, and W. Hirst, "Collective memory from a psychological perspective", *International Journal of Politics, Culture, and Society IJPS*, vol. 22, no. 2, pp. 125–141, 2009.

[124] M. Ferron and P. Massa, "Psychological processes underlying wikipedia representations of natural and manmade disasters", in *Proceedings of the Eighth Annual International Symposium on Wikis and Open Collaboration*, ACM, 2012, p. 2.

[125] D. Allport, "Distributed memory, modular subsystems and dysphasia", *Current perspectives in dysphasia*, pp. 32–60, 1985.

[126] R. West and J. Pfeffer, "Armed conflicts in online news: a multilingual study.", in *ICWSM*, 2017, pp. 309–318.

[127] P. Behnen, R. Kessler, F. Kruse, J. Schoenmakers, S. Zerr, and J. Gómez, "White paper: evidence, scale, and patterns of systematic inconsistencies in google trends data", Jun. 2020. DOI: 10.13140/RG.2.2.26974.66880.

[128] D. M. Russell, *The Joy of Search: A Google Insider's Guide to Going Beyond the Basics*. MIT Press, 2019.

[129] B. Hecht and D. Gergle, "The tower of babel meets web 2.0: user-generated content and its applications in a multilingual context", in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2010, pp. 291–300.

[130] P. Bao, B. Hecht, S. Carton, M. Quaderi, M. Horn, and D. Gergle, "Omnipedia: bridging the wikipedia language gap", in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 1075–1084.

[131] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation", *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[132] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval", *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[133] B. Santorini, “Part-of-speech tagging guidelines for the Penn Treebank Project”, Department of Computer and Information Science, University of Pennsylvania, Tech. Rep. MS-CIS-90-47, 1990. [Online]. Available: ftp://ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz.

[134] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: pre-training of deep bidirectional transformers for language understanding”, *arXiv preprint arXiv:1810.04805*, 2018.

[135] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification”, in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Association for Computational Linguistics, 2017, pp. 427–431.

[136] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: bringing order to the web.”, Stanford InfoLab, Tech. Rep., 1999.

[137] M. H. Ribeiro, K. Gligorić, M. Peyrard, F. Lemmerich, M. Strohmaier, and R. West, “Sudden attention shifts on wikipedia following covid-19 mobility restrictions”, *arXiv preprint arXiv:2005.08505*, 2020.

[138] M. Baker, “Reproducibility crisis”, *Nature*, vol. 533, no. 26, pp. 353–66, 2016.

[139] X. Chen, S. Dallmeier-Tiessen, R. Dasler, S. Feger, P. Fokianos, J. B. Gonzalez, H. Hirvonsalo, D. Kousidis, A. Lavasa, S. Mele, *et al.*, “Open is not enough”, *Nature Physics*, vol. 15, no. 2, pp. 113–119, 2019.

[140] Various. (2013). Snap dataset - wikipedia hyperlink network, [Online]. Available: https://snap.stanford.edu/data/enwiki-2013.html (visited on 01/30/2019).

[141] ——, (2011). Snap dataset - wikipedia top categories, [Online]. Available: https://snap.stanford.edu/data/wiki-topcats.html (visited on 01/30/2019).

[142] C. Klymko, D. Gleich, and T. G. Kolda, “Using triangles to improve community detection in directed networks”, *arXiv preprint arXiv:1404.5874*, 2014.

[143] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, “Local higher-order graph clustering”, in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2017, pp. 555–564.

[144] B. Mitrevski, T. Piccardi, and R. West, “Wikihist. html: english wikipedia’s full revision history in html format”, in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 14, 2020, pp. 878–884.

[145] P. Mernyei and C. Cangea, “Wiki-cs: a wikipedia-based benchmark for graph neural networks”, *arXiv preprint arXiv:2007.02901*, 2020.

[146] C. Consonni, D. Laniado, and A. Montresor, “Wikilinkgraphs: a complete, longitudinal and multi-language dataset of the wikipedia link networks”, in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 13, 2019, pp. 598–607.

[147] T. Zesch and I. Gurevych, “Analysis of the wikipedia category graph for nlp applications”, in *Proceedings of the Second Workshop on TextGraphs: Graph-Based Algorithms for Natural Language Processing*, 2007, pp. 1–8.

[148] L. S. Buriol, C. Castillo, D. Donato, S. Leonardi, and S. Millozzi, “Temporal analysis of the wikigraph”, in *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, IEEE Computer Society, 2006, pp. 45–51.

[149] F. Bellomi and R. Bonato, “Network analysis for wikipedia”, in *proceedings of Wikimania*, 2005.

[150] R. West, A. Paranjape, and J. Leskovec, “Mining missing hyperlinks from human navigation traces: a case study of wikipedia”, in *Proceedings of the 24th international conference on World Wide Web*, International World Wide Web Conferences Steering Committee, 2015, pp. 1242–1252.

[151] H. Singh, R. West, and G. Colavizza, *Wikipedia citations: a comprehensive dataset of citations with identifiers extracted from english wikipedia*, 2020. arXiv: `2007.07022 [cs.DL]`.

[152] H. S. Moat, C. Curme, A. Avakian, D. Y. Kenett, H. E. Stanley, and T. Preis, “Quantifying wikipedia usage patterns before stock market moves”, *Scientific reports*, vol. 3, p. 1801, 2013.

[153] T. Yasseri, R. Sumi, A. Rung, A. Kornai, and J. Kertész, “Dynamics of conflicts in wikipedia”, *PloS one*, vol. 7, no. 6, e38869, 2012.

[154] M. Mestyán, T. Yasseri, and J. Kertész, “Early prediction of movie box office success based on wikipedia activity big data”, *PloS one*, vol. 8, no. 8, e71226, 2013.

[155] N. Aspert, V. Miz, and B. Ricaud. (2019). Wikipedia dataset, [Online]. Available: https://lts2.epfl.ch/Datasets/Wikipedia/ (visited on 01/31/2019).

[156] Wikimedia. (2019). Wikipedia dumps, [Online]. Available: https://dumps.wikimedia.org/ (visited on 01/30/2019).

[157] ——, (2019). Wikipedia pageviews dumps, [Online]. Available: https://dumps.wikimedia.org/other/pageviews/ (visited on 01/30/2019).

[158] A. London, J. Zsibrita, and R. Fear, “A knowledge-graph based taxonomy construction method”,

[159] A. Miyaguchi, S. Chakrabarti, and N. Garcia, “Forecasting wikipedia page views with graph embeddings”, 2019.

[160] K. R. Popper, *Objective knowledge.* Oxford University Press Oxford, 1972, vol. 360.

[161] S. Flaxman, S. Goel, and J. M. Rao, “Filter bubbles, echo chambers, and online news consumption”, *Public opinion quarterly*, vol. 80, no. S1, pp. 298–320, 2016.

[162] D. M. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, *et al.*, “The science of fake news”, *Science*, vol. 359, no. 6380, pp. 1094–1096, 2018.

[163] V. Miller, *Understanding digital culture*. SAGE Publications Limited, 2020.

[164] B. Ricaud, N. Aspert, and V. Miz, “Spikyball sampling: exploring large networks via an inhomogeneous filtered diffusion”, *Algorithms*, vol. 13, no. 11, p. 275, 2020.

[165] N. Aspert, V. Miz, B. Ricaud, and P. Vandergheynst, *SparkWiki: Wikipedia graph dataset and pagecounts pre-processing tools*, version 1.0, Zenodo, Jan. 2019. DOI: 10.1145/3308560.3316757. [Online]. Available: https://doi.org/10.1145/3308560.3316757.

[166] V. Miz, B. Ricaud, and P. Vandergheynst, *Enron email time-series network*, Aug. 2018. DOI: 10.5281/zenodo.1342353. [Online]. Available: https://doi.org/10.5281/zenodo.1342353.

[167] M. Volodymyr, B. Kirell, R. Benjamin, and V. Pierre, *Wikipedia. Events and collective memory detection dataset*, Zenodo, Sep. 2017. DOI: 10.5281/zenodo.886951. [Online]. Available: https://doi.org/10.5281/zenodo.886951.

Volodymyr MIZ

mizvol | in volodymyrmiz | blog.miz.space | @ mizvladimir@gmail.com

WORK EXPERIENCE

EPFL | Doctoral Assistant
Mar 2016 – Dec 2020 | Lausanne, Switzerland

- Selected as one of 5 Ph.D. students to do consulting in Machine Learning for IMD business school based on academic success and presentation skills.
- Led a team of 4 MSc students. Resulted in a publication [3].
- Gave lectures on data visualization for a class of ∼150 students (JavaScript, Gephi, ML on graphs).

FIRMENICH | ML Reseach Consultant
Jan 2020 - Jul 2020 | Lausanne, Switzerland

- Designed and implemented a Machine Learning model for odor classification (Python, PyTorch).
- Increased performance by 10-20% compared to conventional classification models.

IST LISBON | Research Intern
Apr 2018 – Jul 2018 | Lisbon, Portugal

- Optimized a network inference algorithm achieving 3x speedup over an existing implementation (Python).

ECHOSTAR | Software Engineer
Sep 2012 – Mar 2016 | Kharkiv, Ukraine

- Developed remote control services for 2M customers worldwide. Was promoted to Project Lead (SQL, Java).
- Worked on UI and features for an automated hardware testing suite (C#, JavaScript, HTML/CSS, OpenCV).

AWARDS

HACKERNEWS | Top 10, 1 Oct 2017
Oct 2017 | https://hckrnews.com/

- My research **blog post** on Wikipedia was highlighted in Top 10 on Hacker News and received 10K visits over 24h.

IT KHARKIV | Best startup in Cloud Computing
Nov 2013 | Kharkiv, Ukraine

- Selected among 4 out of 73 participants for Best Startup Presentation Award.

NURE | Faculty award
Jul 2013 | Kharkiv, Ukraine

- Selected as one of 21 students for exceptional success during graduate studies.

MICROELECTRONICS OLYMPIAD | Best result
Oct 2012 | Yerevan, Armenia

- Selected as one of 43 finalists worldwide and received Best Result(Ukraine) Award.

EDUCATION

EPFL
PhD in Electrical Engineering
Mar 2016 – Dec 2020 | Lausanne, Switzerland

UNIVERSITY OF RADIO ELECTRONICS
BSc, MSc in Computer Engineering
Sep 2008 - Jul 2013 | Kharkiv, Ukraine

RESEARCH

PATTERN DETECTION IN DYNAMIC GRAPHS | Researcher (PhD thesis)
Jan 2017 – Dec 2020

- Created and implemented an ML framework for dynamic pattern detection in large-scale dynamic networks [4] (Scala, Python, Apache Spark).

WIKIPEDIA RESEARCH PROJECTS | Researcher, Open Source Contributor
Sep 2016 – present

- **SparkWiki.** Co-created a large-scale data processing framework to facilitate research on Wikipedia graph and time-series data (Scala, Apache Spark, Neo4J) [1].
- **WikiTrends. [Video] [Demo]** Designed and implemented a distributed engine for trend detection in different Wikipedia language editions (Scala, Apache Spark, NLP) [3].
- **Collective memory.** Studied collective behavior of Wikipedia visitors, in particular the way people remember past events (Scala, Apache Spark, Python) [2].

DISINFORMATION IN MEDIA | Data Scientist
Jul 2018 – Sep 2020

- Implemented a web-based tool for Swiss journalists (RTS) to discover filter bubbles in YouTube recommendations on controversial content (Python, JavaScript).

OLFACTORY ML MODEL | ML Researcher
Apr 2019 – Dec 2020

- Created and implemented a Machine Learning model to classify odors. The model outperforms 5 baselines by up to 20% (Python, PyTorch, Deep Learning).

SKILLS

TECHNICAL SKILLS
• Python • Scala • JavaScript • SQL • Graph DBs • Statistics • Apache Spark • Visualization • Machine Learning Algorithms • Graph Theory • Research • Statistics • Data Mining • R • PyTorch • Deep Learning • NLP • Knowledge Graphs

SOFT SKILLS
• Leadership • Teaching • Public Speaking and Presentation • Multicultural Communication • Team player

LANGUAGES

Foreign: • English (C2) • French (B2) • German (A1)
Native: • Russian • Ukrainian

HOBBIES

• Basketball. Finalist of Ukrainian National League U16 (2007), champion of Swiss Vaudoise League (2018) and Cup (2019)

• Playing piano • Skiing • Mountaineering

SELECTED PUBLICATIONS

[1] N. Aspert, V. Miz, B. Ricaud, and P. Vandergheynst. A graph-structured dataset for wikipedia research. In *Companion Proceedings of The Web Conference 2019*, pages 1188–1193, 2019.

[2] V. Miz, K. Benzi, B. Ricaud, and P. Vandergheynst. Wikipedia graph mining: dynamic structure of collective memory. *arXiv preprint arXiv:1710.00398*, 2017.

[3] V. Miz, J. Hanna, N. Aspert, B. Ricaud, and P. Vandergheynst. What is trending on wikipedia? capturing trends and language biases across wikipedia editions. In *Companion Proceedings of The Web Conference 2020*. ACM, 2020.

[4] V. Miz, B. Ricaud, K. Benzi, and P. Vandergheynst. Anomaly detection in the dynamics of web and social networks using associative memory. In *Proceedings of The Web Conference 2020*, pages 1290–1299, 2019.

See full list on Google Scholar 🎓