PADERBORN UNIVERSITY
The University for the Information Society

Doctoral Dissertation

Vandalism Detection in Crowdsourced Knowledge Bases

A dissertation presented
by
Stefan Heindorf
to the
Faculty for Computer Science,
Electrical Engineering and Mathematics
of
Paderborn University

in partial fulfillment of the requirements
for the degree of
Dr. rer. nat.

Paderborn, Germany
November 2019

Dissertation

Vandalism Detection in Crowdsourced Knowledge Bases
Stefan Heindorf, Paderborn University
Paderborn, Germany, 2019

Reviewers

Prof. Dr. Gregor Engels, Paderborn University
Jun.-Prof. Dr. Martin Potthast, Leipzig University
Prof. Dr. Hannah Bast, University of Freiburg

Doctoral Committee

Prof. Dr. Gregor Engels, Paderborn University
Jun.-Prof. Dr. Martin Potthast, Leipzig University
Prof. Dr. Axel-Cyrille Ngonga Ngomo, Paderborn University
Jun.-Prof. Dr. Henning Wachsmuth, Paderborn University
Dr. Theodor Lettmann, Paderborn University

Abstract

Vandalism Detection in Crowdsourced Knowledge Bases

Information systems, such as question answering systems and web search engines, increasingly rely on crowdsourced knowledge bases to answer questions and to display important information about entities. While crowdsourcing enables the collection of vast amounts of information, it also brings along the problem of vandalism and damaging contributions. In this thesis, we focus on Wikidata, the largest structured, crowdsourced knowledge base on the web, and develop novel machine learning-based vandalism detectors to reduce the manual reviewing effort. To this end, we carefully develop large-scale vandalism corpora, vandalism detectors with high predictive performance, and vandalism detectors with low bias against certain groups of editors. We extensively evaluate our vandalism detectors in a number of settings, and we compare them to the state of the art represented by the Wikidata Abuse Filter and the Objective Revision Evaluation Service by the Wikimedia Foundation. Our best vandalism detector achieves an area under the curve of the receiver operating characteristics of 0.991, significantly outperforming the state of the art; our fairest vandalism detector achieves a bias ratio of only 5.6 compared to values of up to 310.7 of previous vandalism detectors. Overall, our vandalism detectors enable a conscious trade-off between predictive performance and bias and they might play an important role towards a more accurate and welcoming web in times of fake news and biased AI systems.

Zusammenfassung

Vandalism Detection in Crowdsourced Knowledge Bases

Informationssysteme wie Frage-Antwort-Systeme und Websuchmaschinen verwenden zunehmend crowdsourcing-basierte Wissensdatenbanken, um Fragen zu beantworten und wichtige Informationen über Entitäten anzuzeigen. Crowdsourcing ermöglicht zwar die Sammlung großer Informationsmengen, bringt aber auch das Problem von Vandalismus und schädlichen Beiträgen mit sich. In dieser Arbeit betrachten wir Wikidata, die größte strukturierte, crowdsourcing-basierte Wissensdatenbank im Web und entwickeln neuartige Vandalismusdetektoren mittels maschinellem Lernen, um den manuellen Prüfaufwand zu reduzieren. Dazu entwickeln wir große Vandalismuskorpora, Vandalismusdetektoren mit hoher prädiktiver Performanz und Vandalismusdetektoren mit geringer Voreingenommenheit gegenüber schützenswerten Editorengruppen. Wir evaluieren unseren Ansatz umfassend in zahlreichen Situationen und vergleichen ihn mit dem Stand der Technik, der durch den Wikidata Abuse Filter und den Objective Revision Evaluation Service der Wikimedia Foundation repräsentiert wird. Unser bester Vandalismusdetektor erreicht eine Fläche unter der Kurve der Receiver Operating Characteristics von 0,991 und übertrifft damit deutlich den Stand der Technik; unser fairster Vandalismusdetektor erreicht ein Bias-Verhältnis von lediglich 5,6 im Vergleich zu Werten von bis zu 310,7 vorheriger Vandalismusdetektoren. Insgesamt ermöglichen unsere Vandalismusdetektoren einen gezielten Kompromiss zwischen hoher prädiktiver Performanz und geringem Bias und sie könnten in Zeiten von Fake News und voreingenommenen KI-Systemen eine wichtige Rolle für die Richtigkeit der Informationen im Web spielen und zu einem freundlicheren Klima für Editoren beitragen.

Acknowledgments

This dissertation was written at Paderborn University in the research group Database and Information Systems of Prof. Dr. Gregor Engels. I would like to thank all the people who were part of my PhD journey. I would especially like to thank my supervisor Prof. Dr. Gregor Engels for his continuous guidance throughout my PhD, Jun.-Prof. Dr. Martin Potthast for the long-term collaboration, and Prof. Dr. Hannah Bast for serving as my reviewer. I thank Prof. Dr. Axel-Cyrille Ngonga Ngomo, Jun.-Prof. Dr. Henning Wachsmuth, and Dr. Theodor Lettmann for being members of my doctoral committee. I would like to thank my co-authors and colleagues for many valuable discussions. Thanks especially to Prof. Dr. Benno Stein, Dennis Wolters, and Yan Scholten. Finally, I would like to express my gratitude to my parents, family, and friends for their continuous support. Thank you all!

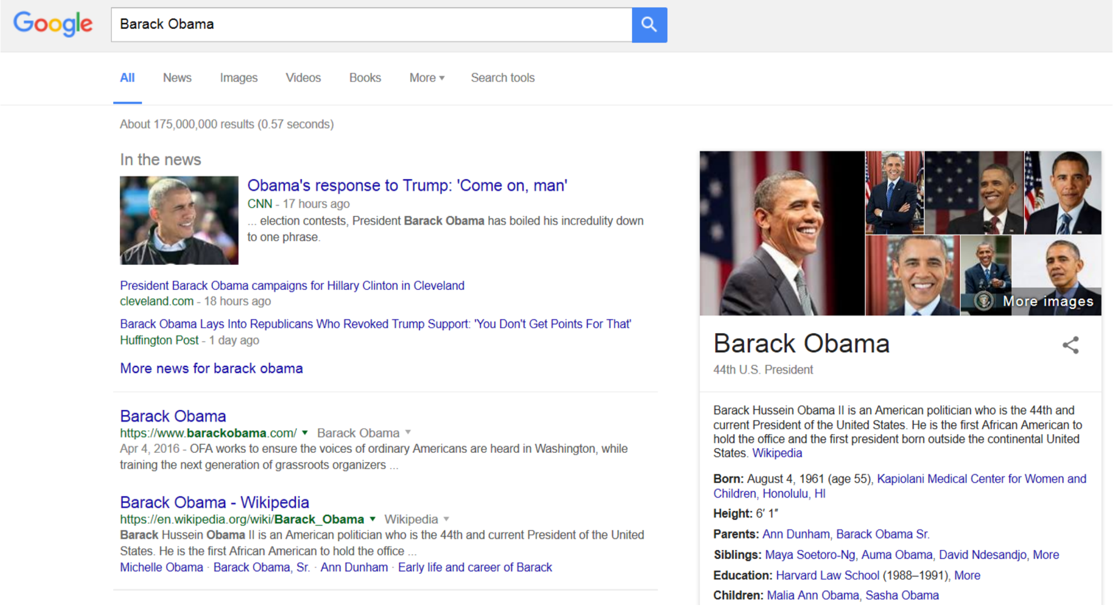
Contents

1
Introduction

Apple's Siri called chancellor Merkel a pig; Yahoo called Obama the founder of ISIS; Google insulted the Cardinals baseball team.[1] How can incidents like this happen? Tracing their causes leads to information systems, such as web search engines and question answering systems relying on crowdsourced knowledge bases. While crowdsourced knowledge bases offer vast amounts of valuable information, they sometimes get vandalized. Manually reviewing millions of contributions every month, however, places a high burden on the community of a knowledge base. For combatting vandalism and supporting reviewers, we develop automatic vandalism detectors in this thesis.

Section 1.1 outlines how crowdsourced knowledge bases are used by information systems and why we focus on the knowledge base Wikidata in this thesis; Section 1.2 describes the problem of vandalism in crowdsourced knowledge bases; Section 1.3 argues why we need novel approaches to detect vandalism automatically and what their requirements are; Section 1.4 identifies research challenges; and finally, Section 1.5 describes our contribution of novel machine learning-based vandalism detectors to detect vandalism automatically. This cumulative thesis is based on our publications (Heindorf et al., 2015, 2016, 2017a,b, 2019a,b).

[1] The sources of the examples are provided in Section 1.2.



Barack Obama

About 175,000,000 results (0.57 seconds)

In the news

Obama's response to Trump: 'Come on, man'
CNN - 17 hours ago
... election contests, President **Barack Obama** has boiled his incredulity down to one phrase.

President Barack Obama campaigns for Hillary Clinton in Cleveland
cleveland.com - 18 hours ago

Barack Obama Lays Into Republicans Who Revoked Trump Support: 'You Don't Get Points For That'
Huffington Post - 1 day ago

More news for barack obama

Barack Obama
https://www.**barackobama**.com/ Barack Obama
Apr 4, 2016 - OFA works to ensure the voices of ordinary Americans are heard in Washington, while training the next generation of grassroots organizers ...

Barack Obama - Wikipedia
https://en.wikipedia.org/wiki/**Barack_Obama** Wikipedia
Barack Hussein **Obama** II is an American politician who is the 44th and current President of the United States. He is the first African American to hold the office ...
Michelle Obama · Barack Obama, Sr. · Ann Dunham · Early life and career of Barack

Barack Obama
44th U.S. President

Barack Hussein Obama II is an American politician who is the 44th and current President of the United States. He is the first African American to hold the office and the first president born outside the continental United States. Wikipedia

Born: August 4, 1961 (age 55), Kapiolani Medical Center for Women and Children, Honolulu, HI
Height: 6′ 1″
Parents: Ann Dunham, Barack Obama Sr.
Siblings: Maya Soetoro-Ng, Auma Obama, David Ndesandjo, More
Education: Harvard Law School (1988–1991), More
Children: Malia Ann Obama, Sasha Obama

Figure 1.1: Google's search result page for the query 'Barack Obama' showing an example of a knowledge panel on the right-hand side (retrieved on October 15, 2016).

1.1 Crowdsourced Knowledge Bases in Information Systems

Information systems heavily rely on knowledge bases: web search engines display knowledge panels; virtual assistants answer questions; online encyclopedias display infoboxes; social networks normalize user input about cities, colleges, and movies; newspapers tag their articles; fact checkers double-check the correctness of information. Figure 1.1 shows one of the largest deployments of knowledge bases: Google extracts information about entities, such as people, places, movies, and books, from Wikipedia and Wikidata, and displays the information in knowledge panels. This often suffices to answer queries directly without users having to browse the links. Users have grown accustomed to these mechanisms, and expect the information to be correct.

In the following, we give a brief overview of what a knowledge base is, how knowledge bases can be categorized, and why we focus on the knowledge base Wikidata in this thesis. A knowledge base is "a database designed to meet the complex storage and retrieval requirements of computerized knowledge management, especially in support of artificial intelligence or expert systems."[2] Originally, knowledge bases were primarily employed by expert systems; nowadays, their primary applications have shifted towards

[2] https://en.wiktionary.org/wiki/knowledge_base

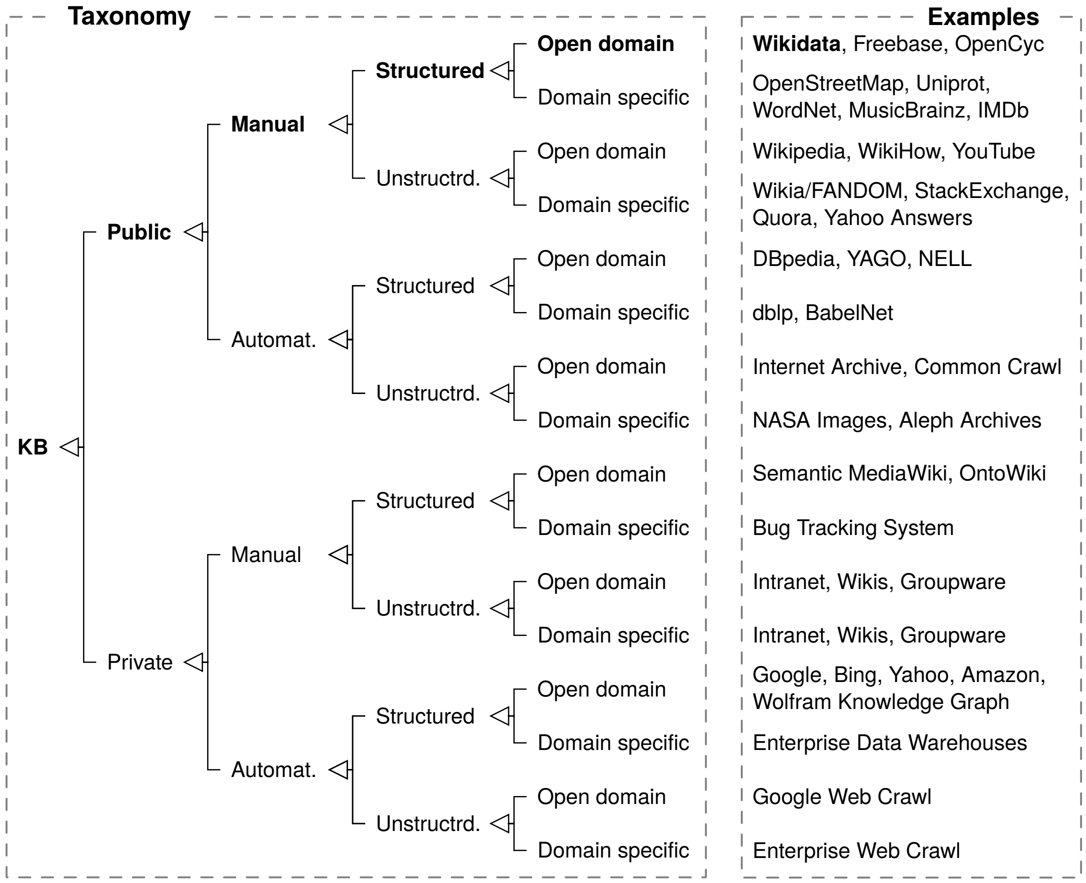

Taxonomy
Examples
KB
Public
Private
Manual
Automat.
Manual
Automat.
Structured
Unstructrd.
Structured
Unstructrd.
Structured
Unstructrd.
Structured
Unstructrd.
Open domain
Domain specific
Wikidata, Freebase, OpenCyc
OpenStreetMap, Uniprot, WordNet, MusicBrainz, IMDb
Wikipedia, WikiHow, YouTube
Wikia/FANDOM, StackExchange, Quora, Yahoo Answers
DBpedia, YAGO, NELL
dblp, BabelNet
Internet Archive, Common Crawl
NASA Images, Aleph Archives
Semantic MediaWiki, OntoWiki
Bug Tracking System
Intranet, Wikis, Groupware
Intranet, Wikis, Groupware
Google, Bing, Yahoo, Amazon, Wolfram Knowledge Graph
Enterprise Data Warehouses
Google Web Crawl
Enterprise Web Crawl

Figure 1.2: Taxonomy of knowledge bases (KBs)

information retrieval by web search engines and question answering by virtual assistants. Perhaps the most widely used knowledge bases as of today are Wikipedia and Wikidata, as they are publicly available and cover a wide range of domains relevant to a large audience. Figure 1.2 shows examples of knowledge bases and their categorization according to four key characteristics:

1. **Availability** It can be distinguished whether a knowledge base is *publicly* available to the research community or not. For example, Wikipedia and Wikidata are publicly available under permissive licenses, whereas the Google Knowledge Graph is not. *Private* knowledge bases, however, often include data from public knowledge bases; hence, improving the latter helps the former, too.

2. **Construction paradigm** Knowledge bases can be constructed—to various degrees—*manually* or *automatically.* For example, Wikidata and Freebase are manually constructed by the crowd, OpenCyc is manually constructed by (paid) editors, while DBpedia and YAGO are automatically constructed by extracting information from Wikipedia. Proprietary knowledge graphs such as those by Google, Bing, and Yahoo automatically combine data from multiple sources, including Wikipedia and Wikidata. Besides their primary construction principles, real-world knowledge bases often employ a combination of construction principles. For example, some Wikidata editors automate routine tasks with bots, and DBpedia's editors manually contribute extraction rules. Proprietary knowledge graphs that are constructed automatically offer facilities to manually overwrite results, e.g., to immediately correct high-profile errors making headlines in the news. Both manual and automatic knowledge bases face quality problems, albeit for different reasons. Manually constructed knowledge bases rely on the trustworthiness of their editors. Automatic knowledge bases rely on complex and error-prone heuristics. Many automatic knowledge bases extract data from Wikipedia, which is manually created by the crowd, thus inheriting the errors from the crowd and adding extraction errors. Manual knowledge bases often serve as high-quality training data for machine learning-based information extractors (Mintz et al., 2009; Dong et al., 2014).

3. **Knowledge representation** Knowledge bases can be distinguished by whether they contain *structured* data, e.g., in the form of subject-predicate-object triples, or *unstructured* data, such as web pages, natural language texts, audio, or video. Prominent examples of structured knowledge bases include Wikidata, DBpedia, and YAGO, whereas Wikipedia is an example of an unstructured knowledge base. Quora, YouTube, web crawls, and the Internet Archive can be considered unstructured knowledge bases, too. While unstructured knowledge bases can contain a range of information, this information is hardly accessible by machines.

4. **Domain specificity** Knowledge bases can be categorized by their domain specificity. While some knowledge bases cover *specific* topics, such as maps, proteins, movies, or music, *open domain* knowledge bases cover a wide range of domains, which are often relevant to web search engines and the general public.

In this thesis, we focus on Wikidata (Vrandečić and Krötzsch, 2014), the largest structured, crowdsourced knowledge base on the web. Structured knowledge bases are widely used in many applications ranging from Wikipedia infoboxes to web search engines and question answering systems. Manually creating a large-scale structured knowledge base, requires substantial effort: one of the first and most ambitious structured knowledge bases, CyC, was created by *experts* in over 900 person-years (Lenat and Guha, 1989; Lenat, 2008). For better scalability, a trend towards *crowdsourcing* knowledge can be observed—where the crowd refers to "a large group of people and especially from the online community rather than from traditional employees or suppliers."[3] For example, over 20,000 person-years were required for the English Wikipedia until 2011 alone (Geiger and Halfaker, 2013). Similarly, the *structured* knowledge base Wikidata is built by the crowd, and despite its recent launch in 2012, as of October 2019, Wikidata already contains over 60 million entities created in over 1 billion edits,[4] surpassing the English Wikipedia's 6 million articles created in about 900 million edits.[5] Until recently, another prominent knowledge base in this category was Freebase (Bollacker et al., 2008), which was originally developed by Metaweb and acquired by Google. However, Google has discontinued the project, and the data is being integrated into Wikidata (Pellissier Tanon et al., 2016). Similarly, the public subset of CyC, OpenCyc, is no longer available as of 2017, further raising the relevance of Wikidata.

Compared with other public knowledge bases such as DBpedia, YAGO, OpenCyc, and NELL, Wikidata contains the most entities and triples (Ringler and Paulheim, 2017), and although Freebase had 60 million entities and 3 billion triples when shut down,[6] Färber et al. (2017) found Wikidata to be more complete with respect to relevant entities from a golden set. In fact, Wikidata "shows an excellent performance for both well-known and rather unknown entities" (Färber et al., 2017), and it was found to have the highest overall quality among the public knowledge bases DBpedia, Freebase, OpenCyc, and YAGO, comparing them along quality dimensions, such as accuracy, completeness, consistency, and timeliness (Färber et al., 2017).

[3] https://www.merriam-webster.com/dictionary/crowdsource

[4] https://www.wikidata.org/wiki/Special:Statistics

[5] https://en.wikipedia.org/wiki/Special:Statistics

[6] https://web.archive.org/web/20160501004947/http://www.freebase.com/

1.2 Vandalism in Crowdsourced Knowledge Bases

Crowdsourcing projects such as Wikipedia have shown that the crowd can be trusted to create one of the largest encyclopedias in the world gathering "the sum of all human knowledge".[7] Allowing everybody to edit the knowledge base with little barriers—even without registering—encourages many people to contribute. However, this freedom-to-edit model occasionally leads to vandalism, which Wikidata defines as "deliberate attempt to damage or compromise the integrity of Wikidata."[8] This definition distinguishes intentional from unintentional damage, excluding the latter. However, for data consumers, the correctness of the data counts regardless of the intention of the editor. Hence, the literature often generalizes vandalism detection to damage detection (Kiesel et al., 2017). We follow this approach and use the terms damage and vandalism synonymously in this thesis. While some previous work focuses on detecting *vandals* (Kumar et al., 2015), we follow a more fine-grained approach and focus on *vandalism*, i.e., damaging *edits* instead of damaging *editors*, since edits are of varying quality and Wikidata data consumers are primarily interested in the quality of the data. Notable examples of vandalism that originated in a knowledge base and spread to information systems include:

- Apple's Siri calls chancellor Angela Merkel a pig[9]
- Yahoo's knowledge panels call Barack Obama the founder of ISIS[10]
- Google's knowledge panels insult the Cardinals baseball team.[11]

All of these examples were caused by vandalism in the underlying knowledge bases and hurt the companies with bad publicity. The examples show that vandalism often contains vulgar and bad words. Other types of vandalism include the removal of valuable content, the insertion of random keystrokes, sneaky vandalism that is hard to spot, as well as sloppy mistakes.

[7] https://en.wikiquote.org/wiki/Jimmy_Wales

[8] https://www.wikidata.org/wiki/Wikidata:Vandalism

[9] http://www.spiegel.de/netzwelt/gadgets/siri-beleidigt-angela-merkel-gefaelschter-wikipedia-eintrag-a-1054790.html

[10] https://www.mercurynews.com/2016/08/22/president-obama-founded-isis-according-to-yahoo/

[11] https://www.riverfronttimes.com/newsblog/2013/10/28/some-12-year-old-boy-has-hacked-the-cardinals-wikipedia-page

Wikidata shares the same problem of spreading false information to its data consumers, but its structured data intensifies the problem: (1) The data is used for many novel applications ranging from knowledge panels and question answering to fact-checking. (2) Question answering systems return a single answer without offering an easy opportunity for double-checking the provided information, thus increasing demands on data correctness. (3) Structured data is often used to infer new facts, e.g., by traversing the type hierarchy, the family tree of people, and biological taxonomies. Errors encountered on the path accumulate and must be avoided as far as possible. Examples of Wikidata vandalism include:

- Changing Barack Obama's description from "44th President of the United States of America" to "worst president ever"[12]
- Changing Barack Obama's type from "human" (Q5) to "extraterrestrial life" (Q181508)[13]
- Changing Barack Obama's spouse from "Michelle Obama" (Q13133) to "Peter Piper" (Q7176398)[14]

The first example affects the textual description of an item, which is part of an item's head. The second and third examples affect subject-predicate-object triples, which are part of an item's body. Wikidata does not enforce any constraints on head and body content and accepts everything that syntactically fits its data model.

Vandalism examples might be categorized according to different criteria. Regarding the content that is vandalized, we found that vandalism often affects famous people, such as politicians, soccer players, and musicians, whereas geographic places, such as cities, regions, rivers, and mountains are seldom vandalized. Regarding the registration status of editors, we found that the majority of vandalism originates from anonymous, unregistered users. However, banning anonymous users would not be a solution, since (1) most edits by anonymous users are benign,[15] (2) many new contributors start their editing career anonymously before registering,[16] (3) vandals could easily register, and (4) banning anonymous users contradicts the founding principles of Wikimedia.[17]

[12] https://www.wikidata.org/w/index.php?diff=prev&oldid=7375872

[13] https://www.wikidata.org/w/index.php?diff=prev&oldid=318726704

[14] https://www.wikidata.org/w/index.php?diff=prev&oldid=48347290

[15] https://en.wikipedia.org/wiki/Wikipedia:IPs_are_human_too

[16] https://en.wikipedia.org/wiki/Wikipedia:Perennial_proposals

[17] https://meta.wikimedia.org/wiki/Founding_principles

Further categorizations of vandalism might be performed according to data quality dimensions. Data quality can be broken down into quality dimensions, such as accuracy, completeness, consistency, and timeliness (Wang and Strong, 1996; Zaveri et al., 2013). Vandalism primarily affects the accuracy dimension of data quality, but other dimensions are affected, too. For example, removing valuable content reduces the completeness of the knowledge base, and the accuracy must always be assessed with respect to time, since, for example, the position held by a politician changes over time.

Another categorization might be based on motivational factors and personality traits of vandals. Shachaf and Hara (2010) studied people performing vandalism on Wikipedia (called trolls) by investigating examples of vandalism and interviewing Wikipedia administrators that often encounter vandalism. They identified three motivational factors: (1) boredom, attention seeking, and revenge, (2) fun and entertainment, and (3) damage to the community and other people. Buckels et al. (2014) investigated the relationship of internet trolls[18] to personality traits such as sadism, Machiavellianism, narcissism, psychopathy, finding the strongest association with sadism. These motivational factors and personality traits might be used to categorize vandals; however, we do not follow this route, as the distinction is rather fluid, and we focus on fine-grained *edits* instead of coarse-grained *editors*.

1.3 A Need for Machine Learning-Based Vandalism Detection

Detecting vandalism means detecting damaging contributions to a crowdsourced knowledge base. Until recently, the Wikidata community had to rely on two suboptimal solutions to detect vandalism: (1) manually reviewing edits and (2) a rule-based abuse filter. In the following, we point out their limitations, why we experiment with machine learning in this thesis, and what the requirements of our approach are.

Reviewers often inspect the (recent) edits in the knowledge base's edit history, and when a damaging edit is encountered, the edit is reverted. Manually reviewing millions of edits every month, however, imposes a heavy workload on the community of a knowledge base. It leads to delays in the reviewing process in which vandalism is widely visible. Moreover, the time that volunteers spend with tedious reviewing might be better spent to improve the knowledge base in other ways, such as adding and updating

[18] Trolls are vandals who intent to provoke an angry reaction in other users (https://en.wikipedia.org/wiki/Wikipedia:Vandals_versus_Trolls).

content. While reverting vandalism is the most common reaction, two additional measures are blocking users and protecting pages. However, both user blocks and page protections contradict the open philosophy of Wiki projects, and they are not allowed as preemptive measures.[19] Thus, regardless of the countermeasure, vandalism has to be detected first.

In addition to manual reviewing, Wikidata administrators can add rules to an abuse filter, which automatically tags revisions that are likely vandalism. The current rule-based abuse filter, however, has only a small number of rules and detects only a small fraction of vandalism cases. Extending the number of rules, only partially helps, since creating and continuously updating a large number of fine-grained rules leads to maintainability issues due to the variety of vandalism.

To circumvent these limitations, the aim of this thesis is to explore machine learning for automatic vandalism detection in crowdsourced knowledge bases and in how far it outperforms currently available alternatives. "Machine learning algorithms can figure out how to perform important tasks by generalizing from examples. This is often feasible and cost-effective where manual programming is not" (Domingos, 2012). In recent years the use of machine learning has increased rapidly, and it has been successfully used for applications such as spam filters, credit scoring, and fraud detection (Domingos, 2012).

A Need for a Vandalism Corpus

In order to develop machine learning-based approaches to detect damaging edits, an appropriate dataset is required. The corpus should cover a *real-world*, current knowledge base in order to be useful in practice and to cover current vandalism patterns, which might evolve over time; the corpus should be *large* to enable machine learning approaches, which benefit from large amounts of training data; the corpus should be *labeled* to enable supervised machine learning, which generally outperforms unsupervised approaches; the corpus should be labeled in a way that is *robust* against manipulations by vandals who might try to circumvent detection; and the corpus should be *self-contained* to enable reproducible results.

[19] https://en.wikipedia.org/wiki/Wikipedia:Blocking_policy
https://en.wikipedia.org/wiki/Wikipedia:Protection_policy

A Need for a Vandalism Detector with High Predictive Performance

For reducing manual reviewing effort, a vandalism detector with high predictive performance is necessary. (1) Vandalism detectors should assign each edit a *vandalism score* denoting the likelihood of vandalism. Edits with a high score can be reverted fully automatically; edits with a medium score can be ordered by their score to prioritize reviewing efforts; edits with low scores might not be reviewed at all. (2) Vandalism detectors should score edits *as soon as possible* such that vandalism can be reverted in a timely manner, and does not spread to a large audience. (3) Vandalism detectors should have a *high predictive performance* across a wide range of operating points to make them suitable for automatic detection with high precision, for semi-automatic detection with high recall, as well as for ranking.

A Need for a Vandalism Detector with Low Bias

Although the discrimination of anonymous editors has long been condemned by the community,[20] both rule-based and machine learning-based approaches have not been optimized for fairness yet. We found, for example, that *benign* edits by anonymous editors receive vandalism scores over 300 times higher than *benign* edits by registered editors raising multiple issues: (1) Newcomers often start their editing career anonymously before registering,[21] and reverting benign edits by newcomers severely affects newcomer retention (Halfaker et al., 2011, 2013; Schneider et al., 2014), thus jeopardizing the long-term sustainability of crowdsourced projects like Wikidata. (2) Such a widespread discrimination of editors undermines the founding principles on which Wikimedia's projects are built:[22] the ability of anyone to edit articles, and the creation of a welcoming environment. (3) It might violate the "Ethics Guidelines for Trustworthy AI" by the European Union as well as similar guidelines by IEEE, and large companies, such as Google, Microsoft, and IBM, with respect to fairness principles.[23] Generally, the fairness of machine learning models recently gets considerable attention from policy

[20] https://en.wikipedia.org/wiki/Wikipedia:IPs_are_human_too

[21] https://en.wikipedia.org/wiki/Wikipedia:Perennial_proposals

[22] https://meta.wikimedia.org/wiki/Founding_principles

[23] https://ec.europa.eu/futurium/en/ai-alliance-consultation/
https://ethicsinaction.ieee.org/
https://ai.google/principles/
https://www.microsoft.com/en-us/ai/our-approach-to-ai
https://www.ibm.com/watson/ai-ethics/

makers and the general public—as evidenced by ethics guidelines and investigative journalism[24]—making it increasingly important to address such issues.

Overall, vandalism detectors should treat important groups of editors *fairly*, e.g., anonymous and new users, such that these groups do not abandon the project because they feel treated unfairly. To this end, it is necessary to analyze how much existing vandalism detectors are biased, how their biases can be reduced, and what the trade-offs in terms of bias and predictive performance are.

1.4 Challenges of Machine Learning-Based Vandalism Detection

Our central research question can be stated as

> *Q: How to detect damaging contributions to structured, crowdsourced knowledge bases automatically?*

As motivated before, in this thesis, we tackle the question with machine learning and break it down into three sub-questions:

> *Q1: How to construct a vandalism corpus for structured, crowdsourced knowledge bases?*
>
> *Q2: How to detect damaging contributions to structured, crowdsourced knowledge bases with high predictive performance?*
>
> *Q3: How to detect damaging contributions to structured, crowdsourced knowledge bases with low bias?*

In the following, we briefly overview the state of the art related to these questions.

Lack of a Vandalism Corpus

Sarabadani et al. (2017) compiled a vandalism dataset simultaneously to us. They took a sample of manual edits, automatically determined reverted edits, and applied additional heuristics. Their dataset, however, is not based on Wikidata's entire history and contains only 500,000 edits from 2015, including less than 700 vandalism examples (in contrast to 200,000 of our corpus; Heindorf et al., 2017a). While their list of edits is publicly available, they have not published their data as a self-contained corpus to enable

[24] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

reproducible results, and they provide little insights into the characteristics of vandalism. Tan et al. (2014) compiled a dataset of low-quality contributions to Freebase, but the usefulness of the data is diminished by the fact that Google has shut down the project, and no new edits are being made. To the best of our knowledge, no other *labeled* datasets for *structured* knowledge bases are available. While Neis et al. (2012) develop a rule-based vandalism detector for OpenStreetMap, and Truong et al. (2018) a clustering-based detector for OpenStreetMap, neither of them constructs a *labeled* dataset.

Lack of a Vandalism Detector

The Wikimedia Foundation developed a machine learning-based vandalism detector for Wikidata simultaneously to us (Sarabadani et al., 2017). They employ a random forest model with 14 features taking into account the edited content, the editor, and the edit operation (Sarabadani et al., 2017; Heindorf et al., 2017b). We include their features in our candidate set for feature selection and add many more features. Moreover, we experiment with multiple-instance learning, and demonstrate that our approach significantly outperforms theirs in a number of settings ranging from different types of content to different points in time. Before, the Wikimedia Foundation had to rely on substantial manual reviewing efforts and a rule-based abuse filter, whose performance had never been systematically evaluated. Tan et al. (2014) tackled the problem of automatically detecting low-quality contributions to Freebase, which has been shut down and whose data model and user community differ from Wikidata, leaving it unclear how well vandalism detection can be detected automatically in today's largest active crowdsourced knowledge base, Wikidata. Our model integrates Tan et al.'s best-performing features, complements them by further features tailored to Wikidata, and evaluates them on a large-scale, up-to-date dataset.

Lack of a Vandalism Detector with Low Bias

None of the existing vandalism detectors goes beyond optimizing predictive performance and aims for fair predictions. Most vandalism detectors rely on biased user features, such as the geolocation of IP addresses, the age of user accounts, or the language of edited content (West et al., 2010; Adler et al., 2011; Heindorf et al., 2017b). While these features are easy to obtain, they do not directly assess the quality of an edit and harm some benign editors.

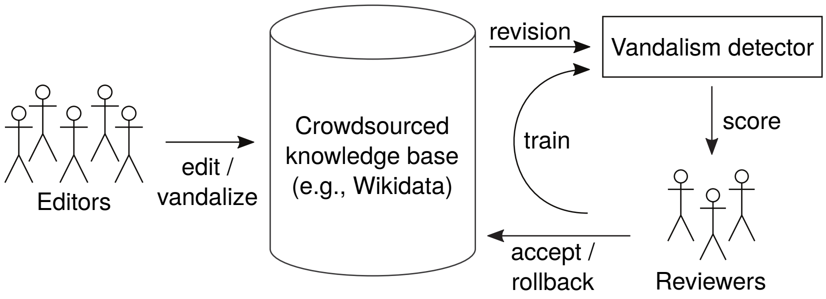


Figure 1.3: Vandalism detector in the context of a crowdsourced knowledge base.

Berk et al. (2018) survey the state of the art of fair machine learning: They give an overview of different notions of fairness, hint at potential means of increasing fairness, and outline what is known about trade-offs between fairness and predictive performance. In our work, we employ the fairness notion *equality of opportunity* (Hardt et al., 2016), argue why it is suitable for our problem, and adapt it to continuous scores. Countermeasures for debiasing can be categorized as pre-processing, in-processing, and post-processing (Berk et al., 2018): pre-processing includes modifications of datasets, weighting of training samples, and modification of feature sets; in-processing includes modifications of machine learning algorithms; post-processing includes modifications of predictions. Since it is not clear what method works best for our problem of vandalism detection, we experiment with many of the methods and evaluate them in terms of predictive performance and bias. While trade-offs between *accuracy* and fairness are explored by Berk et al. (2018), Kleinberg et al. (2017), Corbett-Davies et al. (2017), and Chouldechova (2017), performance measures for imbalanced datasets such as $\mathrm{PR}_{\mathrm{AUC}}$ and $\mathrm{ROC}_{\mathrm{AUC}}$ are not considered. Wikidata makes for an interesting case study for fair vandalism detection, since its subject-predicate-object triples allow to pay particular attention to the content of an edit rather than its meta data.

1.5 Overview of Contributions and Publications

Our envisioned vandalism detector in the context of a crowdsourced knowledge base is depicted in Figure 1.3: A large number of editors from the crowd edit the knowledge base, sometimes vandalizing it. Each edit results in a new revision within the knowledge base's revision history, and revisions are scored immediately by a vandalism detector. Based on the scores, reviewers inspect revisions and rollback damaging ones. Reviewers' decisions might serve for training new vandalism detectors.

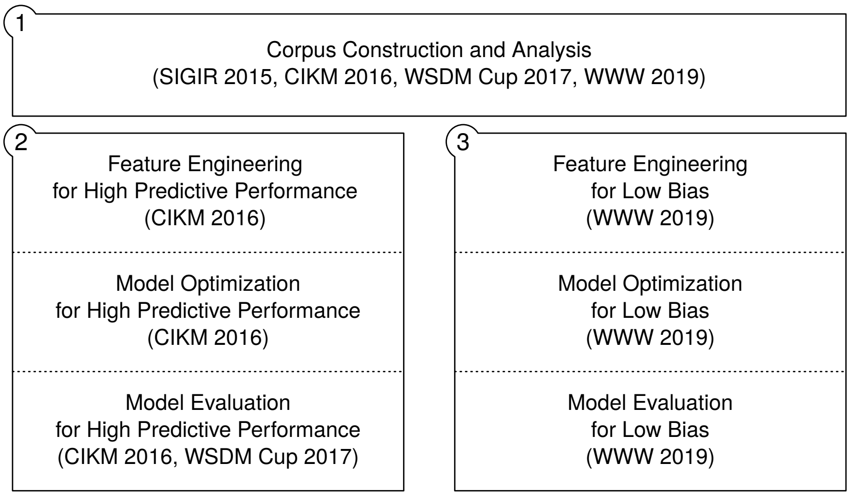

1
Corpus Construction and Analysis
(SIGIR 2015, CIKM 2016, WSDM Cup 2017, WWW 2019)
2
Feature Engineering
for High Predictive Performance
(CIKM 2016)
Model Optimization
for High Predictive Performance
(CIKM 2016)
Model Evaluation
for High Predictive Performance
(CIKM 2016, WSDM Cup 2017)
3
Feature Engineering
for Low Bias
(WWW 2019)
Model Optimization
for Low Bias
(WWW 2019)
Model Evaluation
for Low Bias
(WWW 2019)

Figure 1.4: Overview of the three main contributions of this thesis: (1) We construct and analyze a vandalism corpus. (2) We develop a machine learning approach with high predictive performance. (3) We develop a machine learning approach with low bias.

The three main contributions of this thesis regarding the novel task of vandalism detection in the crowdsourced, structured knowledge base Wikidata are shown in Figure 1.4: (1) We construct the first large-scale vandalism corpus and perform an analysis of the corpus. (2) We develop a new machine learning approach for vandalism detection with high predictive performance by engineering features, optimizing models, and evaluating them. (3) We develop a vandalism detector that increases fairness towards anonymous users, thus promoting the retention of editors and the long-term sustainability of the knowledge base. All data and source code underlying our research is available as open source to enable the reproducibility of our results and to enable future research.[25] In the following, we detail our contributions.

Corpus Construction and Analysis

We compile the first large-scale corpus of Wikidata vandalism called Wikidata Vandalism Corpus WDVC-2015, which comprises over 24 million revisions of which 100,000 are labeled as vandalism (Heindorf et al., 2015). Moreover, we conduct a corpus analysis to investigate what content is vandalized, who the vandals are (Heindorf et al., 2015), and what potential features for vandalism detection are (Heindorf et al., 2016). We

[25] http://www.heindorf.me/wdvd

carefully split the dataset into subsets for training, validation, and testing to avoid "information leaks" (Heindorf et al., 2016). Moreover, we construct an updated version of our corpus, called WDVC-2016 (Heindorf et al., 2017b), for evaluating the submissions to the WSDM Cup 2017, the data science challenge that we organized to drive progress on the vandalism detection task.

Vandalism Detection with High Predictive Performance

In Heindorf et al. (2016), we contribute a new machine learning-based approach for vandalism detection in Wikidata, called Wikidata Vandalism Detector (WDVD). We develop and carefully analyze features taking into account both content and context information of a Wikidata revision to obtain a set of 47 high-performing features. Experimenting with different machine learning algorithms and carefully tuning their hyperparameters, we find multiple-instance learning—which exploits the dependence of consecutive edits by the same editor on the same item (i.e., within an editing session)—to outperform all other models that we tried. Our best model based on multiple-instance learning on top of bagging and random forests achieves an area under the curve of the receiver operating characteristic ($\mathrm{ROC}_{\mathrm{AUC}}$) of 0.991 and significantly outperforms the state of the art represented by the rule-based Wikidata Abuse Filter, FILTER (0.865 $\mathrm{ROC}_{\mathrm{AUC}}$), and the machine learning-based Objective Revision Evaluation Service by the Wikimedia Foundation, ORES (0.859 $\mathrm{ROC}_{\mathrm{AUC}}$), on the Wikidata Vandalism Corpus WDVC-2015. We extensively evaluate our approach in a number of settings, including algorithms, hyperparameters, content types, feature groups, and performance over time.

In Heindorf et al. (2017a,b), we report the results of the WSDM Cup 2017. We compare our approach to the submissions of the five participating teams in terms of features, model variants, and predictive performance. Only two teams were able to slightly outperform our approach in terms of $\mathrm{ROC}_{\mathrm{AUC}}$; none of the teams outperformed our approach in terms of $\mathrm{PR}_{\mathrm{AUC}}$, thus stressing the strength of our approach. Moreover, we developed an ensemble of all approaches, outperforming them in terms of $\mathrm{ROC}_{\mathrm{AUC}}$.

Vandalism Detection with Low Bias

To the best of our knowledge, we present the first machine learning approach for detecting damaging contributions to online communities aiming to make *fair* predictions (Heindorf et al., 2019a,b). We carefully analyze biases in state-of-the-art Wikidata

Table 1.1: Overview of publications underlying this thesis in terms of venue, Core ranking, number of pages, and corresponding section in thesis.

Publication	Venue	Core	Pages	Main contribution
Heindorf et al. (2015)	SIGIR	A*	4	2.1 Corpus construction
Heindorf et al. (2016) ☆	CIKM	A	10	2.2 High predictive performance
Heindorf et al. (2017a)	WSDM	A*	2	2.2 High predictive performance
Heindorf et al. (2017b)	WSDM Cup	–	9	2.2 High predictive performance
Heindorf et al. (2019a)	WWW	A*	11	2.3 Low bias
Heindorf et al. (2019b)	INFORMATIK	–	2	2.3 Low bias

☆ ACM Best Paper Award at CIKM 2016

vandalism detectors, and develop two novel models that have a low bias against anonymous users, who may withdraw from the project if treated unfairly: Our model FAIR-E employs graph embeddings, focusing on the content of an edit rather than biased user information. Our model FAIR-S selects the most predictive hand-engineered features under the constraint that no user features are used. For comparison, we experiment with two transformations of the state-of-the-art vandalism detector WDVD: post-processing scores and weighting training samples. We evaluate our models on a subset of the large-scale Wikidata Vandalism Corpus 2016 and find that FAIR-E and FAIR-S significantly reduce the bias ratio to only 5.6 and 11.9, respectively, from over 310.7 in case of WDVD. Compared to WDVD's transformations based on post-processing scores and weighting training samples, our models FAIR-E and FAIR-S are significantly simpler, hence, better suitable to explain predictions to editors, and achieve roughly similar trade-offs in terms of predictive performance and bias.

Publications

Table 1.1 overviews the publications underlying this thesis. Heindorf et al. (2015) describe our corpus construction and analysis. Heindorf et al. (2016) introduce our Wikidata vandalism model, and report on its optimization and evaluation. Heindorf et al. (2017a) and Heindorf et al. (2017b) evaluate our approach by comparing it with third-party submissions to the WSDM Cup 2017. Heindorf et al. (2019a) and Heindorf et al. (2019b) analyze biases of vandalism detectors and develop novel approaches to significantly reduce biases against anonymous editors.

Structure of Thesis

Having motivated our work in this chapter, Chapter 2 summarizes our three main contributions and presents them in a coherent way. Chapter 3 concludes the thesis with a summary and outlook on future research directions.

2
Contributions

This chapter describes our three main contributions: in Section 2.1, we construct vandalism corpora; in Section 2.2, we create vandalism detectors with high predictive performance; in Section 2.3, we create fair vandalism detectors with low bias against certain groups of editors. We discuss our findings in Section 2.4. This chapter is based on our publications (Heindorf et al., 2015, 2016, 2017a,b, 2019a,b) and provides a coherent presentation of our main findings.

2.1 Corpus Construction and Analysis

Following a data-driven approach for developing vandalism detectors, we need suitable datasets for engineering features, optimizing machine learning models, and evaluating the models. In this section, we describe our corpus construction methodology, validate the methodology, analyze what content is vandalized in Wikidata, who the vandals are, and describe our dataset split for training, validating, and testing models.

2.1.1 Corpus Construction Methodology

Our goal is to derive large-scale, labeled vandalism corpora that are suitable for analyzing vandalism in Wikidata and training supervised machine learning-based vandalism detectors. Moreover, we aim for a construction methodology that is robust against manipulation by vandals. We derive our vandalism corpora from Wikidata database

Table 2.1: Construction of the Wikidata Vandalism Corpus 2015 (WDVC-2015) from the Wikidata edit history.

Wikidata revisions until October 2014	167,802,227	100%
w/o revisions on meta pages	1,211,628	1%
w/o revisions on special items	11,167	0%
w/o revisions by automatic bots	142,574,999	85%
WDVC-2015	24,004,433	14%

dumps which contain the full revision history of Wikidata and are provided by the Wikimedia Foundation under a permissive CC0 license.[1] Our ground truth labels—*benign* or *vandalism*—are derived from the decisions of Wikidata administrators and privileged users who review edits and revert them if they find vandalism. For reverting edits, the Wikidata software offers a *rollback*[2] feature that is explicitly meant to revert vandalism and allows privileged users to revert damaging edits with one click. The rollback operation is recorded in Wikidata's edit history and allows to determine what revisions were reverted, thus yielding examples of vandalism and benign revisions. Moreover, we restrict our dataset to *manual* revisions on Wikidata *items* (see Table 2.1): we filter out revisions on *meta pages*, such as user talk pages and property pages, revisions on *special items*, such as test items, and revisions by *automatic bots*, e.g., for simple maintenance tasks. We believe damaging edits by bots are more systematic and should be dealt with separately, e.g., by improving the reviewing process of bots and the bots themselves. Our corpora are publicly available to enable reproducible results.[3]

2.1.2 Corpus Validity

To validate our corpus construction methodology, which is based on the rollback decisions by Wikidata administrators and privileged users, we manually double-checked a random sample of 1,000 revisions that were rollback reverted and 1,000 revisions that were not reverted. About 86% of the rollback revisions turned out to be vandalism while only about 1% of the non-reverted revisions. We experimented with other means of constructing the corpus, e.g., using the undo/restore feature that cannot only be used

[1] https://dumps.wikimedia.org/legal.html

[2] https://www.wikidata.org/wiki/Wikidata:Rollbackers

[3] https://www.heindorf.me/wdvd

by privileged users but all users. However, we found the rollback feature to yield a significantly larger amount of vandalism and that at a significantly higher precision—possibly due to administrators and privileged users having a lot of experience and being familiar with the intricate details of Wikidata. Moreover, focusing on administrators and privileged users makes our corpus construction methodology robust against vandals trying to manipulate our training data since people need to earn the trust of the Wikidata community before being granted the right to rollback revisions.

2.1.3 Corpus Analysis

Using the corpus construction methodology described above, we identified about 100,000 out of 24 million manual revisions as vandalism in Wikidata's history from October 2012 to October 2014 (Wikidata Vandalism Corpus WDVC-2015) and about 200,000 vandalism revisions from October 2012 until June 2016 (Wikidata Vandalism Corpus WDVC-2016). Figure 2.1 shows our data over time. The total number of edits per month is increasing over time (top) with about 2 million edits per month towards the end of WDVC-2015 and about 5 million edits per month towards the end of WDVC-2016. The number of vandalism edits per month varies without a clear trend (bottom) and stays at around 5 thousand vandalism edits per month. Moreover, we break the total number of edits down by content type: Head content is shown on the top of a Wikidata page and includes labels, descriptions, and aliases in up to 375 supported languages. It is used for rendering the data in human-readable form, for example, on Wikidata pages and in Wikipedia infoboxes, thus being visible to a large audience. Body content is shown below on the Wikidata page and includes statements and sitelinks and makes up the core of the actual knowledge graph. We attribute the increases in edits per month around May 2014 to the emergence of semi-automatic editing tools for Wikidata, such as Wikidata Game, allowing to make large amounts of edits in a short amount of time, e.g., by confirming or refusing edits suggested by simple rule-based scripts. We attribute the drop in head vandalism in April 2015 to the redesign of Wikidata's user interface around this time, which makes it less obvious to edit head content and might deter many drive-by vandals. Before May 2013, Wikidata's statements often had no automatically generated comment ("Misc" in Figure 2.1) that we employ in our analysis.

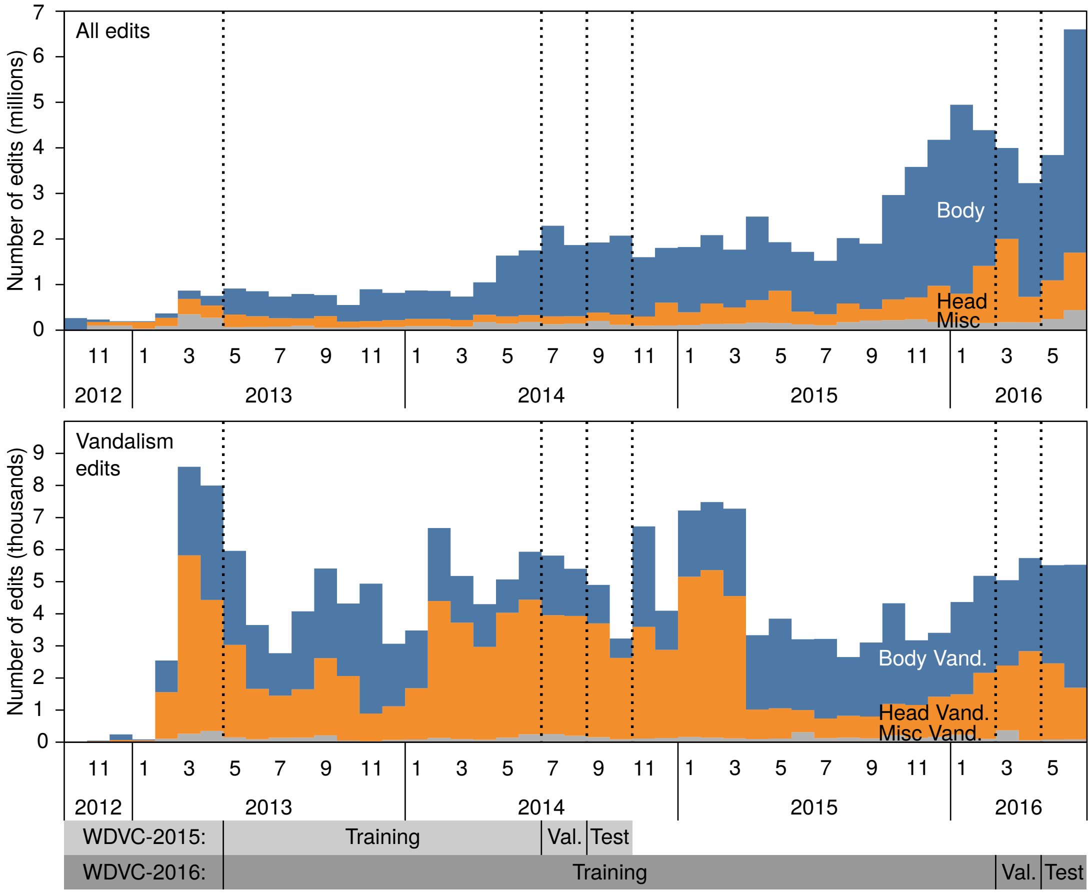


Figure 2.1: Vandalism corpora WDVC-2015 and WDVC-2016 over time in terms of total edits per month (top) and vandalism edits per month (bottom). Colors indicate different content types. Dashed lines indicate the split into training, validation, and test sets.

We analyzed what content is vandalized and who the vandals are. Table 2.2 (left and middle) displays the topmost vandalized items and item categories (in terms of editing sessions). The table shows that Wikidata items about people—particularly famous soccer players and musicians—are often vandalized, whereas items about places, such as cities, mountains, rivers, are relatively seldom vandalized. Regarding the part of an item (Table 2.2, right) that is vandalized, we found that there are about 4 times more edits affecting the item body consisting of statements and sitelinks than edits affecting the item head consisting of labels, descriptions, and aliases. The total number of vandalism edits affecting item heads and bodies is similar, requiring a vandalism detector to work well on both parts of an item. Overall, about 0.4% of revisions are labeled vandalism, 1.4% of head revisions and 0.2% of item revisions.

Table 2.2: Vandalism analysis in the Wikidata Vandalism corpus WDVC-2015. Top vandalized items (left); top vandalized categories and top edited categories in a sample of 1000 items each (middle). Vandalism by item part and user registration status (right).

Cases	Item title
47	Cristiano Ronaldo
43	Lionel Messi
43	One Direction
41	Portal:Featured content
34	Justin Bieber
33	Barack Obama
29	English Wikipedia
29	Selena Gomez

Category	Vand.	All
Culture	20%	12%
People	20%	21%
Society	16%	9%
Nature	14%	15%
Meta items	13%	8%
Technology	9%	4%
Places	8%	31%
Other	1%	1%

Item Part	Vand.	Total
Head	58,868	4,296,817
Body	41,475	17,201,518
Misc	2,862	2,506,098
Total	103,205	24,004,433

Users	Vand.	Total
Anonymous	88,592	768,027
Registered	14,613	23,236,406
Total	103,205	24,004,433

Investigating who vandalizes Wikidata (Table 2.2, bottom right), we found that about 86% of vandalism on Wikidata originates from anonymous users (88,592 of 103,205). Nevertheless, only about 12% of edits by anonymous users are vandalism (88,592 of 768,027). Together with the fact that the open philosophy of Wikidata encourages edits by anonymous users, this is no justification to block all edits by anonymous users, and such edits must be considered in a more differentiated way.

2.1.4 Datasets for Training, Validating, and Testing Models

"The fundamental goal of machine learning is to generalize beyond the examples in the training set. This is because, no matter how much data we have, it is very unlikely that we will see those exact examples again at test time" (Domingos, 2012). Therefore, we create different datasets for training models, optimizing models, and finally estimating their performance on new, unseen data. We split our corpus *by time*, since our goal is to detect vandalism as soon as it happens, and we may score an edit only based on *earlier* edits in order to prevent leaks "from the future." Although sometimes done in related work (e.g., Sarabadani et al., 2017), scoring an edit based on *later* edits, e.g., done on the same item or by the same user, would be false, since this information "from the future" would not be available in practice. All in all, we split our data into three parts by time: the *training set* is used for training models; the *validation set* is used for optimizing models; the *test set* is not used until the very end for the final evaluation of models after their optimization. As shown in Figure 2.1, both corpora start with the release of Wikidata in October 2012 and contain two months of edits for validation and two months of edits for testing. For our models, we chose to start the training set not before May 2013, since Wikidata's data model and serialization format was not stable yet.

2.1.5 Summary of Main Contribution

We summarize our main contribution 'corpus construction and analysis' as follows:

1. **Vandalism corpora** We construct large-scale corpora for an important, novel machine learning task—vandalism detection in the crowdsourced, structured knowledge base Wikidata. Our datasets serve as basis for our vandalism detectors as well as for the WSDM Cup.

2. **Vandalism characteristics** We carefully analyze vandalism characteristics and find that items about famous people are vandalized particularly often, whereas items about places seldom. We find that a lot of vandalism originates from anonymous editors, and we argue that this is no justification for banning them.

3. **Corpus split** We carefully split the dataset into subsets for training, validation, and testing to avoid information leaks "from the future."

2.2 Vandalism Detection with High Predictive Performance

For reducing the manual reviewing effort in crowdsourced knowledge bases, we develop vandalism detectors with high predictive performance: we engineer features, experiment with machine learning algorithms, optimize models, and evaluate them in a number of settings.

2.2.1 Feature Engineering

For engineering features, we studied vandalism characteristics, manually inspected individual vandalism and benign edits, and took related work into account. This resulted in a list of over 100 candidate features that we implemented and use as basis for feature selection: we selected our final set of features as a local optimum such that adding or removing features from our candidate set of features did not improve performance on the validation set of our vandalism corpus WDVC-2015. Table 2.3 gives an overview of

Table 2.3: Features of the Wikidata Vandalism Detector (WDVD) distributed across feature groups. 40 of the 47 features have not been previously evaluated for Wikidata (Heindorf et al., 2016). One feature is partially assigned to two groups and shown as ½.

Features	WDVD	(new)
Content	27	(24)
Character	11	(11)
Word	9	(7)
Sentence	4	(4)
Statement	3	(2)

Features	WDVD	(new)
Context	20	(16)
User	10½	(9½)
Item	2	(2)
Revision	7½	(4½)

our final set of 47 features, grouped into 27 features characterizing the *content* of an edit and 20 features characterizing the *context* of an edit. To the best of our knowledge, 40 of the 47 features have not been previously evaluated for Wikidata. In the following, we introduce the two feature groups before providing statistics for selected features.

Content Features Content features mainly target edits on item heads, i.e., on labels, descriptions, and aliases. They can be subdivided into features operating on the character level, word level, and sentence level. On the character level, for example, we observed that many vandals do not use proper capitalization (e.g., everything is capitalized, or nothing is capitalized). On the word level, we found that vandals often use bad words, including vulgar and offensive language, as well as literal strings of languages. On the sentence level, we identified, for example, edits of suspicious length. Moreover, for subject-predicate-object triples, we found some predicates to have a higher prior of being vandalism than others.

Context Features Context features target the context of an edit such as the user performing the edit, the edited Wikidata item, and revision metadata. We distinguish, for example, edits by registered users from edits by anonymous users. For users with multiple edits, we capture how experienced a user is, e.g., in terms of edits performed; for anonymous users, our dataset contains their IP address allowing us to derive their geolocation (for registered users, IP addresses are withheld by Wikimedia for privacy reasons). We capture the popularity of items, e.g., in terms of distinct users having edited the item. Regarding revisions, we consider revision meta data such as the edit operation performed ("update statement", "add description", ...). One feature—`revisionTags`—belongs to both the group of user and revision features, since some

Table 2.4: Statistics on selected features (`revisionTags`, `languageWordRatio`, `revisionLanguage`, and `userCountry`). The tables show the number of vandalism revisions, total revisions, and the empirical vandalism probability. Rows are ordered by vandalism revisions. Numbers are given in thousands.

`revisionTags`	**Vand.**	**Total**	**Prob.**
Rev. with tags	52	8,619	0.60%
By abuse filter	49	122	39.90%
By editing tools	3	8,496	0.03%
Rev. w/o tags	52	15,386	0.34%

`languageWordRatio`	**Vand.**	**Total**	**Prob.**
Rev. with comment	102	23,304	0.44%
Ratio equals 0	79	22,955	0.34%
Ratio greater than 0	23	349	6.61%
Rev. w/o comment	1	700	0.21%

`revisionLanguage`	**Vand.**	**Total**	**Prob.**
Rev. with lang.	92	8,747	1.05%
English	40	1,664	2.43%
Spanish	4	370	1.11%
Hindi	3	28	11.51%
German	3	865	0.31%
French	2	623	0.38%
Other languages	39	5,196	0.75%
Rev. w/o lang.	12	15,258	0.08%

`userCountry`	**Vand.**	**Total**	**Prob.**
Rev. by unreg. users	88	705	12.42%
USA	13	65	20.85%
India	11	31	35.29%
Japan	5	46	11.39%
United Kingdom	3	20	14.60%
Germany	3	45	6.09%
Other countries	52	498	10.49%
Rev. by reg. users	16	23,299	0.07%

tags from the Wikidata Abuse Filter contain user information (e.g., "new user removing something") and other tags indicate meta data of a revision (e.g., "revision created with semi-automatic editing tool"). The Wikidata Abuse Filter tags revisions according to simple rules created by Wikidata administrators. Semi-automatic editing tools tag revisions by their authentication method, distinguishing them from regular edits.

Feature Statistics During feature engineering, we analyzed vandalism characteristics on the training set. Table 2.4 shows statistics on selected features. For example, tags by the Wikidata abuse filter (`revisionTags`) often signal vandalism, while tags by semi-automatic editing tools signal benign edits (39.90% vs. 0.03% empirical vandalism probability). Moreover, we found features regarding countries and languages to provide a strong vandalism signal. For example, edits containing the literal string of a language, such as "English" or "German," often point to vandalism (`languageWordRatio` greater than 0 in Table 2.4). Apart from such literal strings, vandalism probabilities vary depending on the language of an edit (`revisionLanguage`): while edits in the English language have a vandalism probability of about 2.43%, German edits have a vandalism probability of only 0.31% and Hindi edits of about 11.51%. Looking at the countries of anonymous editors, which we can geolocate through their IP addresses (`userCountry`), yields a similar picture: most edits come from the U.S. and have a medium vandalism probability, whereas edits from Germany have a low and edits from India a high vandalism probability. These findings suggest that Wikidata provides new opportunities for studying cultural differences in crowdsourced knowledge bases, but they might also hint at undesirable biases of vandalism detectors against certain groups of editors. Last but not least, the feature `languageWordRatio` points at potential issues in Wikidata's user interface, since we found a number of cases where editors submitted the word "English" when the user interface suggested "enter a description in English."

2.2.2 Experimental Setup

Before reporting our evaluation results, we briefly describe our experimental setup in terms of baselines, evaluation metrics, learning algorithms, datasets, and implementation details for reproducibility.

Baselines We employ two state-of-the-art baselines for Wikidata vandalism detection: Wikimedia's Objective Revision Evaluation Service (ORES; Sarabadani et al., 2017) and revision tags (FILTER).[4] ORES is a machine learning approach developed by the Wikimedia Foundation to provide machine learning as a service for Wikimedia Projects. ORES' vandalism model for Wikidata was developed concurrently to our Wikidata Vandalism Detector (WDVD) and is based on a random forest with 14 features, many of which are shared with our approach, since, initially, we made a list of features available to the authors (Sarabadani et al., 2017; Heindorf et al., 2017a). The FILTER baseline is based on revision tags, which can be divided into the two groups: (1) tags by the Wikidata Abuse Filter, and (2) tags by semi-automatic editing tools. Although the abuse filter has been in use for a long time, to the best of our knowledge, its performance has never been evaluated.

Evaluation Metrics We evaluate our approach according to the metrics area under the precision-recall curve (PR_{AUC}) and area under the receiver operating characteristics (ROC_{AUC}). Both metrics are widely used for problems with imbalanced classes and take a wide range of operating points into account. Each point on one curve

[4] https://www.wikidata.org/wiki/Special:AbuseFilter

corresponds to one point on the other (Davis and Goadrich, 2006). While the former emphasizes points in the high precision range, the latter emphasizes points in the high recall range. Hence, the former is particularly meaningful for fully automatic operation, while the latter is particularly suitable for semi-automatic operation. Moreover, both metrics can be interpreted as ranking metrics. $\mathrm{PR}_{\mathrm{AUC}}$ is essentially equivalent to the well-known ranking metric average precision (AP; Manning et al., 2008), while $\mathrm{ROC}_{\mathrm{AUC}}$ can be interpreted as the probability that the score of a randomly chosen vandalism revision ranks higher than the score of a randomly chosen benign revision (Fawcett, 2006).

Learning Algorithm We use random forests (Breiman, 2001) as learning algorithm for our experiments. In a pilot study, we experimented with different learning algorithms and their hyperparameters, including naive Bayes and logistic regression, and we found random forests to outperform all other algorithms that we tried. Our findings are corroborated by the facts that random forests have been found to outperform other algorithms for Wikipedia vandalism detection (Javanmardi et al., 2011; Tran and Christen, 2015; Martinez-Rico et al., 2019), random forests are the algorithm of choice of the ORES baseline (Sarabadani et al., 2017), and the winner of the WSDM CUP 2017 employs a tree-based algorithm (Crescenzi et al., 2017). WSDM Cup participants experimenting with logistic regression and neural networks report significantly worse results (Yamazaki et al., 2017; Zhu et al., 2017)—possibly due to problems with encoding high-cardinality features and class imbalance (Micci-Barreca, 2001; Khoshgoftaar et al., 2007; Wang et al., 2016). We leave it for future work to further experiment in the direction of logistic regression, neural networks, and deep learning.

Dataset Unless otherwise mentioned, we perform our experiments on the Wikidata Vandalism Corpus WDVC-2015 using the revisions from May 2013 till June 2014 for training, the revisions from July and August 2014 for validation, and the revisions from September and October 2014 for testing, as depicted in Figure 2.1.

Reproducibility The source code and data to reproduce our results are publicly available.[5] We implemented our feature extraction in Java with Wikidata Toolkit[6] and our machine learning models in Python with scikit-learn (Pedregosa et al., 2011).[7]

[5] https://www.heindorf.me/wdvd

[6] https://www.mediawiki.org/wiki/Wikidata_Toolkit

[7] https://scikit-learn.org

2.2.3 Model Optimization

We optimize the predictive performance of our vandalism models in a series of experiments comparing them with state-of-the-art baselines. Table 2.5 gives an overview of our evaluation results and shows our models of increasing complexity. Starting with scikit-learn's default random forest,[8] we optimize its hyperparameters, and we experiment with bagging and multiple-instance learning increasing its performance in terms of PR_{AUC} by 44%. We perform the same optimizations for our model WDVD as well as for the baselines FILTER and ORES. Our best model WDVD based on multiple-instance learning achieves 0.991 ROC_{AUC} at 0.491 PR_{AUC} and outperforms the baselines by factors ranging from 1.9 to 3.6. Figure 2.2 shows the corresponding precision-recall curves. Regarding WDVD, up to 30% of vandalism can be reverted fully automatically at over 70% precision; the workload of Wikidata reviewers can be reduced by a factor of ten (precision 2% instead of 0.2% in the test dataset) while still identifying over 98.8% of all vandalism; edits in between can be ranked by their vandalism score and manually reviewed in this order. In the following, we describe our models in detail.

Optimized Random Forest We systematically optimized our random forest model by varying its hyperparameters on the validation set: We optimized the maximal depths of the trees, the number of features per split, and the number of trees. Varying the maximal depths in the range $\{1, 2, 4, 8, 16, 32, 64, \infty\}$ and the maximal features per split in the range $\{1, 2, \text{'log2'}, \text{'sqrt'}\}$, we achieved the best results in terms of PR_{AUC} with 'sqrt' features per split and a maximal depth of 8 for WDVD, and a maximal depth of 16 for FILTER and ORES. While increasing the number of trees per forest and jointly optimizing the other hyperparameters hardly increases predictive performance, the runtime increases linearly. Hence, we stick to scikit-learn's default number of trees (10). Table 2.5 shows the resulting performance on the test set. Both PR_{AUC} and ROC_{AUC} significantly improve. WDVD improves by 19% (from 0.342 PR_{AUC} to 0.406 PR_{AUC}), ORES by 35%, whereas FILTER does not improve.

Bagging Investigating the amount of training data required, we observed that the predictive performance of WDVD on the validation set can be increased by training our model on a random sample of the training set. We believe this effect to be due to

[8] 10 trees per forest, no maximal depth of the trees, and 'sqrt' features per split

Table 2.5: Evaluation results of our Wikidata vandalism detector, WDVD, and of the two baselines FILTER and ORES. Performance measures are the area under curve of the receiver operating characteristic (ROC_{AUC}), and the area under the precision-recall curve (PR_{AUC}). Performance values are reported for the entire test dataset as well as divided by item part. The darker a cell, the better the performance.

Classifier	**Item head**		**Item body**		**Entire item**	
Optimization	ROC_{AUC}	PR_{AUC}	ROC_{AUC}	PR_{AUC}	ROC_{AUC}	PR_{AUC}
WDVD (our approach)						
Multiple-instance	**0.985**	**0.575**	**0.981**	**0.216**	**0.991**	**0.491**
Bagging	0.980	0.521	0.879	0.175	0.960	0.430
Optimized random forest	0.980	0.487	0.942	0.171	0.978	0.406
Default random forest	0.922	0.451	0.800	0.087	0.894	0.342
FILTER (baseline)						
Multiple-instance	0.819	0.345	0.893	0.020	0.900	0.218
Bagging	0.768	0.297	0.816	0.014	0.865	0.201
Optimized random forest	0.770	0.351	0.816	0.015	0.865	0.257
Default random forest*	**0.770**	**0.358**	**0.816**	**0.015**	**0.865**	**0.265**
ORES (baseline)						
Multiple-instance	0.962	0.269	0.946	0.132	0.975	0.228
Bagging	0.956	0.197	0.900	0.124	0.960	0.169
Optimized random forest	0.953	0.214	0.896	0.111	0.960	0.182
Default random forest*	**0.882**	**0.176**	**0.749**	**0.058**	**0.859**	**0.135**

*These approaches represent the state of the art; to the best of our knowledge, the outlined optimizations have not been tried with ORES and FILTER before.

many similar training samples resulting in reduced variance of the trees and overfitting. Hence, we experimented with bagging (**b**ootstrap **agg**regat**ing**; Breiman, 1996) to increase predictive performance: We employ an ensemble of 16 random forests each trained on 1/16 of the training set containing 8 trees per forest. Performing a new grid search for random forest hyperparameters, we obtain a maximal tree depth of 32 with 2 features per split for WDVD, a maximal tree depth of 8 with 1 feature per split for FILTER, and a maximal depth of 16, and 'sqrt' features per split for ORES. Bagging improves PR_{AUC} of WDVD by 6% on the test set (from 0.406 PR_{AUC} to 0.430 PR_{AUC}) compared to optimized random forests. ROC_{AUC} and the baselines do not improve.

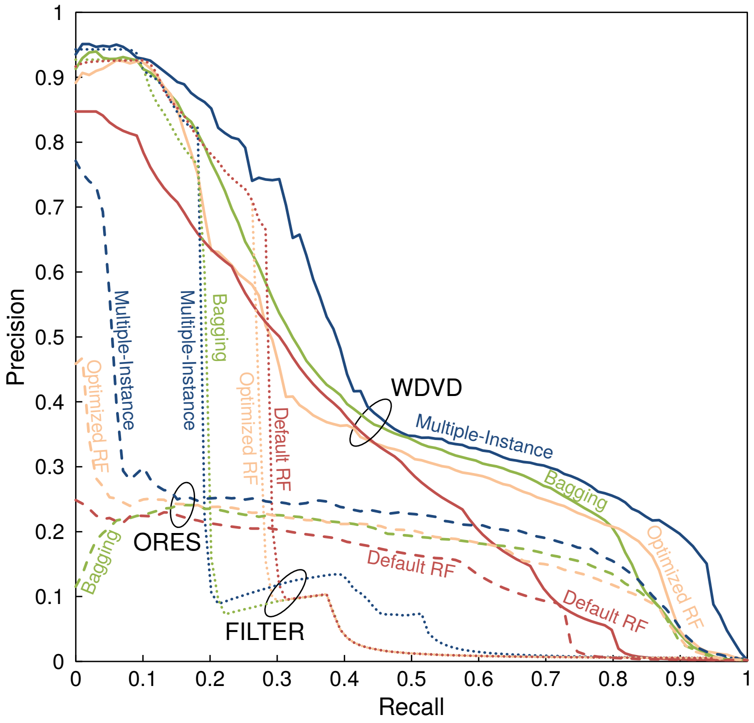


Figure 2.2: Precision-recall curves of the models shown in Table 2.5 on the test set.

Multiple-Instance Learning We noticed that consecutive edits by the same user on the same item, which we refer to as an editing session, are often closely related. Hence, we experiment with multiple-instance learning to classify sessions instead of single revisions (Amores, 2013; Gärtner et al., 2002). Our best model on the validation set combines two approaches: single-instance learning (SIL) and simple multiple-instance learning (Simple MI).

SIL operates in the instance space and assigns the same average score to all revisions within a session: Formally, let $\mathbf{x}^i = \left(x_1^i, \ldots, x_n^i\right)$ be the feature vectors within the editing session $X = \left(\mathbf{x}^1, \ldots, \mathbf{x}^d\right)$ with d revisions. Let $c_{SIL} \colon \mathbb{R}^n \to \mathbb{R}$ be a classifier that is trained on single revisions and assigns a score to every revision. Then SIL assigns the same average score $C_{SIL}\left(X\right)$ to each revision in the session:

$$C_{SIL}(X) = \frac{1}{d} \sum_{i=1}^{d} c_{SIL}\left(\mathbf{x}^i\right) \ .$$

Simple MI (SMI) operates in an embedded space. The single, fixed-size feature vector $\bar{\mathbf{x}}$ of an editing session is obtained by concatenating the element-wise minima and maxima of the sessions' feature vectors $\mathbf{x}^i$: Let $a_j = \max_{i \in \{1,\ldots,d\}} x_j^i$ and $b_j = \min_{i \in \{1,\ldots,d\}} x_j^i$ for $j \in \{1, \ldots, n\}$. Then the embedded vectors $\bar{\mathbf{x}} = (a_1, \ldots, a_n, b_1, \ldots, b_n)$ are used for training the classifier $c_{SMI} \colon \mathbb{R}^{2n} \longrightarrow \mathbb{R}$ and assigning the same score

$$C_{SMI}(X) = c_{SMI}(\bar{\mathbf{x}})$$

to each revision in the session.

Finally, we combine both approaches to obtain our final prediction C for every revision within session X:

$$C(X) = \frac{C_{SIL}(X) + C_{SMI}(X)}{2} \, .$$

As classifiers c_{SIL} and c_{SMI}, we employ the bagging model based on random forests as described above—one time trained on single instances and one time trained in embedded space. Multiple-instance learning improves $\mathrm{PR}_{\mathrm{AUC}}$ of WDVD by 14% compared to bagging. FILTER and ORES improve, too.

Multiple-instance learning brings along a minor limitation: Edits cannot be scored immediately, but only after a session has ended—possibly due to a timeout. Alternatively, multiple-instance learning can be applied in an online variant, where the set of revisions X is continuously updated, and the most up-to-date version is used for immediate classification. We employ the latter variant in the WSDM Cup 2017, which is described in the next section.

2.2.4 Model Evaluation

We evaluate our approach by content type, by feature group, for different points in time, and we compare it with submissions to the WSDM Cup 2017.

Head Content vs. Body Content

Table 2.5 reports the performance of our approaches divided by content type where *head content* refers to the head of an item page and includes labels, descriptions, and aliases, and *body content* refers to the body of an item page and includes statements and sitelinks. Investigating a sample of edits revealed different kinds of vandalism in head vs. body content: head content attracts rather obvious vandalism on the lexical level, e.g., bad words or wrong capitalization, whereas body content attracts rather sophisticated vandalism on the semantic level, e.g., incorrect information; head content is predominantly edited by anonymous users, whereas body content is predominantly edited by registered users (Heindorf et al., 2015); head content is needed to represent the knowledge base in human-readable form, e.g., in infoboxes, on item pages and in search suggestions, whereas body content makes up the core of the knowledge base, the actual knowledge graph. While the number of edits affecting body content is larger, both head and body content contain about the same number of vandalism edits (Figure 2.1). Overall, for a vandalism detector, it is important to detect vandalism in both head and body content. WDVD significantly outperforms the baselines both on head and body content. Moreover, both WDVD and the baselines perform better on head content than body content (with the exception of FILTER in terms of $\mathrm{ROC}_{\mathrm{AUC}}$). This might be explained by more obvious vandalism in head content and a larger vandalism fraction in head content making it easier to achieve a high $\mathrm{PR}_{\mathrm{AUC}}$,[9] but this also hints at opportunities to engineer more advanced features for body content, e.g., using graph embeddings as we do in Section 2.3.2.

Content Features vs. Context Features

We evaluate the performance of content and context features for our four approaches from Table 2.5. Figure 2.3 shows the corresponding precision-recall curves per approach and feature group. It shows that context features generally outperform content features with context features particularly contributing to a high recall and content feature contributing to a high precision. The best performance is achieved by combining content and context features (except for a small range of recall values with the "Optimized Random Forest"). Employing more advanced algorithms such as multiple-instance learning improves the performance of both content and context features, with the effect being particularly strong for content features.

[9] $\mathrm{PR}_{\mathrm{AUC}}$ varies as class imbalance changes (Davis and Goadrich, 2006; Boyd et al., 2012), whereas $\mathrm{ROC}_{\mathrm{AUC}}$ does not (Fawcett, 2006).

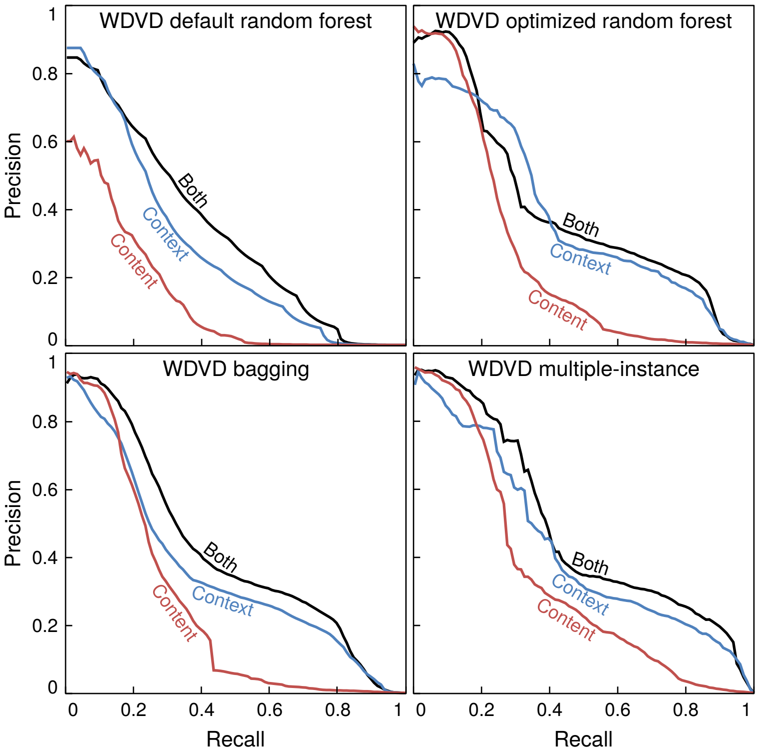


Figure 2.3: Precision-recall curves for content features, context features, and both combined on the test dataset by classifier.

Online Learning and Evaluation over Time

Vandalism detection can be considered an online learning problem where newly arriving revisions are scored immediately, and the model is dynamically updated as soon as new vandalism cases are identified. In an initial experiment, we experimented with scikit-learn's online learning algorithms, including stochastic gradient descent and naive Bayes. However, the online algorithms performed significantly worse than random forests applied in a batch setting. Hence, we stick to our random forest model and evaluate it in a setting where it is regularly retrained and evaluated. Figure 2.4 shows the performance on the corresponding pseudo-test sets in intervals of two months: We use our best-performing model based on multiple-instance learning, train it from May 2013 until the start of our pseudo-test sets, which we vary in two-month increments from July 2013 to September 2014. This way, our penultimate pseudo-test set corresponds to our actual validation set and our ultimate pseudo-test set to our actual test set. The plot shows that the performance of WDVD varies between 0.46 $\mathrm{PR_{AUC}}$ in July & August 2013 and 0.69 $\mathrm{PR_{AUC}}$ in March & April 2014, while ORES

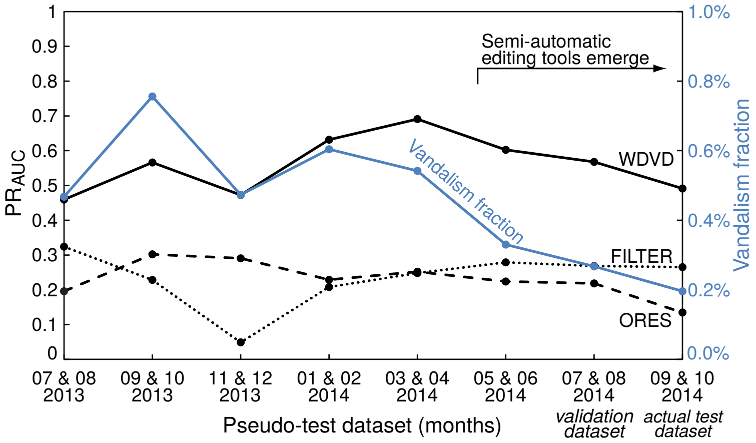


Figure 2.4: Performance over time of our best vandalism detector WDVD (multiple-instance) and the baselines FILTER and ORES. Test datasets vary while classifiers were trained on all preceding revisions. The blue line plot shows the vandalism fraction in the test datasets.

and FILTER fluctuate below 0.32 PR$_{\text{AUC}}$. Hence, WDVD outperforms the baselines at all points in time. Investigating what causes the variance, we identified changing fractions of vandalism in our pseudo-test sets (blue line-plot in Figure 2.4) as a partial explanation: The emergence of semi-automatic editing tools around May 2014 lead to large amounts of benign edits, thus reducing the fraction of vandalism cases and making predictions at high precision more difficult for the machine-learning models WDVD and ORES. Moreover, we suspect changing vandalism patterns over time contributing to the variance, emphasizing the need to regularly retrain the models.

Evaluation of WSDM Cup 2017 Submissions

In order to drive progress on the vandalism detection task and further improve predictive performance, we organized a data science challenge—the WSDM Cup 2017 (Heindorf et al., 2017a,b), which was held in conjunction with the International Conference on Web Search and Data Mining (WSDM 2017). We invited participants from all over the world to contribute novel solutions; we constructed an updated version of our vandalism corpus—the Wikidata Vandalism Corpus WDVC-2016; and we set up an evaluation framework—ensuring the reproducibility of submissions, preventing cheating, and enforcing that the vandalism score is computed in a streaming fashion based on information available at the time of an edit but not on information emerging later. The winner was determined based on ROC$_{\text{AUC}}$. Table 2.6 gives an overview of the submissions, comparing them to WDVD as well as the FILTER and ORES baselines.

Table 2.6: Overview of the WSDM Cup 2017 submissions in terms of features, learning algorithms, and performance. Performance values are reported in terms area under the precision-recall curve ($\mathrm{PR}_{\mathrm{AUC}}$), and area under curve of the receiver operating characteristic ($\mathrm{ROC}_{\mathrm{AUC}}$) on the test dataset of the Wikidata Vandalism Corpus WDVC-2016. The darker a cell, the better the performance. Rows are ordered by $\mathrm{ROC}_{\mathrm{AUC}}$, starting with the best.

Submission	**Features**	**Learning Algorithms**								**Performance**	
		XGBoost	Linear SVM	Logistic Regression	Random Forest	Extra Trees	GBT	Neural Networks	Multiple-Instance	$\mathrm{PR}_{\mathrm{AUC}}$	$\mathrm{ROC}_{\mathrm{AUC}}$
META	ALL	✓	✓	✓	✓	✓	✓	✓	✓	0.475	0.950
Buffaloberry	WDVD + NEW	✓	–	–	–	–	–	–	✓	0.458	0.947
Conkerberry	BoW(WDVD)	–	✓	–	–	–	–	–	–	0.352	0.937
WDVD (baseline)	WDVD	–	–	–	✓	–	–	–	✓	0.486	0.932
Honeyberry	WDVD	–	–	✓	✓	✓	✓	✓	–	0.206	0.928
Loganberry	⊆WDVD	✓	–	–	–	–	–	–	–	0.337	0.920
Riberry	WDVD + $\mathrm{WDVD}^{\mathrm{Overfit}}$	–	–	–	✓	–	✓	–	–	0.174	0.894
ORES (baseline)	ORES	–	–	–	✓	–	–	–	–	0.347	0.884
FILTER (baseline)	FILTER	–	–	–	✓	–	–	–	–	0.227	0.869

All teams build upon our approach WDVD, which we provided as a baseline. The best-performing approach Buffaloberry[10] (Crescenzi et al., 2017) engineered a few new features replacing some of WDVD's, for example, generalizing some of our sentence level features to better work with languages other than English and checking whether edits on labels, descriptions, and aliases are made in the correct language. Conkerberry (Grigorev, 2017) reused many of WDVD's features, but encoded them in a different way: WDVD's features were converted to a string before representing them as a bag-of-words model. Honeyberry's (Yamazaki et al., 2017) feature set is similar to WDVD's but omits some features on bad words and users. Loganberry (Zhu et al., 2017) used a subset of WDVD's features lacking geolocation features, while Riberry (Yu et al., 2017) used many features that we had previously excluded from our model due to overfitting,

[10] All participants of the vandalism detection task were assigned a team name with the -berry suffix.

corroborating our previous findings. For comparison, we report the performance of our baselines WDVD,[11] ORES,[12] and FILTER, and we create an ensemble, called META, of all five submissions and three baselines to estimate the performance that could be achieved by integrating all approaches.

In terms of algorithms, Buffaloberry employs multiple-instance learning in combination with XGBoost, a fast boosting approach based on decision trees (Chen and Guestrin, 2016). Conkerberry trains a linear SVM, slightly outperforming WDVD in terms of $\mathrm{ROC}_{\mathrm{AUC}}$, but not in terms of $\mathrm{PR}_{\mathrm{AUC}}$. Honeyberry builds a complex ensemble of a variety of learning algorithms, but they do not outperform the simpler models of WDVD, Buffaloberry, and Conkerberry—possibly due to their omission of some features and their lack of using multiple-instance learning. Loganberry employs XGBoost, while Riberry employs random forests and gradient boosted trees, with the performance of both approaches seeming limited by their respective feature sets.

In terms of $\mathrm{ROC}_{\mathrm{AUC}}$, two submissions slightly outperform our strong baseline WDVD—Bufalloberry and Conkerberry—supposedly due to slight modifications of features and algorithms. In terms of $\mathrm{PR}_{\mathrm{AUC}}$, WDVD still outperforms all other approaches, including META. Compared to our previous performance values reported on the Wikidata Vandalism Corpus 2015, performance values on the Wikidata Vandalism Corpus 2016 are lower due to an outlier in the new dataset (Heindorf et al., 2017b).

Beyond their final models submitted for evaluation, some WSDM Cup participants report additional experiments, all resulting in lower predictive performance, corroborating our previous findings. Both Grigorev (2017) and Zhu et al. (2017) experimented with different learning algorithms, including logistic regression and ensembles of multiple algorithms. Grigorev (2017) experimented with online learning, as well as undersampling and oversampling for balancing the dataset.

Summarizing the results of the WSDM Cup 2017, the best vandalism model in terms of $\mathrm{PR}_{\mathrm{AUC}}$, i.e., for fully automatic vandalism detection at high precision, is still our approach WDVD based on 47 features, random forests, and multiple-instance learning.

[11] We use the same hyperparameters for this model as reported in Section 2.2.3. To adjust WDVD to the new evaluation setup, where edits must be scored in a streaming fashion, we adjust multiple-instance learning to employ only edits up to the current one.

[12] We use the original hyperparameters by Sarabadani et al. (2017): 80 decision trees with 'log2' features per split using the 'entropy' criterion. While Sarabadani et al. (2017) experimented with balancing the weights of the training examples, we do not do so for the ORES baseline, since it has no effect on performance in terms of $\mathrm{ROC}_{\mathrm{AUC}}$ and decreases performance in terms of $\mathrm{PR}_{\mathrm{AUC}}$.

The best vandalism models in terms of ROC_{AUC} are Buffaloberry and META, slightly outperforming WDVD by employing a couple of new features and employing XGBoost in combination with multiple-instance learning.

2.2.5 Summary of Main Contribution

We summarize our main contribution 'vandalism detection with high predictive performance' as follows:

1. **Content and Context Features** We design a machine learning approach for an important, novel task—vandalism detection in the crowdsourced, structured knowledge base Wikidata. We study vandalism characteristics and derive 47 features taking into account both the content and the context of an edit.

2. **Multiple-Instance Learning** We experiment with machine learning algorithms and their hyperparameters, finding multiple-instance learning on top of bagging and random forests to outperform all other variants that we tried.

3. **High Predictive Performance** We extensively evaluate our approach in a number of settings, including different types of content and different points in time, finding our approach to outperform state-of-the-art baselines in all settings. Our approach turned out to be competitive with the WSDM Cup submissions.

2.3 Vandalism Detection with Low Bias

For treating knowledge base editors fairly, we analyze biases of vandalism detectors and develop two novel models to reduce biases: Our model FAIR-E employs graph *embeddings* to focus solely on the content of an edit instead of biased user features. Our model FAIR-S systematically *selects* the best-performing features under the constraint that no user features are used. Moreover, we experiment with transformations of our best-performing model WDVD: post-processing scores and weighting training samples. We evaluate our novel models on a subset of the Wikidata Vandalism Corpus 2016 and analyze trade-offs between predictive performance and bias comparing our models to state-of-the-art vandalism detectors, including WDVD.

2.3.1 Bias Analysis

Bias can be defined as "inclination towards something; predisposition, partiality, prejudice, [...],"[13] which often leads to discrimination, i.e., "treatment of an individual or group to their disadvantage."[14] In our case, *benign* edits receiving high vandalism scores are more likely to be reverted, and benign editors whose edits are reverted are more likely to withdraw from the project (Halfaker et al., 2011, 2013; Schneider et al., 2014). In this section, we briefly define metrics for measuring bias before analyzing biases.

Measuring Bias

To measure bias of a classifier producing continuous scores, roughly following Kleinberg et al. (2017) and Zemel et al. (2013), we compare the average scores of two groups, which by convention, are called the *protected* and *unprotected* groups (against discrimination). Our goal is to achieve *equality of opportunity* (Hardt et al., 2016), i.e., *benign* edits from both groups should receive similar vandalism scores, giving them similar opportunities of being not reverted. The deviation from this goal, the bias, can be measured in terms of the difference and ratio. We focus on the protected group of anonymous users in this work. Further protected groups based on the time, since registration and country of origin are analyzed in Heindorf et al. (2019a).

Formally, given ground truth labels whether a revision is benign as well as calibrated vandalism scores $y_i \in Y$ with $y_i \approx \Pr(i = \text{vandalism} \mid \mathbf{x}_i)$, where $\mathbf{x}_i$ is revision i's feature vector, we divide the scores of *benign* edits into the two groups of anonymous and registered editors:

$$\begin{aligned} Y^{anon}_{benign} &:= \{y_i \in Y \mid \text{benign edit } i \text{ by anonymous editor}\} \ , \\ Y^{reg}_{benign} &:= \{y_i \in Y \mid \text{benign edit } i \text{ by registered editor}\} \ . \end{aligned}$$

Then the bias can be measured in terms of the difference and ratio of their average scores:

$$\begin{aligned} \text{Diff.} &:= \text{mean}\left(Y^{anon}_{benign}\right) - \text{mean}\left(Y^{reg}_{benign}\right) \ , \\ \text{Ratio} &:= \text{mean}\left(Y^{anon}_{benign}\right) / \text{mean}\left(Y^{reg}_{benign}\right) \ . \end{aligned}$$

[13] https://en.wiktionary.org/wiki/bias

[14] https://en.wiktionary.org/wiki/discrimination

Table 2.7: Number of user features as well as average vandalism scores for anonymous and registered users and bias measured in terms of difference and ratio.

	WDVD	**ORES**	**FILTER**
User features	10½	2	½
mean $\left(Y_{benign}^{anon}\right)$	0.1215	0.1144	0.0978
mean $\left(Y_{benign}^{reg}\right)$	0.0004	0.0009	0.0014
Diff.	0.121	0.114	0.096
Ratio	310.7	133.1	69.2

Biases of Vandalism Models

All existing models exhibit significant biases against anonymous users, newcomers, and users from certain countries (Heindorf et al., 2019a). Table 2.7 shows that the models FILTER, ORES, and WDVD assign benign edits by anonymous users vandalism scores between 69.2 and 310.7 times higher than benign edits by registered users. The bias might be explained by user features that do not take the content of an edit into account to check its correctness. For example, both WDVD and ORES employ the feature `isRegisteredUser`, a simple user feature with high predictive performance, but high bias: 9.00% of edits by anonymous users constitute vandalism whereas only 0.03% of edits by registered users (in the training and validation set of WDVC-2016-Links). As this feature does not take any content information into account, it assigns *benign* edits by anonymous users vandalism scores about 300 times higher than *benign* edits by registered users. Similarly, ORES includes user age (`userAge`); WDVD includes the numbers of revisions and items edited by a user (`userFrequency`, `cumUserUniqueItems`); FILTER assigns tags for “new user changing something” or “new user removing something”—all user features correlated with protected attributes.

2.3.2 Fair Vandalism Detection Models

For developing fair vandalism detection models, we remove user features and strengthen content features, resulting in two novel models: (1) FAIR-E employs graph embeddings for subject-predicate-object triples solely focusing on the content of an edit rather than meta data such as user information, (2) FAIR-S selects the best-performing hand-engineered features under the constraint that no user features are used.

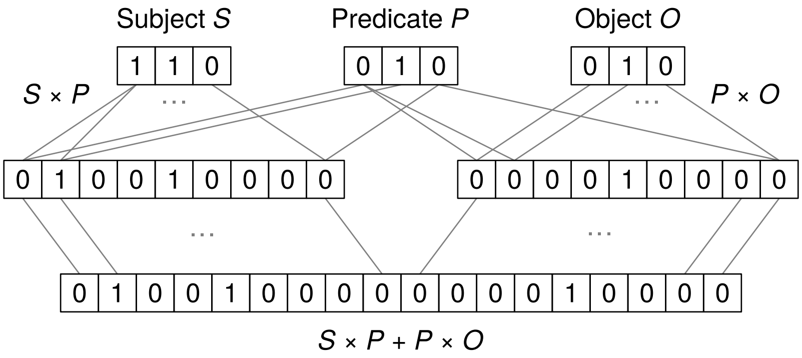


Figure 2.5: Example of a triple embedding. Subject, predicate, and object are represented in 3-dimensional predicate space (top) and combined via outer product (middle) and concatenation (bottom).

FAIR-E: Graph Embeddings for Wikidata

For taking the graph structure into account, we experiment with graph embeddings, and propose a novel embedding FAIR-E, which to the best of our knowledge, has never been used before and results in a low bias. Figure 2.5 illustrates our construction. Given a subject-predicate-object triple,[15] we encode it in predicate space: the *subject* is represented by its set of *outgoing* predicates encoded as a binary vector S; the *predicate* is represented by its binary vector P; the *object* is represented by its set of *incoming* predicates encoded as binary vector O. A subject-predicate-object triple then is encoded as $S \times P + P \times O$ where + denotes the concatenation of vectors and × the outer product, i.e., the pair-wise combination of vector elements. As machine learning algorithm, we employ logistic regression. Our model learns which interactions of subject-predicates, predicates, and object-predicates typically signal vandalism or benign edits. To avoid overfitting, we utilize the top $n := 100$ most frequent predicates of our item graph as determined independently per subject, predicate, and object on our training set. Hence, a subject-predicate-object triple is represented by a $2n^2 = 20{,}000$-dimensional vector.

We experimented with different variants of our model, including different values for n, varying interactions of S, P, and O, and outgoing object predicates instead of incoming. We found the above variant to work best on our validation dataset in terms of predictive performance. We also experimented with different regularization strengths

[15] In Wikidata, an edit usually affects only one triple. In rare cases, when an edit is made via Wikidata's API and affects more than one triple, we employ the edit's main triple as determined by the automatically generated comment.

of logistic regression, which had little effect on predictive performance, but large effect on bias, leading us to disable regularization. We also experimented with existing graph embeddings such as random walks (Dong et al., 2014; Gardner and Mitchell, 2015), but found our graph embeddings to outperform them.

Pointing out limitations, about 15% of triples are represented by the zero vector because their predicates are not among the top n selected; the learning algorithm assigns them all the same, small vandalism probability. We classify new versions of updated triples. Our graph embeddings do not distinguish additions and removals of triples, and we leave it to future work to include edit operations in a way that improves predictive performance. As the set and distribution of predicates evolve over time, the classifier needs to be retrained from time to time.

FAIR-S: Selecting Unbiased Features

As an alternative to graph embeddings, we experimented with feature selection. Starting with a candidate set of features consisting of WDVD's and ORES' features as well as a couple of new features, including FAIR-E's, we omit all user features and features not targeting subject-predicate-object triples. We determine our final set of features as a local optimum such that adding or removing any features from our candidate set does not improve ROC_{AUC}. Our final set of 14 features is described below. Seven features were selected from Heindorf et al. (2016), three features from Sarabadani et al. (2017) as well as four new features (Heindorf et al., 2019a). We employ a random forest algorithm with 32 trees and a maximal tree depth of 16. In a pilot study, all other algorithms, including logistic regression, neural networks, and gradient boosted decision trees performed worse despite tuning their hyperparameters.

Subject Subject features capture the popularity of a subject among editors (`subjectLogCumUniqueUsers`, `subjectLogFrequency`), the amount of information available about this subject (`subjectNumberOfLabels`, `subjectNumberOfAliases`, `subjectPredicateCumFrequency`) as well as the type and complexity of the subject (`subjectLabelWordLength`), for example, proper nouns often have one word in their label, persons two words, and complex topics multiple.

Predicate We encode predicates by the number of times they were used in our training set (`predicateFrequency`).

Object Similarly, we encode the object by the number of times it occurred in the training set (`objectFrequency`), its popularity (`objectPredicateCumFrequency`) as well as its embedding representation derived from our graph embeddings (`objectPredicateEmbedFrequency`): We represent the object in terms of the $n = 100$ most-frequent incoming object predicates in the knowledge graph and count how often the resulting embedding vectors appear in our training set. The idea behind the feature is that incoming object predicates represent in what context the object is typically used.

Edit Additionally, we characterize an edit by the edit operation performed such as “add”, “update”, “remove” (`editActionFrequency`, `editSubactionFrequency`), the previous action performed on the same item (`editSubactionFrequency`), and the number of triples added (`editProportionOfTriplesAdded`) relative to the current number of triples of the subject.

Variants Variants that we experimented with but that did not yield improvements include: Taking the super types of the subject and object according to Wikidata’s *instance of* hierarchy as features. Similar information is already captured by the graph embeddings (e.g., by `objectPredicateEmbedFrequency`). A bag-of-words model of the subjects’ and objects’ labels did not help, either.

2.3.3 Experimental Setup

Before reporting our evaluation results, we briefly describe our experimental setup in terms of baselines, evaluation metrics, datasets, and details for reproducibility.

Baselines We employ the same state-of-the-art vandalism detectors as before: our Wikidata Vandalism Detector WDVD, Wikidata’s machine learning-based vandalism detector ORES, and Wikidata’s rule-based abuse filter FILTER.[16]

Evaluation Metrics For measuring predictive performance, we employ the same metrics as described before: $\mathrm{PR}_{\mathrm{AUC}}$ and $\mathrm{ROC}_{\mathrm{AUC}}$. For measuring bias, we employ the bias difference and bias ratio as described in Section 2.3.1.

[16] WSDM Cup submissions are not considered, since they derive from WDVD and hardly outperform it.

Dataset Our experiments are based on a subset of the Wikidata Vandalism Corpus WDVC-2016, called WDVC-2016-Links. Since our goal is to pay particular attention to the content of an edit and to experiment with graph embeddings, we filter edits not affecting the actual knowledge graph induced by subject-predicate-object triples between entities. Moreover, we filter edits by semi-automatic editing tools because they contain little vandalism, and we believe systematic quality checks should be built directly into them. While WDVC-2016 contains all edits in sequential order, for computing our novel content features, we need to represent the data as a graph and employ Wikidata's static graph ahead of the validation set for this.[17]

Reproducibility The source code and data to reproduce our results are publicly available.[18] We perform our feature extraction in Java and our classification in Python with scikit-learn (Pedregosa et al., 2011).[19] To calibrate classifier scores before computing bias, we use isotonic regression.

2.3.4 Model Optimization and Evaluation

Table 2.8 shows the bias and predictive performance of our models FAIR-E and FAIR-S, which are both based on feature engineering. For comparison, we also experiment with two alternative approaches described below: post-processing scores and weighting training samples. Our models FAIR-E and FAIR-S reduce the bias ratio to only 5.6 and 11.9, respectively, compared to over 310.7 for the state-of-the-art baseline WDVD. Our models FAIR-E and FAIR-S achieve similar predictive performance and bias values as by post-processing scores and weighting training samples, but better explainability.

Debiasing via Feature Engineering

Both our models FAIR-E and FAIR-S focus on the content of an edit rather than biased user features. FAIR-E uses graph embeddings as described in Section 2.3.2. Experimenting with different combinations of subject S, predicate P, and object O embeddings, we generally found the more complex the interactions are, the higher the predictive performance was, but also the bias (Heindorf et al., 2019a). Given the relatively low bias overall, we choose our variant with the highest predictive performance as our model FAIR-E ($S \times P + P \times O$).

[17] https://archive.org/download/wikidata-json-20160229

[18] https://www.heindorf.me/wdvd

[19] https://scikit-learn.org

Table 2.8: Evaluation results in terms of predictive performance and bias on the test dataset of the Wikidata vandalism corpus WDVC-2016-Links.

Debiasing Experiment	**Performance**		**Bias**	
Model	PR_{AUC}	ROC_{AUC}	Diff	Ratio
Feature engineering				
FAIR-E	0.177	0.865	0.016	5.6
FAIR-S	0.316	0.963	0.031	11.9
Post-processing scores				
WDVD with p=3.88	0.230	0.966	0.015	5.3
WDVD with p=3.22	0.340	0.976	0.030	11.8
Weighting training samples				
WDVD with $\alpha = 8.1$	0.160	0.963	0.015	5.3
WDVD with $\alpha = 4.3$	0.314	0.973	0.030	11.5
Baselines				
WDVD	0.547	0.990	0.121	310.7
ORES	0.434	0.965	0.114	133.1
FILTER	0.302	0.924	0.096	69.2

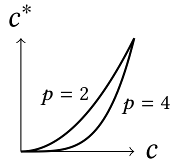
FAIR-S selects the most predictive set of features under the constraint of no user features. The optimization was done on the validation set of WDVC-2016-Links, and the resulting set of features is described in Section 2.3.2. Unlike WDVD, which consists of 47 features, FAIR-S consists of only 14 features, making the model simpler, faster, and easier to explain to editors.

Both FAIR-E and FAIR-S avoid user features, and the small remaining bias can be explained by slight correlations of other features with the protected attribute—an effect known as *indirect discrimination* or *redlining* (Pedreschi et al., 2008). In many cases, it is not even desirable to make all groups have exactly the same average scores, since there might be hidden confounders justifying small differences, and this would be an overreaction often referred to as *affirmative action* (Zliobaite, 2015; Romei and Ruggieri, 2014).

Debiasing via Post-Processing Scores

Alternatively to feature engineering, we can achieve comparable results in terms of bias and predictive performance by post-processing the scores of WDVD. Given the uncalibrated scores $c(i) \in [0, 1]$ for revisions i, we artificially scale down the scores of the protected group, i.e., anonymous users:

$$c^*(i) = \begin{cases} c(i)^p & \text{if revision } i \text{ from anonymous user,} \\ c(i) & \text{if revision } i \text{ from registered user.} \end{cases}$$



After scaling, we apply isotonic regression to calibrate the vandalism scores such that $c^*(i) \approx \Pr(i = \text{vandalism} \mid \mathbf{x}_i)$ in order to prevent lower scores just due to the scores falling in a smaller range. By experimentally determining $p > 1$ such that the bias approximately equals FAIR-E's and FAIR-S', we can compare them in terms of predictive performance. While the resulting predictive performance is similar (Table 2.8), the resulting model consisting of 47 features and a post-processing step is rather a black box, making it more difficult to explain decisions to Wikidata editors. In addition to the polynomial scaling function introduced above, we experimented with other function families, including linear, fractional, and exponential scaling. But polynomial scaling outperformed all others in terms of PR_{AUC}.

Debiasing via Weighting Training Samples

As another means to debiasing, we experimented with adjustments to the training data. We adjust the weights of the training samples as follows: We divide the training data into four groups according to the edit's registration status (anonymous vs. registered) and ground truth (vandalism vs. benign). Then we increase the weights of benign edits by anonymous users and of vandalism edits by registered users, making it more expensive for a vandalism detector to make mistakes on these groups. Moreover, we reduce the maximal tree depth of WDVD from 32 to 16 to increase the number of training samples per leaf, making the weighting to have a larger effect. We adjust the training weight $w(i)$ of a training sample $i \in I$ as follows (with the set of edits I being subdivided by anonymous vs. registered and by vandalism vs. benign edits):

$$w(i) = \begin{cases} \alpha \;/\; \left|I_{benign}^{anon}\right| & \text{if revision } i \in I_{benign}^{anon}, \\ 1 \;/\; \left|I_{vand}^{anon}\right| & \text{if revision } i \in I_{vand}^{anon}, \\ 1 \;/\; \left|I_{benign}^{reg}\right| & \text{if revision } i \in I_{benign}^{reg}, \\ \alpha \;/\; \left|I_{vand}^{reg}\right| & \text{if revision } i \in I_{vand}^{reg}. \end{cases}$$

By experimentally varying the parameter α, we obtain similar trade-offs in terms of bias and predictive performance as our models FAIR-E and FAIR-S (Table 2.8). But the model WDVD is more complex (47 features) and requires adjustments to the training procedure. Hence, we advocate the simpler models FAIR-E and FAIR-S. We also experimented with separate constants α_1 and α_2 per group, but this did not yield better bias-performance trade-offs.

2.3.5 Summary of Main Contribution

We summarize our main contribution 'vandalism detection with low bias' as follows:

1. **Optimization target** We aim for a novel and important optimization target—low bias against certain groups of editors.
2. **Bias Analysis** We analyze biases in Wikidata's state-of-the-art vandalism detectors and find all existing vandalism detectors to be highly biased.
3. **Low Bias** We develop low-biased vandalism detectors with multiple approaches: graph embeddings, feature selection, post-processing scores, weighting training samples. We extensively evaluate our approaches and compare them with state-of-the-art baselines, finding our models to exhibit significantly lower bias.

2.4 Discussion

Having outlined our contributions so far, we discuss limitations and applications of our approach. Potential future work is discussed in Section 3.2.

Corpus Construction

Our ground truth is based on rollback actions of Wikidata administrators and privileged users. This method allows constructing large-scale vandalism corpora that are robust against manipulations by vandals, who cannot easily manipulate the rollback signal. Comparing our ground truth with a manual annotation, we find a reasonable agreement (Heindorf et al., 2015): About 86% of edits rolled back turned out to be vandalism while only about 1% of non-reverted edits. Although we found some bias in the ground truth, e.g., against anonymous editors, a preliminary analysis revealed that vandalism detectors would still be highly biased even if the evaluation dataset were not biased (Heindorf et al., 2019a). Besides the rollback feature, there might be alternatives for corpus construction—each with their own challenges. For example, attempts by the Wikimedia Foundation to crowdsource a corpus from volunteers in a dedicated labeling campaign attracted hardly enough volunteers despite repeated calls on the mailing list.[20] Crowdworkers on Amazon Mechanical Turk might not be familiar with the intricate details of Wikidata, and obtaining a sufficient amount of vandalism cases might be expensive due to the large class imbalance (only a small fraction of edits are vandalism). Techniques based on item states might have a higher recall at the cost of a lower precision (Heindorf et al., 2017b). We leave it for future work to continuously update the dataset to reflect evolving patterns of vandalism and to refine the dataset, e.g., by means of crowdsourcing and statistical denoising.

Vandalism Detection with High Predictive Performance

We develop a vandalism detector with high predictive performance that can be employed in practice. Our best model achieves 0.991 $\mathrm{ROC}_{\mathrm{AUC}}$, significantly outperforming the state of the art and reducing the reviewing effort of volunteers: 30% of vandalism can be reverted fully automatically; 90% of edits can be marked as benign, while still retaining 98.8% of vandalism; the remaining edits can be inspected in the order of their scores to increase the chance of finding vandalism early. Our approach does not require large computational resources: for 24 million revisions, our features can be computed on a standard workstation (16 cores and 64 GB RAM) in less than 2 hours, and training a model only takes 10 minutes. Overall, our vandalism detector can easily be employed in practice and can be of great help to the Wikidata community.

As Wikidata is rapidly growing, and more and more applications of Wikidata are emerging, our vandalism scores might be directly integrated into these applications for hiding vandalized information or warning users of potentially false information. Besides its use by search engines and question answering systems (Dubey et al., 2019; Usbeck et al., 2017), Wikidata is used to populate infoboxes for the different language editions of Wikipedia and for generating natural language summary articles for underserved Wikipedia languages (Kaffee et al., 2018; Vougiouklis et al., 2018). It is increasingly used in the life sciences for integrating biomedical data on genes, diseases, drugs, and symptoms (Mitraka et al., 2015; Turki et al., 2019; Putman et al., 2017) as well as for scientific bibliographic information (Mietchen et al., 2015; Nielsen et al., 2017).

[20] https://labels.wmflabs.org/stats/wikidatawiki/, only 4,854 labels obtained from July 16, 2018 to September 12, 2019.

Vandalism Detection with Low Bias

To the best of our knowledge, we make the first attempt to debias vandalism detection models. As we are breaking new ground, and fairness of machine learning models is still an emerging topic, our approach has a number of limitations. As of today, there are no agreed upon measures for bias yet. Our measure aims at *group fairness*, between the groups of anonymous and registered edits, and our goal is to achieve *equality of opportunity* (Hardt et al., 2016), i.e., we aim to treat *benign* edits fairly. Treating both *benign* and *vandalism* edits fairly would correspond to the stricter fairness notion *equalized odds* (Hardt et al., 2016), which makes it potentially more challenging to find good trade-offs between fairness and predictive performance. Hence, we stick to the former, since it is already well-aligned with our goal of retaining benign editors. For *measuring the deviation* from the goal, a number of measures have been proposed in the literature ranging from simple measures such as differences and ratios between average scores, which we employ in our analysis, to more complex measures such as cost functions (Pleiss et al., 2017) and conditional Kolmogorov distance (Hardt et al., 2016). There are different approaches to *calibrate* vandalism scores before computing bias measures. We perform a single calibration across groups. Other approaches calibrate scores *within groups* (Kleinberg et al., 2017; Pleiss et al., 2017). We focus on the *protected attribute* of *user registration status*. In the future, it might be interesting to explore further groups, e.g., based on time since registration, gender, country, age, etc.

First theoretical findings suggest that it is impossible to build a vandalism detector with both high predictive performance and low bias for many notions of predictive performance and fairness (Berk et al., 2018; Chouldechova, 2017; Corbett-Davies et al., 2017; Kleinberg et al., 2017). Our empirical experiments corroborate these findings (for slightly different notions of predictive performance and bias), thus necessitating difficult trade-offs. Generally, it is hardly desirable to make two groups have exactly the same distribution of scores, since certain differences between the groups might justify some differences, and this would be an overreaction, known as *affirmative action* (Barocas and Selbst, 2016; Romei and Ruggieri, 2014; Zliobaite, 2015). To overcome this problem, other approaches focus on *individual* fairness (Dwork et al., 2012) based on a similarity function between individuals or fairness based on causal modeling (Kilbertus et al., 2017; Kusner et al., 2017; Zhang and Wu, 2017). All in all, our work on debiasing vandalism detection models must be viewed as a first step towards a fair vandalism detector, but there is certainly more research necessary to develop a truly fair vandalism detector and to explore the limits to which this is even possible.

Although vandalism detectors at Wikidata are not operated fully automatically yet and there are still humans in the loop, biases of vandalism detectors are problematic: (1) As the number of revisions per month is rapidly increasing, and reviewers have to review millions of edits every month, they have little time per edit and have to rely on vandalism scores more and more. (2) Since the decisions of reviewers are used to train new vandalism models, this might create a vicious cycle of reinforcing biases (cf., Baeza-Yates, 2018; Heindorf et al., 2019a). (3) Moreover, as the predictive performance of vandalism detectors is getting better over time and biases are getting worse—as we have seen in the past from FILTER over ORES to WDVD—vandalism detectors might soon rollback edits fully automatically and unless fairness constraints are taken into account, this might severely affect the retention of editors and the sustainability of the crowdsourced knowledge base.

Vandalism Detection in Other Crowdsourced Knowledge Bases

In this thesis, we develop vandalism detectors for one of the largest structured, crowdsourced knowledge bases, namely Wikidata. Here, we discuss in how far our results can be transferred to other crowdsourced knowledge bases.

Regarding corpus construction, other knowledge bases have mechanisms to rollback damaging edits, too.[21] However, the degree to which this mechanism is consistently used and recorded might vary. For example, while in Wikidata rollback actions can be clearly identified from automatically generated edit comments, in Wikipedia edit comments are created manually following a convention (Tran and Christen, 2013). Hence, it might be necessary to guide reviewers to follow the convention strictly and rollback edits consistently.

Regarding machine learning models, we expect many of our features and algorithms to be transferable to other crowdsourced knowledge bases. Our content features primarily target labels, descriptions, and aliases, which are available in many knowledge bases. Our context features primarily target meta data such as the user performing an edit, the user's geolocation, or the edit action performed. It seems straightforward to utilize similar features for other crowdsourced knowledge bases, too. Similarly, we would expect random forests and multiple-instance learning to work well for other knowledge bases, since random forests have already been successfully applied for Wikipedia vandalism detection (Adler et al., 2011), and consecutive edits by the same user on the same page are

[21] https://en.wikipedia.org/wiki/Wikipedia:Rollback
https://community.fandom.com/wiki/Help:Vandalism
https://wiki.openstreetmap.org/wiki/Change_rollback

hardly independent, making it promising to experiment with multiple-instance learning. Nevertheless, for the highest predictive performance, features might be adapted to the specifics of a knowledge base, e.g., employing graph-specific features for *structured* knowledge bases as we have done with graph embeddings, or domain-specific features for *domain-specific* knowledge bases (cf., Figure 1.2).

Regarding debiasing efforts, we restrict our dataset to edits affecting subject-predicate-object triples between entities in order to focus particularly on the content of an edit. Our model FAIR-E is based on graph embeddings, and it might be challenging to generalize it to all Wikidata edits or to unstructured knowledge bases. On the other hand, our model FAIR-S selects the best-performing features under the constraint that no user features are used. Such an approach seems to be easily transferable to other crowdsourced knowledge bases. Our alternative debiasing efforts based on post-processing scores and weighting training samples seem to be easily transferable to other knowledge bases, too.

Information Systems, Fake News, and Biases of AI Systems

Viewing our research in a larger context, our vandalism detectors might not only be used by crowdsourced knowledge bases directly, but also as a pre-processing step by information systems importing data from crowdsourced knowledge bases in order to prevent the spread of vandalism. Moreover, in the context of fake news (Lazer et al., 2018; Shu et al., 2017) and misinformation on the web (Kumar et al., 2016; Del Vicario et al., 2016), vandalism in crowdsourced knowledge bases might be considered a special kind of misinformation, and accurate knowledge bases can help to detect fake news (Conroy et al., 2015; Pan et al., 2018). From the perspective of data quality, vandalism affects one of the most important data quality dimensions, namely accuracy (Zaveri et al., 2016; Piscopo and Simperl, 2019; Mora-Cantallops et al., 2019). From the perspective of link prediction (Nickel et al., 2016) and fact-checking (Ciampaglia et al., 2015; Shi and Weninger, 2016), Wikidata edits might serve as real-world data in contrast to artificial data that is often used in this context. While biases of AI Systems are a hot topic among machine learning researchers and policy makers (Hardt et al., 2016; Chouldechova, 2017), Wikidata could serve as an interesting case study, since its permissive license makes it easily accessible by the research community and it allows paying particular attention to the content of an edit rather than to biased user information.

3
Conclusions and Outlook

This chapter concludes the thesis by summarizing our results in Section 3.1 and giving an outlook on future research directions in Section 3.2. It is partially based on our publications (Heindorf et al., 2015, 2016, 2017a,b, 2019a,b), but provides many new ideas for future research.

3.1 Conclusions

Information systems, such as search engines and question answering systems, increasingly rely on crowdsourced knowledge bases, and when crowdsourced knowledge bases get vandalized, this bears the risk of spreading damaging and false information to all their users. In this thesis, we devise the novel machine learning task of vandalism detection in the crowdsourced, structured knowledge base Wikidata and make three main contributions: (1) We construct a large-scale vandalism corpus for vandalism detection. (2) We develop vandalism detectors with high predictive performance. (3) We develop vandalism detectors with low bias.

Corpus Construction

We compiled a large-scale corpus for vandalism detection in the crowdsourced, structured knowledge base Wikidata. Our automatic labeling strategy is robust against manipulations by vandals and allows generating updated versions of the corpus, e.g., for the organization of data science challenges. Our analysis revealed that items about famous people are particularly often vandalized, whereas items about places are not; although a lot of vandalism originates from anonymous editors, we argue that this is no justification to prevent anonymous edits. The varying vandalism prevalence by country and language of editors gives rise to explore cultural phenomena. All in all, our corpora can serve for further analysis of vandalism in crowdsourced knowledge bases and the improvement of machine learning-based approaches.

Vandalism Detection with High Predictive Performance

Our machine learning approach assigns each edit a vandalism score as soon as the edit is made allowing immediate action upon vandalism in three modes of operation: edits with high scores can be reverted fully automatically; edits with medium scores can be manually reviewed in the order of their scores; edits with low scores might not need to be reviewed at all. We engineered 47 features to detect vandalism, taking both content and context information into account. Our best vandalism detector is based on multiple-instance learning on top of bagging and random forests. It achieves 0.991 ROC_{AUC} at 0.491 PR_{AUC} on the Wikidata Vandalism Corpus 2015, thus significantly outperforming the state of the art by factors between 1.9 in case of FILTER and 3.6 in case of ORES. Not only did our approach outperform the baseline on head content, body content, and at all points in time, it withstood the competition of the WSDM Cup, outperforming all approaches in terms of PR_{AUC}, and almost in terms of ROC_{AUC}. Our approach is ready to be employed in practice to save reviewers thousands of hours of work.

Vandalism Detection with Low Bias

Our analysis revealed that today's vandalism detectors are highly biased against certain groups of editors, which may cause a number of problems such as decreased user retention as well as a violation of project and ethics guidelines. We developed two novel machine learning models that significantly reduce bias against edits by anonymous editors. Our model FAIR-E, which is based on graph embeddings, achieves a bias ratio of only 5.6 compared to over 310.7 in case of WDVD. Our model FAIR-S, which is based on selecting hand-engineered features, achieves a bias ratio of only 11.9. We compared our models to two transformations of the state-of-the-art vandalism detector WDVD: post-processing scores and weighting training samples. Regardless of the approach, we found that high predictive performance and low bias cannot be achieved at the same time; however, our models enable a conscious trade-off. Further research on fair vandalism detectors and their perception by editors is needed to create a welcoming environment—not only for Wikidata editors but for many editors on the web.

Overall, our vandalism detectors do not make human reviewers fully redundant yet, but they can certainly help to reduce reviewing efforts. Given the large, emerging body of work on related topics such as fake news, misinformation on the web, data quality of knowledge bases, as well as fairness of AI systems, we are confident that our approach has the potential to influence further work in these directions.

3.2 Outlook

Extensions of our approach might go in multiple directions: improving vandalism detection in Wikidata, detecting vandalism in other online communities and across communities, robust vandalism detection in the presence of powerful adversaries, vandalism detection with explanations, as well as preventing some vandalism with novel user interfaces. Beyond vandalism detection, it might be interesting to improve other quality dimensions of knowledge bases and to explore further options to increase fairness and editor retention.

Vandalism Detection in Wikidata with High Predictive Performance

Ideas to further improve the predictive performance of vandalism detectors include further exploiting the edit history of the knowledge base, the graph structure of the knowledge base, and external data sources. Moreover, it might be possible to refine the training data and evaluation procedures.

What seems particularly promising is the exploitation of the *edit history* for vandalism detection. In his master's thesis, Crescenzi (2018) experimented with additional features taking the history of users and entities into account, improving predictive performance. Similarly, Pellissier Tanon et al. (2019); Nishioka and Scherp (2018) analyze the edit history of knowledge bases to verify edits using features focusing on the graph structure, but not the user history. While all mentioned approaches use classical feature engineering, the predictive performance might be further improved with deep learning techniques, such as sequence-to-sequence models and recurrent neural networks.

Another promising direction might be to further take the *graph structure* into account and to apply link prediction and link classification approaches to the problem of vandalism detection. Given a static snapshot of the graph, link prediction is the task of predicting missing predicates between subject-object pairs. Pairs receiving a high score are likely correct, whereas pairs receiving low scores are likely incorrect. Nickel et al. (2016) survey corresponding approaches, which might be based on neural embeddings (Dong et al., 2014), translational embeddings (Bordes et al., 2013; Wang et al., 2014; Lin et al., 2015; Wang et al., 2017), matrix factorization (Nickel et al., 2012), and explicit paths in the graph (Minkov et al., 2006; Lao et al., 2011; Shi and Weninger, 2016; Gardner and Mitchell, 2015; Ciampaglia et al., 2015). While most approaches focus on predicting links between entities, approaches for predicting attributes are emerging, too (Speck and Ngonga Ngomo, 2019).

Moreover, it seems promising to double-check edits with *external data sources*. During her work as a student assistant and in her master's thesis, Naphade (2018) double-checked Wikidata edits with the DBpedia knowledge base. However, a particular challenge was that DBpedia does not fully cover the same entities and predicates as Wikidata, and only a fraction of Wikidata edits could be mapped to DBpedia, leading to little improvements of predictive performance of vandalism detection. For the future, it seems particularly promising to double-check information via information retrieval on a large-scale web corpus (if challenges of scalability and outdated web corpora can be overcome). For example, Lehmann et al. (2012), Syed et al. (2018), Gerber et al. (2015), and Popat et al. (2018) verify facts with pages retrieved from the web.

Regarding *dataset construction*, it seems promising to experiment with the paradigm of "data programming" (Ratner et al., 2016, 2017). The idea is that domain experts write so-called *labeling functions*, which are simple heuristics applied to subsets of the data or crowdsourced labels obtained for subsets of the data. All *labeling functions* may have different accuracies and may overlap and conflict with each other. They are combined by means of a *noise-aware* generative model to determine the most likely label of each sample. For the task of Wikidata vandalism detection, promising labeling functions might be based on rollbacks, undo/restores, item states, external databases, crowdsourced labeling of small data subsets, as well as fact-checking results obtained via information retrieval. Future research might even make the approach *bias-aware*.

Furthermore, the *evaluation metrics* might be adjusted in the future. Consistent with previous research, we assign the same weight to every edit regardless of how often the edited data is viewed and regardless of whether the edited data affects the instance or schema level of the knowledge base. However, data that is viewed millions of times might be more important and should receive higher weights in the evaluation metrics. Similarly, data that affects the schema of the knowledge base, e.g., in the instance-of-hierarchy, can indirectly affect large amounts of data. Hence, future evaluation procedures might put a higher weight on highly visible data and schema level data. Moreover, the weight of certain edits might also heavily depend on the envisioned use case. For example, some data in Wikidata is heavily used by Wikipedia projects; other data is heavily used by search engines and question answering systems. Hence, the evaluation procedures might be made context-specific tailored to certain use cases. In order to optimize approaches according to the new evaluation metrics, it might be beneficial to devise novel datasets, features, and models.

Vandalism Detection in Other Online Communities

Besides Wikidata, malicious edits by users are a wide-spread problem in many online communities (Kumar and Shah, 2018). It might be interesting to adapt our approach to detect damaging edits to other knowledge bases, such as MusicBrainz and OpenStreetMap, to social networks, such as Facebook and Twitter, to question answering sites, such as StackExchange, to crowdsourcing platforms, such as Amazon Mechanical Turk, to review platforms, such as by Amazon or Tripadvisor, and to software hosting sites, such as GitHub. While in this thesis, we study vandalism in *one* online community, it might be interesting to investigate how one user behaves *across* communities.

Vandalism Detection against Powerful Adversaries

In our work, we focus on vandalism by the occasional vandal, and we took steps to ensure the robustness of our approach against adversaries—by only taking decisions of administrators and privileged users as training data, which are hard to manipulate by the occasional vandal. However, we did not assume a sophisticated thread model where powerful adversaries have advanced knowledge on attacking machine learning models and spend a lot of time on refining their attacks (cf., Szegedy et al., 2013; Goodfellow et al., 2015; Papernot et al., 2017; Zügner et al., 2018). It would be interesting to study in-depth how powerful adversaries can circumvent our vandalism detector, e.g., by creating user accounts with a high reputation before making damaging edits, by combining benign and damaging contributions in an edit, by damaging the knowledge base in novel ways that are not covered by our features, by making large amounts of benign edits appear as vandalism, thus distracting reviewers from the actually damaging edits, and by other novel ways to disrupt the vandalism detection system.

Moreover, we propose to regularly update machine learning models to detect new kinds of vandalism. While we performed experiments to regularly update the vandalism detector, in the presence of powerful adversaries, it might be necessary to update vandalism models more often or to utilize online learning approaches that immediately adapt to changing patterns of vandalism.

Vandalism Detection with Explanations

Today's vandalism detectors compute a vandalism score for each edit without providing explanations to editors how a specific score was obtained. However, explanations might increase the trust of editors in automatic vandalism detectors. Moreover, explanations might guide the development of better vandalism detectors by pointing out reasons for incorrect classifications or pointing out correct classifications due to the wrong reasons, thus jeopardizing the robustness of the approach. Existing work on explaining the predictions of classifiers such as Guidotti et al. (2019); Ribeiro et al. (2016) mainly focuses on textual content and images, and neglects explanations for the classification of subject-predicate-object triples in knowledge bases.

Vandalism Prevention with Novel User Interfaces

In this thesis, we focus on detecting vandalism after it has already happened. Another approach might be to focus on vandalism prevention before it happens. For example, we observed that certain kinds of vandalism are encouraged by the Wikidata user interfaces, and we found that changes to the Wikidata user interfaces correlated with changes to the kinds of vandalism. Hence, future work might further analyze this problem and design user interfaces that particularly discourage vandalism, for example, by making vandalism harder to commit for the occasional user and by educating users.

Quality of Knowledge Bases

Many quality management approaches for knowledge bases concentrate on assessing and managing the quality of knowledge bases as a whole (Zaveri et al., 2016; Färber et al., 2017). While this might be useful for choosing between different knowledge bases, this provides little actionable insights to improve the quality of a knowledge base. Our approach might serve as an example of fine-grained quality management with respect to the quality dimension of accuracy. In the future, other quality dimensions, such as completeness, consistency, timeliness might be managed on a fine-grained level, too.

Regarding completeness, Galárraga et al. (2017) developed an approach to detect gaps in knowledge bases. To fill such gaps in Wikidata, future work might extract the information automatically from the web, or editors might be encouraged to fill the gaps. The automatic extraction might be done with bootstrapping (Agichtein and Gravano, 2000) or distant supervision (Mintz et al., 2009; Dong et al., 2014). For example, in his bachelor's thesis, Scholten (2019) developed a bootstrapping approach to extract one particularly important predicate, *canCause*, from the web. Before, knowledge bases contained little causal information and were often unable to answer questions regarding the causes of diseases or natural disasters. For encouraging editors to fill the gaps, we envision a system making personal edit suggestions based on an editor's editing history.

Regarding consistency, in his master's thesis, Petkovic (2019) developed an approach to rank constraint violations according to their importance, such that the most important constraint violations can be fixed first. Future work might try our graph embeddings for this task and take our vandalism scores into account when assessing the importance of fixing a constraint violation.

Debiasing Vandalism Detection and Editor Retention

While our bias measure is based on group fairness and focuses on the two groups of anonymous and registered users, it would be interesting to explore further groups and to study biases based on individual fairness (Dwork et al., 2012) and causal modeling (Kilbertus et al., 2017; Kusner et al., 2017; Zhang and Wu, 2017).

Having evaluated bias-performance trade-offs in Wikidata, it would be interesting to develop a general framework for bias mitigation in online platforms, e.g., by employing evolutionary algorithms to explore the Pareto front of non-dominated models in bias-performance space. It also seems promising to add fairness metrics to automatic machine learning tools, such as TPOT (Olson et al., 2016) and ML-Plan (Mohr et al., 2018), which are based on evolutionary algorithms and hierarchical planning and could serve as a vehicle for broader adoption of fairness metrics. Similar fairness issues like on Wikidata have recently been found on Reddit, too (Jhaver et al., 2019).

Our motivation for developing fair vandalism detectors includes increased editor retention. While fair vandalism detectors contribute towards this goal, there might be further means such as explaining decisions to editors, improving user interfaces, creating onboarding programs for newcomers, increasing social interactions, and gamification. The effect of each single intervention might be measured in a data-driven process with A/B testing. Alternatively, the system might continuously optimize editor retention with techniques from reinforcement learning.

References

Adler, B. T., L. de Alfaro, S. M. Mola-Velasco, P. Rosso, and A. G. West. Wikipedia Vandalism Detection: Combining Natural Language, Metadata, and Reputation Features. In *CICLing*, pages 277–288. Springer, 2011.

Agichtein, E. and L. Gravano. Snowball: Extracting Relations from Large Plain-text Collections. In *ACM DL*, pages 85–94. ACM, 2000.

Amores, J. Multiple Instance Classification: Review, Taxonomy and Comparative Study. *Artificial Intelligence*, 201:81–105, 2013.

Baeza-Yates, R. Bias on the Web. *Commun. ACM*, 61(6):54–61, 2018.

Barocas, S. and A. D. Selbst. Big Data's Disparate Impact. *Cal. L. Rev.*, 104:671, 2016.

Berk, R., H. Heidari, S. Jabbari, M. Kearns, and A. Roth. Fairness in Criminal Justice Risk Assessments: The State of the Art. *Sociological Methods & Research*, 2018.

Bollacker, K., C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *SIGMOD*, pages 1247–1250. ACM, 2008.

Bordes, A., N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*, pages 2787–2795, 2013.

Boyd, K., V. S. Costa, J. Davis, and D. Page. Unachievable Region in Precision-Recall Space and Its Effect on Empirical Evaluation. In *ICML*, pages 639–646, 2012.

Breiman, L. Bagging Predictors. *Machine Learning*, 24(2):123–140, 1996.

Breiman, L. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

Buckels, E. E., P. D. Trapnell, and D. L. Paulhus. Trolls Just Want to Have Fun. *Personality and Individual Differences*, 67:97–102, 2014.

Chen, T. and C. Guestrin. XGBoost: A Scalable Tree Boosting System. In *KDD*, pages 785–794. ACM, 2016.

Chouldechova, A. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data*, 5(2):153–163, 2017.

Ciampaglia, G. L., P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, and A. Flammini. Computational Fact Checking from Knowledge Networks. *PLOS ONE*, 10(6):1–13, 2015.

Conroy, N. J., V. L. Rubin, and Y. Chen. Automatic Deception Detection: Methods for Finding Fake News. In *ASIST*. Wiley, 2015.

Corbett-Davies, S., E. Pierson, A. Feller, S. Goel, and A. Huq. Algorithmic Decision Making and the Cost of Fairness. In *KDD*, pages 797–806. ACM, 2017.

Crescenzi, R., M. Fernandez, F. A. G. Calabria, P. Albani, D. Tauziet, A. Baravalle, and A. S. D'Ambrosio. A Production Oriented Approach for Vandalism Detection in Wikidata—The Buffaloberry Vandalism Detector at WSDM Cup 2017. In *WSDM Cup*, 2017.

Crescenzi, R. Context Features for Vandalism Detection in Knowledge Bases. Master's thesis, Universidad Austral, 2018.

Davis, J. and M. Goadrich. The Relationship Between Precision-Recall and ROC Curves. In *ICML*, pages 233–240. ACM, 2006.

Del Vicario, M., A. Bessi, F. Zollo, F. Petroni, A. Scala, G. Caldarelli, H. E. Stanley, and W. Quattrociocchi. The Spreading of Misinformation Online. *Proceedings of the National Academy of Sciences*, 113(3):554–559, 2016.

Domingos, P. M. A Few Useful Things to Know About Machine Learning. *Commun. ACM*, 55(10):78–87, 2012.

Dong, X., E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In *KDD*, pages 601–610. ACM, 2014.

Dubey, M., D. Banerjee, A. Abdelkawi, and J. Lehmann. LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia. In *ISWC*, pages 69–78. Springer, 2019.

Dwork, C., M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel. Fairness Through Awareness. In *ITCS*, pages 214–226. ACM, 2012.

Färber, M., F. Bartscherer, C. Menne, and A. Rettinger. Linked Data Quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web*, pages 1–53, 2017.

Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.

Galárraga, L., S. Razniewski, A. Amarilli, and F. M. Suchanek. Predicting Completeness in Knowledge Bases. In *WSDM*, pages 375–383. ACM, 2017.

Gardner, M. and T. M. Mitchell. Efficient and Expressive Knowledge Base Completion Using Subgraph Feature Extraction. In *EMNLP*, pages 1488–1498. ACL, 2015.

Gärtner, T., P. A. Flach, A. Kowalczyk, and A. J. Smola. Multi-Instance Kernels. In *ICML*, pages 179–186, 2002.

Geiger, R. S. and A. Halfaker. Using Edit Sessions to Measure Participation in Wikipedia. In *CSCW*, pages 861–870. ACM, 2013.

Gerber, D., D. Esteves, J. Lehmann, L. Bühmann, R. Usbeck, A. Ngonga Ngomo, and R. Speck. DeFacto - Temporal and Multilingual Deep Fact Validation. *J. Web Semant.*, 35:85–101, 2015.

Goodfellow, I. J., J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. In *ICLR (Poster)*, 2015.

Grigorev, A. Large-Scale Vandalism Detection with Linear Classifiers— The Conkerberry Vandalism Detector at WSDM Cup 2017. In *WSDM Cup*, 2017.

Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.*, 51(5): 93:1–93:42, 2019.

Halfaker, A., A. Kittur, and J. Riedl. Don't Bite the Newbies: How Reverts Affect the Quantity and Quality of Wikipedia Work. In *Int. Sym. Wikis*, pages 163–172. ACM, 2011.

Halfaker, A., R. S. Geiger, J. T. Morgan, and J. Riedl. The Rise and Decline of an Open Collaboration System: How Wikipedia's Reaction to Popularity is Causing Its Decline. *American Behavioral Scientist*, 57(5):664–688, 2013.

Hardt, M., E. Price, and N. Srebro. Equality of Opportunity in Supervised Learning. In *NIPS*, pages 3315–3323, 2016.

Heindorf, S., M. Potthast, B. Stein, and G. Engels. Towards Vandalism Detection in Knowledge Bases: Corpus Construction and Analysis. In *SIGIR*, pages 831–834. ACM, 2015.

Heindorf, S., M. Potthast, B. Stein, and G. Engels. Vandalism Detection in Wikidata. In *CIKM*, pages 327–336. ACM, 2016.

Heindorf, S., M. Potthast, H. Bast, B. Buchhold, and E. Haussmann. WSDM Cup 2017: Vandalism Detection and Triple Scoring. In *WSDM*, pages 827–828. ACM, 2017a.

Heindorf, S., M. Potthast, G. Engels, and B. Stein. Overview of the Wikidata Vandalism Detection Task at WSDM Cup 2017. In *WSDM Cup 2017 Notebook Papers*, 2017b.

Heindorf, S., Y. Scholten, G. Engels, and M. Potthast. Debiasing Vandalism Detection Models at Wikidata. In *WWW*, pages 670–680. ACM, 2019a.

Heindorf, S., Y. Scholten, G. Engels, and M. Potthast. Debiasing Vandalism Detection Models at Wikidata (Extended Abstract). In *INFORMATIK*, pages 289–290, 2019b.

Javanmardi, S., D. W. McDonald, and C. V. Lopes. Vandalism Detection in Wikipedia: A High-Performing, Feature-Rich Model and its Reduction Through Lasso. In *Int. Sym. Wikis*, pages 82–90. ACM, 2011.

Jhaver, S., S. Appling, E. Gilbert, and A. Bruckman. Did You Suspect the Post Would be Removed? Understanding User Reactions to Content Removals on Reddit. In *CSCW*. ACM, 2019.

Kaffee, L., H. ElSahar, P. Vougiouklis, C. Gravier, F. Laforest, J. S. Hare, and E. Simperl. Learning to Generate Wikipedia Summaries for Underserved Languages from Wikidata. In *NAACL-HLT*, pages 640–645. ACL, 2018.

Khoshgoftaar, T. M., M. Golawala, and J. V. Hulse. An Empirical Study of Learning from Imbalanced Data Using Random Forest. In *ICTAI*, pages 310–317. IEEE Computer Society, 2007.

Kiesel, J., M. Potthast, M. Hagen, and B. Stein. Spatio-Temporal Analysis of Reverted Wikipedia Edits. In *ICWSM*, pages 122–131. AAAI Press, 2017.

Kilbertus, N., M. Rojas-Carulla, G. Parascandolo, M. Hardt, D. Janzing, and B. Schölkopf. Avoiding Discrimination through Causal Reasoning. In *NIPS*, pages 656–666, 2017.

Kleinberg, J. M., S. Mullainathan, and M. Raghavan. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *ITCS*, volume 67, pages 43:1–43:23. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

Kumar, S., F. Spezzano, and V. S. Subrahmanian. VEWS: A Wikipedia Vandal Early Warning System. In *KDD*, pages 607–616. ACM, 2015.

Kumar, S., R. West, and J. Leskovec. Disinformation on the Web: Impact, Characteristics, and Detection of Wikipedia Hoaxes. In *WWW*, pages 591–602. ACM, 2016.

Kumar, S. and N. Shah. False Information on Web and Social Media: A Survey. *CoRR*, abs/1804.08559, 2018.

Kusner, M. J., J. R. Loftus, C. Russell, and R. Silva. Counterfactual Fairness. In *NIPS*, pages 4069–4079, 2017.

Lao, N., T. M. Mitchell, and W. W. Cohen. Random Walk Inference and Learning in A Large Scale Knowledge Base. In *EMNLP*, pages 529–539. ACL, 2011.

Lazer, D. M., 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*, 359(6380):1094–1096, 2018.

Lehmann, J., D. Gerber, M. Morsey, and A. Ngonga Ngomo. DeFacto - Deep Fact Validation. In *ISWC*, pages 312–327. Springer, 2012.

Lenat, D. B. and R. V. Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

Lenat, D. B. The Voice of the Turtle: Whatever Happened to AI? *AI Magazine*, 29(2): 11–19, 2008.

Lin, Y., Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning Entity and Relation Embeddings for Knowledge Graph Completion. In *AAAI*, pages 2181–2187. AAAI Press, 2015.

Manning, C. D., P. Raghavan, and H. Schütze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

Martinez-Rico, J. R., J. Martinez-Romo, and L. Araujo. Can Deep Learning Techniques Improve Classification Performance of Vandalism Detection in Wikipedia? *Eng. Appl. of AI*, 78:248–259, 2019.

Micci-Barreca, D. A Preprocessing Scheme for High-Cardinality Categorical Attributes in Classification and Prediction Problems. *SIGKDD Explorations*, 3(1):27–32, 2001.

Mietchen, D., G. Hagedorn, E. Willighagen, M. Rico, A. Gómez-Pérez, E. Aibar, K. Rafes, C. Germain, A. Dunning, L. Pintscher, et al. Enabling Open Science: Wikidata for Research (Wiki4R). *Research Ideas and Outcomes*, 1:e7573, 2015.

Minkov, E., W. W. Cohen, and A. Y. Ng. Contextual Search and Name Disambiguation in Email Using Graphs. In *SIGIR*, pages 27–34. ACM, 2006.

Mintz, M., S. Bills, R. Snow, and D. Jurafsky. Distant Supervision for Relation Extraction Without Labeled Data. In *ACL/IJCNLP*, pages 1003–1011. ACL, 2009.

Mitraka, E., A. Waagmeester, S. Burgstaller-Muehlbacher, L. M. Schriml, A. I. Su, and B. M. Good. Wikidata: A Platform for Data Integration and Dissemination for the Life Sciences and Beyond. In *SWAT4LS*, volume 1546 of *CEUR Workshop Proceedings*, pages 69–73. CEUR-WS.org, 2015.

Mohr, F., M. Wever, and E. Hüllermeier. ML-Plan: Automated Machine Learning via Hierarchical Planning. *Machine Learning*, 107(8-10):1495–1515, 2018.

Mora-Cantallops, M., S. Sánchez-Alonso, and E. García-Barriocanal. A Systematic Literature Review on Wikidata. *Data Technologies and Applications*, pages 250–268, 2019.

Naphade, N. Vandalism Detection in Wikidata's Type Relations. Master's thesis, Paderborn University, 2018.

Neis, P., M. Goetz, and A. Zipf. Towards Automatic Vandalism Detection in OpenStreetMap. *ISPRS International Journal of Geo-Information*, 2012.

Nickel, M., V. Tresp, and H. Kriegel. Factorizing YAGO: Scalable Machine Learning for Linked Data. In *WWW*, pages 271–280. ACM, 2012.

Nickel, M., K. Murphy, V. Tresp, and E. Gabrilovich. A Review of Relational Machine Learning for Knowledge Graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016.

Nielsen, F. Å., D. Mietchen, and E. L. Willighagen. Scholia, Scientometrics and Wikidata. In *ESWC (Satellite Events)*, volume 10577, pages 237–259. Springer, 2017.

Nishioka, C. and A. Scherp. Analysing the Evolution of Knowledge Graphs for the Purpose of Change Verification. In *ICSC*, pages 25–32. IEEE Computer Society, 2018.

Olson, R. S., R. J. Urbanowicz, P. C. Andrews, N. A. Lavender, L. C. Kidd, and J. H. Moore. Automating Biomedical Data Science Through Tree-Based Pipeline Optimization. In *EvoApplications*, pages 123–137. Springer, 2016.

Pan, J. Z., S. Pavlova, C. Li, N. Li, Y. Li, and J. Liu. Content Based Fake News Detection Using Knowledge Graphs. In *ISWC*, pages 669–683. Springer, 2018.

Papernot, N., P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical Black-Box Attacks against Machine Learning. In *AsiaCCS*, pages 506–519. ACM, 2017.

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

Pedreschi, D., S. Ruggieri, and F. Turini. Discrimination-aware Data Mining. In *KDD*, pages 560–568. ACM, 2008.

Pellissier Tanon, T., D. Vrandecic, S. Schaffert, T. Steiner, and L. Pintscher. From Freebase to Wikidata: The Great Migration. In *WWW*, pages 1419–1428. ACM, 2016.

Pellissier Tanon, T., C. Bourgaux, and F. Suchanek. Learning How to Correct a Knowledge Base from the Edit History. In *WWW*, pages 1465–1475. ACM, 2019.

Petkovic, M. Ranking Constraint Violations in Knowledge Bases. Masters's thesis, Paderborn University, 2019.

Piscopo, A. and E. Simperl. What We Talk About When We Talk About Wikidata Quality: A Literature Survey. In *OpenSym*, pages 17:1–17:11. ACM, 2019.

Pleiss, G., M. Raghavan, F. Wu, J. M. Kleinberg, and K. Q. Weinberger. On Fairness and Calibration. In *NIPS*, pages 5680–5689, 2017.

Popat, K., S. Mukherjee, A. Yates, and G. Weikum. DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning. In *EMNLP*, pages 22–32. Association for Computational Linguistics, 2018.

Putman, T. E., S. Lelong, S. Burgstaller-Muehlbacher, A. Waagmeester, C. M. Diesh, N. A. Dunn, M. C. Munoz-Torres, G. S. Stupp, C. Wu, A. I. Su, and B. M. Good. WikiGenomes: An Open Web Application for Community Consumption and Curation of Gene Annotation Data in Wikidata. *Database*, 2017:bax025, 2017.

Ratner, A. J., C. D. Sa, S. Wu, D. Selsam, and C. Ré. Data Programming: Creating Large Training Sets, Quickly. In *NIPS*, pages 3567–3575, 2016.

Ratner, A., S. H. Bach, H. R. Ehrenberg, J. A. Fries, S. Wu, and C. Ré. Snorkel: Rapid Training Data Creation with Weak Supervision. *PVLDB*, 11(3):269–282, 2017.

Ribeiro, M. T., S. Singh, and C. Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *KDD*, pages 1135–1144. ACM, 2016.

Ringler, D. and H. Paulheim. One Knowledge Graph to Rule Them All? Analyzing the Differences Between DBpedia, YAGO, Wikidata & co. In *KI*, pages 366–372. Springer, 2017.

Romei, A. and S. Ruggieri. A Multidisciplinary Survey on Discrimination Analysis. *Knowledge Eng. Review*, 29(5):582–638, 2014.

Sarabadani, A., A. Halfaker, and D. Taraborelli. Building Automated Vandalism Detection Tools for Wikidata. In *WWW (Companion Volume)*, pages 1647–1654, 2017.

Schneider, J., B. S. Gelley, and A. Halfaker. Accept, Decline, Postpone: How Newcomer Productivity Is Reduced in English Wikipedia by Pre-Publication Review. In *OpenSym*, pages 26:1–26:10. ACM, 2014.

Scholten, Y. Towards a Large-scale Causality Graph. Bachelor's thesis, Paderborn University, 2019.

Shachaf, P. and N. Hara. Beyond Vandalism: Wikipedia Trolls. *Journal of Information Science*, pages 357–370, 2010.

Shi, B. and T. Weninger. Discriminative Predicate Path Mining for Fact Checking in Knowledge Graphs. *Knowl.-Based Syst.*, 104:123–133, 2016.

Shu, K., A. Sliva, S. Wang, J. Tang, and H. Liu. Fake News Detection on Social Media: A Data Mining Perspective. *SIGKDD Explorations*, 19(1):22–36, 2017.

Speck, R. and A.-C. Ngonga Ngomo. Leopard — A Baseline Approach to Attribute Prediction and Validation for Knowledge Graph Population. *J. Web Semant.*, 55:102 – 107, 2019.

Syed, Z. H., M. Röder, and A. Ngonga Ngomo. FactCheck: Validating RDF Triples Using Textual Evidence. In *CIKM*, pages 1599–1602. ACM, 2018.

Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing Properties of Neural Networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tan, C. H., E. Agichtein, P. Ipeirotis, and E. Gabrilovich. Trust, but Verify: Predicting Contribution Quality for Knowledge Base Construction and Curation. In *WSDM*, pages 553–562. ACM, 2014.

Tran, K. and P. Christen. Cross Language Prediction of Vandalism on Wikipedia Using Article Views and Revisions. In *PAKDD*, pages 268–279. Springer, 2013.

Tran, K. and P. Christen. Cross-Language Learning from Bots and Users to Detect Vandalism on Wikipedia. *IEEE Transactions on Knowledge and Data Engineering*, 27 (3):673–685, March 2015.

Truong, Q. T., G. Touya, and C. de Runz. Towards Vandalism Detection in OpenStreetMap Through a Data Driven Approach. In *GIScience*, volume 114 of *LIPIcs*, pages 61:1–61:7. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

Turki, H., T. Shafee, M. A. H. Taieb, M. B. Aouicha, D. Vrandečić, D. Das, and H. Hamdi. Wikidata: A Large-Scale Collaborative Ontological Medical Database. *Journal of Biomedical Informatics*, 99:103292, 2019.

Usbeck, R., A. Ngonga Ngomo, B. Haarmann, A. Krithara, M. Röder, and G. Napolitano. 7th Open Challenge on Question Answering over Linked Data (QALD-7). In *SemWebEval@ESWC*, volume 769 of *Communications in Computer and Information Science*, pages 59–69. Springer, 2017.

Vougiouklis, P., H. ElSahar, L. Kaffee, C. Gravier, F. Laforest, J. S. Hare, and E. Simperl. Neural Wikipedian: Generating Textual Summaries from Knowledge Base Triples. *J. Web Semant.*, 52-53:1–15, 2018.

Vrandečić, D. and M. Krötzsch. Wikidata: A Free Collaborative Knowledgebase. *Communications of the ACM*, pages 78–85, 2014.

Wang, R. W. and D. M. Strong. Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems*, 12(4):5–33, 1996.

Wang, Z., J. Zhang, J. Feng, and Z. Chen. Knowledge Graph Embedding by Translating on Hyperplanes. In *AAAI*, pages 1112–1119. AAAI Press, 2014.

Wang, S., W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy. Training Deep Neural Networks on Imbalanced Data Sets. In *IJCNN*, pages 4368–4374. IEEE, 2016.

Wang, Q., Z. Mao, B. Wang, and L. Guo. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.*, 29(12):2724–2743, 2017.

West, A. G., S. Kannan, and I. Lee. Detecting Wikipedia Vandalism via Spatio-temporal Analysis of Revision Metadata. In *EUROSEC*, pages 22–28. ACM, 2010.

Yamazaki, T., M. Sasaki, N. Murakami, T. Makabe, and H. Iwasawa. Ensemble Models for Detecting Wikidata Vandalism with Stacking—Team Honeyberry Vandalism Detector at WSDM Cup 2017. In *WSDM Cup*, 2017.

Yu, T., Y. Zhao, X. Wang, Y. Xu, H. Shao, Y. Wang, X. Ma, and D. Dey. Vandalism Detection Midpoint Report—The Riberry Vandalism Detector at WSDM Cup 2017. University of Illinois at Urbana-Champaign Student Report, not published, 2017.

Zaveri, A., D. Kontokostas, M. A. Sherif, L. Bühmann, M. Morsey, S. Auer, and J. Lehmann. User-driven Quality Evaluation of DBpedia. In *I-SEMANTICS*, pages 97–104. ACM, 2013.

Zaveri, A., A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer. Quality assessment for Linked Data: A Survey. *Semantic Web*, 7(1):63–93, 2016.

Zemel, R. S., Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning Fair Representations. In *ICML*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 325–333, 2013.

Zhang, L. and X. Wu. Anti-Discrimination Learning: A Causal Modeling-Based Framework. *I. J. Data Science and Analytics*, 4(1):1–16, 2017.

Zhu, Q., H. Ng, L. Liu, Z. Ji, B. Jiang, J. Shen, and H. Gui. Wikidata Vandalism Detection—The Loganberry Vandalism Detector at WSDM Cup 2017. In *WSDM Cup*, 2017.

Zliobaite, I. On the Relation Between Accuracy and Fairness in Binary Classification. *CoRR*, abs/1505.05723, 2015.

Zügner, D., A. Akbarnejad, and S. Günnemann. Adversarial Attacks on Neural Networks for Graph Data. In *KDD*, pages 2847–2856. ACM, 2018.