Graph-Linguistic Fusion: Using Language Models for Wikidata Vandalism Detection

Mykola Trokhymovych
Pompeu Fabra University
mykola.trokhymovych@upf.edu

Lydia Pintscher
Wikimedia Deutschland
lydia.pintscher@wikimedia.de

Ricardo Baeza-Yates
Pompeu Fabra University
rbaeza@acm.org

Diego Saez-Trumper
Wikimedia Foundation
diego@wikimedia.org

Abstract

We introduce a next-generation vandalism detection system for Wikidata, one of the largest open-source structured knowledge bases on the Web. Wikidata is highly complex: its items incorporate an ever-expanding universe of factual triples and multilingual texts. While edits can alter both structured and textual content, our approach converts all edits into a single space using a method we call Graph2Text. This allows for evaluating all content changes for potential vandalism using a single multilingual language model. This unified approach improves coverage and simplifies maintenance. Experiments demonstrate that our solution outperforms the current production system. Additionally, we are releasing the code under an open license along with a large dataset of various human-generated knowledge alterations, enabling further research.

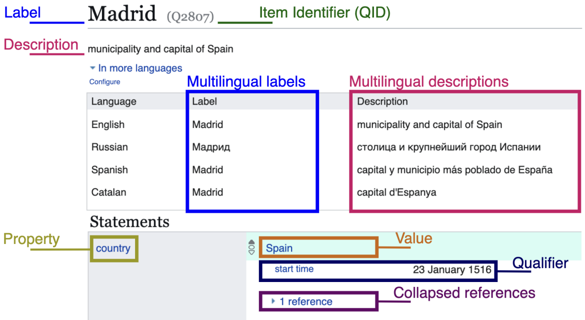
1 Introduction

Wikidata is a large open-source, multilingual knowledge graph that plays a key role in the modern Web. It was designed as a centralized, linked repository of structured data for all Wikimedia projects, including over 300 language versions of Wikipedia (Kent, 2019; Zhao, 2022).

Beyond the Wikimedia ecosystem, Wikidata is extensively used by the most popular web services, such as search engines (Kanke, 2021) and data for digital assistants like *Alexa* and *Siri* (Reagle and Koerner, 2020) as well as for AI models, bots, and scripts. Wikidata facilitates better question answering models, offers more context in search results, links to related sources efficiently, and helps reduce factual errors in large language models (Kent, 2019; Simonite, 2019; Xu et al., 2023).

Wikidata can be described as a document-oriented database focusing on items that represent any named entity (Wikipedia, 2024). Each entity is assigned a unique identifier (ID) and can include textual information such as labels, aliases, and descriptions in multiple languages. Another essential component is the *Statements*, which provides the information necessary to form semantic triples — a key component of the knowledge graph. Triples consist of tuples of *{entity,property,value}*, where the property defines the relationship between entity and the value. Values can be free text, numbers, dates, coordinates, or another entity. A diagram illustrating the key parts of a Wikidata record is presented in Figure 1. Hence, although Wikidata provides structured relationships among entities, the building blocks of this knowledge graph include many components of unstructured data, such as multilingual descriptions or values of various types.

Given its central role in the online knowledge ecosystem, the quality of Wikidata content has relevant implications for very prominent services and products. For example, due to vandalism in Bulgaria's Wikidata Entity in 2017 (see Figure 2), when iPhone users were asking *"What is the national anthem of Bulgaria?,"* the answer was *"Despacito"*, a popular song at that time (Reagle and Koerner, 2020). Vandalism has become more serious when it affects the reputation of people, institutions, or brands (Saez-Trumper, 2019). However, with Wikidata receiving around 10 edits *(a.k.a*



Madrid (Q2807)

municipality and capital of Spain

In more languages
Configure

Language	Label	Description
English	Madrid	municipality and capital of Spain
Russian	Мадрид	столица и крупнейший город Испании
Spanish	Madrid	capital y municipio más poblado de España
Catalan	Madrid	capital d'Espanya

Statements

country

Spain

start time 23 January 1516

1 reference

Figure 1: Diagram with the most important parts of the Wikidata record.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track), pages 284–294
July 28-30, 2025 ©2025 Association for Computational Linguistics

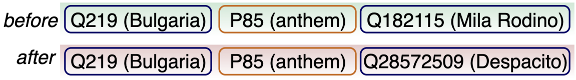


Figure 2: Example of a revision (ID: 593195479) vandalizing the Wikidata entry for Bulgaria. Original triple IDs are mapped to their corresponding English labels.

revisions) per second,[1] it becomes difficult for the human vandalism *patrollers* to analyze every single edit. Therefore, several methods have been proposed to assist the community in this task by using machine learning models. In fact, in 2016, the Wikimedia Foundation developed a system named ORES that is currently supporting the vandalism detection work on Wikidata. Unfortunately, the current ORES model is limited to certain types of edits and entities, and it cannot deal with the complexity of the different data types and topics coexisting in Wikidata.

This paper introduces a new generation model for detecting vandalism in Wikidata that can deal with the aforementioned complexities. A key aspect of the proposed solution is transforming all content changes, including structured data, into their textual equivalents (*Graph2Text*). This approach allows the processing of all types of content changes by transforming them into text and using a single language model that takes advantage of the rich semantic knowledge embedded within it.

The main contributions of this work are: *(i)* The next-generation vandalism detection system for Wikidata, utilizing multilingual language models to improve accuracy and fairness compared to the current production model; *(ii)* System productionalization addressing limitations imposed by resource-constrained infrastructure and product requirements; *(iii)* The publication of a new open benchmark dataset for vandalism detection in Wikidata, containing about 5M unique samples.[2]

2 Related work

2.1 Vandalism detection in Wikipedia

Vandalism detection in Wikidata is closely related to the same problem in Wikipedia. Both services operate within the Wikimedia Foundation ecosystem, share similar editing mechanisms, and have many common users (Sarabadani et al., 2017). Initial research on Wikipedia vandalism detection systems appeared much earlier and laid the groundwork for similar tools in Wikidata.

Early models for Wikipedia vandalism detection were binary classifiers that used generic features, such as the ratio of uppercase letters and term frequency (Potthast et al., 2008). Later studies also explored the relationship between editing behavior, editors' characteristics, link structure, and article quality on Wikipedia (Ruprechter et al., 2020). The most recent work proposed a vandalism detection model for Wikipedia utilizing advanced content change features based on transformer models (Trokhymovych et al., 2023).

Additionally, investigations into vandalism detection on other open-source platforms like Freebase and OpenStreetMap, which analyzed vandalism patterns and proposed various detection approaches, provide valuable insights applicable to our work due to the shared similarities among these platforms (Tan et al., 2014; Neis et al., 2012).

2.2 Vandalism detection in Wikidata

With the launch of Wikidata in 2012, it quickly became one of the most edited projects within the Wikimedia Foundation ecosystem (Vrandečić and Krötzsch, 2014). As with any open-knowledge project, maintaining the content reliable and verifiable has been a challenge. The first research addressing this issue emerged, introducing WDVC-2015, a corpus designed for detecting vandalism based on the entire revision history up to that point (Heindorf et al., 2015). This corpus facilitated the understanding of vandalism patterns on Wikidata and provided a foundation for developing automatic vandalism detection models.

Subsequently, several approaches have been published, introducing revision classifiers to determine whether specific revisions include vandalism. These approaches employed machine learning, using features from both an edit's content and its context (Heindorf et al., 2016; Sarabadani et al., 2017). One of these solutions, WDVD, proposed a model based on an extensive set of 47 content and user features, utilizing the random forest algorithm (Heindorf et al., 2016). Later, the Wikimedia research team introduced the ORES model, designed to function effectively in real-world applications with a much smaller feature set. This feature set was primarily established through community consultations and reflected the key concerns of Wikidata patrollers (Sarabadani et al., 2017).

Morover, the Wikidata Vandalism Detection

[1] https://stats.wikimedia.org/

[2] https://zenodo.org/records/15492678

Task at the WSDM Cup 2017 (Heindorf et al., 2017) introduced a new dataset and received five software submissions, contributing significantly to advancements in the field (Yu et al., 2017; Zhu et al., 2017; Yamazaki et al., 2017; Grigorev, 2017; Crescenzi et al., 2017).

2.3 Bias in vandalism detection

Even though the Wikipedia community is generally open to anyone, editors need specific skills and an understanding of community rules, which poses a challenge for newcomers. Previous research has shown that newcomer retention in Wikimedia projects is significantly affected by the reversion of their edits (Halfaker et al., 2013; Schneider et al., 2014). While newcomers and anonymous users are statistically more prone to mistakes, a biased model that unfairly cancels their edits could result in a long-term decline in the number of editors.

One of the primary reasons for this issue is that earlier models primarily relied on user characteristics and revision metadata, using a very modest set of features to characterize actual content changes. Recent advancements in Wikipedia vandalism detection models have shown that enhancing content change processing can both improve model performance and make the system fairer for anonymous users (Trokhymovych et al., 2023).

Similar to previous research, our focus is on processing content changes to enhance the predictive power of content features and reduce model bias. For evaluation, we employ group fairness metrics such as Disparate Impact Ratio (DIR) and the difference in AUC between privileged and unprivileged user groups (Bellamy et al., 2018).

3 System design

3.1 Design requirements

First, our main goal is to determine if a specific Wikidata edit is vandalism. We frame this as a binary classification problem. In practice, the probability score is often more important than binary prediction, as it enables the prioritization of tasks for patrollers or the automatic reversion of changes by applying stricter thresholds.[3]

Second, we aim to develop a single multilingual model that can process various types of content modifications (e.g., inserts, removes, changes). While Wikidata is largely language agnostic, it includes crucial elements like labels and descriptions that can appear in multiple languages for each record. Single multilingual model allows to extend the range of content edits that the model can effectively handle and reduce the infrastructure costs associated with maintaining multiple models for different content types and languages.

Third, the system requires to be efficient enough to handle a high volume of edits in a production environment. Wikidata receives about 10 edits per second, and our model should be capable of processing all of them. We also aim to develop a system that can operate with the existing resources on the Wikimedia ML Infrastructure, called LiftWing,[4] that currently[5] has no GPU acceleration for inference. This high edit frequency and focus on CPU-based models rule out most LLMs.

Finally, the system must not cause undue harm to good-faith editors. Past work has shown that reverting edits by newcomers can deter new contributors (TeBlunthuis et al., 2018). It is important that any deployed model does not unfairly target these newer editors.

3.2 Architecture overview

Our proposed system receives Wikidata revisions as input and returns a revert-risk score, indicating the probability of a given revision being reverted. The system mainly consists of three main logical steps: *(i)* features preparation; *(ii)* multilingual language model classifier for content processing; *(iii)* final classification model to aggregate content and revision meta-features. The full system schema is presented in Figure 3.

3.2.1 Feature processing

Wikidata entity's content is represented in a complex nested structure of dictionaries and lists. Consequently, parsing content modifications can be quite challenging, as these modifications may involve structural changes (e.g., converting a single value to a list), value edits across various entities (e.g., text in different languages, numerical values, dates), and different types of content modifications (e.g., insertions, deletions, changes). Therefore, feature preparation is a critical component of the system we present.

We distinguish three main types of features. The first type is *revision metadata*, which includes fea-

[3] See `https://www.mediawiki.org/wiki/Moderator_Tools/Automoderator` for an example from Wikipedia.

[4] LiftWing: `https://wikitech.wikimedia.org/wiki/Machine_Learning/LiftWing`

[5] As of March 1, 2025, this fact is valid.

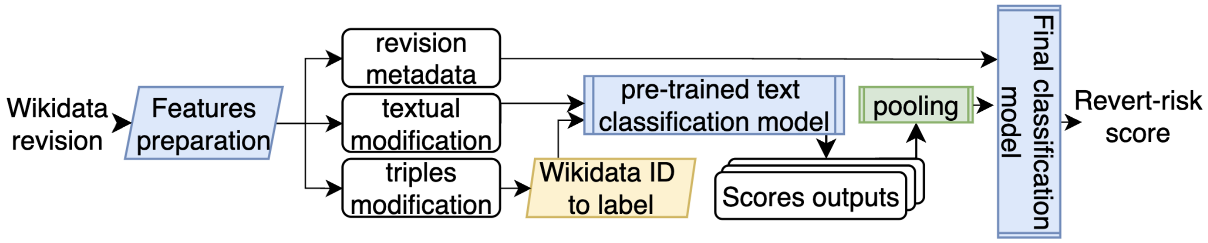


Figure 3: Wikidata vandalism detection system schema.

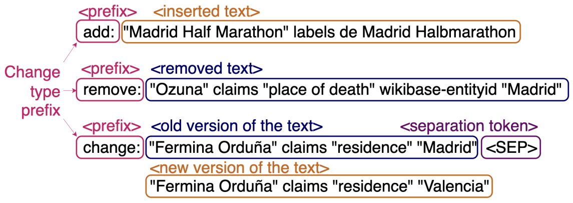

<prefix>
<inserted text>
add:
"Madrid Half Marathon" labels de Madrid Halbmarathon
Change type prefix
<prefix>
<removed text>
remove:
"Ozuna" claims "place of death" wikibase-entityid "Madrid"
<prefix>
<old version of the text>
<separation token>
change:
"Fermina Orduña" claims "residence" "Madrid"
<SEP>
<new version of the text>
"Fermina Orduña" claims "residence" "Valencia"

Figure 4: Text processing schema.

tures that require no additional processing and can be used directly in the final classification model (e.g., editor account creation date, time since previous revision, etc.).

Despite Wikidata's general language-agnostic nature, its entities have textual characteristics of any language. The second feature type represents Wikidata *textual modifications*, which refer to changes in elements such as entity labels, descriptions, or aliases.

The third feature group is *triples modification*. Wikidata triples are composed of three parts: the entity, the property, and the value. The entity and property are represented by their corresponding Wikidata IDs. The value can also be represented by an ID, but it may also be free text, a date, a numeric value, etc. To process this content together with textual changes, we convert the triples into text by mapping the IDs to their corresponding English labels.

It is important to note that both *textual* and *triples modification* can be of different types, such as insert, remove, and change. To process these modifications using a single language model (LM), we prepend a corresponding prefix text to the input sequence (see Figure 4), inspired by the "text-to-text" used in the T5 model (Raffel et al., 2019). This approach allows the LM to distinguish between different types of edits.

3.2.2 Language model classifier

To process content changes, specifically the previously discussed *textual* and *triples modification*, we fine-tune a single multilingual language model for binary classification tasks. Following the experience of a similar model for Wikipedia, we utilize the *bert-base-multilingual-cased*, which was pretrained with approximately 100 languages with the largest presence on Wikipedia (Trokhymovych et al., 2023; Devlin et al., 2019). Each revision may include multiple individual content changes of different types (e.g., a single revision might modify both a description and a factual triplet). During training, each of these changes is treated as an independent sample with the label of the revision. While inference, each of changes is independently processed by the language model classifier (LMC), with the following aggregation using mean pooling.

3.2.3 Final classification model

For the final classification step, we utilize the CatBoost classifier (Dorogush et al., 2017). This model is trained using both the *revision metadata* and the aggregated LMC outputs. The CatBoost classifier then generates a probability score indicating the likelihood of a revision being reverted. Details about the hyperparameters and computational resources can be found in Appendix A.

3.3 Deployment details

The complete system includes the extraction of the content using the Wikimedia API, feature engineering, and final model prediction. The inference pipeline is standardized and published under an open license in a dedicated repository of similar tools.[6] Additional testing with editors and community discussion would still be required prior to deployment.

4 Data preparation

Initially, we collect metadata for all human-created Wikidata revisions between September 1, 2021, and September 1, 2023. It includes information about

[6] https://gitlab.wikimedia.org/repos/research/knowledge_integrity

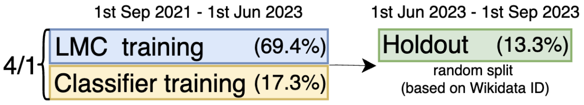

1st Sep 2021 - 1st Jun 2023
1st Jun 2023 - 1st Sep 2023
4/1
LMC training (69.4%)
Classifier training (17.3%)
Holdout (13.3%)
random split
(based on Wikidata ID)

Figure 5: Data splitting logic.

the Wikidata record, the user who performed the change, and specifics of the individual edit. To ensure that the revisions are human-created, we filter for revisions tagged with *Wikidata user interface*. Also, to improve data quality and reduce the noise in the revert signal, which we use as an indicator of vandalism, we additionally filter out several types of revisions (e.g., *self-reverts* and revisions involved in *"edit wars"*).

Wikidata entity's content is saved in the form of JSON. We extract the content for both the current and previous (parent) revisions and then compare them to identify differences. In particular, we employ Deepdiff[7] to extract the fine-grained signals from content modifications. We parse the content differences, getting features in the form of a list of inserts, removes, and changes. This includes but is not limited to alterations in descriptions, labels, and knowledge triples. Additional data processing details and explanations are included in Appendix B.

We utilize a time-based split to allocate the last three months of collected data as the holdout testing set (see Figure 5). This portion of the dataset is reserved exclusively for the final system evaluation. It ensures that our evaluation strategy represents real-world usage scenarios and helps to avoid time-related anomalies. The remaining data are used to train the components needed for the final system.

As the proposed system consists of multiple related and independently trainable components, we divide the training part into two groups following an 80/20 split, to prevent data leakage during training. The larger portion is used for the LMC, and the smaller for the final classifier. Final dataset characteristics details are presented in Appendix C.

5 Evaluation

5.1 Baselines

We compare our proposed model with four different baselines. As a dummy baseline, we build a *Rule-based* model that considers all edits done by anonymous editors as vandalism. In addition, we use two strong baseline models based on subsets of the features used in the final model: the *Metadata-based Classifier (MbC)* that uses only metadata features such as user group and age and the *Content-based Classifier (CbC)*, ignores user characteristics and uses only content modification features. Both are classification models constructed with the same methodology described in Section 3.2.3.

Table 1: System performance on holdout testing set.

Model	AUC	CI	FR@99	FR@90	FR@70
Rule-based	0.760	[0.74, 0.78]	0.0	0.0	0.92
ORES	0.859	[0.84, 0.87]	0.45	0.88	0.94
MbC	0.880	[0.87, 0.89]	0.55	0.89	0.94
CbC	0.876	[0.86, 0.89]	0.60	0.82	0.93
Graph2Text	**0.924**	[0.91, 0.93]	**0.71**	**0.91**	**0.96**

Our main reference model is ORES, the current production model for vandalism detection. This model mainly relies on metadata and includes some basic content features, such as binary indicators for changes in gender, date of birth, or English labels, to detect common vandalism patterns. We compare ORES and the previously mentioned baselines with our proposed system, *Graph2Text*, which integrates advanced content modification features based on the language model, along with revision metadata features and user characteristics.

5.2 System performance

The primary metric we use for model comparison is the Area under the ROC curve, the AUC score. The AUC score can be interpreted as the probability that the model assigns a higher score to a random positive example than to a random negative example. Also, we compute confidence intervals (CI) for our main metric using bootstrapping (see details in Appendix D) (Efron and Tibshirani, 1994).

Additionally, we employ a Filter Rate at the recall level (FR@) as suggested in previous work (Sarabadani et al., 2017). This metric measures the proportion of edits that can be removed from Wikidata patrollers reviewing backlog, with all the remaining revisions containing a specific percentage of all vandalism.

The results of our evaluation are summarized in Table 1. Our system, *Graph2Text*, significantly outperforms all other models across all metrics. Additionally, we observe that incorporating content features significantly improves the metrics compared to the *MbC*, just as adding user features enhances the *CbC*. Notably, the *CbC*, which uses only advanced content features without user characteristics, performs comparably to the *MbC*. This marks

[7] https://github.com/seperman/deepdiff

Table 2: System performance on expert-labeled data.

Model	AUC	CI	FR@99	FR90	FR70
ORES	0.885	[0.879, 0.892]	0.593	0.799	0.881
Graph2Text	**0.932**	[0.926, 0.937]	**0.698**	**0.846**	**0.918**

a significant advancement compared to previous approaches, where such performance was impossible without user characteristics. The performance based at FR99 indicates that with *Graph2Text* (compared to ORES), patrollers will need to analyze nearly half as many revisions to detect 99% of all vandalized samples (29% vs. 55%). Additional experiments, including performance evaluations for various prediction thresholds and use cases, are presented in Appendix E.

5.3 Expert evaluation

In practice, the holdout dataset, based on community-generated data, may include revisions that have not yet been reverted or were mistakenly reverted. To enhance the validity of our evaluation, we created a subsample of 1,000 revisions for expert labeling. We divided the holdout dataset into ten bins based on scores from the ORES and *Graph2Text* models separately. For each model and bin, we randomly selected fifty revisions without replacement. An experienced editor labeled these revisions as *Keep*, *Revert*, or *Not Sure*. Revisions labeled *Not Sure* were excluded from the final evaluation, resulting in 755 labeled revisions. The evaluation results, shown in Table 2, demonstrate that consistent with the performance evaluation using community-generated labels as the ground truth, the *Graph2Text* model significantly outperforms ORES on the expert-labeled data.

5.4 Fairness evaluation

Anonymous user's edits tend to have a higher likelihood of being vandalized compared to those by registered users, primarily due to factors such as a lack of experience in editing pages or intentional identity hiding for committing vandalism. The same situation applies to newly registered users. Nevertheless, it is unacceptable for the model to discriminate based on this characteristic. On the contrary, Wikidata encourages the participation of newcomer editors.

Table 3: System fairness based on holdout testing set.

Model	DIR^{anon}	DAUC^{anon}	DIR^{new}	DAUC^{new}
ORES	5.69	0.035	1.37	-0.193
MbC	4.09	0.097	1.15	-0.155
CbC	**2.07**	-0.04	**1.08**	**-0.027**
Graph2Text	4.43	**-0.01**	1.24	-0.096

To evaluate bias against anonymous users and new editors, we use two metrics: the Disparate Impact Ratio (DIR) and the Difference in AUC score (DAUC). For more details, please refer to Appendix D. In particular, the closer DAUC is to 0, the better. We compare these metrics for anonymous versus registered users and newcomers versus experienced users among the registered group. Table 3 summarizes the results of our evaluations.

Our analysis shows that our proposed model has lower DIR^{anon} and DIR^{new} values, indicating fairer treatment of anonymous and new users compared to ORES. Moreover, the difference in AUC scores between anonymous and registered users is significantly smaller, suggesting our model performs more consistently across these groups.

Although the proposed Graph2Text model demonstrates improved performance over the current ORES system, the CbC baseline, which disregards user attributes, achieves the highest fairness scores. However, our objective is to balance both predictive performance and fairness, while also maintaining applicability in scenarios where content features are not available. Consequently, we selected Graph2Text as our final model.

6 Discussion

To sum up, we present a study focused on developing a new generation of systems for detecting vandalism on Wikidata. The key innovation of the presented approach is the use of a single multilingual language model, which enables the processing of content changes in both structured and unstructured components in multiple languages. We demonstrate that the proposed system significantly outperforms the current production model in terms of both performance and fairness.

In this paper, we cover all the crucial steps needed to build a production-ready system, including the definition of design requirements, data collection and processing, feature engineering, model training, and evaluation.

Finally, we created a new dataset capturing changes made to the Wikidata platform over a two-year period. In addition to metadata, the dataset includes detailed content edits, represented by fine-grained differences between two versions of Wikidata items. We published the dataset and the code[8] under an open license to enable further research in this area.

6.1 Limitations

When interpreting the results, it's important to recognize several limitations of this study. First, the data preparation process can be improved by expanding parsing coverage, such as including changes in qualifiers or rankings. Also, using labels in non-English languages for mapping Wikidata IDs to text may enhance model performance by increasing coverage and diversifying the data.

Although the language model we fine-tuned was initially trained with about 100 languages, it still doesn't cover all of the 300+ languages represented in Wikidata. Considering these factors, we conclude that there are still issues with language diversity. Furthermore, we tested only one language model for our task. We believe that experimenting with more language models could improve the system's performance, which we leave for future research.

6.2 Ethical considerations

We introduce a new dataset designed to train models to predict the risk of reverts in Wikidata changes. The dataset includes metadata about revisions and editors but ensures the protection of Wikidata editors' privacy by not including any private or personally identifiable information.

We use crowd-sourced targets, which can include bias and noise, but we address this by filtering the data to minimize noise and clean the dataset. Moreover, we evaluate the system using the subsample labeled by experts. We also evaluate model fairness and ensure we reduce bias against anonymous users.

The intended use of the model is to detect vandalism edits in Wikidata. One of the risks we care about is over-reliance on automated detection. However, the presented system includes human-in-the-loop by design, meaning human moderators retain final decision-making control while receiving enhanced assistance.

Language models can perform differently across languages (Cotterell et al., 2018). Consequently, there is a potential risk that our system may have worse performance for underrepresented languages. To address this concern, we conducted additional experiments to verify that our system significantly outperforms alternatives on both revisions with English and non-English textual content (see Appendix Section E.2).

Another potential risk of our approach is adversarial exploitation, as open access to the code and dataset could enable bad actors to design edits that bypass detection. However, we select this transparency to promote trust, accelerate further research, and enable the community to review, audit, and improve the system.

Acknowledgments

The work of Mykola Trokhymovych is funded by MCIN/AEI /10.13039/501100011033 under the Maria de Maeztu Units of Excellence Programme (CEX2021-001195-M). This paper was partially supported by the ICT PhD program of Universitat Pompeu Fabra through a travel grant.

References

Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. 2018. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias.

Ryan Cotterell, Sabrina J. Mielke, Jason Eisner, and Brian Roark. 2018. Are all languages equally hard to language-model? In *Proceedings of NAACL'18: Human Language Technologies, Volume 2 (Short Papers)*, pages 536–541.

Rafael Crescenzi, Marcelo Fernández, Federico A. Garcia Calabria, Pablo Albani, Diego Tauziet, Adriana Baravalle, and Andrés Sebastián D'Ambrosio. 2017. A production oriented approach for vandalism detection in wikidata - the buffaloberry vandalism detector at WSDM cup 2017. *CoRR*, arXiv:1712.06919.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the NAACL-HLT'19*, pages 4171–4186.

Anna Veronika Dorogush, Andrey Gulin, Gleb Gusev, Nikita Kazeev, Liudmila Ostroumova, and Aleksandr Vorobev. 2017. Fighting biases with dynamic boosting.

Bradley Efron and R.J. Tibshirani. 1994. *An Introduction to the Bootstrap*, 1st edition edition. Chapman and Hall/CRC, New York.

[8] `https://github.com/trokhymovych/wikidata-vandalism-detection`

Alexey Grigorev. 2017. Large-scale vandalism detection with linear classifiers - the conkerberry vandalism detector at WSDM cup 2017. *CoRR*, arXiv:1712.06920.

Aaron Halfaker, R.Stuart Geiger, Jonathan Morgan, and John Riedl. 2013. The rise and decline of an open collaboration system how wikipedia's reaction to popularity is causing its decline. *American Behavioral Scientist*, 57:664–688.

Stefan Heindorf, Martin Potthast, Gregor Engels, and Benno Stein. 2017. Overview of the wikidata vandalism detection task at WSDM cup 2017. *CoRR*, arXiv:1712.05956.

Stefan Heindorf, Martin Potthast, Benno Stein, and Gregor Engels. 2015. Towards vandalism detection in knowledge bases: Corpus construction and analysis. In *Proceedings of SIGIR '15*, page 831–834.

Stefan Heindorf, Martin Potthast, Benno Stein, and Gregor Engels. 2016. Vandalism detection in wikidata. In *Proceedings of CIKM '16*, page 327–336.

Timothy Kanke. 2021. Knowledge curation work in wikidata wikiproject discussions. *Library hi tech*, 39(1):64–79.

Will Kent. 2019. Why is wikidata important to you? `https://wikiedu.org/blog/2019/06/03/why-is-wikidata-important-to-you/`. Accessed on October 6, 2024.

Pascal Neis, Marcus Goetz, and Alexander Zipf. 2012. Towards automatic vandalism detection in openstreetmap. *ISPRS International Journal of Geo-Information*, 1(3):315–332.

Martin Potthast, Benno Stein, and Robert Gerling. 2008. Automatic vandalism detection in wikipedia. In *Advances in Information Retrieval*, pages 663–668, Berlin, Heidelberg. Springer Berlin Heidelberg.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, arXiv:1910.10683.

Joseph Reagle and Jackie Koerner. 2020. *Wikipedia @ 20: Stories of an Incomplete Revolution*. The MIT Press, Cambridge, MA.

Thorsten Ruprechter, Tiago Santos, and Denis Helic. 2020. Relating wikipedia article quality to edit behavior and link structure. *Applied Network Science*, 5:61.

Diego Saez-Trumper. 2019. Online disinformation and the role of wikipedia.

Amir Sarabadani, Aaron Halfaker, and Dario Taraborelli. 2017. Building automated vandalism detection tools for wikidata. In *Proceedings of WWW '17 Companion*, page 1647–1654.

Jodi Schneider, Bluma S. Gelley, and Aaron Halfaker. 2014. Accept, decline, postpone: How newcomer productivity is reduced in english wikipedia by prepublication review. In *Proceedings of The International Symposium on Open Collaboration*.

Tom Simonite. 2019. Inside the alexa-friendly world of wikidata. `https://www.wired.com/story/inside-the-alexa-friendly-world-of-wikidata/`. Accessed on October 6, 2024.

Chun How Tan, Eugene Agichtein, Panos Ipeirotis, and Evgeniy Gabrilovich. 2014. Trust, but verify: predicting contribution quality for knowledge base construction and curation. In *Proceedings of WSDM '14*, page 553–562.

Nathan TeBlunthuis, Aaron Shaw, and Benjamin Mako Hill. 2018. Revisiting "the rise and decline" in a population of peer production projects. In *Proceedings of CHI '18*, page 1–7.

Mykola Trokhymovych, Muniza Aslam, Ai-Jou Chou, Ricardo Baeza-Yates, and Diego Saez-Trumper. 2023. Fair multilingual vandalism detection system for wikipedia. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 4981–4990, New York, NY, USA. Association for Computing Machinery.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Wikipedia. 2024. Wikidata. Accessed on October 6, 2024.

Silei Xu, Shicheng Liu, Theo Culhane, Elizaveta Pertseva, Meng-Hsi Wu, Sina Semnani, and Monica Lam. 2023. Fine-tuned LLMs know more, hallucinate less with few-shot sequence-to-sequence semantic parsing over Wikidata. In *Proceedings of EMNLP'23*, pages 5778–5791.

Tomoya Yamazaki, Mei Sasaki, Naoya Murakami, Takuya Makabe, and Hiroki Iwasawa. 2017. Ensemble models for detecting wikidata vandalism with stacking - team honeyberry vandalism detector at WSDM cup 2017. *CoRR*, arXiv:1712.06921.

Tuo Yu, Yiran Zhao, Xiaoxiao Wang, Yiwen Xu, Huajie Shao, Yuhang Wang, Xin Ma, and Dipannita Dey. 2017. Vandalism detection midpoint report—the riberry vandalism detector at wsdm cup 2017. http://www.wsdm-cup-2017.org/proceedings.html. University of Illinois at Urbana-Champaign Student Report, not published.

Fudie Zhao. 2022. A systematic review of Wikidata in Digital Humanities projects. *Digital Scholarship in the Humanities*, 38(2):852–874.

Qi Zhu, Hongwei Ng, Liyuan Liu, Ziwei Ji, Bingjie Jiang, Jiaming Shen, and Huan Gui. 2017. Wikidata vandalism detection - the loganberry vandalism detector at WSDM cup 2017. *CoRR*, arXiv:1712.06922.

A Modeling details

To process content changes we utilize the *bert-base-multilingual-cased*[9] (∼178M parameters). We fine-tune the model for five epochs with an initial learning rate of $2e^{-5}$ and a weight decay of 0.01. The batch size during training is set to 8. We reserve random 5% of the training data as the validation set. Throughout the training process, we track the loss and select the checkpoint from the epoch where the model performs best on the validation data as the final model. Training the model requires approximately 30 GPU hours (1x AMD Radeon Pro WX 9100 16GB GPU). The choice of hyperparameter values was guided by previous approaches using similar models that have demonstrated strong performance (Trokhymovych et al., 2023).

As for the final classification model, which aggregates all the revision meta-features and outputs of LMC, we use the CatBoost classifier. We train it with 2500 iterations, a learning rate of 0.005, and a parameter selection strategy that determines the final model weights based on the iteration, achieving the best loss on the validation dataset.

B Data preparation

B.1 Data sources

Our dataset construction process involves extracting data from multiple sources within the Wikimedia Data Lake.[10] In particular, we are utilizing the *mediawiki history* table to collect metadata for all human-created Wikidata revisions and *mediawiki wikitext history* table to get the Wikidata entity's content in the form of JSON. The mentioned data is available under an open license. Also, given the rarity of reverts, the initial dataset is highly imbalanced. To address this issue, we balance the dataset by retaining all reverted revisions and supplementing them with a random sample of unreverted revisions at a ratio of 1:5. The collected and processed dataset is published under an open license on the Zenodo platform to support further research.

B.2 Data filtering

To improve data quality and reduce the noise in the revert signal, which we use as an indicator of vandalism, we apply several filters. Specifically, we filter out *self-reverts*, which are revisions reverted by the same user who created them. These reverts typically occur shortly after the revision's creation and are part of an iterative page editing process. Since self-reverts usually do not indicate vandalism, it is essential to filter them out to avoid falsely marking these cases as potential vandalism. Additionally, inspired by the process proposed in (Trokhymovych et al., 2023), we filter out revisions involved in *"edit wars"*. Edit wars are characterized by sequential revisions that revert one another. In these instances, half of the reverted revisions represent good-faith changes intended to remove vandalism. However, as it is challenging to automatically differentiate between vandalism and good-faith changes, we eliminate all such revisions to reduce noise. Overall, these two filters removed about 57.7% of all revisions initially labeled as "reverted".

B.3 Content processing

Content changes to Wikidata items include alterations in descriptions, labels, and knowledge triples (see examples in Figure 6). To leverage a single language model (LM) for processing all content features, we employ specific data preparation techniques. Textual changes, such as descriptions, can be directly fed into the LM. However, graph-based features, such as knowledge triples, require additional processing. To integrate these into the LM, we convert knowledge triples into textual equivalents by mapping Wikidata IDs to their corresponding English labels. For the approximately 9% of IDs that lack corresponding labels (*i.e.* they have just an ID without a human-readable English equivalent), we map them to a default value, "unknown," which also provides a useful signal to the model. Additionally, as detailed earlier in Section 3.2.1, we prepend action-specific prefixes to all the input data. These prefixes supply the LM with context regarding the type of modification being processed.

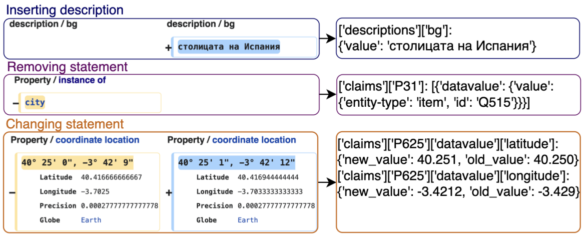
B.4 Data balancing

We use the separate splits to train each of the components of the final system. This split is done randomly, ensuring that all revisions for a specific Wikidata entity are contained within only one of the datasets. This approach is designed to prevent contextual leakage.

Each training dataset part is further divided into separate training and validation sets. For the content model LMC, we use a random split where

[9] https://huggingface.co/google-bert/bert-base-multilingual-cased

[10] https://wikitech.wikimedia.org/wiki/Analytics/Data_Lake



```
['descriptions']['bg']:
{'value': 'столицата на Испания'}
```

```
['claims']['P31']: [{'datavalue': {'value':
{'entity-type': 'item', 'id': 'Q515'}}}]
```

```
['claims']['P625']['datavalue']['latitude']:
{'new_value': 40.251, 'old_value': 40.250}
['claims']['P625']['datavalue']['longitude']:
{'new_value': -3.4212, 'old_value': -3.429}
```

Figure 6: Examples of fine-grained signals extracted from Wikidata content JSON in diverse forms and content types.

Table 4: Data characteristics.

Dataset	# of samples	Period	Anon. rate	Revert rate
Training	4,197,231	21 months	10.7%	7.9%
Hold-out	645,264	3 months	8.3%	6.2%

Table 5: Revert rate by modification type.

Type	Revert rate	# of samples	Anon. rate
Insert	11%	4,603,084	7%
Change	29%	1,093,665	24%
Remove	35%	530,317	14%
More than one type	36%	183,570	14%

5% of the data is allocated for model validation. In contrast, for the final classifier, we employ a time-based split, mirroring the logic of the holdout set, by dedicating all revisions from the last three months for validation.

It is important to note that the obtained datasets are unbalanced. For the LMC model training, we address this imbalance by random downsampling the overrepresented class of non-reverted changes, achieving a completely balanced dataset. For the CatBoost model, we utilize the *class_weights* parameter to adjust the importance of the underrepresented class, increasing its weight according to the level of disproportion.

C Data characteristics

The dataset is divided into two parts: a training set and a hold-out validation set, which is used for the final evaluation presented in Section 5.

The complete dataset contains 4,842,495 revisions spanning 24 months. Key data characteristics are summarized in Table 4. In particular, we report the rate of edits made by anonymous users and the revert rate.

We also analyze the types of modifications made by editors (see Table 5). We found that most revisions involve adding information to a Wikidata entity. This modification type also has the smallest revert rate and the lowest rate of anonymous edits. Revisions that include multiple modification types simultaneously are the most prone to containing vandalism.

It is worth noting that textual changes (modifying Wikidata entity descriptions or labels) in our dataset account for 25% of all revisions and 16.7% of all reverts. While English is the most popular language, it represents only 25% of all textual changes. Other prominent languages in the top 10 include German, French, Spanish, Italian, Russian, Japanese, Swedish, Simplified Chinese, and Dutch, which, along with English, make up 62% of the total. There are about 200 languages represented with at least 100 revisions. Revert rates vary significantly across different languages; for instance, English has a revert rate of 19%, while Swedish has 3.7%.

D Evaluation

D.1 Confidence intervals

To compute confidence intervals for our main metric, we employ a bootstrapping technique (Efron and Tibshirani, 1994). Specifically, we create 10K random samples, each of size 10K, by sampling with replacement. We then calculate the standard deviation of the AUC scores across these 10K bootstrap samples. We report the 5th and 95th percentiles for AUC as the confidence interval (CI).

D.2 Metrics details

For system fairness evaluation, we use the Disparate Impact Ratio (DIR). Equation 1 presents the DIR calculation, where Pr denotes the probability, $\hat{Y}$ is the predicted value, and D represents a group of users. In our setup, registered users are considered the privileged group, while anonymous users and new editors are treated as the unprivileged group.

$$\frac{\Pr(\hat{Y} = 1 \mid D = \text{unprivileged})}{\Pr(\hat{Y} = 1 \mid D = \text{privileged})} \tag{1}$$

E Experiments

E.1 General system performance

As we showed previously in Table 1, our proposed *Graph2Text* system significantly outperforms all other models across all metrics. This is further confirmed by a precision/recall plot (see Figure 7), which shows that our model performs better at any

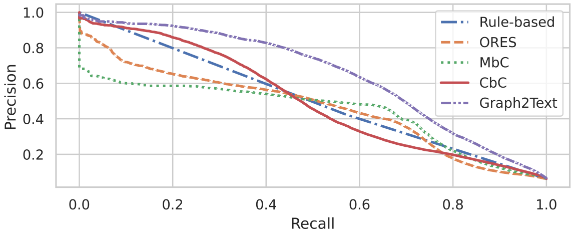


Figure 7: The precision/recall curves for models.

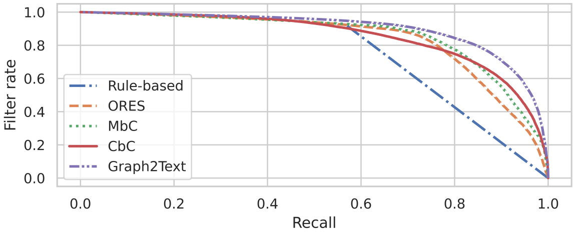


Figure 8: The filter rate/recall curves for models.

threshold. We also support our analysis with a filter rate/recall plot, which highlights the dominance of the presented *Graph2Text* system, especially when a high recall is needed (see Figure 8).

E.2 Use case analysis

Additionally, we analyze how the models perform in different scenarios to understand their strengths and weaknesses and to define steps for future development and improvement.

First of all, we analyze the performance for anonymous users group. Many newcomers begin their editing as anonymous users. Retaining these new users is a priority, as they often transition into active registered editors. Failure to do so could result in a long-term decline in the number of active editors, which could significantly impact the Wikimedia environment in the future. Therefore, incorporating a bias analysis into our model evaluation is an essential step before deploying similar models in real-world contexts.

We present our findings in Figure 9. Specifically, we evaluate the model separately for anonymous and registered users. Our analysis shows that the proposed *Graph2Text* system outperforms the existing ORES model for both groups. Notably, the performance difference is considerably larger for models that include content features when evaluating revisions made by anonymous users.

Wikidata contains pages about various types of entities, but pages about humans receive the most edits, accounting for about 34% of all edits. Furthermore, modifications to human pages are more exposed to vandalism, with a 46% higher revert rate compared to non-human pages. We compared model performance for revisions of human and non-human Wikidata entities and concluded that the proposed system outperforms the current model for both groups. Additionally, all tested systems perform better on revisions of pages about humans.

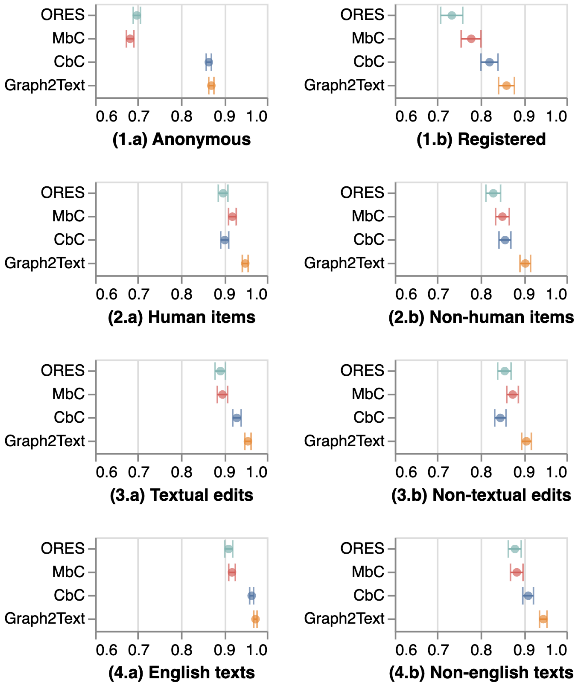


Figure 9: Models performance (AUC) comparison across various Wikidata edit characteristics: (1) Edit source: (a) anonymous, (b) registered users; (2) Entity type: (a) human, (b) non-human; (3) Content type: (a) textual, (b) non-textual; (4) Textual content language: (a) English, (b) non-English.

We have tested model performance on revisions with and without textual changes. As expected, even a basic content model without user features performs significantly better than the current model for handling textual edits. We also compared model performance on English and non-English textual content edits. Our findings indicate that the proposed *Graph2Text* configuration is better for both groups. However, the improvement is significantly greater for English content, suggesting that the largest gains are still within English. At the same time, revisions of non-English content have over double the revert rate, and instances of vandalism persist more than twice as long for this content in Wikidata. This highlights the need to enhance vandalism detection for non-English content in the future.