UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

Language-Agnostic detection of Current Events across Wikipedia

Author:
Alisa ANTYPOVA

Supervisor:
Diego SAEZ-TRUMPER

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2023

Declaration of Authorship

I, Alisa ANTYPOVA, declare that this thesis titled, "Language-Agnostic detection of Current Events across Wikipedia" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 2023/06/12

"The Brave Always Have Happiness."

Ivan Bahrianyi

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

Language-Agnostic detection of Current Events across Wikipedia

by Alisa ANTYPOVA

Abstract

Currently, English Wikipedia alone includes over 6,660,000 articles and it averages 550 new articles per day. To assist the readers in identifying pages that cover recent noteworthy occurrences the Current Events portal was implemented. However, this portal is maintained manually with notable quality differences across languages. The main goal of this work is to establish the task of supervised event detection in Wikipedia and propose a language-agnostic solution to address this problem. This is an important milestone towards improving the quality of the Current Events Portal for the languages with not many active editor communities. In this work, we reviewed existing research on this topic, and by combining and enriching those existing solutions, we proposed a current event detection dataset based on the Current Events Portal updates and Wikipedia pages' features. Also, we developed a language-agnostic event detection model and reported its performance in English, German, and Polish languages, showing that is possible to automatize this task. The outcome of this work can be used to assist Wikipedia editors to keep the Current Event portal updated, saving time and the human effort used on this task.

Acknowledgements

I am grateful to The Armed Forces of Ukraine for the opportunity to live, study and write this thesis.

I want to thank my supervisor Diego Saez-Trumper, who supported me in writing this work and showed me how exciting the academic world can be.
I could not have undertaken this journey without my family and my boyfriend, who were with me in the happiest and hardest times, always believing in me.
I am grateful to Ukrainian Catholic University and Kyiv National University for surrounding me with brilliant people and ideas.
Lastly, I'd like to mention the company I work for - Preply for challenging me and providing opportunities to grow.

Contents

List of Figures

List of Tables

List of Abbreviations

NN	**N**eural **N**etwork
CE	**C**urent **E**vent
CEP	**C**urent **E**vents **P**ortal
SQL	**S**tructured **Q**uery **L**anguage
API	**A**pplication **P**rogramming **I**nterface
STEM	**S**cience, **T**echnology, **E**ngineering, and **M**athematics
XGBoost	e**X**treme **G**radient **B**oosting
SHAP	**SH**apley **A**dditive ex**P**lanations
TF-IDF	**T**erm **F**requency - **I**nverse **D**ocument **F**requency
TP	**T**rue **P**ositive
TN	**T**rue **N**egative
FP	**F**alse **P**ositive
FN	**F**alse **N**egative
PR curve	**P**recision **R**ecall curve

Dedicated to the fast-changing world and the people who explore it.

Chapter 1

Introduction

In this chapter, our goal is to share the importance of event detection in Wikipedia, its state now, open problems, and our research goals to solve them.

1.1 Importance of Events Detection in Wikipedia

Wikimedia Foundation's mission is to empower and engage people around the world to collect and develop educational content under a free license or in the public domain, and to disseminate it effectively and globally[1]. Currently, just the English Wikipedia includes 6,660,000 articles and it averages 552 new articles per day. Moreover, Wikipedia develops at a rate of over 2 edits every second, performed by editors from all over the world.[2] Since the online encyclopedia was established 1,150,235,533 page edits have been made. [3]
Wikipedia is maintained at its core by its readers, as anyone can become an editor and provide new information to the existing pages or create new ones.

To improve readership further, Wikipedia introduced the Current Events portal (CEP) [4]. CEP increases the readership of Wikipedia by attracting readers to the pages related to the topics that are big at the moment - Current Events (CE) according to [Singer et al., 2017]. The example of CE that happened on January 22, 2022, is shown in Figure 1.1. The Current Events Portal [4] is a daily archive of events as recorded by Wikipedia editors. Its main purpose is to highlight significant ongoing events that deserve to be tracked as Wikipedia pages [5].

While it covers recent or ongoing events, it has major differences from the regular news. Wikipedia is not a news service, but a news backgrounder [5]. This means that Wikipedia Current Events is a section dedicated to the facts only, maintaining a core focus both on the area of event and reliability of data and lowering author bias in the materials.

The Current Events section is not updated automatically and is maintained by the community, with different groups of maintainers for the different languages. As a result, less represented languages sections of the CEP get almost no updates and so don't help to achieve Wikipedia's big mission.

[1] https://wikimediafoundation.org/about/mission/

[2] https://en.wikipedia.org/wiki/Wikipedia:Statistics#:~:text=While%20you%20read%20this%2C%20Wikipedia%2Cbe%20analyzed%20in%20many%20ways

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

[4] https://en.wikipedia.org/wiki/Portal:Current_events

[5] https://en.wikipedia.org/wiki/Wikipedia:How_the_Current_events_page_works

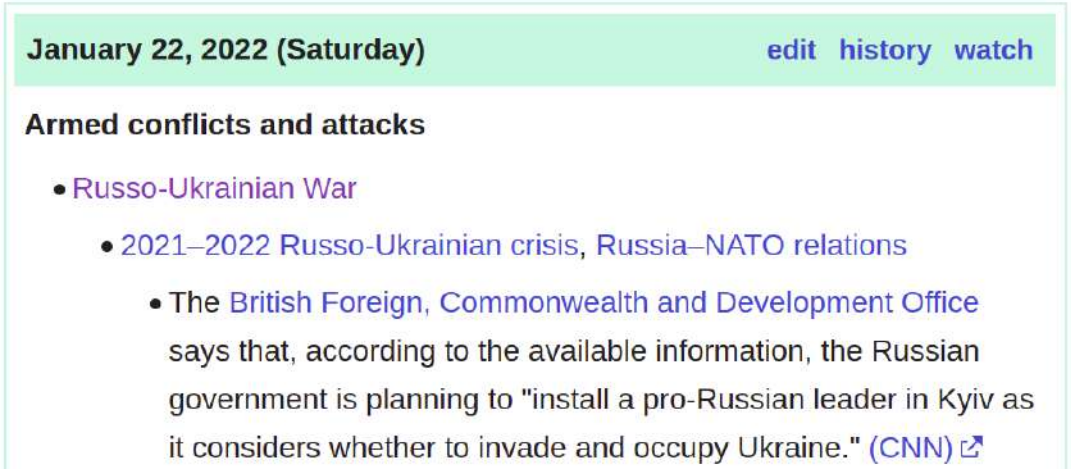

FIGURE 1.1: Example of the CE at the Current Events portal

1.2 The Current Events Portal Specifics

In this work, we want to investigate methods for automatic updates of the CEP based on event detection in Wikipedia. Every day only in English Wikipedia 100.000 pages are edited[6]. Finding events among all edited pages manually seems unrealistic, finding events from news sources also takes time and introduces biases. However, automatic event detection can help to reduce the workload of editors who have to manually search for them now.
Language Agnostic Approach to detecting Current Events can do even more. It can propose Current Events in languages, where no editors are doing this manually. Also, automatic event detection can help to identify important events that may have gone unnoticed by editors. This can help to ensure that Wikipedia's CEP content is comprehensive and inclusive, and can also help to raise awareness about important events that may have been overlooked by traditional news sources.

There are guidelines for users to detect events that can be posted in the Current Events section[5]:

- events should correspond to one of the following sections: Armed conflicts and attacks, Arts and culture, Business and economy, Disasters and accidents, Health and environment, International relations, Law and crime, Politics and elections, Science and technology, Sports;[5]
- the Current Events are meant to cover facts related to international events with extra attention put on the precision of description in terms of information quality and time of the event (additional interest is put into international events rather than local ones).

Wikipedia amplifies differences between Current Events and news by having a separate portal for the news itself (Wikinews), with the latter being targeted to events that don't meet the criteria of events worthy of a separate Wikipedia page.

In this work, we propose an approach for language-agnostic Current Events detection that allows enriching less represented Wiki languages with events minimizing needed manual work. For this, we form a dataset that contains information about Wikipedia pages and their features. In addition to that, CEs from the Current

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

Events portal are used as ground truth in this work, meaning that for each page we have a label for whether it became a part of CEP, or not and this is the label we want to predict based on information in features. The fact that the portal is filled by Wikipedia editors, brings some advantages and disadvantages for using it as true labels.

The main advantages are:

- The CEP has a high precision for defining events, cause it is manually curated by a large community of editors;
- the large community of editors also reduces personal bias from defining events;
- using CE solves the problem of costly, time-consuming, and biased annotation;
- it is suitable for evaluating the proposed solution as well as the baselines for event detection at the page level;
- it's maintained in different languages, which allows checking cross-language performance transferability;
- the portal is quite popular - it is viewed by more than 1 million users each month in 2022[7], which makes its improvement valuable for people.

On the other hand, we can list the following disadvantages:

- quality of the section significantly depends on the size of the active community, for less active languages the Current Events section would be less representative and more biased or not maintained at all;
- usage of the Current Events portal as a ground truth undermines the number of events, as the Current Events portal might not contain all the events. This means CEP has a recall gap in tracking significant occurrences.

Given all of that, we decided to move on with Current Events from the portal as the ground truth for our research.

1.3 Motivation

Event detection is a fast-growing field with a variety of studies and real-life applications. The progress in the field is pushed by advantages in data science and the ability to process large datasets. Wikipedia incorporates the CEP as part of the infrastructure to facilitate the involvement of the people in the Wiki but does not currently benefit from the advances in the field.

That's the gap we want to address to make Wikipedia a better place to discover events and big movements that are happening in the world. With our work, we want to provide a tool to support editors by partially automatizing Current Event detection in Wikipedia and doing it not only for popular languages like English but also for languages with less active Wikipedia communities.

In addition to that, we want to contribute to the science community with the page-level Wikipedia event detection dataset with data from the Current Events portal as ground truth, and data about Wiki pages with community activity such as revisions and views on them.

[7] https://wikimedia.org/api/rest_v1/metrics/pageviews/per-article/en.wikipedia/all-access/all-agents/Portal:Current_events/monthly/2016010100/2016123100

1.4 Open Problems and Research Goals

1.4.1 Open Problems

Existing works that propose methods for Event detection at Wikipedia are evaluated either using social networks plausibility checks [Steiner et al., 2013], or manually (for example, [Keegan et al., 2013] labeled over 3,000 breaking news events; [Wang et al., 2020] proposed MAVEN dataset labeling text mentions from 4,480 Wikipedia documents; [Pouran Ben Veyseh et al., 2022] proposed the largest dataset on the topic "MINION", which contains 31,226 text segments that are manually annotated by following an annotation guideline). Both approaches introduce uses their specific definitions of what is considered to be an event, which can lead to some degree of biases and makes them difficult to compare.

At this point, there is no standard dataset for detecting which pages of Wikipedia contain updates about happening events. There are attempts for establishing it for text sentences by [Wang et al., 2020] and [Pouran Ben Veyseh et al., 2022] works, but they work with sentence-level data, not with a page level. As there is no standard dataset there is also no established benchmark for the event detection on Wikipedia. We cover this problem by introducing baseline algorithms to compare developed models with them.

From the Current Event Portals point of view, the main open problems that we noticed are:

- It is currently updated in a fully manual way with no automation, which requires a lot of editors' effort; [8]
- The portal is maintained for a limited number of languages, due to small editor communities for some of them. While the main page is available in 114 languages, [9] the data about Current events for January 2022 is available only for 21 languages, being the most popular month through the year in terms of the number of languages at the portal [10]. Also, there are drastic changes in the level of maintenance of existing pages, which are discussed in Section 4.2.1.

Lastly, while there are efforts in multilingual event detection, there is a limited amount of work in the language-agnostic event detection domain. As [Mredula et al., 2022] mentioned in a review of the trends in event detection *"very few works consider language independence in their models"*.

1.4.2 Research Goals

The research goal of this project is to establish the task of supervised event detection in Wikipedia and propose a language-agnostic solution to it. Our hypothesis is that based on language-agnostic features we can build a model that will be able to predict events across different languages with good enough precision and recall to be useful in real-life settings.

We defined the following research tasks that are covered in this work to help in achieving this goal:

1. Create a dataset with edited pages and labels if pages got to the Current Events portal.

[8] https://en.wikipedia.org/wiki/Wikipedia:How_the_Current_events_page_works
[9] https://en.wikipedia.org/wiki/Portal:Current_events
[10] https://en.wikipedia.org/wiki/Portal:Current_events/January_2022

2. Develop and score a language-agnostic model for predicting Current Events in English.

3. Check the transferability of the model by checking its performance in other languages where the CEP is maintained.

1.5 Thesis Structure

In Section 2, we provide a review of the related work in the field of event detection. In Section 2.1, we cover the overall landscape of advances in the event detection task in multiple fields, focusing on the ways to represent candidates for events and models used to detect events. In Section 2.2 we go through works that apply event detection specifically for Wikipedia and cover existing language-agnostic event detection approaches. In Section 3.1 we describe the specifics of the dataset collection based on Wikipedia data, In Section 3.2, we share the exploratory data analysis for the collected Wikipedia event detection dataset. In Section 3.3 we present the features we used for the event detection, as well as methods used to preprocess it. After this, in Section 4 we explain the proposed Wikipedia event detection solution and report the results of a series of experiments we did.

In this Section, we cover:

- Performance of several baselines and supervised algorithms for predicting Current Events (Section 4.1);

- Models transferability across different languages (Section 4.2).

Finally, in Section 5 we give the conclusions about the contribution of the work that we did, its limitations, and propose future work directions.

Chapter 2

Related Works

In the previous chapter, we showed that millions of the Current Event portal users can benefit from the improvement delivered by event detection automation. The goal of this work is to establish the task of supervised event detection and propose a language-agnostic solution to it. The goal of this chapter is to understand the current state of the automated event detection task and language-agnostic approaches to solving it.

2.1 Event Detection Overview

Event detection is quite a popular task with a variety of applications ranging from marketing analysis to threat intelligence. At the core of the event detection task is the assumption that real-world (or virtual) events would result in changes on social media, media, or in our case Wikipedia and CEP. The second important assumption is that those changes can be detected automatically.

The most straightforward event detection task is event detection in a news feed. An example of such a task would be work by [Azeemi et al., 2021]. Each separate news article by nature covers some event, and the challenge comes less from the detection of events than from the detection of important events. Also, news portals are centralized and tend to introduce more perception bias into the events. By nature, event detection in the news operates in one language or set of languages, which makes it impractical to apply language-agnostic approaches to the task.

A lot of research work in event detection in social media has been done during the last years. Only [Mredula et al., 2022] reviewed 67 articles in the domain of event detection. Those works differ in how they represent events: text, images, graph representation of links to Wikipedia pages, etc. And also event detection approaches are using different ML algorithms for event detection modeling. We cover event representation and feature selection approaches in Section 2.1.1 and in Section 2.1.2 we discuss models used for event detection.

2.1.1 Event Representation

When working with textual data one of the basic approaches to be used is TF-IDF [Salton and Buckley, 1988]. This approach helps to assess the frequency of the word in a document in comparison to an inverse frequency in some defined document corpora. It is very effective for lightweight text classification tasks.

[Azeemi et al., 2021] in their work "RevDet: Robust and Memory Efficient Event Detection and Tracking in Large News Feeds" goes one step further. They understood that for the event detection type, it is beneficial to use word occurrence data

about locations, names, and general topics together with counters related to the specific keywords extracted directly from the text as features. In this work, every news article is represented as a sparse vector of counts of the key terms in the article text and its metadata.

The advantage of the approach is the ability to directly merge all informative features into a vector, but it comes with a major downside as well: big sparse vectors result in additional computational strain on the models and limit the applicability of such event representation approaches with the clustering and NN models.

This downside however can be overcome with the introduction of the sentence embedding approaches. Sentence embedding builds on top of language modeling techniques such as word2vec proposed by [Mikolov et al., 2013], FastText by [Bojanowski et al., 2017], or more recently BERT by [Devlin et al., 2019]. Language models represent each word in the form of a fixed-size vector with the idea that close in meaning and usage words would produce similar vectors (with a minimal distance).

The simplest way to go from the word embedding level to a sentence embedding level is to add vectors of the words in a sentence and average the result by the count of words. This technique, while being pretty basic, shows great results in a short sentence classification task, which is useful for event detection on Twitter. Examples of such approaches to sentence embeddings can be found in works by [Dabiri and Heaslip, 2019]. Recent advances in language modeling such as BERT allow the generation of cross-language word embeddings presented in a work by [Feng et al., 2022]. There more precise sentence embeddings with an attention mechanism are used instead of simple averaging of the words in a sentence.

There is also an interesting middle-ground between word frequency-based features and sentence embeddings presented in a work by [Leskovec, Backstrom, and Kleinberg, 2009]. This work came before the wide adoption of the NN-based word and sentence embedding but utilizes similar ideas nevertheless. To trace the popularity of topics in media and blogs authors take important phrases about events and citations by people, convert them into a similar form with the same meaning, and then use this set of original phrases and variations to perform a search in media and blogs and track popularity of the event through time.

The core requirement for a language-agnostic approach is for features and models to not be hardwired to one particular language. One can argue, that multilingual language models can be considered language agnostic, but for our purposes of this work, it's not true. If the language model is multilingual it means it was trained in multiple languages and can work on them, it still does not mean easy transferability to another language, cause it may require language-specific knowledge. While language-agnostic models are not specific to any language, so they can work with any language without the need for language-specific adaptation. One example of a language-agnostic approach has been presented by [Buntain, 2017]. While at its core presented approach works with text, model words with words as with independent tokens. Researchers work with time windows and monitors for bursts of tokens that would be different from the usual statistics higher than the threshold. This method does not need to know what tokens mean, just how many times are they usually used in a feed. The approach was tested against multiple sports event broadcasts. Sports events are perfect for such an approach because no matter the language when the team scores, the words "score" and "goal" would spike immediately in the feed and so would be easily detectable. The downside of such an approach in application to Wikipedia event detection is that it relies on the dynamic stream of data about the

specific topic, like a Twitter feed for a specific hashtag, or live broadcast, which is not the case for the Wikipedia revisions, that covers a vast variety of topics.

2.1.2 Event Detection Modeling Approaches

As was shown in 2.1.1 the event detection task can deal with a lot of different modalities: text, graph relations, frequencies of term occurrences, or reposts. It results in a wide variety of multistep approaches being used for the task.

Works such as [Azeemi et al., 2021], [Steiner et al., 2013], and [Leskovec, Backstrom, and Kleinberg, 2009] use clustering for the events tracking in a media. The idea is that different sources that cover the same event would result in similar representations of the event in a selected feature space. This by itself facilitates further analysis and filtering of the event occurrences within the sources. Notable here is that in work by [Steiner et al., 2013] clustering is performed on the Wikipedia pages by a simple feature of pages being transcripts for different languages. This was eventually used to track events between different languages.

As an extension of the clustering approach for the event detection task, some scholars use topic modeling for event occurrence tracking. Topic modeling is an unsupervised machine-learning task that deals with textual information. Separate words or in some cases phrases are called topics and documents are compared and clustered based on the sets of topics they possess. To apply topic modeling to the event detection task [Oghaz et al., 2020] introduced a new method for dynamic topic modeling called "Narratives Over Categorical Time" or NOC. Authors solve the task of narrative detection, which is similar to event detection but focuses on changes in public perception of events or big themes. In this task, NOC is a tool that allows tracking changes in a narrative based on the set of used topics and detecting when a set of topics changed enough to say that a narrative has changed.

One of the major challenges that come with the unsupervised approaches is related to an absence of ground truth. Each such approach introduces a separate metric of similarity between pages or events, based on the used features set which makes it harder to compare solutions one to another.

Supervised approaches resolve this issue by the introduction of ground truth labels for the events. [Johnson, Gerlach, and Sáez-Trumper, 2021] use data about Wikipedia Talk pages as a label for the training of the language agnostic topic prediction model for the topic tagging of the Wikipedia pages.

Another such example is presented in a work by [Yagcioglu et al., 2019] where cybersecurity events are detected on Twitter. Authors manually annotated 2000 tweets in the cybersecurity domain and used this data for training and evaluation of the LSTM-based neural network classifier (the foundation of LSTM architecture was proposed by [Hochreiter and Schmidhuber, 1997]) that predicts if text represents an event or not. This work showcases one of the interesting solutions that are used for Twitter as a data source. While a separate tweet is a short message, concatenating a set of tweets together and treating them as the text allows higher quality text classification and event detection because a concatenated text provides the model with more necessary context for robust predictions.

2.2 Event Detection in Wikipedia

Event detection in Wikipedia has some coverage and approaches that in a way unique to Wikipedia in comparison to event detection on social media.

[Gildersleve, Lambiotte, and Yasseri, 2022] in their work "Between News and History: Identifying Networked Topics of Collective Attention on Wikipedia", provide an excellent bird's eye view of the topic. While the paper itself works only with English Wikipedia and is not fully aligned with the subject of our work, it tackles interesting questions regarding the relations between Wikipedia and social media nowadays. They state that social media is not a classical media and trends there are driven by the interests of the people. On the other hand, Wikipedia is not a classical encyclopedia: it's dynamic, people-driven, and operates on a significantly larger scale of topics than a classical encyclopedia used to cover.

The thing in common between social media and Wikipedia is being driven by the people, which besides differences between the Wikipedia editor community and the social media community results in interesting effects. First, despite the intent of the Current Event Portal to cover international events, authors have detected geographical bias in the portal when for English Wikipedia Current Event Portal would have asymmetrically higher coverage of events that happened in the UK and USA. The second effect is related to the personality focus. When an event related to a famous person or a politician happens, it is common to see spikes of edits and interest related to this person and closest relatives and inner circle of this person on Wikipedia. The third observation is related to the "Media storm" effect described by [Boydstun, Hardy, and Walgrave, 2014]. In this case, the media maintains focus on some subject for a long time (for example coverage of tension on the North-South Korean border). This effect takes place in social media and is also present in Wikipedia and CEP.

In the work "MJ no more: Using Concurrent Wikipedia Edit Spikes with Social Network Plausibility Checks for Breaking News Detection" [Steiner et al., 2013] authors argue that event detection on Wikipedia is a way to reduce false positive rate compared to the event detection on the social media. Authors use Wiki page edits frequency as a feature for the event detection model. And having in mind, that edits must be validated by a small community of editors and each edit requires the source to be mentioned authors infer that the level of false positives on Wikipedia would be lower than on social media event detection where no validation for the sources is required.

In the work "WikipEvent: Leveraging Wikipedia Edit History for Event Detection" [Tran et al., 2014] detect events from Wikipedia based on users' edit history 2014. They perform the extraction of a connected component (Local Temporal Constraint algorithm) based on the graph, with the nodes corresponding to Wikipedia entities and edges corresponding to their relationships.

In the work "Between News and History: Identifying Networked Topics of Collective Attention on Wikipedia" [Gildersleve, Lambiotte, and Yasseri, 2022] established representations of topics beyond individual articles, on Wikipedia. Rather than detecting Current Events out of all pages (that is the focus of this thesis), authors look for topics that connect different events that were part of the Current Events portal.

They collect clickstream networks and page view time series data to train a model. Then they use the content of the page (structure) and attention to the page (views dynamics) to find pages that are both well connected and exhibit similar page view time series.

Common topics for events are detected by calculating the correlation matrix between events representation and using the Leiden algorithm to divide the received graph into subgraphs, which correspond to topics.

Chapter 3

Data Exploration

In our work, we aim to create a language-agnostic approach for the prediction of Current Events based on the analysis of Wikipedia pages. In the previous chapter, we discussed that different works on the neighboring topics have used datasets with page views history, page structure [Gildersleve, Lambiotte, and Yasseri, 2022], edits history [Tran et al., 2014], and links to other pages [Johnson, Gerlach, and Sáez-Trumper, 2021]. However, no established benchmark is available for validation of the language-agnostic event detection approaches for Wikipedia pages. To facilitate experimentation, and show the performance of our approach in a language-agnostic setup we collected our benchmark dataset from Wikipedia. Our dataset includes data about revisions, views, and topics. To formulate our research task as a supervised task and compare different event detection algorithmic approaches we also collected CE from the Current Events portal for 1 year, to use it as a ground truth. We also performed data collection for three different languages to provide an opportunity for the evaluation of the cross-language transferability of models. In total, we used 5 different data sources to collect the dataset (Current Events Portal[1], Quarry[2], MediaWiki API[3], Wikimedia Pageview API client[4], Wikipedia Content Tagging project[5]). They are discussed in more detail in Section 3.1. In this chapter, our goal is to share the data collection process we used, present exploratory data analysis of our Wikipedia CE detection dataset, and explain feature definitions.

3.1 Data Collection

3.1.1 Current Events Portal Data

For collecting events from the Current Events portal we used a data scraper proposed by [Gildersleve, Lambiotte, and Yasseri, 2022]. In the paper, the authors used the data from the portal to identify networked topics of collective attention. In other words, to find groups of events connected with each other by the same topic and see how attention to the topics changes through time.

We modified it to be able to scrap an updated Wikipedia layout for the English portal and wrote our scrapper to be able to get the data from German and Polish versions of the portal as well. Those languages were chosen as ones, that have a constant coverage of events, describing several events each day during 2022. We collected all the event entities from English, German, and Polish portals that were

[1] https://en.wikipedia.org/wiki/Portal:Current_events
[2] https://quarry.wmcloud.org
[3] https://www.mediawiki.org/wiki/API:REST_API/Reference
[4] https://github.com/Commonists/pageview-api
[5] https://wiki-topic.toolforge.org/

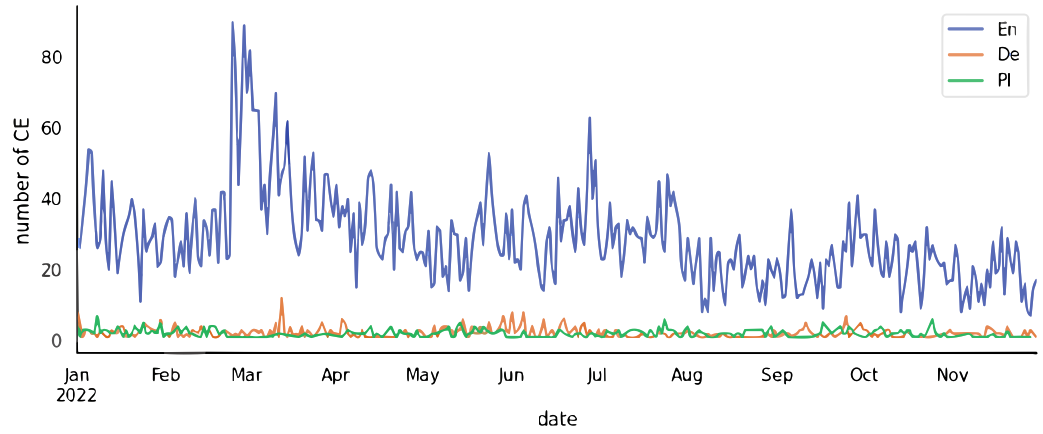

FIGURE 3.1: Number of Current Events per day during 2022 for different languages

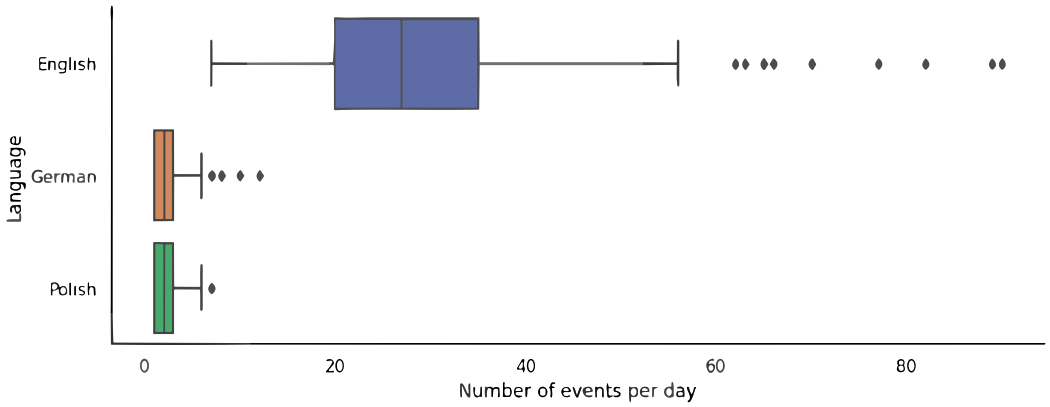

FIGURE 3.2: The box plot for the number of events per day distribution for different languages

mentioned there in 2022. We received 9,674 events for the English portal, 688 events for the German portal, and 512 events for the Polish portal.

The distribution of Current Events through the year can be seen in Figure 3.1. For English Wiki most of the days have from 20 to 40 events, with a spike of up to 90 events that occurred on 24 February 2022 because of the "Russo-Ukrainian War" [6].

The quartiles of the number of events per day for different languages can be found in Figure 3.2. The median number of events per day is shown as a line in the boxes, for English is 27, and both for German and Polish is 2.The lower quartile for English is 20, and the upper is 35. For both German and Polish, the lower quartile is 1 and the upper is 3. We can see that difference in the maintenance of the portals is huge, with English being the most maintained one and German and Polish being quite similar in terms of the number of events captured.

Each Current Event entity in our dataset has the date when it appeared in the CEP, the link to the Wiki page the event is referencing, as well as the text description of the event. For all events captured the value of the 'is_event' column in the dataset

[6] https://en.wikipedia.org/wiki/Russo-Ukrainian_War

is 1. The code for scraping and the dataset with all events for 2022 is available in the Github repo [7].

3.1.2 Pages Data

To be able to differentiate regular pages from pages, where CE occurred we need to collect data for both types of pages. We already discussed how we collected the pages with CEs and dates when they occurred in the CEP.

Next, we collected all pages that had at least one revision on June 1, 2022, using a query to Quarry [8] - a service that allows running SQL queries against the Wikipedia Database Replicas[9]. Ideally, we would want to collect features from the whole year for predictions, but due to the limitations of the data collection, we selected a specific day as a method to collect a subset of the 'no event' class pages for the dataset. For CE we use the whole year to let the model learn more differences between the two classes, as there are much fewer CEs that pages edited on a specific date (more details about numbers of pages and CEs are discussed in 4.2.1).

Then we marked each page that was CE at a given date with 1 in the column 'is_event' and we marked each page that is not in a CE with 0 in the 'is_event' column.

Notice that this way we shrink the space of the search from all Wikipedia pages, we decided to consider only pages that had at least one revision the day before the page could possibly appear in the CEP.

This limits the field of search drastically:

- for English: from 6,503,749 [10] existing pages as of June 1, 2022, to 98,629 [10] pages edited during June 1, 2022;
- for German: from 2,710,103 existing pages as of June 1, 2022, to 16,897 pages edited during June 1, 2022;
- for Polish: from 1,526,228 existing pages as of June 1, 2022, to 4,788 pages edited during June 1, 2022.

At the same time, 18.4% of English, 33.9% of German of 61.9% of Polish CE pages were not edited before the day they appeared in the CEP.Those events can not be captured by this work and it is a limitation of the chosen approach.

3.1.3 Revisions Data

The data about revisions are stored in the "revision" table at Quarry [8]. The same data can be accessed via MediaWiki REST API "Get page history counts"[11]. We collected a number of revisions per different time periods for both pages and events. The time periods we queried are the day before, the day before the day before, the week before, the month, and the year before.

All time periods are taken in relation to the date a page appeared in the Current Event portal (for events); or in relation to the date, that is one day after the day when a page was edited (for pages, that are not CE).

[7] https://github.com/AlisaEdu/current_events_detection
[8] https://quarry.wmcloud.org
[9] https://wikitech.wikimedia.org/wiki/Wiki_Replicas
[10] https://stats.wikimedia.org
[11] https://www.mediawiki.org/wiki/API:REST_API/Reference

3.1.4 Views Data

To add to the collected data information about the number of views we used the Wikimedia Pageview API client [12]. With its help, we collected information about views for specific time periods that were interesting for us: previous day views, previous week views, previous month views, and previous year's views.

For views, we also do not collect the statistics for the day when we want to know if it is a Current Event or not. We do it to protect our models from data leakage that could occur due to a change in the number of revisions/views because of the occurrence of a page at the CEP.

3.1.5 Topics Data

We used the Language-Agnostic Wikipedia Content Tagging project [13] to add features related to the page topic. The tool that we used is called Language-agnostic Topic Classification and it predicts a page topic based on page links to other Wikipedia articles and returns information about a topic, subtopic, and a predicted score of the topic. The result of prediction is accessible via API when the language and title of the page are specified. Our hypothesis for using information about the topic is that some topics are more likely to contain CEs than others.

In the current version, we use the page topic as of the date we collected data, not as it was on the day we consider it in the dataset. It is a limitation of this work.

3.2 Exploratory Data Analysis

In this work, we deal with datasets for English, German, and Polish Wiki pages that have features related to the number of revisions and views they had in different time periods, and their topic.

Since we are interested in developing the language-agnostic model, let's check if features are similarly distributed across languages. For exploratory data analysis, we use both pages with and without CE together

For revision distribution shown in Figure 3.3, the pattern is similar - most of the pages have less than 10 revisions on the previous day and less than 50 revisions during the previous year with no major differences between languages.

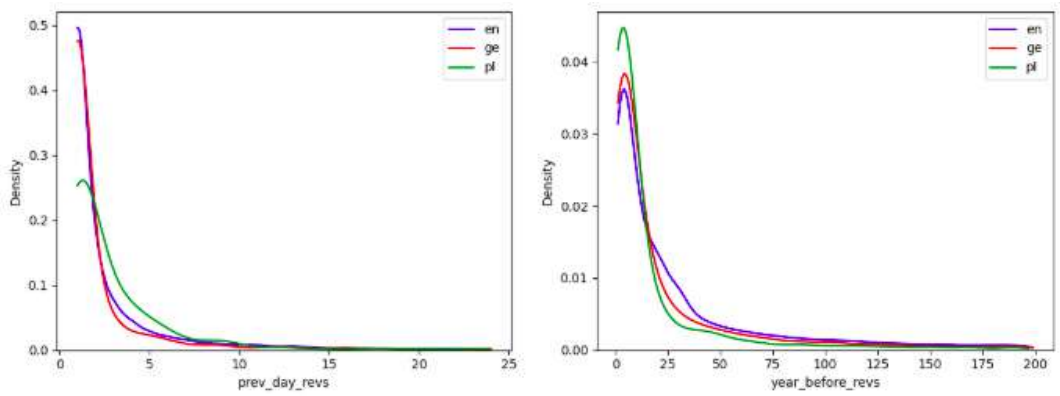

FIGURE 3.3: The distribution of 'prev_day_revs' and 'year_before_revs' across languages for all pages

[12] https://github.com/Commonists/pageview-api
[13] https://wiki-topic.toolforge.org/

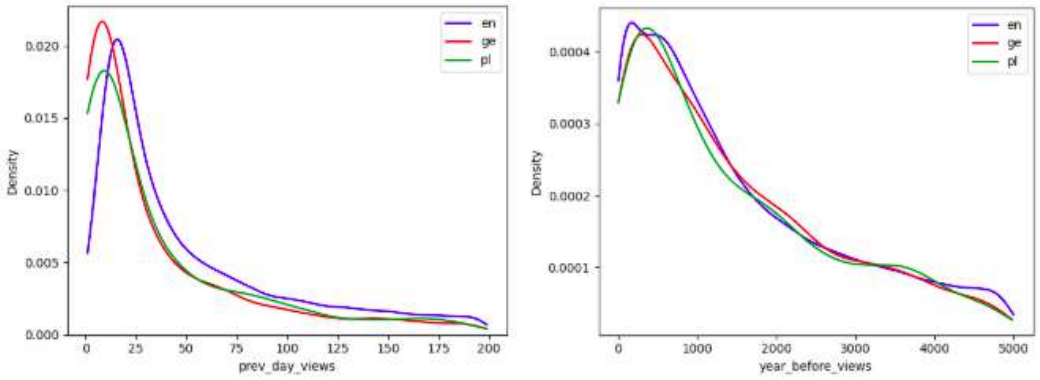

FIGURE 3.4: The distribution of 'prev_day_views' and 'year_before_views' across languages for all pages

Looking at the "prev_day_views" and "year_before_views" distribution in Figure 3.4 we can see that the general trend for all languages is similar as well - a lot of pages have a small number of views followed by a long tail of pages that have more views.

In Figure 3.5 we can see that the most popular topic language-wide is"Culture", followed by "Geography", "History_and_Society" and "STEM". The largest difference between languages is that German has much more pages with a not defined topic (37%, compared to 23% in English and 14% in Polish), and Polish pages have a much bigger percentage of the "STEM" topic (15%, compared to 8% in English and 6% in German).

EDA shows that for the three investigated languages overall trends within the selected features are similar. All three languages have similar distribution with regard to revision. Yet, the Polish language has a fat tail in regard to previous day revisions which may indicate a more uniform editors community that contributes across all the Wikipedia. Distributions related to views are also pretty similar, we can see that English viewership is a bit higher and the distribution has a fatter tail for the previous day and last year's views, compared to the German and Polish. And topic distribution has some differences across languages, mostly related to the percentage of the "Not defined" topic.

3.3 Feature Description

We did several steps of data processing to convert collected features to features suitable for data modeling:

1. Introduced individual columns with flags for "Culture", "Geography", "History_and_Society", "STEM", and "Other" topics (if a topic of a page corresponds to the name of the column, the value in this column for this page is 1, otherwise 0).

2. Transforming topic and subtopic columns from the names to the share of the specific topic/subtopic name in the training data.

3. Scaling all features using MinMaxScaler. We did it for each language separately, to preserve feature distributions, but scale numbers to the same range. The motivation is that the range of a number of views and revisions can differ

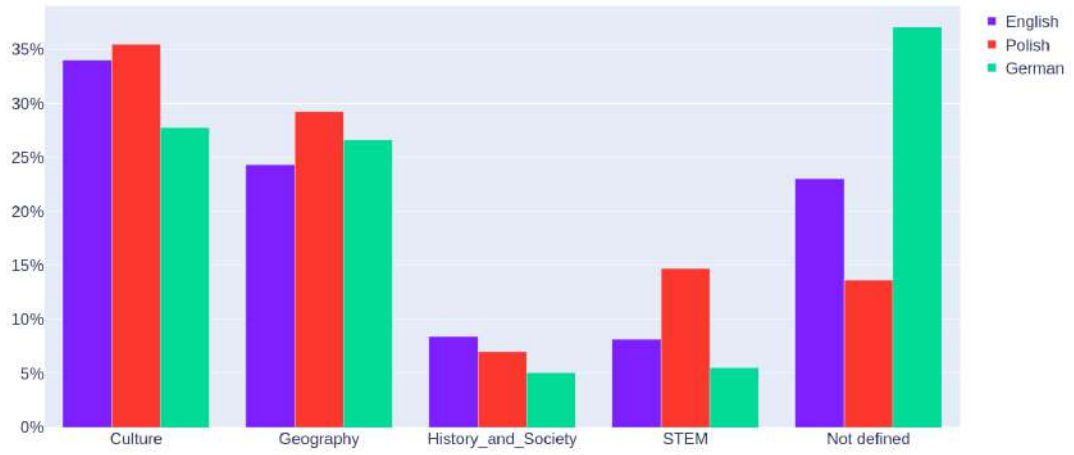

FIGURE 3.5: The distribution of topics across languages for all pages

across languages. Converting absolute numbers to range from 0 to 1 for each language solves this problem. [14].

A detailed description of all features is in Table 3.1.

Table 3.1: Feature description

Feature name	Description	Value	Source
prev_day_views	Scaled count of views of the page in the previous day	Float number from 0 to 1	Views data
week_before_views	Scaled count of views of the page in the previous week	Float number from 0 to 1	Views data
month_before_views	Scaled count of views of the page in the previous month	Float number from 0 to 1	Views data
year_before_views	Scaled count of views of the page in the previous year	Float number from 0 to 1	Views data
part_views_week	'prev_day_views' divided by 'week_before_views'	Float number from 0 to 1	Views data
part_views_month	'prev_day_views' divided by 'month_before_views'	Float number from 0 to 1	Views data
prev_day_revs	Scaled count of revisions of the page in the previous day	Float number from 0 to 1	Revisions data
pre_prev_day_revs	Scaled count of revisions of the page in the day before the previous day	Float number from 0 to 1	Revisions data
week_before_revs	Scaled count of revisions of the page in the previous week	Float number from 0 to 1	Revisions data

Continued on next page

[14] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

Table 3.1: Feature description (Continued)

month_before_revs	Scaled count of revisions of the page in the previous month	Float number from 0 to 1	Revisions data
year_before_revs	Scaled count of revisions of the page in the previous year	Float number from 0 to 1	Revisions data
part_revs_week	'prev_day_revs' divided by 'week_before_revs'	Float number from 0 to 1	Revisions data
part_revs_month	'prev_day_revs' divided by 'month_before_revs'	Float number from 0 to 1	Revisions data
topic_score	Topic score based on Language-Agnostic Wikipedia Content Tagging	Float number from 0 to 1	Topics data
Topic	Share of the page's topic in the train data	Float number from 0 to 1	Topics data
subtopic	Share of the page's subtopic in the train data	Float number from 0 to 1	Topics data
Culture	1, if the topic is "Culture", else 0	0 or 1	Topics data
Geography	1, if the topic is "Geography", else 0	0 or 1	Topics data
History_and_Society	1, if the topic is "History_and_Society", else 0	0 or 1	Topics data
STEM	1, if the topic is "STEM", else 0	0 or 1	Topics data
Not defined	1, if the topic is "Other", else 0	0 or 1	Topics data

All the views and revisions features have information only before the day that page appeared, or could have appeared in the CEP. We define them this way to prevent data leakage that can happen because the page already appeared in the CEP which can give additional attention to the page.

Chapter 4

Experiments

In the previous section, we discussed how we collected and defined the dataset for the event detection task. In this section, our goal is to describe our experiments on event detection. First, we start showing our experiments on English Wikipedia. Next, we describe the work on cross-language transferability.

4.1 Classification Models for English Wikipedia

This section contains the results of the experiments of event detection on English Wikipedia. All experiments are performed on the dataset collected by us and described in Section 3. In all of the experiments, we utilize CEP data as ground truth to classify pages as events or not.

4.1.1 Experiment Setup

In this set of experiments, we developed models for predicting CE for English Wikipedia. However, models described in this section are language-agnostic by the nature of their features, none of which are directly related to the language. We use English for the basis of experiments because CEP that we use for ground truth is highly maintained by the community in this language. It has on average 26 events per day, a higher number compared to 1.9 in the German portal and 1.4 in the Polish one (see Section 3.2).

Here we formulate an event detection task as a binary classification problem with unbalanced classes (10.2 pages without CE corresponds to 1 page with CE). Objects to classify are Wikipedia pages on a particular day represented by features described in 3.2 (edits, views, and page topic). One row represents the state of a Wiki page on a specific day.

The target that we predict is the answer to the question: will this page be presented on the CEP on the next day? This way,

- Class 1 - Wiki page on a particular day that was added to the CEP the day after (shortly, an event);
- Class 0 - Wiki page on a particular day that was not added to the CEP the day after (shortly, a page not corresponding to an event).

Taking into account that data is tabular by nature, we compared three models that are used for tabular data classification:

- Logistic regression [Cox, 1958][1]

[1] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

- Random forest [Ho, 1995][2]
- XgBoost [Chen and Guestrin, 2016][3]

We selected Logistic regression as a baseline model. Its computationally lightweight, which is beneficial for our application use case. It is also important that logistic regression is a model that is based on log odds of a linear combination of features, which means that if features are linearly separable we would be able to get good metrics, and if not performance of the logistic regression model would be a good indicator of the non-linear nature of the data. We selected Random Forest and XgBoost as algorithms with non-linear nature of the separate plane as both of them use a decision tree as a base classifier. The difference, however, comes from the approach to combining tree models. Random Forest utilizes bagging - multiple simpler trees are built on the data, and results are averaged to get the prediction of the class. In the XGBoost ensemble mechanism is called gradient boosting. In this case, a series of classifiers are trained on the initial data, with each new classifier being trained to solve the data the previous classifier failed to properly classify. It allows for more complex decision boundaries that might be able to incorporate corner cases better than Random Forest.

To have a benchmark to compare models with, we evaluated not only the three models that we mentioned earlier but also two baselines:

1. Sorting based on the number of revisions on the previous day
2. Sorting based on the number of views on the previous day.

We call those methods sorting, cause they are based on giving the order to the samples according to specified features. For baseline 1 we take the number of revisions on the previous day and divide it by the largest number of revisions on the previous day in the dataset. This way, a page with a maximum number of revisions received the score 1 - meaning that based on this simple model we predict it to be the Current Event. While, a page with 0 revisions received the score 0 - meaning this model is sure it is not the CE. For all values in between the logic is the same with the only rule - the more revisions - the more the baseline model believes this page will be captured in the Current Events portal. In other words, the number we receive is the probability predicted by the baseline model of the page to belong to the CEP.

The same logic we use for baseline 2, changing the number of edits to the number of views. All the models will be trained on a train set (67% of the dataset, chosen randomly[4]) and tested on a test set (33% of the dataset that is different from the train set).

4.1.2 Experiment Results

For result evaluation, let's start with a Recall-Precision chart that works well for unbalanced classification evaluation. In Figure 4.1 we can see the performance of 3 models discussed before as well as 2 baselines. Each point of the Precision-Recall graph represents a possible precision and recall metric that we can achieve by each

[2] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[3] https://xgboost.readthedocs.io/en/stable/get_started.html

[4] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

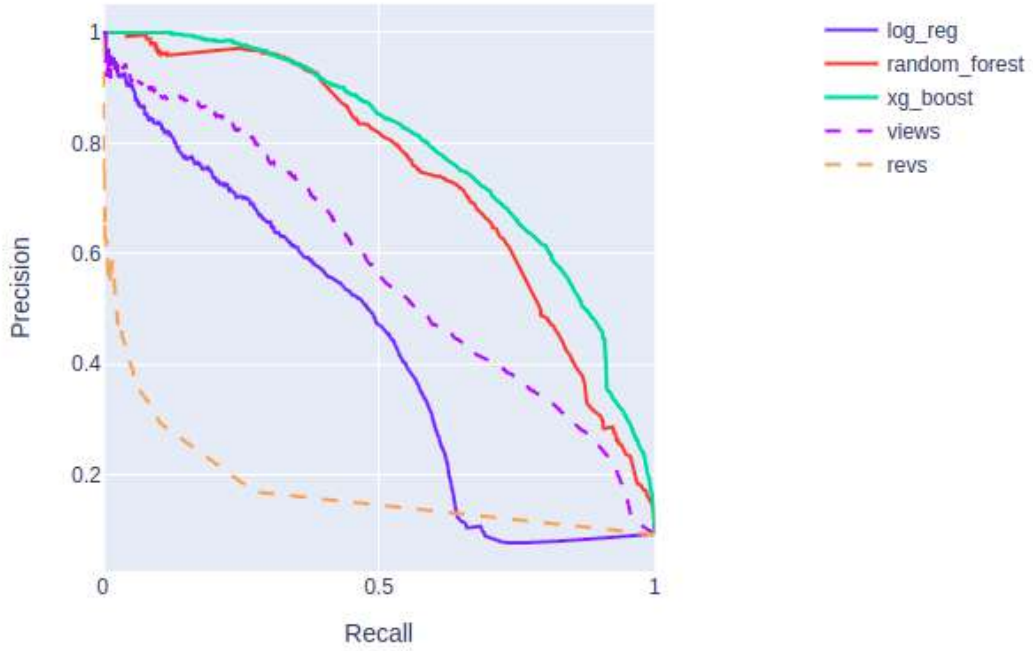

FIGURE 4.1: Precision-Recall plot for the 3 models and 2 baselines (for English dataset)

model while setting a threshold at a specific place. Note that when the threshold for a model is 1 we predict as events samples that have a prediction of the model more than 1. Given that model predictions vary from 0 to 1, we will predict all samples as pages without events. In this case, we receive recall 0 (we do not capture any real events) and precision 1, cause we don't have any false positives (samples that are predicted to be an event, but they are not in reality). On the other side, when the threshold is 0, we predict all samples to be events. In this case, we receive recall 1 (we capture all events) and precision that equals the portion of events to all samples in test data (in our case 0.09).

In Figure 4.1 metrics for two baselines are presented in the dashed lines, and metrics for three models are presented as solid lines. The perfect model would look like a line that starts at recall 0 and precision 1 and then goes to recall 1 and precision 1. And generally, the bigger recall and precision we can get - the better. In Figure 4.1 we can see that sorting by views outperforms sorting by revisions at any given point. Meaning that the views have larger prediction power and most probably will give more information for the models. As for the models, the XgBoost approach outperforms all models and baselines at any given point. So later, we will iterate and check transferability based on this approach. Random Forest is the second-best model and Logistic regression fails to capture feature relationships and performs worse, than a baseline sorting based on views.

Now, let's find an optimal threshold for each model and compare them based on this. For evaluation, we checked precision and recall, and we optimized those values based on the F1 score. It is the harmonic mean of precision and recall, which measures the trade-off between the two of them.[5] For the optimal value search, we

[5] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

implemented a two-step approach based on the binclass-tools library[6].

The algorithm we used:

1. Search for the optimal (in terms of F1-score) threshold at the range from 0 to 1 with step 0.01.
2. Search the best threshold at range +-0.01 from the best value from the first step with step 0.0001.

The best result we received in terms of the F1-score is given by the XgBoost model with a 0.13 threshold. Using this model, we can predict Current Events with 0.73 precision, and 0.72 recall. It results in a 0.73 F1-score. At this point, we predict 3170 pages to be events. Out of them, 2302 are real events. And we capture 2302 out of 3180 events that we have in a test set.

Results for all models in the point of optimal F1-score point are presented in table 4.1. We report precision, recall, F1-score, and threshold for which they are calculated as well as AUC-score [7], which does not depends on the threshold and measures the area under ROC (receiver operating characteristic) curve (but it is sensitive to class imbalance, so should be used with care in our case).

Table 4.1: Results of models in the optimal F1-score point (for English dataset)

Model	Precision	Recall	F1-score	AUC-score	Threshold
Revisions sorting baseline	0.17	0.28	0.21	0.57	0.0001
Views sorting baseline	0.52	0.55	0.53	0.89	0.0004
Logistic regression	0.47	0.50	0.49	0.65	0.5005
Random forest	0.63	0.67	0.65	0.94	0.2900
XgBoost	0.73	0.72	0.73	0.96	0.13

As we view this model as a filter for the further proposed events, the community review threshold for the classifier can be moved higher or lower to control the count of event candidates for the review. But getting a threshold for the higher recall would typically result in a higher false positive rate (and lower precision) and vice versa.

In our case, considering only the top 100 pages with the highest predicted results to be CE, we will be right with predictions in each case, for the top 1000 pages predicted to be events, 968 of them are real events. In Figure 4.2 there are more examples of possible confusion matrices we can receive by setting different thresholds.

4.1.3 Results Interpretation

Even though we used different models, some of which are considered to be not well interpretable, we are still interested in understanding how our models work. To discover that we used SHAP (SHapley Additive exPlanations) approach proposed

[6] https://github.com/lucazav/binclass-tools

[7] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html

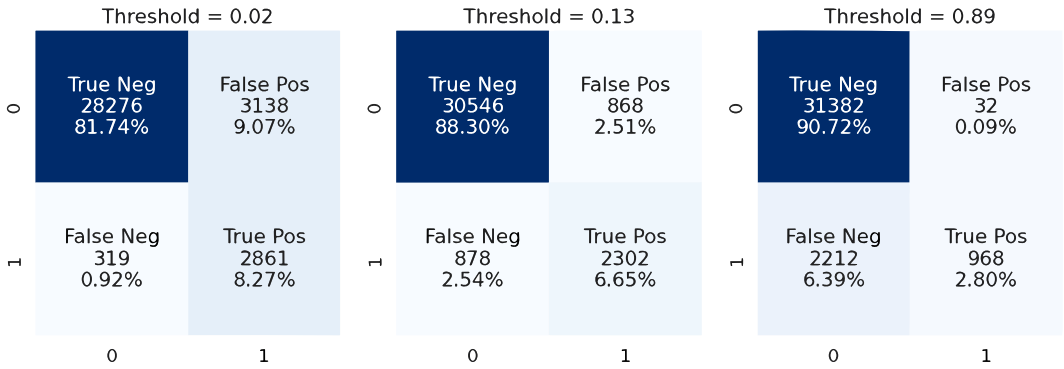

FIGURE 4.2: Confusion matrix for different threshold values (for English dataset, XgBoost model)

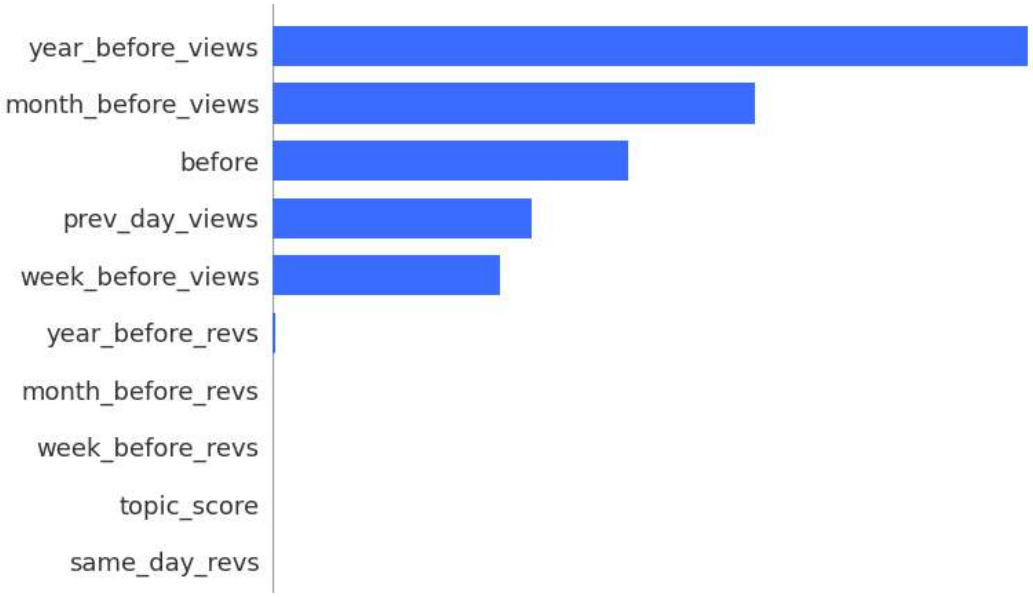

FIGURE 4.3: Feature importance for 10 the most important features of the Logistic regression model (for English dataset)

by [Lundberg et al., 2020]. It can explain the output of any machine learning model by assigning importance values to each feature that contributed to the prediction.

SHAP values explain how much a model prediction would change given the change in a feature. There are several advantages of SHAP values in relation to regular feature importance:

1. it allows both global and local interpretability (feature importance across samples and for an individual sample)[8];
2. SHAP values are additive, so the final prediction for a sample can be explained as the average prediction plus the SHAP values of all features[8].

Let's start by checking the SHAP-based feature importance for all models.

In Figures 4.3, 4.4, 4.5 we can see that there is one feature that is in the top-5 of every model - the count of page views on the previous day. Also, for Random Forest

[8] https://medium.com/dataman-in-ai/explain-your-model-with-the-shap-values-bc36aac4de3d

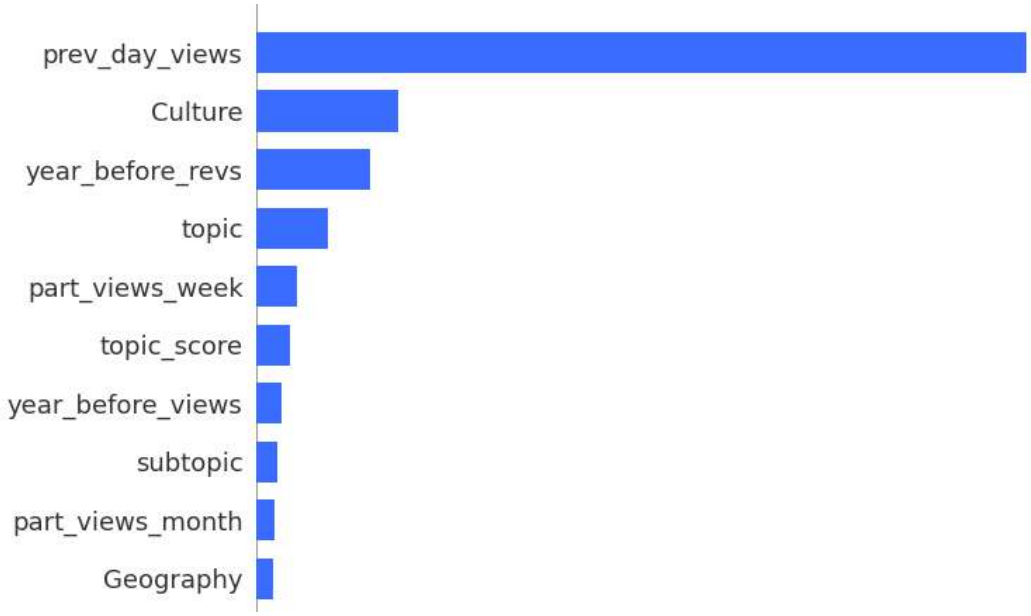

FIGURE 4.4: Feature importance for 10 the most important features of the Random forest model (for English dataset)

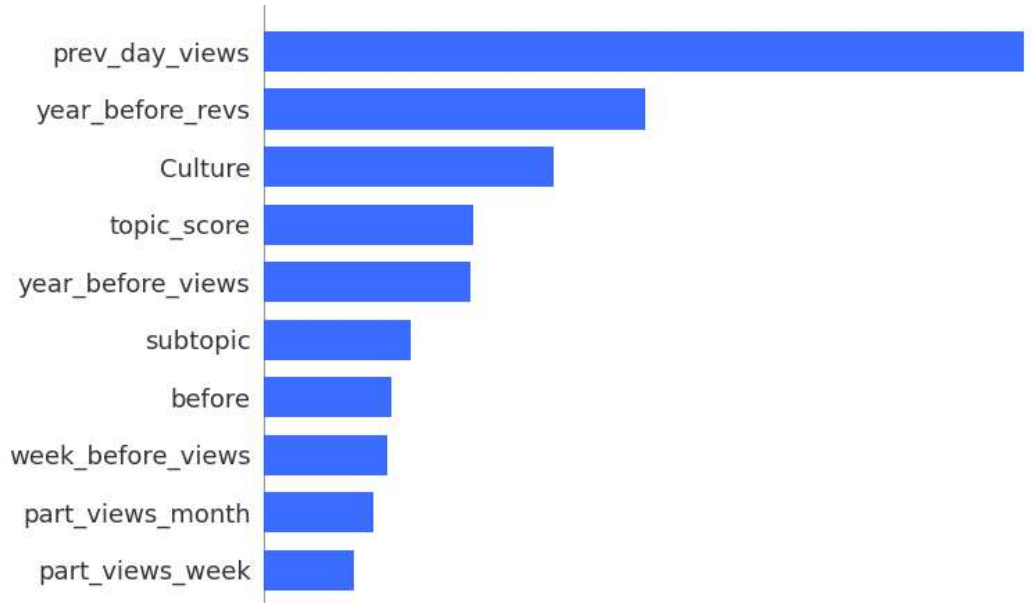

FIGURE 4.5: Feature importance for 10 the most important features of the XgBoost model (for English dataset)

and XgBoost, top-3 features are the same - 'prev_day_views', 'year_before_revs', and 'Culture'.

We can note that for all models views-based features are at the very top. That is aligned with the results from baseline sorting, where we saw that the sorting based on views works much better than the sorting based on revisions.

To explore concrete samples, in Table 4.2 we share examples of pages on which our final model (XgBoost) works well, and on which it fails.

For model prediction and real class, we reference current event tern in a short form (CE). CE in a "Model prediction column" means that model predicted that the page is going to be CE, for real class CE means that the page is CE in reality.

TABLE 4.2: Example of page titles with their real and predicted by XgBoost classes (for English dataset)

Model prediction	Real class	Page titles
CE	CE	'Kyiv_offensive_(2022)', '2022_Nauruan_parliamentary_election', 'Siege_of_Mariupol', '2022_Sri_Lankan_protests', 'Baloch_Nationalist_Army'
Not CE	CE	'Journal_of_Vertebrate_Paleontology', Ministry_of_Transport_and_Communications_(Peru)', 'Verdal', 'Bam_Province', 'Point_Bonita_Lighthouse'
CE	Not CE	'Prithviraj_Chauhan', 'Australian_rules_football_in_South_Africa', 'Metaverse', '2022–23_UEFA_Nations_League_B', 'Paris_Agreement'
Not CE	Not CE	'2017_Women's_Rugby_World_Cup_squads', 'Extended_Constructed_Response', 'If_It_Ain't_Love_and_Other_Great_Dallas_Frazier_Songs', 'Christine_Scipio-O'Dean', 'Rafael_Matos'

Also, using SHAP values we can see how exactly one or other value influences the model prediction.

Let's take a look at one example, where the XgBoost model predicted a high (0.9735) probability of a page being on the Current Events (in Figure 4.6, left side). The page is "Southern_Ukraine_offensive", as of 18/03/2022. We can see two features that influenced model decision the most are previous day views and the year before revisions. While for previous day views high value is associated with being a Current event, for a year before revisions, low value is associated with this. So, in this case, the main factor for the model to consider a page to be in the Current Event portal is attention from viewers the day before, while no revisions in the past year (does not include the previous day).

For the "San_Diego_Climate_Action_Plan" page the model predicted a low probability (0.00018) of the page to be on the Current Events as of 01/06/2022 (in Figure 4.6, right side).

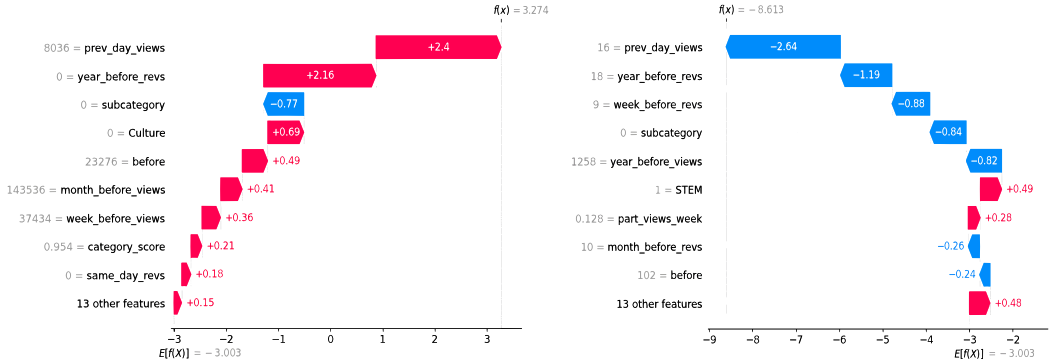

FIGURE 4.6: Examples of SHAP values for individual examples (for English dataset, XgBoost model)

The main factors for the decision are the same as in the previous example (previous day views and year before revisions). But the value of previous day views is much lower, while the number of the year before revisions is higher.

4.2 Cross-language Transferability

4.2.1 Experiment Setup

In this section, we are investigating the cross-language transferability of the XgBoost model described in 4.1. In this section, we review the performance of the XgBoost model trained in English and applied to the Polish and German datasets. We selected English, German, and Polish as the three languages with the constantly active Current Events communities. For the German portal, in 2022 there were 688 events, while for the English portal, there were 9,674, and 512 events for the Polish portal. More details about users' activity in portals are presented in Table 4.3.

TABLE 4.3: Comparison of the Current Event portal activity in different languages

	English	German	Polish
# revisions per day	98,629	16,897	4,788
# events for 2022	9,674	688	512
Average # events per day in 2022	26.5	1.9	1.4
Imbalanced class ratio	10.2/1	24.6/1	9.3/1

In this section, our goal is to check if a model trained in one language produces reliable predictions in other languages without tuning. To achieve this we:

1. Applied to German and Polish datasets the XgBoost model - our best performing model - trained on the English dataset and evaluated its performance.

2. Evaluated two baselines that we used before for English - a sorting based on the number of revisions on the previous day and a sorting based on the number of views on the previous day.

3. Using the same features we utilized for the XgBoost model for English we trained and evaluated XgBoost models for German and Polish datasets separately (only using the data for one language at a time).

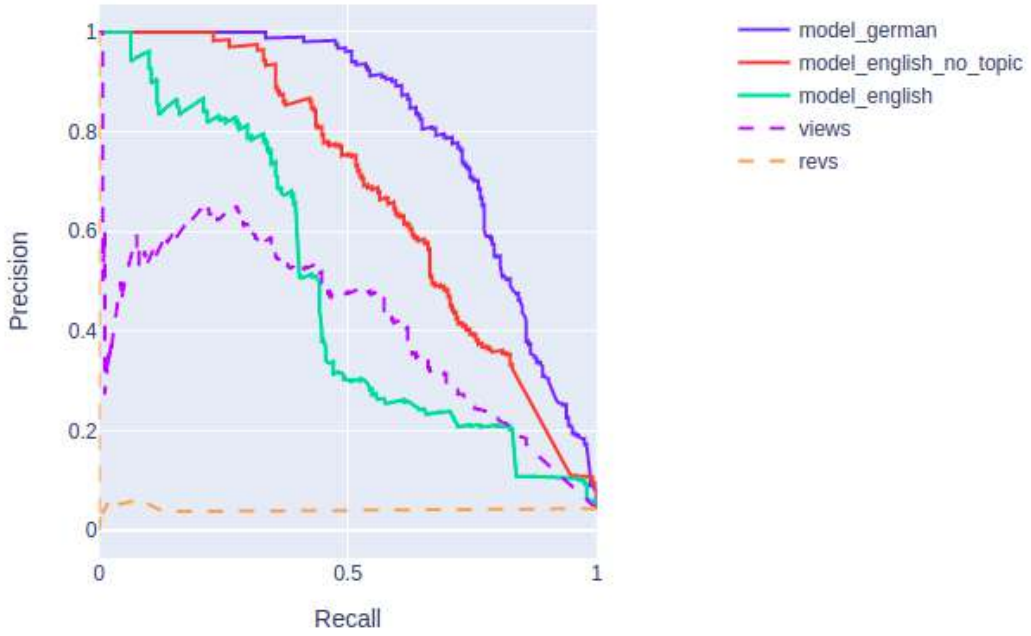

FIGURE 4.7: Precision-Recall plot for German dataset

The target that we predict is the same - the answer to the question - will the page be presented on the CE portal the next day?

4.2.2 Experiment Results

In Figures 4.7 and 4.8 precision and recall metrics for two baselines are presented in the dashed lines, and metrics for three models are presented as solid lines. The lines are:

- 'model_german', 'model_polish' - results of XgBoost model trained based on features described in Section 3.3 on specified languages only;
- 'model_english_no_topic' - results of XgBoost model trained based on features described in Section 3.3, excluding features related to page topic ('topic_score', 'topic', 'subtopic', 'Culture', 'Geography', 'STEM', 'History_and_Society', 'Not defined') and English language;
- 'model_english' - results of XgBoost model trained based on features described in Section 3.3 and English language (described in Section 4.1);
- 'views' - results of baseline model that sorts pages by the number of views on a previous day;
- 'revs' - results of baseline model that sorts pages by the number of revisions on a previous day;

Note that precision doesn't have to decrease monotonically with higher recall. We see it in Precision-Recall plots for both German and Polish.

A little reminder: precision is calculated as TP/(TP+FP), and recall is calculated as TP/(TP+FN), where TP is True positive, FP is False Positive, FN is False negative and TN is True Negative.

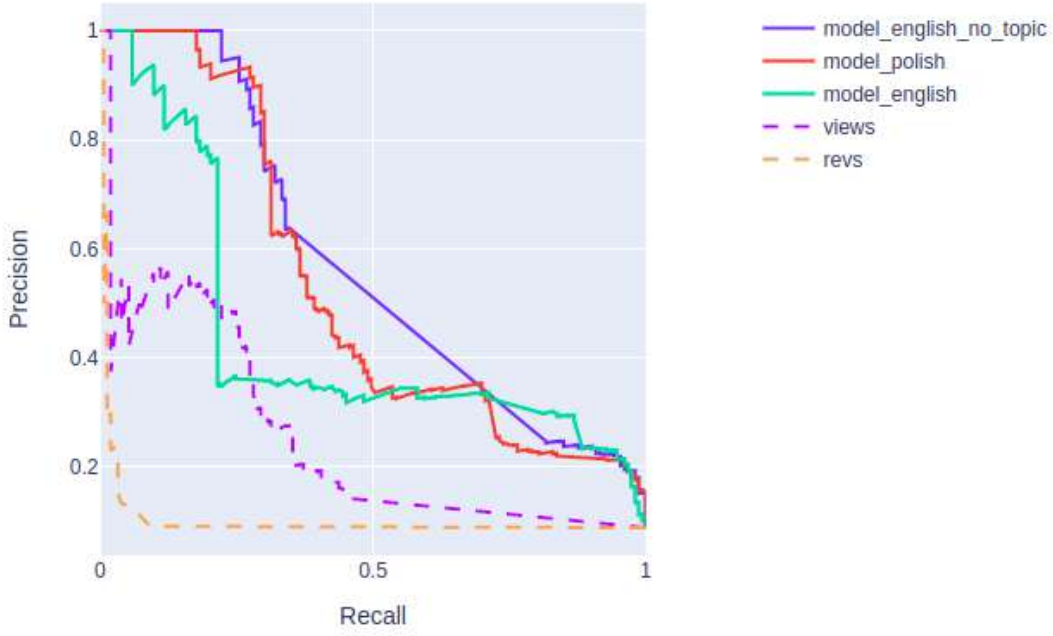

FIGURE 4.8: Precision-Recall plot for Polish dataset

An example: for the Polish language and sorting based on views:

- With threshold 0.06 we have TP=5, FP=5, FN=148, TN=1560. So, the precision is 0.5, and the recall is 0.0032.
- With threshold 0.047 we have TP=6, FP=5, FN=147, TN=1560. So, the precision is 0.55, and the recall is 0.0038.

As we see, in the example both precision and recall grew simultaneously, even though often we need to sacrifice one to improve another one.

As for model performance, we can see that for German, the model trained in German performs the best, followed by the model trained in English without topic features. The model trained in English based on all features shows a big degradation in performance, compared to the results in English. For the baselines, similarly to English, sorting based on the previous day's views works much better than the one based on the previous day's revisions.

For Polish, a model trained in English without topic features performs the best in a majority of cases, followed by model trained or Polish data. We can also see that performance degraded for Polish much more, it can be explained less maintained Polish Wiki and CE portal, compared to English and German. As for baseline models - the situation is the same with other languages, meaning that the sorting based on views is the most successful one.

One more step that we did is the introduction of the model trained in English without topic features based on SHAP values' importance for different languages (described in more detail in Section 4.2.3) we detected that the influence of those features changes the most from language to language. That's why we decided to train and evaluate the XgBoost model in the same setup described in Section 4.1, with the only difference - removing all the topic-related features from the dataset. In the figures we reference this model as 'model_english_no_topic'.

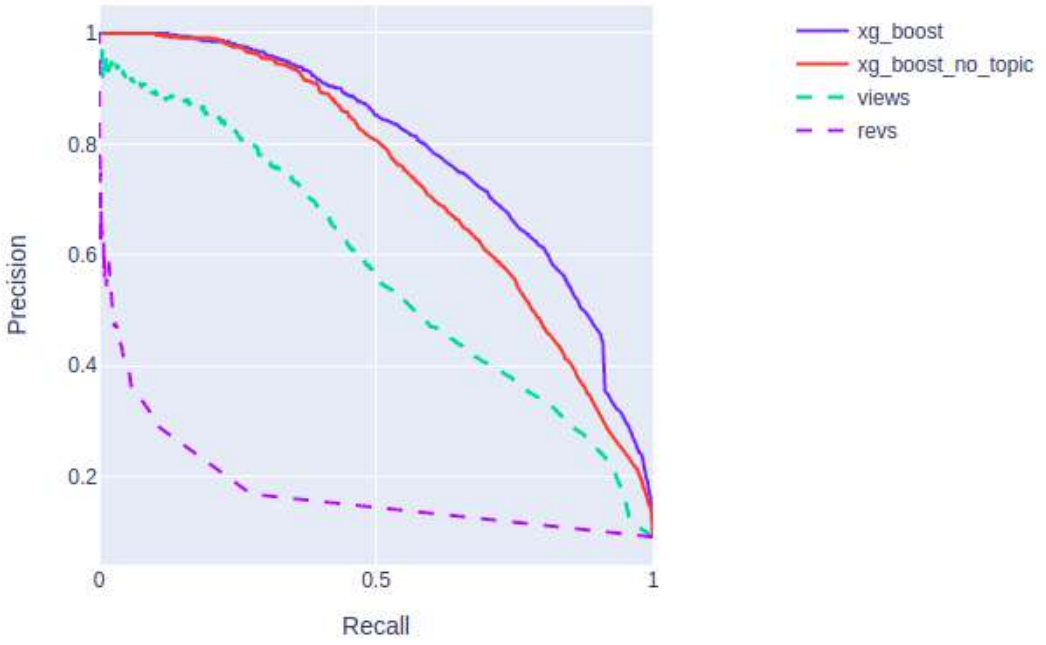

FIGURE 4.9: Precision-Recall plot for XgBoost based on all features, and XgBoost based on all features, excluding topic features (for English dataset)

In Figure 4.9 we share the difference in performance of the XgBoost model based on all features, XgBoost model based on all features, excluding topic features together with 2 baselines for comparison for the English language.

We can see that performance for the English language dropped quite consistently for different thresholds. To dive deeper we present the results of different models at the point with the best F1-score in Table 4.2.2.

Table 4.4: Results of models in different languages in the optimal F1-score point

Model	Precision			Recall			F1-score			Threshold		
	DE	**PL**	**EN**	**DE**	**PL**	**EN**	**DE**	**PL**	**EN**	**DE**	**PL**	**EN**
Revisions baseline	0.04	0.09	0.17	0.01	1	0.28	0.02	0.16	0.21	0.0398	0	1e-4
Views baseline	0.48	0.46	0.52	0.56	0.25	0.55	0.51	0.32	0.53	4e-4	0.0025	4e-4
Model from English	0.67	0.34	0.73	0.39	0.71	0.72	0.49	0.46	0.73	0.17	0.0525	0.13
Model from English, no topic	0.75	0.71	0.68	0.52	0.33	0.63	0.61	0.45	0.65	0.4296	0.3371	0.34

Continued on next page

Table 4.4: Results of models in different languages in the optimal F1-score point (Continued)

Language-specific model	0.79	0.88	0.73	0.71	0.29	0.72	0.75	0.44	0.73	0.0256	0.6646	0.13

We can see that in terms of F1-score, English and German models trained only in the specified language work the best, while for Polish the best performance has a regular model from English.

For all languages the second model in terms of performance in the model trained in English, without topic features. For Polish it performs almost the same as a winning model, while for English and German, the drop is notable.

Still, considering that we target to find a model that works the best across different languages, our absolute winner is the model trained on English without topic features, as its results are consistent across languages.

Let's also see how the chosen model (English, no topic) will perform in different languages given the same threshold.
We can note that for English and Polish, the optimal thresholds are almost the same ≈ 0.34. So we took it for further investigations and reported the performance of the chosen model across languages with the 0.34 threshold in Table 4.5.

TABLE 4.5: Results of XgBoost model trained on English, without topic with fixed threshold=0.34 for different languages

	Precision	Recall	F1-score
English	0.68	0.63	0.65
German	0.69	0.54	0.61
Polish	0.71	0.33	0.45

We can see that the F1-score is the same as optimal.
So, the best model in terms of cross-language transferability that we received is XgBoost, trained in English, without topic features with the threshold = 0.34.

4.2.3 Results Interpretation

When we evaluated results for language-specific models and the model transferred from English to German and Polish datasets, we decided to check changes in feature importance. The motivation to do this is to find features with different SHAP feature importance for different languages. Because they may prevent the model trained on English data from being well transferable.

Table 4.6: Importance difference between model transferred from English and specified language-specific model (for top 10 features with largest average importance difference)

Feature	Importance diff, GE	Importance diff, PL	Importance diff, avg
STEM	100%	82%	91%

Continued on next page

Table 4.6: Importance difference between model transferred from English and specified language-specific model (for top 10 features with largest average importance difference) (Continued)

Not defined	82%	81%	81%
topic	93%	36%	65%
week_before_revs	90%	35%	62%
Culture	93%	29%	61%
Geography	64%	53%	59%
prev_day_revs	100%	17%	59%
part_views_week	28%	89%	59%
History_and_Society	100%	11%	55%
week_before_views	40%	66%	53%

In Table 4.2.3 we present the difference between the importance of features in the language-specific model compared to the English one in percentages for the top 10 features with the largest average difference in importance.

We noticed that the top-3 features with the largest difference in feature importance are related to the topic. Also, 5 of the top-6 features are related to a sample topic as well. Meaning, that topic features have different importance between languages. Also, from the analysis in Section 3.2, we know that the distribution of topics across languages is different.

Based on that we decided to introduce and evaluate XgBoost model trained in English, but without topic features. As we already discussed in Section 4.2.2 it turned out to work better in terms of transferability.

Chapter 5

Conclusions and Future Work

In the previous chapter, we described our experiments on event detection. Now, we present conclusions for the whole work that we did.

5.1 Conclusions

The main goal of this work was to establish the task of supervised event detection in Wikipedia and propose a language-agnostic solution to it.

In this work we:

1. Collected and constructed a dataset for the Wikipedia event detection task that contains information about views, revisions, topics of the Wikipedia pages, and ground truth for the event detection task obtained from the Current Events Portal.

2. Developed and evaluated a language-agnostic model for predicting Current Events in the English dataset, reaching a 0.73 F1-score, outperforming all the baselines.

3. Checked cross-language transferability of the proposed approach by assessing its performance in German and Polish languages. After the removal of topic-related features, we have trained the model that shows a 0.65 F1-score in English, while at the same time maintaining a 0.61 F1-score in German and a 0.45 F1-score in Polish.

The dataset and code are available publicly in the GitHub repo[1].

5.2 Discussion

In this work, we developed a language-agnostic approach for event detection in Wikipedia data, evaluating different algorithms and feature sets in three languages. Below we present key observations from our work.

5.2.1 XGboost outperforms Logistic regression model

Experiments show that Logistic regression has poor performance on our data, while the best results were obtained with the XGBoost model, followed by Random Forest. We hypothesize that features related to the topic, views, and revisions of the pages have no linear patterns that separate pages with CEs from pages without CEs.

[1] https://github.com/AlisaEdu/current_events_detection

5.2.2 Previous day views is the most important feature across all experiments

This observation is interesting because it highlights Wikipedia's role as a news backgrounder. In our work, we used features related both to the revision history of the pages and their views history. A high amount of revisions for the page indicates the activity of the editorial community of Wikipedia associated with an event. At the same time, a high amount of views indicates readers' activity. Information flow in the last case can go this way: a reader learns about an event from a source that seems not reliable enough and decides to check on Wikipedia if the information they learned is true or just a rumor. Because Wikipedia has much more viewers than editors, spikes in viewership may be more indicative of detecting new occurring events. However, our experiments show that models that utilize only view information perform worse than those that use information both about views and revisions of the features, which once again indicates that for event detection tasks there is a complex dependency between the features that describe classes separability.

5.2.3 Topic features limit cross-language model transferability

One of the experimental setups that we investigated was cross-language model transferability. In Section 4.2 we transferred models trained in English to German and Polish to validate them in those languages. Major observations made after feature importance differences across languages analysis was that the usage of the features related to the page topics decreased the effectiveness of the models in a cross-language setup. At the same time, we saw that features related to the topic of the page are informative for the event detection in each language separately, as some topics have more CE than others. But we can conclude that those patterns are different across languages.

5.2.4 Cross-language model transferability

Experiments from Section 4.2 have shown that model trained on the English Wikipedia works better than Polish specific model for the Polish dataset (0.45 F1-score for the transferred model, compared to 0.44 F1-score for language specific), and shows 0.61 F1 score on German, which is close to the 0.65 F1-score that model gives in the English language that it was originally trained on. We view these as data points that support the cross-language transferability of our approach.
Still, even though the model uses language-agnostic features and shows good results, the language-specific models still work better for German and we can see different performances of the selected model in different languages. Our hypothesis is that this happens because each language segment of the Wiki is maintained and read by a different audience. Different audiences may have different behavior patterns in interaction with Wikipedia, which results in a bit different meanings of the same features.

5.3 Limitations

Also, there are some important limitations to mention.

1. The data collection mechanism we established is not optimized for real-time processing. To collect data for our work we utilized 5 different data sources

and APIs within the Wikipedia infrastructure. One of which is Quarry, which does not have API integration, so the data can only be requested by the website, and downloaded manually. It creates a bottleneck for full automation and API takes a considerable amount of time, limiting the amount of data we were able to collect and process. The same data can be found in Wikipedia dumps, but it requires a massive amount of available resources and additional engineering efforts to collect and parse the data on the scale.

2. The limitations in the data collection also led to a limited size of the datasets collected. While data for CE is collected for the whole year, data on pages edited is collected only for a day.

3. Topic API returns the topic of the page as of the moment it was requested, and it does not have the ability to specify the time at which we want to know the topic (when we consider a page to be a candidate for CE). The topic of a page is something that can change over time, meaning, that for purposes of our task of CE prediction, this change of page topic might introduce noise in the data. To overcome this limitation one can train models without the topic as a feature. Our experience showed that this will negatively affect model performance in English, but at the same time improve cross-language transferability.

4. The usage of the Current Events as a ground truth is a trade-off between the cost of data collection and the number of events captured, as the CEP might not contain all the events. In other words, the ground truth has good precision but unknown recall. To mitigate this issue, manual annotation and verification of the events can be done.

5.4 Future Work

In future work, our core area of improvement is to set up a fully automated real-time data collection process. It will facilitate the following key points:

- It will allow the collection of bigger datasets to level up the quality of the model across the languages by capturing more dependencies and evaluating the performance with higher confidence;
- It will enable the introduction of the tool that can be easily used by any Wiki editors for events suggestions.

To improve the data we consider adding features related to editors (number of unique editors, percentage of edits done by bots); network features (changes in the number of links from and to the page, vectorized representation of a page by links). The introduction of the real-time data stream will also allow us to investigate recurrent neural networks and other approaches, that can benefit from analysis of the dynamics of the feature.

5.5 Reproducibility statement

We shared the main code and datasets used for writing this work in the "current_events_detection" Github repository[2].

[2] https://github.com/AlisaEdu/current_events_detection

The datasets for English, German, and Polish languages are located in the "data" folder. Final models in pickle format are stored in the "models" folder.

All models are created and evaluated with the code stored in the "modeling" folder (using data from the "data" and models from the "models" folders). Data analysis is stored in the "analysis" folder (using data from the "data" folder). Queries used to collect data from Quarry[3] service are stored in "quarry.docx".

The data collection code can be found in the "data_colection" folder. There we present all functions used for the data collection. The events collection can be reproduced by notebooks "get_events_[lang].ipynb". For the full dataset generation, one should collect events as described in "get_events_[lang].ipynb", and request data about pages from Quarry (query is in "quarry.docx").
Then, for events follow the logic in "get_events_features_ge.ipynb"; for pages follow "get_pages_features_ge.ipynb".
The way to merge datasets is described in "join_dfs_ge.ipynb".

Several remarks here:

- we share the data collection for the German language only, but for others, the process is the same (make sure to change a language parameter, if API uses one)
- we collected data for pages with and without events separately, following the same process, with the only difference being that for pages we requested revisions features from Quarry to speed up the process. We additionally checked that data from Quarry and MediaWiki REST API "Get page history counts"[4] is the same.

Additional data, such as more events features (that we did not use, but collected, train/test split) can be found at the Google Drive [5].

[3] https://quarry.wmcloud.org

[4] https://www.mediawiki.org/wiki/API:REST_API/Reference

[5] https://drive.google.com/drive/folders/118klLVhF-mWtjLsFODzpZhnATHtUxnth?usp=sharing

Bibliography

Azeemi, Abdul Hameed et al. (Mar. 2021). "RevDet: Robust and Memory Efficient Event Detection and Tracking in Large News Feeds". In: URL: https://arxiv.org/abs/2103.04390.

Bojanowski, Piotr et al. (2017). "Enriching Word Vectors with Subword Information". In: arXiv: 1607.04606 [cs.CL].

Boydstun, Amber E., Anne Hardy, and Stefaan Walgrave (2014). "Two Faces of Media Attention: Media Storm Versus Non-Storm Coverage". In: *Political Communication* 31, pp. 509 –531.

Buntain, Cody (2017). "Learning to Discover Key Moments in Social Media Streams". In: *ACM Transactions on Intelligent Systems and Technology* 8.6, 83:1–83:23. DOI: 10.1145/3131269.

Chen, Tianqi and Carlos Guestrin (2016). "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 22.2, pp. 785–794. DOI: 10.1145/2939672.2939785.

Cox, David R (1958). "The regression analysis of binary sequences". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 20.2, pp. 215–232.

Dabiri, S. and K. Heaslip (2019). "Developing a Twitter-based traffic event detection model using deep learning architectures". In: *Expert Systems with Applications* 118, pp. 425–439. DOI: 10.1016/j.eswa.2018.10.017.

Devlin, Jacob et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: arXiv: 1810.04805 [cs.CL].

Feng, Fangxiaoyu et al. (2022). "Language-agnostic BERT Sentence Embedding". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 878–891. DOI: 10.18653/v1/2022.acl-long.62.

Gildersleve, Patrick, Renaud Lambiotte, and Taha Yasseri (Nov. 2022). "Between News and History: Identifying Networked Topics of Collective Attention on Wikipedia". In: DOI: 10.48550/arXiv.2211.07616.

Ho, Tin Kam (1995). "Random Decision Forests". In: *Proceedings of the 3rd International Conference on Document Analysis and Recognition* 1, pp. 278–282.

Hochreiter, Sepp and Jürgen Schmidhuber (Dec. 1997). "Long Short-term Memory". In: *Neural computation* 9, pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

Johnson, Isaac, Martin Gerlach, and Diego Sáez-Trumper (2021). "Language-agnostic Topic Classification for Wikipedia". In: arXiv: 2103.00068 [cs.CY].

Keegan, Brian et al. (2013). "Hot off the Wiki". In: *American Behavioral Scientist* 57.5, pp. 595–622. DOI: 10.1177/0002764212469367.

Leskovec, Jure, Lars Backstrom, and Jon Kleinberg (June 2009). "Meme-Tracking and the Dynamics of the News Cycle". In: pp. 497–506. DOI: 10.1145/1557019.1557077.

Lundberg, S.M. et al. (2020). "From local explanations to global understanding with explainable AI for trees". In: *Nature Machine Intelligence* 2.1, pp. 56–67. DOI: 10.1038/s42256-019-0138-9.

Mikolov, Tomas et al. (2013). "Efficient Estimation of Word Representations in Vector Space". In: arXiv: 1301.3781 [cs.CL].

Mredula, Motahara Sabah et al. (2022). "A Review on the Trends in Event Detection by Analyzing Social Media Platformsrsquo; Data". In: *Sensors*. DOI: 10.3390/s22124531.

Oghaz, Toktam A. et al. (2020). "Probabilistic Model of Narratives Over Topical Trends in Social Media". In: *Proceedings of the 31st ACM Conference on Hypertext and Social Media*. DOI: 10.1145/3372923.3404790.

Pouran Ben Veyseh, Amir et al. (2022). "Minion: A Large-Scale and Diverse Dataset for Multilingual Event Detection". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. DOI: 10.18653/v1/2022.naacl-main.166.

Salton, G. and C. Buckley (1988). "Term-weighting approaches in automatic text retrieval". In: *Information Processing Management* 24.5, pp. 513–523. DOI: 10.1016/0306-4573(88)90021-0.

Singer, Philipp et al. (2017). "Why We Read Wikipedia". In: *Proceedings of the 26th International Conference on World Wide Web*. DOI: 10.1145/3038912.3052716.

Steiner, Thomas et al. (2013). "MJ No More". In: *Proceedings of the 22nd International Conference on World Wide Web*. DOI: 10.1145/2487788.2488049.

Tran, Tuan et al. (2014). "WikipEvent: Leveraging Wikipedia Edit History for Event Detection". In: *Springer International Publishing*, pp. 90–108. DOI: 10.1007/978-3-319-11746-1_7.

Wang, Xiaozhi et al. (2020). "Maven: A Massive General Domain Event Detection Dataset". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. DOI: 10.18653/v1/2020.emnlp-main.129.

Yagcioglu, Semih et al. (2019). "Detecting Cybersecurity Events from Noisy Short Text". In: arXiv: 1904.05054 [cs.CL].