A Mastodon Corpus to Evaluate Federated Microblog Search

Matti Wiegmann[1,†], Jan Heinrich Reimer[2,†], Maximilian Ernst[2], Martin Potthast[3], Matthias Hagen[2] and Benno Stein[1]

[1]Bauhaus-Universität Weimar, 99423 Weimar, Germany
[2]Friedrich-Schiller-Universität Jena, 07743 Jena, Germany
[3]Leipzig University and ScaDS.AI, 04109 Leipzig, Germany

Abstract

In this paper, we present the Webis Mastodon Corpus 2024, a collection of about 733 million public posts from the timelines of 1,015 Mastodon nodes across 61 days. Mastodon is a federated open-source microblogging platform that gained a lot of attention in 2023 as an alternative to Twitter (now rebranded as X). However, searching Mastodon is not straightforward due to its federated architecture. This presents an interesting new challenge for federated IR research, and our corpus is meant as a starting point for the new direction of federated microblog search. To ensure privacy, we host the corpus on TIREx, where it can be processed but neither read nor downloaded, with the goal of developing a shared task and a public leaderboard. We also publish our parallelized and polite Mastodon crawler alongside this paper.[1]

Keywords

Microblog Search, Federated Search, Mastodon, Fediverse, Open Social Media

1. Introduction

In the wake of Twitter's self-inflicted demise, several new competing microblogging services have emerged. They took the chance to grow by inviting the users that wanted to leave Twitter onto their platforms. Many of these users divided across three platforms: Mastodon, Bluesky, and Threads.[2,3] Mastodon is unique among them in that it implements the federated open-source social networking protocol ActivityPub [1], a W3C standard, which forms the basis of the Fediverse, the federated "universe" of social networks that can communicate with each other via this protocol.[4] The Fediverse in general, and Mastodon in particular, are proclaimed to be less vulnerable to platform decay [2] than proprietary platforms. Their core principles, namely openness, federation, and independence of the attention economics, give users a lot of agency, from creating Mastodon nodes with custom rules to options for controlling content visibility. Mastodon is highly interesting for researchers, because, similar to Wikipedia, the inner mechanisms of a large social media platform are publicly visible, and because promising research results can be more directly transferred to practice.

This is also true for search and retrieval on Mastodon, which presents new and unique challenges due to the federation of the platform. In September 2023, a consensual, node-level search functionality has been integrated into the platform, which enables search on a given Mastodon node but not across

WOWS'24: 1st Workshop on Open Web Search at ECIR 2024, March 28, 2024, Glasgow, United Kingdom

[†]These authors contributed equally.

matti.wiegmann@uni-weimar.de (M. Wiegmann); heinrich.reimer@uni-jena.de (J. H. Reimer); maximilian.ernst@uni-jena.de (M. Ernst); martin.potthast@uni-leipzig.de (M. Potthast); matthias.hagen@uni-jena.de (M. Hagen); benno.stein@uni-weimar.de (B. Stein)
https://sigmoid.social/@mattiwiegmann (M. Wiegmann); https://mastodon.acm.org/@jhreimer (J. H. Reimer); https://idf.social/@potthast (M. Potthast); https://idf.social/@matthias (M. Hagen)
0000-0002-3911-0456 (M. Wiegmann); 0000-0003-1992-8696 (J. H. Reimer); 0009-0009-5954-7725 (M. Ernst); 0000-0003-2451-0665 (M. Potthast); 0000-0002-9733-2890 (M. Hagen); 0000-0001-9033-2217 (B. Stein)

[1]Source code and data available online: https://github.com/webis-de/mastodon-search

[2]https://joinmastodon.org, https://bsky.app, https://threads.net

[3]For instance, Mastodon doubled its long-term active user base in 2023 to over 1 million according to https://fedidb.org/software/mastodon, January 15, 2024.

[4]Both Bluesky and Threads have announced they will be connected to the Fediverse in the future: https://docs.bsky.app/blog/feature-bridgyfed, https://help.instagram.com/169559812696339

Table 1
Number of posts crawled from each node's federated (🏙), local (🏠), or remote (🏢) timeline, in comparison to the number of available posts of each crawled (🗄) or discovered (🌐) node. Sorted by the number of posts in the node's federated timeline. Comparisons of post counts are given for 🏙← a nodes's remote/local timeline with the federated timeline, 🗄↓ a node's timeline to the deduplicated corpus (i.e. the centrality), and 🌐→ a node's local timeline in the corpus with the node's total. Deduplicated counts are approximate via HyperLogLog++ [3].

Node	Crawled posts								Available	
	🏙 Fed.	🗄↓	🏢 Remote	🗄↓	🏙←	🏠 Local	🗄↓	🏙←	🌐→	🏠 Local	🌐↓
mastodon.social	15,668,157	44%	12,542,364	50%	80%	3,125,793	31%	20%	4%	81,205,325	10%
mastodon.online	9,857,093	28%	9,570,010	38%	97%	287,083	3%	3%	4%	7,759,995	1%
mstdn.social	9,638,803	27%	9,267,685	37%	96%	371,118	4%	4%	2%	14,989,224	2%
ohai.social	8,754,882	25%	8,735,250	34%	100%	19,632	0%	0%	2%	1,180,837	0%
mas.to	8,208,704	23%	8,034,084	32%	98%	174,620	2%	2%	2%	7,021,649	1%
mastodon.world	8,149,459	23%	7,968,251	31%	98%	181,208	2%	2%	4%	4,912,465	1%
universeodon.com	7,489,453	21%	7,399,674	29%	99%	89,779	1%	1%	3%	2,852,641	0%
social.vivaldi.net	7,194,763	20%	7,055,731	28%	98%	139,032	1%	2%	8%	1,709,786	0%
techhub.social	7,121,898	20%	7,047,603	28%	99%	74,295	1%	1%	5%	1,404,970	0%
toot.community	6,660,323	19%	6,630,708	26%	100%	29,615	0%	0%	2%	1,303,738	0%
⋮ 1,005 others	644,299,245	—	638,818,357	—	87%	5,480,888	55%	1%	1%	261,449,365	32%
🗄 1,015 crawled	733,042,780	—	723,069,717	—	99%	9,973,063	100%	1%	3%	385,789,995	47%
↳ deduplicated	35,300,568	100%	25,327,505	100%	72%	9,973,063	100%	28%	3%	385,789,995	47%
🌐 10,354 discovered	—	—	—	—	—	9,973,063	—	—	1%	823,560,767	100%

nodes. Searching the federated network of Mastodon nodes as a whole, however, is still difficult. A federated search would require the commitment and reliability of all participating nodes, and to balance effectiveness with efficiency while respecting user and node preferences for visibility and consent.

In this paper, we create the foundation for research on federated search on Mastodon by creating the Webis Mastodon Corpus 2024 (Section 3) and by analyzing it with respect to the perspectives and limitations for search on Mastodon (Section 4). Our collection consists of about 733 million posts from the local and federated timelines of 1,015 diverse instances, spanning 61 days worth of Mastodon traffic (see Table 1). We offer limited, privacy-preserving access to this collection by hosting it on TIREx [4], The Information Retrieval Experiment Platform, which is implemented on top of TIRA [5], the TIRA Integrated Research Architecture. We already invite researchers to contribute retrieval systems for a future shared task on federated microblog search on Mastodon.

2. Related Work

Mastodon has attracted a fair amount of scholarly attention since its inception in 2017, but not comparable in volume to research on Twitter, Facebook, or Reddit. Existing work is interested in social relations and structures formed in the federated scenario [6, 7, 8], in content moderation [9] and governance [10], and, very recently, in migration patterns towards the Fediverse [11, 12, 13].

While there is some prior work in information retrieval on Mastodon (e.g., account recommendation [14]), most work on search is driven by community initiatives like `propulsion.social` to search nodes. However, none of these initiatives search for posts, and those that attempted to do so, like `search.noc.social` and `fedsearch.io`, were shut down citing "extreme backlash from the community". A 2022 paper on content moderation was retracted for similar reasons: the statement of removal [15] cites GDPR violations in the analyzed and redistributed user content. The focus on consent and protection of user content aligns with the developer's earlier stance on search [16]. However, Mastodon introduced post search on the federated timeline as a per-account opt-in feature in September 2023 [17]. This new means of consent allows us to study post search on Mastodon in typical Cranfield experiments, especially under TIRA's [5] protection of the index. TIRA as an EaaS platform allows the evaluation of retrieval systems on privately held indexes, where only the evaluation results reach the public.

Regarding the search for microblogs, we can build on substantial prior work on task definition and evaluation from the TREC 2011–2015 Microblog tracks [18, 19, 20, 21, 22]. The tracks introduced essentially two paradigms of tasks: (1) the (temporally-anchored or real-time) ad-hoc search task (2011–2014) and the filtering task (2012), where posts created before or after a time T should be retrieved and scored regarding their relevance to a given topic, and (2) the "summarization" tasks, where a summary of relevant and novel posts should be retrieved, either up to a time T (timeline generation in 2014; as "daily digest" style in 2015) or starting at a time T as a stream filtering task. The latter paradigm (2) was continued as the "real-time summarization" track at TREC 2016–2018 [23, 24, 25]. Outside of these TREC tracks, there is little work on creating new tasks or collections but the existing ones are still used frequently to evaluate new technologies [26, 27, 28]. From 2013 onwards, the TREC Microblog tracks also made their collections available through an EaaS system like TIRA to avoid issues with distributing large collections of sensitive and protected data. The post search as currently implemented in Mastodon equates to TREC's "ad-hoc search" task, so that will be our immediate focus. It should be noted though that the utility of this paradigm is limited and a "summarization"-style search is the natural continuation for Mastodon search, too.

Regarding the search in federated systems, we can build on prior work regarding tasks and, to a degree, systems and evaluation as presented in a recent survey of federated web search [29]. The TREC 2013–2014 Federated Web Search tracks [30, 31] introduce three tasks. The first task is "Resource Selection", where users should determine the best resource to query given the results of several prior queries. The TREC track considers different search engines that are heterogeneous in content type, search collections, and retrieval systems. Mastodon, on the other hand, is almost homogeneous: all resources implement the same protocol (i.e., ActivityPub), all have the same content type (posts) and similar retrieval systems (although there might be different software versions). The challenges for resource selection instead are efficiency and politeness (see Section 4). The second task is "Results Merging", where the given SERPs from various sources should be merged, and the relevance of the resulting list is scored. Since all posts can be ordered by post time, this task will be reduced to a microblog search. The third task is "Vertical Selection", where the best set of topics, genres, and media types should be determined for each query. Although verticality might become relevant for search on Mastodon at some point (e.g. for balancing text, images, and videos in the results or to extend the search across the Fediverse), we decided to ignore it for now. For now, we assume that the resource selection aspect combined with ad-hoc microblog search form a well-grounded basis for a first evaluation campaign.

3. Constructing a Collection of Mastodon Posts

We created our collection of 733 million posts by capturing the local and federated timeline of 1,015 nodes concurrently for 61 days, between December 12, 2023 and February 21, 2024. Our corpus is built in three steps: (1) We sample a set of diverse nodes to crawl, including large and general nodes as well as small communities, (2) we crawl the posts from these nodes, and (3) we bundle the crawled posts in a re-usable document collection.

3.1. Node Sampling

There are many ActivityPub nodes on the Fediverse and many of them are small, inactive, or do not concern Microblogs. To limit the acquisition and filtering load, we sampled a subset of all discoverable nodes in a principled way. First, we got the 22,178 discoverable ActivityPub nodes through a public and up-to-date list.[5] Second, we download each node's general statistics[6] and their weekly activity[7] across the three months prior to our crawling and subsequently discarded the 11,822 non-Mastodon nodes without these endpoints. Third, from the remaining 10,354 candidate nodes, we sampled 1,000 Mastodon

[5] https://nodes.fediverse.party (crawler source code available online: https://github.com/Minoru/minoru-fediverse-crawler)
[6] https://nodeinfo.diaspora.software/
[7] https://docs.joinmastodon.org/methods/node/#activity

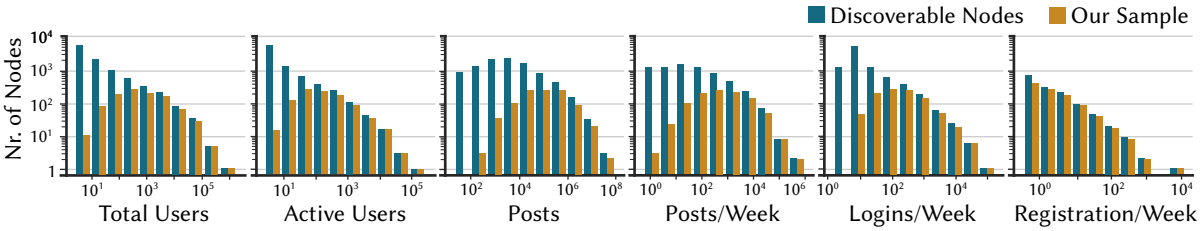

Figure 1: Histograms showing the numbers of nodes (log scale) according to the (log) activity measures used for sampling: total users, monthly active users, total posts, average weekly posts, average weekly logins, and average weekly registrations. Shown are the 10,354 discoverable Mastodon nodes on the Fediverse (blue) and the 1,000 nodes sampled for corpus construction (yellow).

Figure 2: Pairwise Spearman's ρ correlation of the node activity measures used for sampling: total users, monthly active users, total posts, average weekly posts, average weekly logins, and average weekly registrations.

nodes based on the six activity measures shown in Figure 1: (1) total users, (2) monthly active users, (3) total posts, (4) average weekly posts, (5) average weekly logins, and (6) average weekly registrations.

The goal of our sampling strategy is to include most (moderately) large and active nodes while also representing smaller and less active nodes. However, the histograms in Figure 1 show that the activity across nodes is roughly log-normally distributed but skewed towards low activity, so less active nodes would dominate a (uniform) random or a (by size and activity) stratified sample. Instead, we apply weighted sampling, where the weight w of each node $k \in K$ is discounted by the joint probability of observing its activity statistics $P(X_a); a \in A$. We define a random variable for each of the six activity measures: $X_1 :=$ total users, $X_2 :=$ monthly active users, $X_3 :=$ total posts, $X_4 :=$ average weekly posts, $X_5 :=$ avg. weekly logins, and $X_6 :=$ avg. weekly registrations, where each variable is $X_a \sim \text{Lognormal}(\mu_a, \sigma_a^2)$. We fit the mean μ_a and standard deviation σ_a^2 of each log-normal distribution X_a on the measures of the candidate nodes. For computational simplicity, we assume the independence of $X_1, \ldots, X_6$, although that is generally not the case (see Figure 2). We calculate the sampling probability as:

$$P(k) \approx \frac{w_k}{\sum_{i \in K} w_i} \quad \text{with} \quad w = \frac{1}{P(X_1, \ldots, X_6)} \quad \text{and} \quad P(X_1, \ldots, X_6) = \prod_{a \in A} P(X_a)$$

In other words, nodes whose activity scores are very likely under the log-normal distributions X_i are less likely to be sampled. In the candidate nodes, this affects nodes with low activity much stronger and so the final samples (see Figure 1, yellow bars) are less skewed.

Online Resampling During the corpus construction, 15 nodes became unavailable to the crawler.[8] They may have gone offline, which is not uncommon for smaller nodes, or they might have blocked our crawler. Since this effect is relevant for resource selection, we decided to replace these nodes on the fly with the node with the closest sampling weight. Consequentially, the timelines of some nodes are only available until a time T, and others are only available starting from T.

```
Input: Mastodon node n, Elasticsearch index i.
function crawl_batched(node n, index i, should stop early) is
    Determine last crawled post p_0 of n from index i        ▷ If node n has not been crawled before, p_0 is not set.
    if p_0 is unknown then
        Fetch the latest timeline batch from node n.
    else
        Fetch next timeline batch after p_0 from node n.
    if batch is empty then                                   ▷ Crawling has "catched up" with the timeline.
        if should stop early then
            return                                           ▷ Stop crawling, e.g., to continue with streaming crawling.
    for post p in batch do
        if noindex flag is not set then
            Save post p to index i.
    crawl_batched(n, i, should stop early)                   ▷ Continue crawling the next batch.
function crawl_streaming(node n, index i) is
    crawl_batched(n, i, true)                                ▷ Run batched crawling until we "catch up" with the timeline.
    for post p in timeline stream of n do
        if noindex flag is not set then
            Save post p to index i.
        if unrecoverable error has occurred then
            crawl_batched(n, i, false)                       ▷ Fall back to batch crawling.
            return
crawl_streaming(n, i)                                        ▷ Start streaming/crawling.
```

Algorithm 1: Pseudocode of a crawler process for a single node. Crawling either uses Mastodon's streaming or batch APIs. One crawler process is run for each of the 1,000 sampled nodes (see Section 3.1).

3.2. Crawling

We implemented a parallelized, polite, and privacy-respecting crawler that fetches all public posts from each node's federated timeline and stores the posts in an Elasticsearch index. Mastodon nodes offer a streaming API that pushes new posts to listening applications via a long-lived HTTP or WebSocket connection,[9] and a REST endpoint for fetching batches of posts.[10] Our crawler uses the streaming API by default (since it causes less load on the nodes) and, if it is unavailable, falls back to the REST API (see Algorithm 1). We also use the REST API to fill "gaps" caused by eventual crawler downtimes. We schedule one crawler process for each of the 1,000 sampled nodes as self-restarting jobs on our Kubernetes cluster (1,620 CPU cores, 25 TB RAM). To balance the indexing load and to be future-proof (see Section 4), we create one index per month of crawling (i.e., December, January, and February) and distribute each index across 20 shards.

Considering politeness, our crawler identifies itself via a custom `User-Agent`,[11] respects server-imposed wait times (i.e., `Retry-After` headers), and limits subsequent requests (e.g., due to connection errors) politely with an exponential backoff (minimum: 14 seconds). Considering privacy, we explicitly remove all crawled posts without the users' permissions, i.e. we remove posts where the `noindex`[12] flag is set.

Stored Data per Post The collection contains ca. 88 GB (index size) of post data per day across 61 days, allocating a total of 6 TB of storage. Table 1 shows an overview of the fields stored for each post. By default, we store all fields returned by the API using the exact same field names to maximize compatibility with Mastodon and its documentation. We also adopt Mastodon's handling of optional and conditional fields: we omit empty optional fields like `cards` and store empty values for conditional

[8] List of the re-sampled nodes: https://github.com/webis-de/mastodon-search/blob/main/data/nodes_resample.txt
[9] https://docs.joinmastodon.org/methods/streaming/#public
[10] Up to 40 posts per request; https://docs.joinmastodon.org/methods/timelines/#public
[11] User agent: `Webis Mastodon crawler (https://webis.de/, webis@listserv.uni-weimar.de)`
[12] https://docs.joinmastodon.org/entities/Account/#noindex

Table 2
Overview of the essential fields contained in the collection for each post. Some (Nr.) similar fields were combined or omitted (Additional fields) for brevity. Fields that are not required (Req.) are omitted in the exported JSON when empty (following the behavior of the Mastodon API). The fields in the document collection exactly follow the naming convention of the Mastodon API but have been renamed here for readability.

Field	Nr.	Type	Req.	Description
Fields of the post				
URI	1	`URI`	✓	The unique address of the post; can serve as federated ID.
Node	1	`string`	✓	The node of the creator of the post.
Crawled from	1	`string`	✓	The node from whose timeline the post was crawled.
Dates	3	`date`	✓	The date of post creation, last edit, and when the post was crawled.
Content	1	`string`	✓	The text content of the post with HTML formatting.
Reply to	1	`ID`	✓	If this post is a reply, this is the ID of the original post.
Is local	1	`boolean`	✓	If true, this post is from the local timeline, else from the federated timeline.
Is sensitive	1	`boolean`	✓	If true, the post is hidden by default and requires a confirmation to be seen.
Spoiler	1	`string`	✓	The text shown before a sensitive post.
Tags	2	`list`	✓	Text and reference of the hashtags used in the post.
Mentions	4	`list`	×	Name of and references to mentioned users.
Poll	8	various	×	If a poll was attached, the options of the poll and their respective votes.
Reblog	2	various	×	If the post is a boost, contain the reference to the original post.
Additional fields	12	various		
Fields of the author or booster of the post (account)				
Handle	1	`string`	✓	The "username@node" handle.
Display name	1	`string`	✓	The username as it should be displayed.
Note	1	`string`	✓	The user's text biography with HTML formatting.
Activity	3	`integer`	✓	The follower count, following count, and number of posts.
Dates	3	`date`	✓	The date of account creation and of the last post.
Is bot	1	`boolean`	✓	If true, then the account is automated.
Is discoverable	1	`boolean`	✓	If true, the account wants to participate in discovery services.
Is locked	1	`boolean`	✓	If true, the account only accepts followers after manual review.
Verification	3	`list`	×	The websites that verify the user's identity.
Additional fields	14	various		
Fields of the optional linked content preview (card)				
URL	1	`URL`	×	The source that is referenced by the linked content.
Title	1	`string`	×	The embedded title of the linked content.
Description	1	`string`	×	The embedded preview snippet of the linked content.
Type	1	`keyword`	×	The content type (link, video, etc.).
Additional fields	11	various		
Fields of the optional media attachment				
URL	1	`URL`	×	The source of the media attachment.
Description	1	`string`	×	The alt-text of the attachment.
Type	1	`keyword`	×	The content type of the attachment (video, image, etc.).
Media metadata	11	various	×	The metadata of the attachment (duration, size, aspect ratio, bitrate, etc.).
Additional fields	4	various		

fields like `reply_to` and `spoiler`. We add 4 fields: the handle (username@node), the crawl date, the node from whose timeline the post was crawled, and a universally unique ID (based on hashing the crawled node and post ID) that is used for Elasticsearch indexing and random access.

3.3. Bundling and Release

We make our document collection accessible to other researchers on TIREx (see Section 2 for a discussion) by extending the `ir_datasets` library. We publicly release a development dataset of about 1,000 posts to demonstrate the export format. We export the collection via newline-delimited JSON files[13]

[13] https://jsonlines.org/

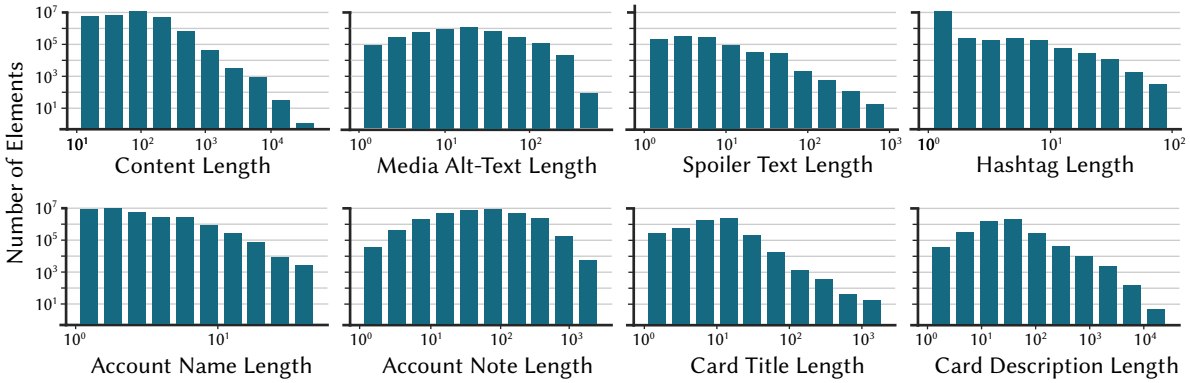

Figure 3: Text length distributions (in indexed tokens; Elasticseach's default tokenizer) of the post contents, media alt texts, spoiler texts, hashtags, account names, account notes, card titles, and card descriptions.

where each line is individually compressed with GZIP.[14] The files that contain the (compressed) posts are partitioned by node in individual directories and sub-partitioned in files of up to 1 GB (e.g., `./mastodon.social/0001.jsonl.gz`) to allow for efficient filtering of nodes and in-memory reads of individual files. An index file that maps post IDs to the corresponding file location is included to allow for random access to individual posts by their IDs.

With `ir_datasets`, researchers can efficiently access the posts from all crawled nodes via the `mastodon` dataset ID, or just the post from a single node (e.g., via the `mastodon/sigmoid.social` dataset ID). In both cases, our `ir_datasets` extension implements random access by post IDs.

3.4. Results and Analysis

After crawling posts for 61 days, we have accumulated a total of 733 million posts from 1,015 nodes (due to re-sampling, see Section 3.1), allocating 6 TB of storage on our Elasticsearch cluster. Here, we quantitatively analyze several noteworthy properties of the document collection.

Text Length We measure the text length in tokens[15] of the eight text fields that can appear in the posts (see Figure 3). Most fields have a length as would be expected for microblogging: Most hashtags have just one or very few tokens, account notes 100s, alt-texts 10s, and account names and spoiler text in the single digits. However, there are some irregularities. First, there are many hashtags with 10s of tokens, which are mostly Japanese and Chinese, where long phrases are regularly used as tags (e.g. "If 15 people call your name, they love you."). Second, while most contents are in the typical range of 10s or low 100s of tokens, the are also many posts with 1,000s of tokens, up to about 10,000. Upon manual examination of a random sample of 10 posts with at least 1,000 tokens, we find that these posts are mostly spam or contain excessively many hashtags (that also count towards the content length).

Contributing Nodes The timelines of our collected nodes contain many posts from remote nodes (72% of all posts, see Table 1). A substantial part of these posts originates from 16,655 distinct Fediverse nodes, despite us having only crawled from 1,015 Mastodon nodes (6%). Table 3 shows the source nodes with the most contributed posts and their characteristics. Notably, half of the shown nodes are not Mastodon servers themselves. Instead, `misskey.io`, `live-theater.net`, and `misskey-square.net` are nodes from the Misskey[16] microblogging network that is popular in Japan. The other two non-Mastodon nodes (`sportsbots.xyz` and `rss-parrot.net`) are hubs for automated accounts only. Apart from `mstdn.jp`, our sample (see Section 3.1) contained all of the top-10 contributing Mastodon nodes.

[14] https://datatracker.ietf.org/doc/html/rfc1952
[15] As indexed in Elasticsearch, see https://elastic.co/guide/en/elasticsearch/reference/current/token-count.html
[16] https://misskey-hub.net/en/

Table 3
Nodes contributing the most posts to our corpus, whether they were sampled for crawling (; see Section 3.1), whether it is a Mastodon () or other ActivityPub server, the covered timespan (; 1% to 99% percentile), number of unique posts, avg. redundancy (; same post crawled from multiple timelines), and proportions: reply posts (), boosting posts (), posts with media (), with hashtags (#), with linked content (), with a spoiler (»), with sensitive content (), posts from bot accounts (), from locked accounts (), from indexable accounts (), and from discoverable accounts (). Unique counts are approximate (HyperLogLog++ [3]).

Node			Days	Unique					#		»					
mastodon.social	✓	✓	61d	4,274,826	28	15%	14%	26%	28%	37%	2%	4%	20%	6%	98%	56%
misskey.io	×	×	61d	1,797,527	5	2%	0%	13%	11%	10%	2%	5%	4%	5%	100%	95%
mstdn.jp	×	✓	60d	839,146	5	3%	1%	10%	10%	8%	1%	5%	7%	5%	100%	30%
mstdn.social	✓	✓	61d	744,583	32	14%	37%	11%	20%	23%	1%	2%	19%	6%	99%	63%
live-theater.net	×	×	60d	699,158	12	0%	0%	6%	11%	6%	2%	2%	0%	16%	100%	96%
sportsbots.xyz	×	×	61d	667,792	13	8%	0%	58%	27%	22%	0%	0%	100%	0%	100%	100%
rss-parrot.net	×	×	46d	628,638	1	0%	0%	0%	0%	35%	0%	0%	100%	0%	100%	0%
fedibird.com	✓	✓	61d	605,355	18	6%	1%	9%	14%	14%	1%	2%	4%	18%	88%	35%
mastodon.online	✓	✓	61d	474,488	40	15%	27%	16%	33%	38%	1%	8%	26%	5%	98%	73%
misskey-square.net	×	×	61d	427,559	5	1%	0%	8%	17%	10%	4%	5%	0%	40%	100%	97%
16,655 sources	6%	—	61d	35,300,568	21	13%	15%	19%	21%	24%	2%	4%	18%	12%	96%	61%

Table 4
Top-10 hashtags (case-sensitive), languages (ISO 639), applications (name and OS: web (), Android (), iOS (), and API ()), and authoring user accounts from posts across all nodes. Unique counts via HyperLogLog++ [3].

Hashtag	Uniq. posts	
#news	254,354	0.7%
#press	235,316	0.7%
#News	202,197	0.6%
#nowplaying	146,348	0.4%
#nsfw	96,020	0.3%
#bot	76,058	0.2%
#ukraine	61,792	0.2%
#photography	55,239	0.2%
#music	54,225	0.2%
#art	41,924	0.1%

Lang.	Unique posts	
en	12,315,678	35%
ja	7,802,875	22%
de	1,731,484	5%
zh	596,070	2%
fr	582,844	2%
es	533,447	2%
nl	265,121	1%
zh-CN	237,492	1%
pt	185,635	1%
it	161,729	0%

Application	OS	Unique posts	
Web		2,317,376	6.6%
Mastodon		454,959	1.3%
Tusky		381,749	1.1%
Mastodon		373,691	1.1%
dlvr.it		365,610	1.0%
iembot		310,318	0.9%
Jetpack		276,705	0.8%
RSS bot		202,894	0.6%
Ivory		167,653	0.5%
CheapBots...		160,016	0.5%

Account	Uniq.
my24group	97,831
europesays	43,603
g1_globo	39,055
rawchili	37,466
prtimes	37,018
htTweets	35,133
rogue_corq	31,782
usluck	28,867
realTuckFrumper	27,057
dnc	24,367

Furthermore, 13% of the posts in our collection are replies, and 15% are boosts. About a fifth of the posts contain media attachments or hashtags, respectively. A quarter of the posts link to external content, while only a few posts contain spoilers (2%) or sensitive content (4%). Bot accounts contributed a fifth of all posts in our document collection. Most users did not opt-out from being indexed, and 61% of the posts were authored by accounts that explicitly opted in to search and discovery services. Locked accounts (i.e., follow requests manually approved) are most popular on non-Mastodon nodes (an extreme case being misskey-square.net; 40%), while sensitive content warnings are more popular on Mastodon nodes. The largest contributor to our document collection, mastodon.social closely approximates the post characteristics observed for the whole collection; the most notable differences being a higher share of posts with linked content (37% vs. 24%) and a lower share of posts from locked accounts (6% vs. 12%).

Frequent Hashtags, Languages, Accounts, and Apps Finally, Table 4 shows the most frequently used hashtags, languages, and applications, as well as the most active accounts in the document collection. The top hashtags are the typical tags for news (e.g., #news, #press) and hobbies (e.g., #nowplaying, #photography) besides some irregularities: The #nsfw hashtag (commonly used to indicate sensitive content) is relatively popular; it also seems to be somewhat common to mark bot posts with a hashtag (i.e., #bot); and compared to general news hashtags, the Russian invasion of Ukraine (i.e., #ukraine) is a dominant news topic. The language distribution is relatively diverse, with only about a third of the posts

tagged as English, closely followed by Japanese (22%). Generally, the Fediverse seems to be popular in Japan, as also five of the top-10 nodes are Japanese. The remaining top-10 languages are European languages and Chinese.[17] The users in our document collection use a large variety of applications to create posts. While many posts were authored on the Web (7%) or various Android or iOS apps, a large proportion of the posts also report which bot was used, which might be a useful ranking feature for search. Regarding user activity, we find that some accounts post excessively and the most "active" users contributed tens of thousands of posts to our collection. For example, the account "my24group" posts more than once every minute, on average. The top-7 most-posting accounts and "realTuckFrumper" are news bots; the remaining two are bots posting memes ("usluck") or recipes ("dnc").

4. Perspectives and Limitations for Search on Mastodon

Mastodon is highly similar to other microblogging platforms regarding content, interactions, metadata, and information needs but there are several unique constraints. We particularly look at how centrality and politeness influence task design and evaluation, how visibility and content policy influence the creation of collections, and how the federation influences the interaction features.

4.1. Centrality

Each Mastodon node has two server-wide timelines: (1) the local timeline (equivalent to ActivityPub's outbox) contains all public posts created and reblogged by the node's accounts and (2) the federated timeline (equivalent to ActivityPub's outbox and inbox combined) contains all public posts created or reblogged by the node's accounts and anyone they follow. As shown in Table 1, the federated timelines of nodes with high centrality (i.e. many accounts) will capture a large part of the complete network traffic. In our sample, for example, the largest node received 44% of the total posts and the ten next largest nodes received 19–28%.

For a search task, this centrality effect has two implications. First, selecting the federated timeline of the largest or most central nodes is likely very efficient and effective. It may be Pareto optimal for small nodes to forward all search requests to the largest known node, which is not polite (see Section 4.2) and goes against the idea of a federated social network. Additionally, for users on very large nodes, it is likely efficient and effective to only search the node's federated timeline, which would introduce a bias: it excludes small and isolated communities (which might have relevant expertise on a topic) and risks creating echo chambers. Hence, the diversity and specificity (i.e. selecting small but specialized nodes over large ones) should be considered in the task design and evaluation, either by penalizing the usage of federated timelines or by penalizing the reuse of large nodes for every topic.

Second, searching the federated instead of the local timeline will be more effective, since there are more documents in the index, but it will also be less effective since the index will be larger and there will be (many) duplicates across federated timelines which will have to be removed. This difference should be considered when evaluating efficiency.

4.2. Politeness

The Fediverse is a large federated network with well over 10,000 nodes that could be queried. This means that both the network and per-node load would be extreme even if just 1% of the nodes are selected for each query. Minimizing the number of selected resources is especially important since most nodes may not be able to afford a steep increase in traffic or compute cost. This implies that a certain politeness is required and the evaluation must consider computational and network efficiency. This could be as simple as applying a penalty function on the number of selected resources instead of just scoring the ranking in a resource selection task. An alternative design could also consider the Pareto optima between effectiveness and efficiency.

[17] News reports suggest that Chinese users move to Mastodon to avoid censoring and punishment [32, 33].

However, rating the politeness via the number of selected resources more strongly promotes the problems of centrality (see Section 4.1), for example by only querying the one largest node. That means the efficiency measures need to penalize the (ab)use of few central nodes.

4.3. Visibility and Consent

Two of the stand-alone features of Mastodon are post-visibility control and consent to process. Visibility can be controlled on 4 levels: public (visible to everyone), unlisted (visible to followers or via direct link), followers only, and direct messages. It is paramount to collect and index only public posts, which is the default for the public (not authenticated) API we used to create our collection.

Visibility is enforced through the public key authentication integrated into all Mastodon nodes. Since the origin nodes decide to which inbox any outgoing messages are going to be delivered, this authentication also controls visibility in case nodes de-federate or block certain actors. This is why a central search is unwelcome by many users: it would counteract these self-protection mechanisms by circumventing the node's authentication.

Consent to process is expressed through two opt-in features that are included in every post's ActivityPub message: discoverable and indexable (since Version 4.2). Discoverable indicates if the account can appear in discovery services like recommenders or user search. Indexable indicates if a post can be indexed and searched for. In our document collection, ca. 49% of posts did not opt-in to search and were not indexed and analyzed. An additional 35% of posts originated from non-Mastodon software which has no `noindex` flag and which we included in our index for quantitative analysis.

4.4. Content Policy and Moderation

Mastodon timelines contain posts from different Mastodon nodes, but also from other Fediverse software (like Misskey, Lemmy, or Pixelfeed) that federate via ActivityPub (cf. Table 3). Since these are independently operated, the allowed content differs between nodes in, for example, maximum text length, desired topics, disallowed content or topics, or mandate for content warnings and alt-text. In addition, all moderation efforts are up to the node and bad actors may exist.

For a search task, the content policy has two implications. First, some nodes may more often produce relevant content because they allow longer posts, which makes those nodes more attractive for a retrieval system with detrimental effects on diversity (see Section 4.1) and politeness (see Section 4.2). Second, the search can not rely on the nodes for moderation. That means it can not ignore blocked or de-federated nodes or retrieve harmful content even though it exists in the sources. However, the harmful content can be removed from the document collections to separate moderation and search.

4.5. Interactions

Interaction information, replies, boosts (reposts), and favorites (likes), are essential features for microblog search that behave differently on Mastodon [34]. The most obvious is that favorites are not federated and each node keeps individual counters, except that every favorite is announced to the origin node. Similarly, boosts are only announced to the origin and the followers of the booster, so an node's boost counter equates to the number of times a post has entered the node's federated timeline. The propagation of replies is more reliable, although it is not guaranteed that all replies in a tree are the same for every node (refer to Jambor [34] for a precise explanation).

These differences mean that posts from the origin node will have higher interaction counts than the same posts from the federated timeline, especially those of smaller nodes. Our collection does not contain any interaction data since we crawled new posts from the streaming API.

5. Conclusion

We have presented a new collection of microblog posts from the federated social media platform Mastodon to be used in information retrieval research. The collection contains about 733 million public posts from the federated timelines of 1,015 diverse Mastodon nodes across 61 days. We offer access to the document collection in a privacy-preserving manner via TIREx and we provide our parallelized and polite Mastodon crawler as part of the code of this publication.

Analyzing the collection with respect to search on Mastodon, we identified several challenges for retrieval systems. First, the centrality of nodes will have a large effect on retrieval systems and their evaluation regarding effectiveness, efficiency, resource selection, and politeness. Second, visibility and consent is less of an issue than we previously assumed: 61% of the public, federated posts are opted into search, which already are over 10 million unique posts per month. Third, although microblogs on the Fediverse are structurally similar to those of well-researched sites, the content differs in length (even between nodes) and interaction statistics are unreliable.

There are two notable limitations to our collection. First, most post interaction statistics are missing since we collect the posts, usually, directly after they have been created. We might later add those statistics. Second, our document collection only consists of posts from the timelines of Mastodon nodes as, to our knowledge, no other Fediverse software implements a confirmation mechanism for search. If these mechanisms become available, we plan to extend our document collection with generic ActivityPub-compliant software and networks.[18]

Acknowledgments

Partially supported by the European Union's Horizon Europe research and innovation programme under grant agreement No 101070014 (OpenWebSearch.eu).

References

[1] C. Lemmer-Webber, J. Tallon, E. Shepherd, A. Guy, E. Prodromou, ActivityPub, W3C Recommendation, W3C, 2018. URL: https://w3.org/TR/2018/REC-activitypub-20180123/.

[2] C. Doctorow, As platforms decay, let's put users first, 2023. URL: https://eff.org/deeplinks/2023/04/platforms-decay-lets-put-users-first.

[3] S. Heule, M. Nunkesser, A. Hall, HyperLogLog in practice: Algorithmic engineering of a state of the art cardinality estimation algorithm, in: Proceedings of EDBT/ICDT 2013, ACM, 2013, pp. 683–692. doi:10.1145/2452376.2452456.

[4] M. Fröbe, J. Reimer, S. MacAvaney, N. Deckers, S. Reich, J. Bevendorff, B. Stein, M. Hagen, M. Potthast, The information retrieval experiment platform, in: Proceedings of SIGIR 2023, ACM, 2023, pp. 2826–2836. doi:10.1145/3539618.3591888.

[5] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous integration for reproducible shared tasks with TIRA.io, in: Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), LNCS, Springer, 2023, pp. 236–241. doi:10.1007/978-3-031-28241-6_20.

[6] M. Zignani, S. Gaito, G. P. Rossi, Follow the "Mastodon": Structure and evolution of a decentralized online social network, in: Proceedings of AAAI 2018, volume 12, AAAI, 2018, pp. 541–550. doi:10.1609/icwsm.v12i1.14988.

[7] D. Zulli, M. Liu, R. W. Gehl, Rethinking the "social" in "social media": Insights into topology, abstraction, and scale on the Mastodon social network, New Media Soc. 22 (2020). doi:10.1177/1461444820912533.

[18] For example, Meta considers connecting their Threads platform to ActivityPub, while projects like fed.brid.gy attempt to bridge ActivityPub with Nostr and Bluesky's AT Protocol.

[8] L. La Cava, S. Greco, A. Tagarelli, Information consumption and boundary spanning in decentralized online social networks: The case of Mastodon users, Online Social Networks and Media 30 (2022). doi:10.1016/j.osnem.2022.100220.
[9] A. Z. Rozenshtein, Moderating the Fediverse: Content moderation on distributed social media, J. Free Speech L. 3 (2023) 217–235. URL: https://heinonline.org/HOL/P?h=hein.journals/jfspl3&i=217.
[10] R. W. Gehl, D. Zulli, The digital covenant: Non-centralized platform governance on the Mastodon social network, Information, Communication & Society 26 (2023) 3275–3291. doi:10.1080/1369118X.2022.2147400.
[11] L. La Cava, L. M. Aiello, A. Tagarelli, Get out of the nest! drivers of social influence in the #TwitterMigration to Mastodon, arXiv 2305.19056, 2023. doi:10.48550/arXiv.2305.19056.
[12] U. Jeong, P. Sheth, A. Tahir, F. Alatawi, H. R. Bernard, H. Liu, Exploring platform migration patterns between Twitter and Mastodon: A user behavior study, arXiv 2305.09196, 2023. doi:10.48550/arXiv.2305.09196.
[13] J. He, H. B. Zia, I. Castro, A. Raman, N. Sastry, G. Tyson, Flocking to Mastodon: Tracking the great Twitter migration, in: Proceedings of IMC 2023, ACM, 2023, pp. 111–123. doi:10.1145/3618257.3624819.
[14] J. Trienes, A. T. Cano, D. Hiemstra, Recommending users: Whom to follow on federated social networks, arXiv 1811.09292, 2018. doi:10.48550/arXiv.1811.09292.
[15] Statement of removal. Mastodon content warnings: Inappropriate contents in a microblogging platform, in: Proceedings of ICWSM 2022, volume 13, 2022. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/22003.
[16] E. Rochko, Cage the Mastodon: An overview of features for dealing with abuse and harassment, 2018. URL: https://blog.joinmastodon.org/2018/07/cage-the-mastodon/.
[17] E. Rochko, Mastodon 4.2: A new search experience and more!, 2023. URL: https://blog.joinmastodon.org/2023/09/mastodon-4.2.
[18] I. Ounis, C. Macdonald, J. Lin, I. Soboroff, Overview of the TREC 2011 microblog track, in: Proceedings of TREC 2011, volume 500-296 of *NIST Special Publication*, NIST, 2011. URL: https://trec.nist.gov/pubs/trec20/papers/MICROBLOG.OVERVIEW.pdf.
[19] I. Soboroff, I. Ounis, C. Macdonald, J. Lin, Overview of the TREC-2012 microblog track, in: Proceedings of TREC 2012, volume 500-298 of *NIST Special Publication*, NIST, 2012, p. 7. URL: https://trec.nist.gov/pubs/trec21/papers/MICROBLOG12OVERVIEW.pdf.
[20] J. Lin, M. Efron, Overview of the TREC-2013 microblog track, in: Proceedings of TREC 2013, volume 500-302 of *NIST Special Publication*, NIST, 2013. URL: https://trec.nist.gov/pubs/trec22/papers/MB.OVERVIEW.pdf.
[21] J. Lin, Y. Wang, M. Efron, G. Sherman, Overview of the TREC-2014 microblog track, in: Proceedings of TREC 2014, volume 500-308 of *NIST Special Publication*, NIST, 2014. URL: https://trec.nist.gov/pubs/trec23/papers/overview-microblog.pdf.
[22] J. Lin, M. Efron, G. Sherman, Y. Wang, E. M. Voorhees, Overview of the TREC-2015 microblog track, in: Proceedings of TREC 2015, volume 500-319 of *NIST Special Publication*, NIST, 2015. URL: https://trec.nist.gov/pubs/trec24/papers/Overview-MB.pdf.
[23] J. Lin, A. Roegiest, L. Tan, R. McCreadie, E. M. Voorhees, F. Diaz, Overview of the TREC 2016 real-time summarization track, in: Proceedings of TREC 2016, volume 500-321 of *NIST Special Publication*, NIST, 2016. URL: https://trec.nist.gov/pubs/trec25/papers/Overview-RT.pdf.
[24] J. Lin, S. Mohammed, R. Sequiera, L. Tan, N. Ghelani, M. Abualsaud, R. McCreadie, D. Milajevs, E. M. Voorhees, Overview of the TREC 2017 real-time summarization track, in: Proceedings of TREC 2017, volume 500-324 of *NIST Special Publication*, NIST, 2017. URL: https://trec.nist.gov/pubs/trec26/papers/Overview-RT.pdf.
[25] R. Sequiera, L. Tan, J. Lin, Overview of the TREC 2018 real-time summarization track, in: Proceedings of TREC 2018, volume 500-331 of *NIST Special Publication*, NIST, 2018. URL: https://trec.nist.gov/pubs/trec27/papers/Overview-RTS.pdf.
[26] J. Rao, W. Yang, Y. Zhang, F. Türe, J. Lin, Multi-perspective relevance matching with hierarchical ConvNets for social media search (2019) 232–240. doi:10.1609/AAAI.V33I01.3301232.

[27] W. Yang, H. Zhang, J. J. Lin, Simple applications of BERT for ad hoc document retrieval, arXiv 1903.10972, 2019. doi:10.48550/arXiv.1903.10972.
[28] A. Dusart, K. Pinel-Sauvagnat, G. Hubert, TSSuBERT: Tweet stream summarization using BERT, arXiv 2106.08770, 2021. doi:10.48550/arXiv.2106.08770.
[29] A. Garba, S. Wu, S. Khalid, Federated search techniques: An overview of the trends and state of the art, Knowl. Inf. Syst. 65 (2023) 5065–5095. doi:10.1007/s10115-023-01922-6.
[30] T. Demeester, D. Trieschnigg, D. Nguyen, D. Hiemstra, Overview of the TREC 2013 federated web search track, in: Proceedings of TREC 2013, volume 500-302 of *NIST Special Publication*, NIST, 2013. URL: https://trec.nist.gov/pubs/trec22/papers/FEDERATED.OVERVIEW.pdf.
[31] T. Demeester, D. Trieschnigg, D. Nguyen, D. Hiemstra, K. Zhou, Overview of the TREC 2014 federated web search track, in: Proceedings of TREC 2014, volume 500-308 of *NIST Special Publication*, NIST, 2014. URL: https://trec.nist.gov/pubs/trec23/papers/overview-federated.pdf.
[32] M. Haldane, Chinese social media users are flocking to the decentralised Mastodon platform to find community amid crackdown at home, 2022. URL: https://sc.mp/mjx8.
[33] D. Maung, Social media platform Mastodon gains thousands of new Chinese users amidst Beijing's security pressures, 2022. URL: https://visiontimes.com/2022/09/21/social-media-platform-mastodon-gains-thousands-of-new-chinese-users-amidst-beijings-security-pressures.html.
[34] S. Jambor, Understanding ActivityPub part 3: The state of Mastodon, 2022. URL: https://seb.jambor.dev/posts/understanding-activitypub-part-3-the-state-of-mastodon/.