Published in Distrib. Parallel Databases (2018) 36:47–79
(https://doi.org/10.1007/s10619-017-7211-3)

Web-scale Provenance Reconstruction of Implicit Information Diffusion on Social Media

Io Taxidou · Sven Lieber · Peter M. Fischer · Tom De Nies · Ruben Verborgh

Abstract Fast, massive, and viral data diffused on social media affects a large share of the online population, and thus, the (prospective) *information diffusion* mechanisms behind it are of great interest to researchers. The (retrospective) *provenance* of such data is equally important because it contributes to the understanding of the relevance and trustworthiness of the information. Furthermore, computing provenance in a timely way is crucial for particular use cases and practitioners, such as online journalists that promptly need to assess specific pieces of information. Social media currently provide insufficient mechanisms for provenance tracking, publication and generation, while state-of-the-art on social media research focuses mainly on *explicit* diffusion mechanisms (like retweets in Twitter or reshares in Facebook).The *implicit* diffusion mechanisms remain understudied due to the difficulties of being captured and properly understood. From a technical side, the state of the art for provenance reconstruction evaluates small datasets after the fact, sidestepping requirements for scale and speed of current social media data.

In this paper, we investigate the mechanisms of implicit information diffusion by computing its fine-grained provenance. We prove that explicit mechanisms are insufficient to capture influence and our analysis unravels a significant part of implicit interactions and influence in social media. Our approach works incrementally and can be scaled up to cover a truly Web-scale scenario like major events. We can process datasets consisting of up to several millions of messages on a single machine at rates that cover bursty behaviour, without compromising result quality.

Io Taxidou, Peter M. Fischer
University of Freiburg, Georges Köhler Allee 51, 79110, Freiburg, Germany
Tel.: +497618125
E-mail: taxidou@informatik.uni-freiburg.de

Sven Lieber, Tom De Nies, Ruben Verborgh
Ghent University – imec
IDLab
Technologiepark Zwijnaarde 19, AA-tower, B-9052 Ghent, Belgium

By doing that, we provide to online journalists and social media users in general, fine grained provenance reconstruction which sheds lights on implicit interactions not captured by social media providers. These results are provided in an online fashion which also allows for fast relevance and trustworthiness assessment.

Keywords provenance · information diffusion · incremental clustering · social media · influence

1 Introduction

Social networks, micro-messaging services, or sharing sites (e.g., Facebook, Twitter, or Instagram) provide the virtual space in which a significant part of social interactions take place. Many real-life situations, such as elections [28] or natural disasters [32], are reflected by these social media. In turn, social media shape these situations by forming opinions and strengthening trends, or by spreading news on emerging situations faster than conventional media. Most importantly, social media provide a huge audience where information can be easily spread and consumed by others. In this paper, we propose a scalable way to expose the origin – or *provenance* – of such information.

Provenance is a significant aspect to consider when judging the *relevance* and *trustworthiness* of information. On social media, tracing the provenance of information in a timely way is as challenging as it is crucial, given the information speed, the large audiences it reaches, and the multiple sources that might have produced it: anybody can write anything, without it being verified. The knowledge of information diffusion processes – including the sources, the intermediate forwarders, and the modifications that this piece of information has undergone on the way – provides valuable context to assess its relevance, validity, and trust.

For example, online journalists need to analyze the flow of information in a prompt way, by assessing the sources along with intermediate forwarders and determining the impact of their own publications. The detection of rumors and false information is crucial for them, since the propagation of trustworthy information in a timely way determines their own value as professionals. Rumors are not detected accurately by only identifying the sources, but also by analyzing the properties of the diffusion process, including the intermediate steps and any information modifications [25].

In addition to journalists, online users in general will benefit from such analysis with regard to *fake news*. Fake news aim at deliberately distorting the truth in order to mislead the public opinion and gain benefits. By providing the sources and intermediate steps, we facilitate the understanding of how a message reached its current state and by whom it has been propagated and modified. This knowledge will help in the prompt decision whether a piece of information is truthful or not. For example, if multiple official sources published such piece of information and any modifications on the way have not affected the content, then it is highly likely that it is credible. In general, identifying the provenance of information is a key aspect for judging its truthfulness [21].

Furthermore, the rapid spread of information on social media is often exploited in order to propagate negative opinions. The reputation of companies, politicians or celebrities is harmed by such negative information or rumors, which might be

consumed and re-shared by millions. Consequently, they need to react promptly by understanding who is propagating certain information and who is influencing others. As researchers, we thus need to view the information flow in both a prospective way (i.e., its diffusion: *where will it end up?*) as well as in a retrospective way (i.e., its provenance: *where does it come from?*).

In contrast to typical information diffusion, provenance has received limited attention in the context of social media [18]. Likewise, existing models of information diffusion are insufficient to capture provenance [39], while social media offer limited or no mechanisms to its users to judge the received information [6]. For example, in case of retweets on Twitter, the source of information is provided in the metadata but not the intermediate forwarders. However, it has been shown that forwarders play an equally important role in the outcome of information diffusion [5] and, as a result, in the reverse process of provenance. In the cases of quotes and replies only the previous step is given, but not the source of diffusion. There are also physical limitations to the extent social media providers can support provenance. When mechanisms of social media (like *retweet*, *quote* or *reply*) are used, some basic provenance is provided and further analysis can be implemented to identify possible sources and previous steps. However, we have previously shown that users might sidestep such mechanisms, and use their own conventions for crediting the sources [40]. To complicate further matters, users might provide no provenance information, e.g., by copying messages, which renders any further analysis challenging.

The speed and scale at which messages propagate through social media imposes further challenges in the process of identifying provenance. In recent work [14, 40], we investigated message similarity as a means of determining possible provenance. While repeatedly shown to be conceptually sound [13–15, 40], our initially proposed provenance reconstruction algorithm leads to a quadratic time and space complexity in terms of messages to consider. The scale and speed of contemporary social media renders those methods inapplicable for real-life scenarios. In this paper, we propose an approach to solve this challenging scaling problem.

The contributions of this work are the following:

(i) We implement methods to unravel and evaluate the fine-grained provenance of information in social media, especially when no information from social media providers is given. By doing so, we provide a deep analysis over human behaviour patterns in information propagation, validating our previous ideas [40]. We show that by considering only explicit means (from social media providers) a large share of implicit interactions and influence is ignored.

(ii) We provide web-scale, incremental provenance reconstruction of information diffusion to accommodate streaming use cases, such as fast and accurate online journalism. We identify the trade-offs of performance and result quality, and leverage assumptions including the limited user attention span to limit the space complexity. Our results show that it is possible to reconstruct provenance of messages not captured by social media providers at scale and speed without compromising in result quality.

2 Methodology

We describe our methodology in three parts: first, we explain the concepts behind implicit interactions on social media in Section 2.1. Second, we describe our approach for similarity-based provenance reconstruction in Section 2.2. Last, the technical contributions follow that lead to the system's scale and speed follow in Section 3. Note that although we focus on Twitter here, our method can be applied to any type of text-based social and news media. We validate that by using a news dataset which differs significantly from Twitter data.

2.1 Provenance of Implicit Interactions

As discussed in Section 1, messages on social media might carry explicit information diffusion metadata as provenance, such as the known retweets or quotes. These types of *explicit interactions* have been studied in literature as well as by us, for example in [37], where information cascades are reconstructed, and in [39], where we introduced the PROV-SAID model to represent these cascades as interoperable provenance. Therefore, we focus on reconstructing *implicit interactions* in this paper, as they cannot be captured by social media providers, and thus tend to elude traditional methods.

As a starting point for implicit interactions, we shed light on information diffusion patterns and users' conventions of credit attribution. Table 1 summarizes the types of influence indicators that we leverage for reconstructing provenance, originally introduced in [40]. Note that similarity between messages is a prerequisite for computing these indicators, as further explained in Section 2.2.

User influence is expressed by explicitly mentioning another user (*explicit credit*) or identified by the existence of a social graph connection since users are exposed to the messages of their friends (*without credit*). However, expanding on [40], we identify and evaluate a new, additional type of user influence here: i.e., *mentioning* a user in order to expose some information to them. We assume that such information is relevant for the mentioned user, and there is a high probability that the latter was influenced and propagated it, thereby further decreasing the uncertainty of the suspected provenance.

In addition to user influence, there might also exist an external event which drives the high similarity among certain messages, without the existence of any additional provenance indicators. In this case, there is no user influence but *external influence*, which is hard to capture without any event identification analysis. Note that there is a grey area of *external influence* and *user influence without credit*, where it is unclear whether users are influenced by their connections or by an external event. Likewise, users might propagate messages without any obvious connection with the original authors, or any other indicator. In Section 4, we show that users are indeed mentioning others, without having an explicit connection. It is possible that further research into event identification sheds light on *external influence*, but this is out of scope for this work.

Lastly, we observed that in order to re-expose their audience to their own content, prolific users often promote it again (*self-influence/promotion*). Additionally, certain

social media such as Twitter lack an *edit* message function, leaving users with no other option than *deleting and rewriting* their messages when making corrections.

User Influence		
with explicit credit	*without credit*	*by mention*
manual mention of user	no user mentioned but social graph connection exists	expose information and user reacts

External Influence	**Self-Influence**	
	delete and rewrite	*promotion*
no user mentioned and no social graph connection	oldest messages get deleted	none of the messages get deleted

Table 1: Types of Implicit Influence

To express these implicit interactions as interoperable provenance, we use the PROV-SAID model for provenance of information diffusion [39, 40], extending the W3C PROV Data Model [30]. The advantage of this model is that it allows a Web-native and interoperable format, which facilitates cases where data needs to be combined from different (social media) sources that do not share the same concepts and notations. In what follows, we revise our suggested provenance reconstruction flow [40] to account for these implicit interactions.

2.2 From Similarity to Provenance

As in [14], our main underlying hypothesis is that *"if two messages are highly similar, there is a high probability that they share some provenance"* (H1). More specifically: *"the higher the similarity between two messages, the lower the uncertainty of the provenance they share"* (H2). To test these hypotheses, we proposed a provenance reconstruction algorithm, as shown in Figure 1.

First, we determine a subset of messages which we want to consider in scope for our provenance reconstruction (e.g., the stream of messages for a conference). Then, we compute similarity between all the messages in this subset. To group similar messages – which we assume share some provenance (H1) – we rely on clustering. The procedure to cluster messages and reveal their provenance is the following. First we tokenize the text of the messages and we remove stop-words. We index the messages using a feature model (e.g., Tf-Idf by [33]) and semantic similarity function *Sim* (e.g., cosine similarity), and we compute the similarity matrix of all messages which are in scope. Next, we apply the similarity-based clustering algorithm SimClus [2] to divide the messages into (possibly overlapping) clusters of messages that share some similarity higher than a predetermined threshold T_S. This threshold is dependent on the content type, and should be empirically determined. We provide further discussion on selecting the lower bound of similarity in Section 4.1.

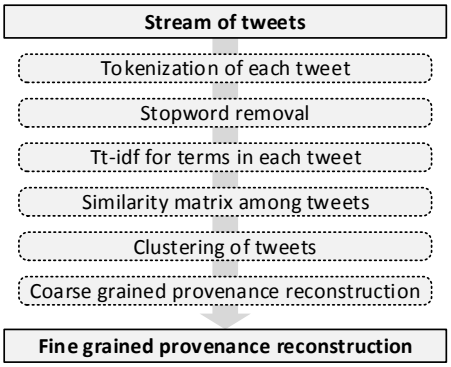


Fig. 1: Flow of Provenance Reconstruction

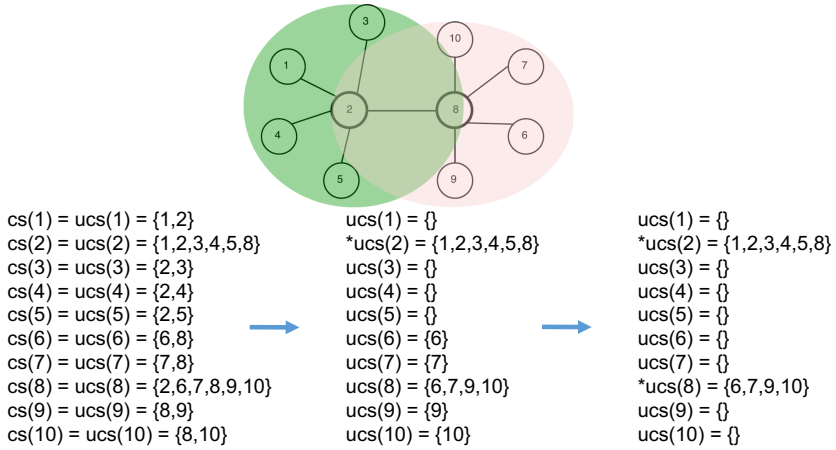


Fig. 2: SimClus Example

In short, the SimClus algorithm works as follows: first, the *coverset (cs)* is determined for each element of the dataset. The $cs(x)$ of a document x includes all documents y for which $Sim(x, y) \geq T_S$. The *uncovered coverset* $ucs(x)$ of a document x then includes all documents $\in cs(x)$ which are not part of a cluster yet.

Next, the document with the largest ucs is chosen as the first cluster center. All elements of its *coverset* are removed from the other *uncovered* coversets, and the center is marked to no longer be considered in further iterations. This process is repeated for the document with the next largest ucs that is not part of a cluster yet, until there are no more uncovered documents, which is when the algorithm terminates. For example, in Figure 2, the algorithm terminates after two iterations, when documents 2 and 8 are chosen as cluster centers.

The advantage of this algorithm is its computational efficiency, while it provides a natural way to cluster documents in overlapping clusters where the number of clusters should not be specified in advance.

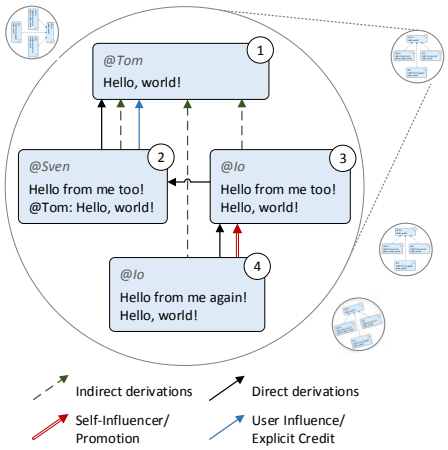


Fig. 3: Fine grained Provenance Reconstruction. Numbers in messages indicate temporal order. The *dashed arrows* in the figure identify *indirect derivations*, generated by connecting all messages to the oldest. The solid black arrows indicate *direct derivations*, created by maximizing the similarity in each cluster by identifying the most similar pairs of messages, while respecting the temporal order.

To reconstruct *coarse grained provenance* from this, we identify the oldest message for each cluster as the *root message* of that cluster. We assume that all other messages are derived from the root through an unknown number of n steps, in other words: that they are *indirectly derived* from the oldest message in the cluster.

Until now, the algorithm does not consider pairwise similarity among other messages in the same cluster, other than the root. Since our goal is to reconstruct *fine grained provenance*, we need to identify the most similar (and chronologically older) candidate for every message, which might not be the root. Therefore, we maximize the similarity in each cluster by connecting each message with its most similar and assume that the newest message is *derived directly* from the oldest through 1 step. By doing this, we decrease the uncertainty of the reconstructed provenance (H2). An example of the fine-grained provenance resulting from this process is shown in Figure 3.

However, as explained in Section 2.1, there might also exist particular indications, such as user conventions of assigning credit or connections on the social graph, that provide additional information and decrease the uncertainty of the reconstructed provenance even further. For example, the blue arrow in Figure 3 identifies a message that contains a mention (`@username`), and thus indicates *influence with explicit credit*. For these messages, we identify whether the mentioned user has emitted a similar message in the past by checking the direct derivations between these two users. If the mentioned user is emitting a similar message in the future then we observe *influence by mentioning*. Finally, the double red arrow indicates *self-influence/promotion*, and is generated by identifying pairs of messages among the direct derivations that share the same author. In case the oldest message is deleted then it is *self-influence/delete and re-write*.

2.3 Provenance on Social Media Streams

After we have described the background for provenance reconstruction, we proceed a step further and lay the foundations for provenance computation over streams of messages. The problem we tackle here is *similarity computation over infinite streams of data*.

Besides a massive volume (for which we will provide optimizations in Section 3), real-life social media exhibits a strong temporal component that needs to be addressed:

Most social media present information to their users in reverse chronological order, both activity of their connections and search results. This has also profound impact on the temporal scope of influence and thus provenance.

As [11] show for the space of explicit interactions, over 80% of replies and 60% of retweets are reactions to the 50 most recent tweets in a user's feed. For implicit interactions, this observation has been taken into account by [7]. For the identification of influence, this work considers the 100 last messages from friends presented on a users' timeline, since users generally have a limited attention span to react.

Likewise, complete information diffusion processes (as opposed to individual users) are typically modeled as processes with exponential decay [42], exhibiting a long tail of low activity, yet significant differences in overall duration, ranging from few minutes to weeks or months [38].

We form the following hypothesis to incorporate such a temporal component into our method: *"users are most likely to react on recently seen information or information that is getting enough additional support"*(H3). This includes older messages that have gone viral, attracting further attention. H3 entails that we can safely expire messages that are not being seen by users any more. In contrast to [2], instead of expiring individual documents, we expire clusters. The underlying assumption is that if a cluster (that corresponds to a topic) gets sufficient support, the oldest messages might become relevant again, because users might explicitly search for them. The recency of messages gets updated if they receive retweets (the time of the most recent retweet is considered), which means that such messages are getting further visibility. Keeping around the entire cluster contents ensures that the long tails of "older" processes get support, while the overlapping nature of SimClus ensures that such messages are not being consumed only by old clusters.

Yet, this stream clustering workload does not fit the bill well on existing stream clustering methods such as [24]: These methods leverage statistical summaries over the data, and provide a hierarchical decomposition into clusters. Their goal is to identify concept drift and adapt to the streaming data speed (leading to a less clear clustering result). For our requirements, we do not need to pay the overhead of computing concept drift, but rather need to carry all the active documents available for fine grained provenance computation. Additionally, a hierarchical clustering decomposition is not sufficient since we desire to identify meaningful clusters in a natural way that can reveal provenance information. Dynamic topic models such as dynamic LDA [8] provide such clusters, but are out of scope for this work, as their main focus is to identify topics and topic evolution by employing complex generative statistical model. Specifically, LDA clusters according to the latent representation of documents, while we consider the explicit Tf-Idf. Latent topics might help for the reconstruction

of fine-grained provenance but we desire not to add another layer of abstraction. We leave this for future work to identify whether LDA based clustering contribute to provenance identification. In general, our focus is not on topic detection, but a lightweight topic identification is a by-product of document clustering.

2.4 Use Case: Ranking of Influence

Lastly, we describe how provenance reconstruction can be used in practice. Such provenance information with additional indicators complements influence computation analysis, which was in most cases studied with explicit means (e.g. retweets, reshares, replies etc). Online journalists, scientists and online users in general are equipped with a powerful tool that unravels influence, beyond explicit interactions. Apart from the fine-grained provenance reconstruction in section 2.2, we provide ranked lists of influential messages, according to different influence metrics. We consider implicit, explicit means and their combinations. We also consider cumulative influence: for that, we compute influence paths by traversing the direct derivations from the leaves to the sources. We consider the following metrics of influence:

Retweets shows how many retweets a message has attracted.

Single-hop direct derivations shows how many messages were implicitly influenced.

Single-hop direct derivations with retweets adds implicit and explicit influence that a message evokes.

Multihop derivations shows the cumulative influence by traversing influence paths.

Multihop derivations with retweets accumulates implicit and explicit influence along influence paths.

3 Reconstruction on a Web-Scale

3.1 Overview

Existing related work like [14] and [7] reconstruct implicit provenance on a finite amount of data by accessing the available dataset in a blocking manner or in reverse temporal order. However, this is of limited use when working with web scale streaming data, since a massive amount of data arrives incrementally in a temporal order with high rates and without an obvious end. While the temporal nature and the decay on relevance dampen the size of the overall working set, long periods (ranging from hours to days) still need to be taken into consideration to find matching candidates (as our evaluation shows in Figure 10 for tweets and Figure 18 for news). As a result, even when constraining ourselves to specific user relationships and/or events, we end up with a working set consisting of several 10Ks for news media to millions of messages for social networks which gains and sheds hundreds or thousands of messages per second.

The overall pipeline outlined in Figure 1 refers to established and well-researched methods for each step, but combining and applying them on such a workload leads to the following challenges:

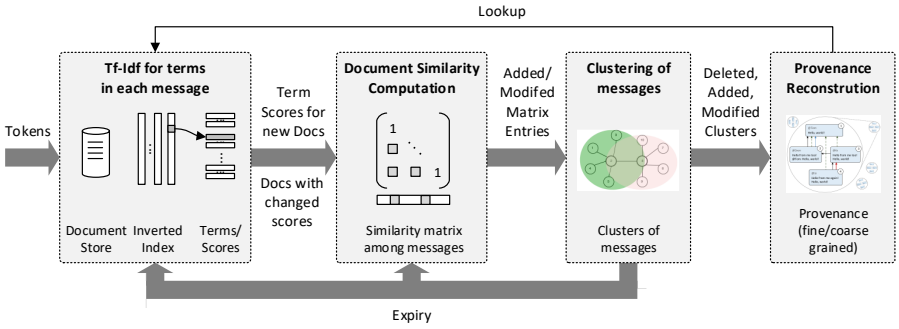


Fig. 4: Architecture to compute implicit provenance on web-scale data

- The Tf-Idf scores of terms used to determine similarity depend on frequency in the document set, yet this document set is constantly shifting due to arriving and expiring messages. Precisely reflecting these changes will lead to widespread score changes and thus permanent re-computations of the similarity matrix on existing values.
- Determining the similarity between all document pairs is a costly undertaking in terms of space and time. Even without score changes every new document needs to be compared against every existing document in the working set.
- Fully re-computing the results in clustering as well as the intra-cluster PROV derivation (as introduced in Section 2.2) is clearly infeasible at the requested speeds, considering that the computational cost of these methods are also clearly super-linear in size.

To tackle these challenges, we pursue a number of strategies that can be grouped in the following directions:

- Wherever possible, we use or develop incremental methods for the individual steps of the pipeline in order to re-use existing state, turning this into an architecture where every stage consumes and produces incremental results (Figure 4).
- We perform semantic optimizations by exploiting the properties of the "down-stream" algorithms and the workload to reduce the number of computations and the amount of storage needed.
- Similar to query processing of database systems, we heavily rely on *indexing* to limit the scope of documents to consider when performing operations as well as on early stopping once results could no longer be produced.

The combinations of these techniques enable us to advance significantly beyond the current state-of-the-art and allow for optimizations that speed up the computations by several orders of magnitude.

3.2 Architecture for Incremental Evaluation

As outlined above, a key element of a scalable solution on streaming data is the ability to perform incremental computations. We turn the conceptual pipeline outlined in

Figure 1 into a scalable architecture that is shown in Figure 4. Each stage corresponds to a pipeline step, yet refined to fit the requirements of infinite, massive-volume data streams. Instead of computing the full results at each stage and the passing them on, the computation is performed gradually on available data. Each stage stores a certain, limited amount of state, consumes the changes produced by the previous stage and produces new changes that are propagated to the next stage. This state is being pruned by expiring documents that are no longer covered by any cluster. In turn, documents are expired after a certain time has elapsed since last activity and thus cannot generate any provenance, as outlined in Section 2.3.

In more detail, this means that we submit batches of newly arrived messages to the system, which are turned in a set of tokens for each added message after stopword removal and, if applicable, stemming. These documents are then stored as long as needed, as some of their contents (including the author ids and the texts) are required to generate the provenance serialization. Furthermore, Tf-Idf scores are computed on the basis of these tokens. New documents as well as documents whose scores change, e.g., by shifts in the term distribution, are forwarded to the similarity matrix computation. In that stage these documents are compared to the set of active documents to compute the cosine similarity on the score-weighted word vectors. The result is a set of document pairs whose similarity values are new (either by addition or value change). For the similarity clustering algorithm, we picked up the extension for incremental updates working on this set of new and changed similarities described by [2]. Most of the conceptual contribution in this space lies in making the intra-cluster influence derivations also incremental. As a foundation, the clustering algorithm was extended to report changes by returning three sets after each computation. (1) The clusters which were updated, (2) the clusters which were added and (3) the clusters which were deleted. Newly added clusters are not only the result of new documents. The possible reshaping of the clustering (former centers become non centers and vice versa) also leads to the deletion and addition of new clusters. In case of addition or deletion of clusters, the incremental provenance algorithm simply computes the derivations of any new cluster; respectively deletes the derivations corresponding to a cluster which no longer exists. For the clusters which were updated, all the direct (single-step) derivations need to be recomputed as the similarity may change or a previous seen message now is more similar to a newly added document (message). If a message was added which is older than the other messages, all the indirect (n step) derivations need to be updated as now all messages within the cluster are derived indirectly by the newly added message.

Overall, this kind of architecture lends itself well to parallelization and distribution, even though we have not exploited this yet. One natural direction is pipeline parallelism, as each stage of our pipeline can be executed individually with little coordination due to the notion of propagated changes. Furthermore, most stages – with the exception of the clustering step – lend themselves well for data parallelism: tokenization, score computation, similarity matrix entry computation, and fine-grained provenance computation in clusters provide clear means for partitioning. Yet, this is a rather brute force approach and in this work we are looking into two means to increase the performance and scalability of this architecture on a more conceptual level: (1) Semantic optimizations to reduce both the amount of state in each stage as

well as changes propagated between the stages without impairing provenance results (2) Reducing computational cost at each of the stages using techniques from database implementation. We will now describe these optimizations in more detail.

3.3 Semantic optimizations to reduce changes and state

A key insight we gained when designing our system to compute provenance is that even when producing precise provenance not all data from the previous stages is needed. As a result, computation and storage in the earlier stages can be avoided, in particular for Tf-Idf score changes and similarity matrix entries.

In order to limit the impact of score/Idf changes due to newly arriving or expiring documents, we only propagate such changes when they exceed a certain threshold. This is driven by the insight that the utmost majority of these changes is small. When documents arrive or expire, updated Tf-Idf values ($tfidf_{current}$) are computed for all documents with the same terms as these documents, utilizing the inverted index. We then determine the amount of change from the value propagated before ($tfidf_{old}$) as follows:

$$\frac{|tfidf_{current} - tfidf_{old}|}{tfidf_{old}} \geq \epsilon$$

The values are only propagated once they exceed the change threshold ϵ. This approach provides several advantages: First and foremost, it guarantees a bounded error that will also not lead to unlimited error propagation. The trade-off on accuracy and cost is tunable, and yields significant benefits even at fairly conservative values, as the evaluation will show. Likewise, the resulting errors are very small even at aggressive settings. Finally, while computing all updated Tf-Idf values does not come for free, this cost is rather moderate and the computation could be easily parallelized.

Likewise, to overcome the prohibitive cost of fully computing and materializing the similarity matrix, our main observation is that the distribution of those values is strongly skewed towards low values, as many documents have very few terms in common. In turn, only similarity values above a certain threshold are actually needed for clustering and fine-grained provenance. After the obvious optimization to store only those values above the threshold to save space, the main pursuit is to avoid as many irrelevant computations as possible while not compromising the results. Significant benefits towards this step stem from the fact that we are dealing with very short documents, often containing less than ten terms after stop-word removal. When applying this technique on datasets with larger texts, the effects are less pronounced, as the distribution of values is more even. In particular, there are many data points that are actually zero.

3.4 Indexing and Early Stop to speed up computations

Considering the cost of score and similarity matrix computation, we are also employing techniques from database query processing, namely indexing and lazy computation of results, as outlined in Figure 5.

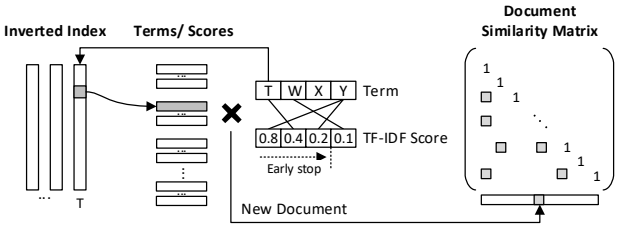


Fig. 5: Similarity Computation and Early Stop Optimizations

An important observation is that on only documents with common terms will produce non-zero similarities and changes in the Tf-Idf scores. While we were already saving the storage using the optimizations outlined before, these unnecessary values still need to be computed. Given the sparsity of term in a short-text workload, we expect only a small number of matches. Therefore we can effectively utilize the inverted index to retrieve the union of all documents that contain at least one common term (as otherwise the sum and thus the score would be 0). Instead of comparing this document against all documents in the working set, we only compare it against the members of this union which is typically much smaller for workloads with short texts.

Furthermore, we can also sort the terms by their score and exploit an early-stop approach to only consider those terms with high scores, since documents that have only low-score terms in common will not make it above the threshold. In addition, this will help in pruning out long word lists, as terms with low scores tend to be contained in more documents (due to their lower Idf). This is particularly useful to deal with frequent terms for which no stop-word filtering is effective, e.g., because they have periodic behavior.

To understand the gist of our optimizations, recall the definition of Cosine Similarity we are using to determine the similarity of two documents:

$$\frac{\sum_i^N a_i * b_i}{\sqrt{\sum_i^N (a_i)^2} * \sqrt{\sum_i^N (b_i)^2}}$$

where the non-zero parts of the sum in the numerator stem from common terms.

The idea rests on the possibility to estimate the expression $a_i * b_i$ (and also the denominator) from the information in A only, turning it into $(a_i)^2$ and $\sum_i^N (a_i)^2$, respectively. This means that the scores (and Euclidian norms in the denominator) should be roughly symmetric, which tends to be the case since the small number of terms per documents allows only for small variations in the Tf part. Lastly, a good stopping condition for term/score at *currentPosition* is

$$\frac{(a_i)^2 * (|terms| - currentPosition)}{\sum_i^N (a_i)^2} < stopThreshold$$

since earlier terms have already been covered and all later terms may at most have the same score as the current term. Choosing *stopThreshold* depends on the (a)symmetry in scores and Euclidian norms. For the short-text workloads, setting *stopThreshold* as

θ/k, where k is greater than 2.5 ensured that there were no differences in the results while achieving quite significant saving in cost. When dealing with longer texts, the long terms lists lead to values for *stopThreshold* that are closer to θ, and additionally a filter for the raw score values (e.g., 0.02) to clip of the long tail.

4 Evaluation

We evaluate our approach on three levels. In Section 4.1, we confirm the conceptual soundness of our provenance reconstruction method on a *small-scale* dataset gathered at the ISWC 2015 conference. Note that automated evaluation is very challenging – if not impossible – here, due to the non-existence of a ground truth. Instead, we evaluate the correctness of the provenance edges by a thorough manual investigation using qualitative methods. We study user interactions in detail, and compute influence according to different metrics from Section 2.4.

In Section 4.2, we confirm the method's quality and scalability by applying it to a *large-scale social data* dataset collected during the 2012 Olympics, measuring the run-times and consistency of the reconstruction results.

Lastly, in section 4.3, we validate the method on a dataset taken from a news feed, providing a significantly different text corpus with a large number of words/tokens per message. For each section, we will first determine the relevant parameters of the pipeline introduced and discussed in the previous sections:

- ϵ: degree of Tf-Idf deviation allowed before score change propagation;
- θ: minimum similarity value to materialize, needs to be at most as high as the clustering threshold;
- clustering threshold: lower bound of the cluster similarity;
- expiry: duration since the last activity of a cluster before discarding it.

4.1 Small-scale empirical evaluation

For the empirical evaluation, we used a controlled and complete dataset in order to compute fine-grained interactions. The dataset contains 3909 messages, consisting of 2068 retweets, 198 quotes, and 93 replies. By excluding messages that carry explicit provenance by Twitter, we end up with 1550 distinct messages.

We applied the pipeline outlined in Figure 1 on this data set, relying on non-incremental computation due to the small data size. Therefore, no values for ϵ, θ and expiry needed to be set; we will investigate them in more detail in Section 4.2 and Section 4.3. In order to find an appropriate clustering threshold for our dataset, we need to define an objective function which assesses the clustering result. Since SimClus allows the overlap of clusters, which is desirable in our set-up (messages might belong to more than one topic and might have multiple sources), we do not need to account for the distances among clusters, as other clustering quality metrics like the Silhouette [31]. We select the RMSSTD (root–mean–square standard deviation) index [23, 34] which measures the distance of data items within each cluster. RMSSTD produces values from $[0, 1]$, where values lower than 0.5 imply a good clustering result.

Figure 6 shows how the RMSSTD metric and the number of singleton/non-singleton clusters are influenced by different clustering thresholds (shown as numbers at the data points). We observe that clustering thresholds above 0.4 are acceptable, as the quality increases linearly with the increase in the clustering threshold. Furthermore, the number of non-singleton clusters decreases with higher thresholds/lower RMSSTD, making clustering thresholds above 0.8 unsuitable due to the sparsity of results. We quantify sparsity with the number of singletons. We also desire to have larger clusters (more than two messages), so that connections can be developed within each cluster. This also confirms our empirical observations for appropriate clustering thresholds in [13] and [40].

In general, there is not a clear lower bound for clustering. This figure is indicative of how the number of clusters and singletons behave for different clustering thresholds, so that each use case can leverage the "most fitting clustering threshold". This threshold depends from the amount of external influence that each dataset contains. The higher the external influence, the higher the threshold should be set in order to avoid the "overeager" similarity connections that the algorithm makes. Another significant factor to consider is the size of the dataset. Small datasets (in the range of thousands) with very high thresholds yield mostly singletons. On the contrary large datasets (in the range of millions) and low thresholds result in large clusters that are hard to evaluate. Empirically a 0.7 threshold is a good trade off for this particular dataset that has external but also internal influence.
In practice, different similarity thresholds do not affect the fine-grained provenance reconstruction (direct derivations) much. Every document will be connected with its most similar and in the majority of the cases the pairwise similarity within clusters is much higher than the similarity threshold. From the perspective of the light-weight topic identification, evaluating different similarity thresholds provides a hierarchical decomposition of topics. In reality, there is no "hard clustering": a corpus of documents can be described by general topics which in turn are divided in sub-topics. It is left to the individual application to decide for the desired level of topic granularity.

By applying the provenance algorithm with lower bound of similarity 0.7, we identified 67 clusters and 81 direct derivations. The cluster distribution is presented in Figure 7. We observe that the majority of clusters are singletons or clusters with two messages, given the rather high threshold. Despite possibly missing some lower similarity pairs of messages, we opted for this threshold to simplify the manual evaluation by reducing the number of clusters and edges to be inspected and filtering out possibly irrelevant edges. A quantitative evaluation of this dataset using a less restrictive clustering threshold 0.4 is described in [40] for implicit and explicit diffusion means and their interactions.

In order to verify the correctness and soundness of our approach, we followed two evaluation strategies. First, we manually inspect the results of the clustering algorithm and report observations for human interactions and influence. Such empirical observations shed light on the human behaviour patterns that are not captured by automated tools. Second, we compare the influence of messages with explicit and implicit means, which also provide indicators for the soundness of our approach.

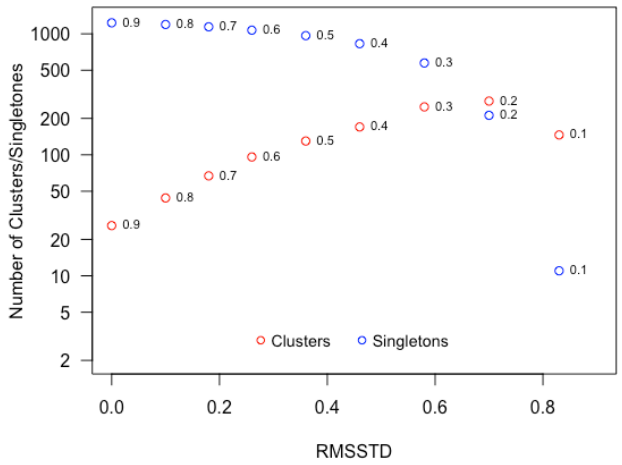


Fig. 6: Determining suitable clustering thresholds (at data points) via RMSSTD and cluster count for ISWC dataset

4.1.1 Empirical manual analysis and observations

We performed a manual, qualitative inspection of the 81 pairs of messages for which our algorithm indicated that they share some provenance. The goal was to understand the truthfulness of the provenance generated and creating a stand-in for actual ground truth. An interesting challenge in this analysis is the role of external influence, i.e., events affecting several users without traveling through the social network.

Several complementary means were employed to aid this analysis – some of which could eventually be turned into rules for automated, deep classification.

1. We considered explicit means of attribution, e.g. user mentions (see Section 2.1 and [40]) or keywords that show influence, e.g. “via”, “RT”.

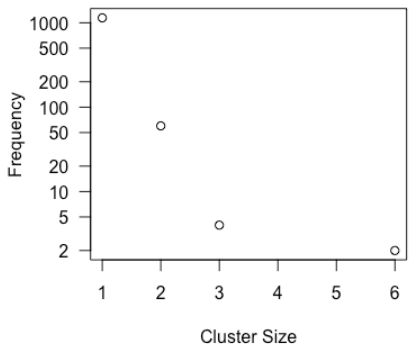


Fig. 7: Distributions of Cluster Sizes for ISWC dataset. The cluster size distribution has more than 1000 singletons, 60 clusters have 2 messages, 4 clusters have 3 messages, 2 clusters have 6 messages.

2. We looked at social connections among the authors - their absence might indicate external influence.
3. We also observed shared content in the messages, e.g., memes, URLs or photos. We identified that users might share the same URLs, but also might emit the same photo under a different URL. Unfortunately, the latter are indiscernible without image analysis algorithms, which are out of scope for this work.
4. Finally, we considered additional interactions among the authors, e.g. likes, retweets or replies that might strengthen our provenance assumptions (interaction-based observations). For those interaction-based indicators, the authors often imply influence and provenance by interacting additionally through explicit means. We observe messages sharing some provenance, while additionally, the influenced author retweets or likes the source message in order to highlight the influence. Additionally, the source author might identify their influence by interacting with the influenced author in similar explicit ways.

Those indicators are computed on top of the clustering algorithm to strengthen the reconstructed provenance. They are Twitter-specific in a sense that they support our observations mainly from Twitter. We desire that our algorithm is general enough so that it can be applied to any text-based social media, and for that the core of the algorithm remains as simple as possible. The indicators 1,2 that are applicable to other social media like mentions or the existence of social connections are already systematically computed. Indicator 3 is very hard to capture with rules and computationally expensive (image identification or URL dereferencing). Also the amount of messages that demonstrates such provenance is very low, which does not provide enough reasons to fully incorporate such provenance. Indicator 4 is bound to data restrictions. For example, likes in Twitter cannot be crawled (i.e., who liked which message). We identified those cases manually using the Twitter web interface. Furthermore, indicators like additional retweets or replies could be incorporated given

that those messages have been crawled. The systematic analysis of indicator 4 is left for future work.

In terms of truthfulness of provenance, these observations are certainly not perfect since they have been discovered by extensive observations and manual investigations, not formal grounding. Furthermore, they may not apply to other datasets. However, we have been observing some recurring patterns and user conventions and we assume that there must be a qualitative basis to claim that these indicators take place in social media. We also have observed that online users leverage multiple means to convey influence, not always explicit, like the older generations who still use mentioning instead of retweeting. In order to minimize any possible uncertainties over these indicators, we may crowdsource these tasks to a larger group of experts in the future. Alternatively, we can implement surveys where we ask the authors of messages, how they got influenced.

Considering explicit means, interactions and shared content, we could categorize 54 (out of 81) derivations that our algorithm connected as in-network provenance edges, and thus deem them correct. Out of these messages, 29 share the same author: 11 are promoting the same content and 18 show editing behavior. Already by looking only at the user information we can reason for 36% of the implicit interactions (direct derivations). 16 messages mention their influencer by giving some explicit credit to them. 7 messages include particular content or interaction based indicators, which is hard to model and automatically identify.

In terms of the social ties, besides the 29 derivations which are from the same author we observed the following: 4 pairs share information of their friends (users whom they follow) , while 13 pairs of messages demonstrate bidirectional relationships. 4 users are forwarding messages of their followers. This number ties with our observations that sometimes users identify their own influence on others by interacting with their messages. Lastly, 26 messages share no social relationship. These pairs of messages may be accounted to the external influence (the conference). For 5 pairs of messages the social graph for at least one of their authors could not be retrieved. In general, these results align with our hypothesis that users are influenced by information from their social connections to a considerable extent [37].

We classified 27 derivations as demonstrating external influence, meaning that no particular indicators were identified to strengthen the generated provenance. The presence of social links alone is not sufficient for a resolution, as the close-knit community of people in the ISWC conference knows each other well in real life, yet their behavior does not imply the usage of the social network in those cases.

Expressing external influence as provenance derivations is not “wrong” per se, as we still capture an apparent influence, and can express it as an “unidentified entity” [40]. Since our algorithm does not know the actual external events, it connects those messages with their most similar predecessor. In future work, we aim at identifying external events with a topic detection algorithm, create nodes for such events and connect them with those messages. However, such analysis is out of scope for this paper.

In summary, this manual analysis showed that no incorrect provenance edges were generated, demonstrating the high precision of our approach with this cluster threshold. Unfortunately, we cannot assess the recall without a ground truth, as it

would be infeasible to scale the manual analysis to include to all possible message combinations.

4.1.2 Influence ranking of implicit vs explicit interaction means

We also analyzed the soundness of our approach by contrasting the generated influence with the explicit means influence, e.g., retweets. In the absence of a ground truth, we rely on the retweet count as a prominent means of attributing influence [9]. In particular, we are interested in the most influential messages computed by both methods (implicit: our algorithm, explicit: retweets) and we would like to find out whether there are substantial differences in the influence inflicted by both methods.

We compare the ranking of the same documents according to different objectives: explicit and in implicit influence. We are interested in the relative order that each document has in both lists. For the ranked list with implicit means, we counted how many documents are influenced by each document by means of direct derivations. These documents are ordered and the top 10 most influential documents are selected. For those documents, we also retrieve their retweet counts. We used the rank-biased overlap (RBO) which computes how close two (possibly) infinite ranked lists are [41]. RBO is useful when comparing lists that do not contain the same elements and are not equally sized (disjointedness problem), which is not possible using metrics like Kendal Tau. In Section 4.1.3 we compare lists that might not contain the same elements. The result show that these two lists are indeed very close with an RBO value of 0.92, which confirms the suitability of our provenance reconstruction algorithm for relative influence computation. This also illustrates one of the possible in-use scenarios of our approach: identifying the most influential messages in a dataset.

4.1.3 Influence ranking of the combination of explicit and implicit means

Table 2: ISWC dataset Influence Rankings for top30

Comparison of provenance types	**RBO value**
single-hop (2) vs retweets (1)	0.20
single-hop (2) vs multi-hop (3)	0.87
single-hop (2) vs single-hop with retweets (4)	0.38
multi-hop (3) vs multi-hop with retweets (4)	0.48

After confirming that the relative influence computed by our algorithm complies with the "ground truth" influence given by Twitter (Section 4.1.2), we investigate the impact of our implicit influence results. Here, we evaluate the implicit influence computed by our method and we compare it with explicit influence, i.e. retweets. In order to do that, we compare the ranking of messages according to their *implicit* and *explicit* influence. In contrast to 4.1.3 where the two lists contained the same elements and the goal was to compare the same documents with different influence means, here focus on identifying the purely ordered lists according to different means.

If the rankings are very close, then it means that explicit means in isolation can capture the relative influence of messages, and implicit means can be disregarded. In contrast, if the correlation of ranked lists is weak, it means that implicit influence has a significant impact on the results, providing additional value over purely explicit influence rankings. Additionally, we want to investigate the impact of accumulating and combining influence with regard to implicit and explicit methods. That will help us to better understand the factors that need to be consider when calculating influence (implicit, explicit, accumulated).

We compare the lists of messages that are ranked according to: (1) explicit influence: number of messages influenced by retweeting, (2) implicit influence: number of messages influenced directly (single-hop direct derivations reconstructed within the clusters, which are formed by the clustering algorithm in 2), (3) cumulative implicit influence: number of messages influenced implicitly thought n hops (multi-hop). (4) implicit influence combined with explicit: the sum of single or multi-hop derivations with the number of retweets.

In order to compute multi-hop direct derivations, we need to construct the influence paths within the clusters. For the ISWC dataset, we observed paths up to length 3. The results are shown in Table 2: The first line concerns comparison of ranked lists of messages according to single-hop and retweets. The difference with the evaluation in Section 4.1.1 is that previously we considered the same messages ranked by two different means (single-hop derivations vs retweets). Now we consider the absolute ranking of top 30 messages according to different means. As the results show, the correlation of 0.2 is now quite weak. This means that computing influence solely relying on retweets lacks the impact of implicit diffusion.

In the second line, we observe that extending direct derivations to arbitrary many hops does not change the ranking significantly: messages that are ranked high for direct implicit influence (single-hop), have also similar ranking for cumulative influence (multi-hop). Note here that this relatively high correlation derives also from the small size of the dataset and the short influence paths. However, in the third and fourth lines, when implicit (both single-hop and multi-hop) and explicit influence are combined the shifts in rankings are significant, which further proves the merit of our contribution.

4.2 Large-scale Twitter Dataset Evaluation

For the large-scale Twitter data evaluation and also for the news data evaluation, we implemented the system described in the Section 3 in Java, utilizing an OpenJDK 8 64 bit JVM on an Amazon EC2 r4.2xlarge instance with a 2.3 Ghz Intel Xeon (Broadwell) CPU and 60 GB RAM. All steps of the pipeline were executed on a single thread. The initial stages of our pipeline build on well-known components: Tweets texts are split using the Twokenize library, stopwords are taken from the Python NLTK toolkit and amended with common Twitter expressions such as emoticons and a sample of frequent, domain-specific words taken from the first batch of messages in the dataset.

The dataset we are using was recorded during the 2012 summer Olympics in London using terms like “olympics”, “london2012”. For the evaluation, we used a

prefix of this data spanning four days (August 3 to 7th, 2012, afternoon to afternoon) and containing more than 4.6 million messages.

Given the large size of this dataset (3 orders of magnitude bigger than the previous), setting the parameters correctly for incremental processing is of great importance. We used batches of 2500 messages that we fed continuously into the system. Larger batch sizes generally would lead to somewhat higher throughput but also carry higher latency due to the queuing times when building the batches. For the clustering threshold, we selected a slightly higher similarity threshold (0.75) since this data set is much noisier than the previous, also due to the presence of spam. Additionally, there is much more external influence than in the ISWC dataset and we increase the threshold to remove those external influence edges.

With this setting, the RMSSTD value starts off at 0.12 in the first batch, then increases to around 0.18 within the next 250K messages and then stays almost stable (never exceeding 0.19) until the end of the data set (Figure 8). In turn, we set the value of θ to 0.65 so that there is enough headroom for provenance computation in clusters. For ϵ, we settled on a value 0.5. The impact of these values will be studied in more detail in Section 4.2.3. To capture the temporal dynamics and limit the growth of the working set, we enabled expiry with an activity timeout of 12 hours, i.e., a cluster not getting updates to any of its messages for 12 hours will be discarded. To determine an appropriate expiry time, we looked at times between the last and the second-last activity in a cluster, as this would determine if a cluster would have been discarded too early. The distributions of those values showed that doubling the expiry interval would cover about 10-15 percent more cluster "tails", up to a duration of 12 hours, leveling off after. 12 hours was also a convenient time to give room for the roughly 18-24 hours news cycle present in this data set (most events taking place in the afternoon and evening).

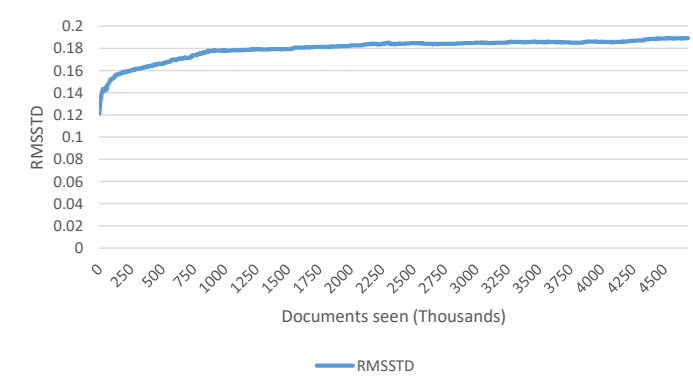


Fig. 8: Evolution of Clustering Quality for Olympics dataset

4.2.1 Provenance Analysis

Figure 9 and 10 shows the qualitative characteristics of the clusters produced with these settings.

Figure 9 shows the cluster distribution of the last batch. We can observe that the cluster size demonstrates a skewed distribution, with the largest cluster containing almost 21K messages. The median size is 15, which shows that there are many small clusters and a few large ones (corresponding to trending topics). The lifetimes in Figure 10 show a more balanced distribution, since small clusters with limited activity expire quickly. Values on the top 25th percentile are more spread, demonstrating a variety of longer lifetimes. The mean value is approximately 8.7 hours (median 8.2).

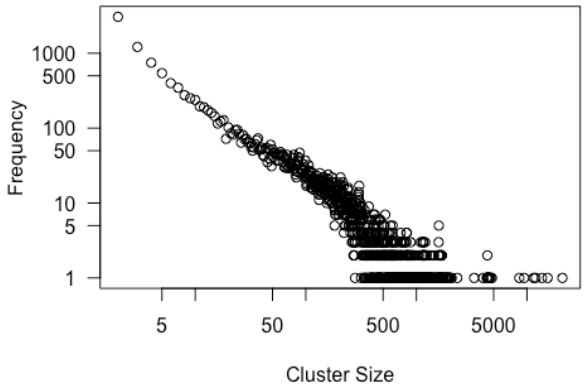


Fig. 9: Distributions of Cluster Sizes for Olympics dataset

The clusters show good coherence, which means that in most cases, the provenance structure will form a single tree; in rare cases, the similarity was too uneven, which will result in several disconnected provenance subtrees. The provenance structure will be quite complex, since we observed paths with up to 14 hops.

Analyzing the social connection on the derivations proved do be difficult for two reasons: In order to not slow down provenance generation, fast access to the full social graph is needed, requiring massive amounts of main memory. Even more limiting for our study was the completeness and quality of the social graph information we had collected for [37]: since its purpose was to support analysis of explicit interactions (retweets), it lacked the coverage of user involved in implicit interactions. As a result, we could not get conclusive and reliable results.

We did get reliable results on the rankings of influence, in order to verify and compare the results from Table 2, with an additional emphasis on how the rank changes over time. As shown in Figure 11a, we observe that the single-hop ranks are even less

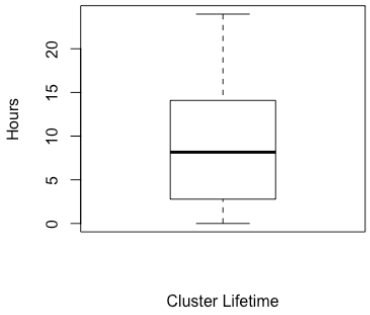
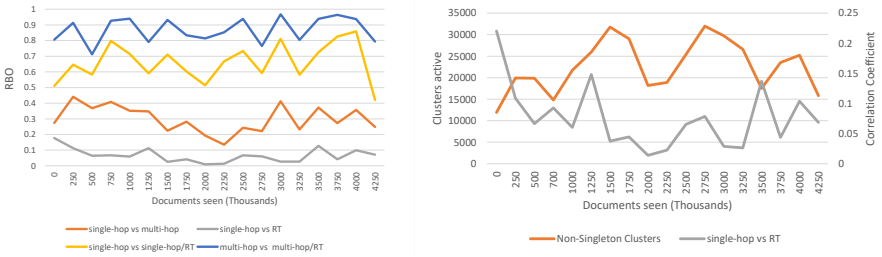


Fig. 10: Distributions for Cluster Lifetimes for Olympics dataset



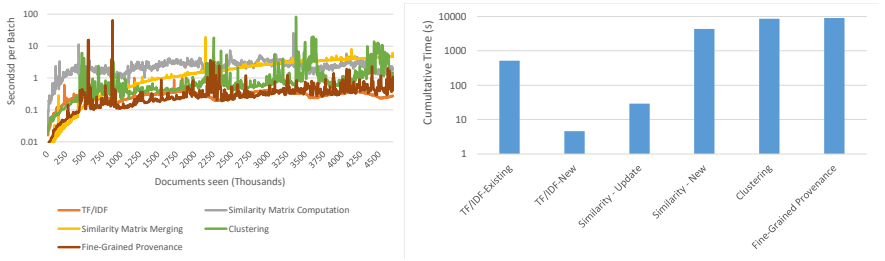
(a) Rankings for Different Influence Metrics **(b)** Ranking vs Activity over Time

Fig. 11: RBO Rankings for Olympics dataset

correlated with the retweets, proving again that retweets alone are not a reliable indicator of influence. Single-hop vs multi-hop show medium correlation (compared to Table 2). The reason for that is the denser dataset and greater size of clusters which results in influence being aggregated over longer paths. The single-hop vs single-hop with retweets is correlated medium to high: we observe that when considering only single-hop implicit influence, we get a significant share of influence. Similar trends (even stronger) can be observed for multi-hop vs multi-hop with retweets. Figure 11b provides some insights into causes of the fluctuation over time. Comparing the number of active non-singleton clusters against the correlation of implicit influence vs retweets, we see that the correlation is stronger on periods of low activity (somewhat obscured by the expiry delays). In such periods the number of retweets remains rather low, thus aligning stronger with implicit diffusion.

4.2.2 Performance analysis

Our first set of performance experiments presents the timing and cost contributors, determining its suitability to produce quick results. Figure 12a shows the cost per batch for the main components of our computational pipeline: Tf-Idf computation, simi-



(a) Computational Times per # of Documents **(b)** Overall Computational Times

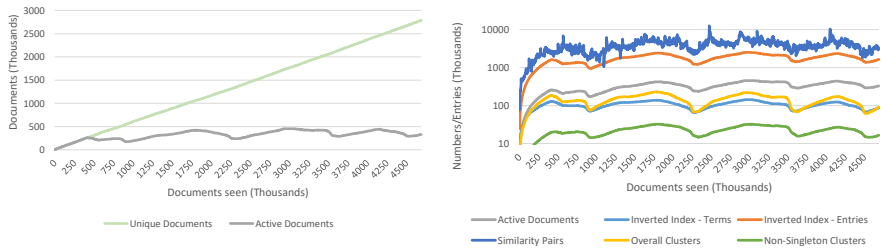
Fig. 12: Computational Times and Costs

larity matrix computation, merging of similarity matrix deltas with the main matrix, similarity clustering and computation of fine-grained provenance on all clusters. One can clearly see we are able to maintain a rate off several hundred to thousand messages per second, taking a few seconds to process a batch of 2500 messages. Furthermore, we can observe that the cost for Tf-Idf and similarity computation becomes stable rather quickly, while the cost of merging (similarity post-processing) keeps growing for longer - the reasons will become clear with the next experiment. As expected, the cost for fine-grained provenance dominates since it tries to optimize the connections among each cluster. The second Figure 12b presents the overall costs breakdown, showing that computing the similarity of new documents, incremental clustering and provenance computation dominate, while the cost of updates has been dampened.

The second set of experiments covers the working state of the system to understand where scaling limits exist. Figure 13a shows that expirations and changes in trending behavior keep the set of the active documents to a very small fraction of the unique (observed minus retweeted) documents. One can see as in the previous figure that the number of active documents fluctuates over time, as the underlying activity fluctuates. The second figure 13b provides a deep dive into the state induced and needed by our computations. The number of clusters as well as the number of distinct terms and total index entries in the inverted index become stable soon and fluctuate with the activity, capturing four days of news cycle with slightly different activity per cycle. Overall, the space consumption of the inverted index is rather modest, given the challenges of the "noisy" language use in Twitter. The main limiting factor is the number of similarity pairs used - both in the matrix and the clustering algorithm.

4.2.3 Optimizations and Tunables

The final set of experiments highlights the benefits of our optimizations and the sensitivity to the tuning settings. Figure 14 shows the benefits of the score change mitigation introduced in Section 3.3. Propagating updated scores only when the changes exceed a ratio (epsilon) of the last propagated has a tremendous effect. Without any mitigation, every newly arriving documents triggers more than 20 changes in existing documents. Even when just allowing deviations of 10 percent (0.1), the number of changes produced is reduced to roughly the same number of changes as new doc-



(a) Number of Documents over Time for Olympics dataset

(b) Working Set Sizes

Fig. 13: Set Sizes over time for Olympics dataset

uments. Increasing the allowed deviation to 0.3 and 0.5 brings again an order of magnitude reduction, beyond that the gains become very marginal. We computed the mean square error of the similarity matrix and the differences in clustering, both of which turned out to be negligible for the thresholds tested.

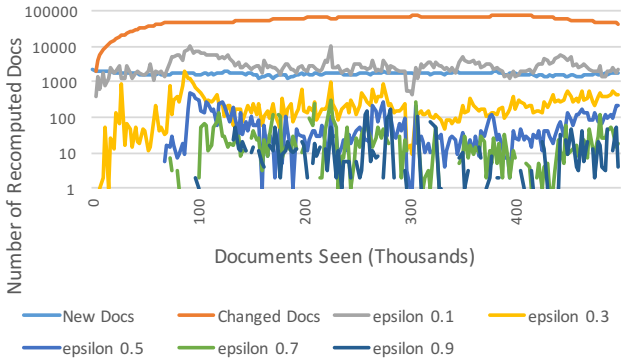


Fig. 14: Reducing Score Update Frequency by Change Thresholds

The next optimization presented in Section 3.3 is covered by Figure 15, which concerns the means to store and compute the contents of the similarity matrix. Each optimization introduced brings about one to two orders of magnitude reduction - storing only values above the relevant threshold, indexing and early stop. It should be noted that early stop is most effective on short documents like tweets. In turn, most sources with larger texts would not produce the rates we are observing here, side-stepping the need for this final optimization.

Last, Figure 16 shows the large gains of incremental computations compared to re-computations from scratch for clustering and provenance computations, yielding up to four orders of magnitude speedup. Note here that the quality of results of the incremental computations compared to full re-computation is not being affected. We tested this by computing the Jaccard Index for the clusters produced in both cases.

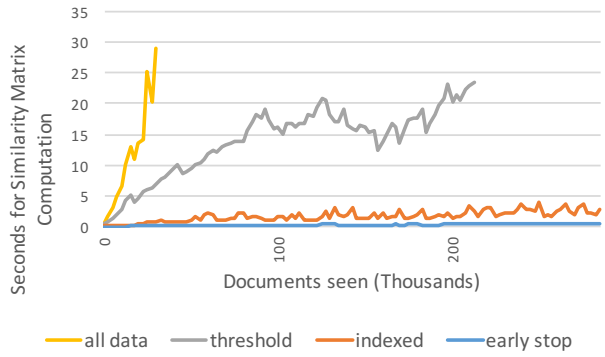


Fig. 15: Impact of Similarity Matrix Optimizations for Olypics dataset

The Jaccard Index measures the similarity between finite sets and is defined as as "the size of the intersection divided by the size of the union of the sample sets". The methodology is the following:

Two variants of the clustering algorithm were computed for equally sized batches: a) incremental computations, and b) full re-computations in every batch. These two variants resulted in particular clusterings which we need to compare. In particular, the clusters of each variant have to be examined for similarity. In order to identify matching cluster among both variants we computed the Jaccard index for all pairs of clusters among the two variants and selected the most similar for every cluster. This Jaccard index value was averaged for all pairs of similar clusters and was 0.95. As a result, we can safely assume that the clustering results of the incremental case are very similar with the corresponding results of the full re-computation. That lead us to the conclusion that we can trust the results of the incremental variant.

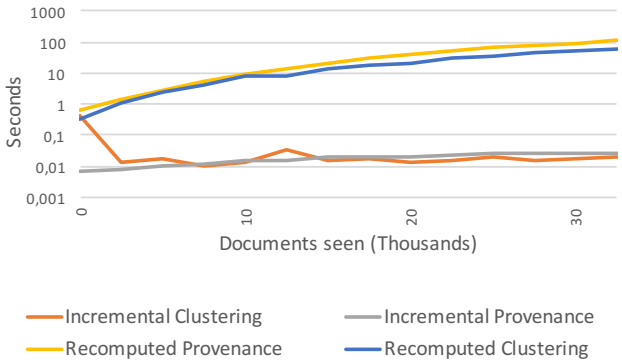


Fig. 16: Performance Gains for Incremental Computations vs Recomputations for Olympics dataset

4.3 News Media Dataset Evaluation

Our second large-scale dataset stems from the IJS newsfeed aggregator[1] and combines the data of 75K RSS feeds, covering more than 40 languages. For this study, we focused on English messages which make about around 50 % of this data set. Compared to the Twitter data evaluated so far, there are two significant differences: 1) The text contents of these messages are significantly larger after tokenization and stemming, on average around 280 terms and up to 30K terms. 2) The message rates are lower, with around a 100-150K messages per day. For consistency, this data set also spanned several days (January 2nd to 4th, 2017, full days) and contained 277500 news articles. The same setup in terms of hardware and software was used as the large-scale Twitter dataset (Section 4.2). The only substantial change was a different Tokenizer, replacing the Twitter-specific library with the open-source Lucene Tokenizers and Stemmmers.

Given the lower overall message rates, we now processed batches of 100 messages. Furthermore, we set a lower similarity threshold for clustering (0.5) and θ (0.4) since this data set is much more sparse than the large-volume Twitter dataset.

As a result, the RMSSTD values turned out to be slightly higher, fluctuating between 0.3 and 0.35, which is within the bounds for an acceptable clustering quality.

For ϵ, we increased the value slightly to 0.7 as score changes were more pronounced. In terms of the expiry timeout, we observed that doubling the expiry interval would cover about 15 % more cluster "tails", as more than 85 % of the clusters would have less than half of the time between the last and second-last activity and would not have been expired. Similar to the previous workload we set an activity timeout of 12 hours, i.e., a cluster not getting updates to any of its messages for 12 hours will be discarded.

4.3.1 Provenance Analysis

In order to evaluate the clustering result, we inspected manually the clusters produced, which accurately cover the news topics of those days. Examples of topics include the North Carolina's transgender bathroom law, the expulsion of 35 suspected Russian spies, the terror attack at Riena night club in Istanbul, and smaller scale events, like sport's matches, etc. After deeper inspection of the fine-grained provenance reconstruction, news updates could be matched to the previously published articles and republishing of the same article could be linked to the original sources.

Figures 17 and 18 demonstrate some characteristics of the clusters produced with the settings from 4.3. Figure 17 shows the cluster distributions of the last batch. The total number of non singleton clusters is 8808. We can observe that the cluster size demonstrates a skewed distribution, with the largest cluster containing almost 15K messages. The mean size is 5 (median: 1), which shows that there are many small clusters and singletons and a few large ones (corresponding to emergent news).

The cluster lifetimes in Figure 18 show a more balanced distribution, since small clusters with limited activity expire quickly. The median value is approximately 6 hours (mean 7.3), which reflects the large amount of smaller range topical news that

[1] http://newsfeed.ijs.si/

are not interesting on a large scale. However, emergent news, like updates for the attack at Reina in Istanbul, follow the typical 24 hours news cycle.

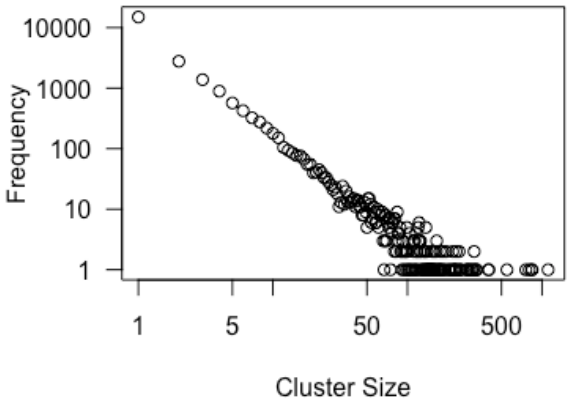


Fig. 17: Distributions of Cluster Sizes for Newsfeed dataset

Fig. 18: Distributions of Cluster Lifetimes for Newsfeed dataset

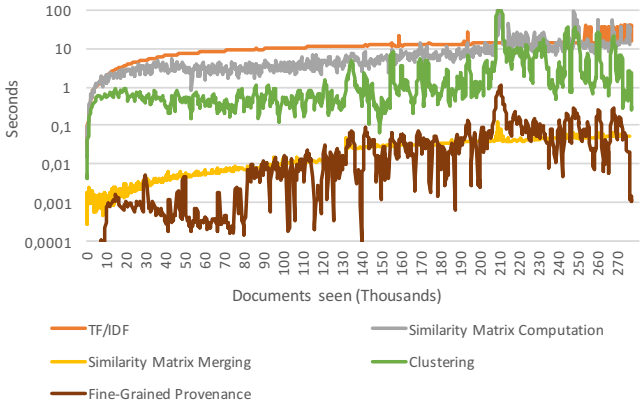


Fig. 19: Computational Times per # of Documents for Newsfeed dataset

4.3.2 Performance analysis

As in the previous dataset, we show the cost per batch for the main components of our computational pipeline Figure 19. Again, the cost of all operations becomes stable after a small number of batches, and the processing rates are clearly higher than messages rates: Processing a batch of 100 messages takes up to 10 seconds, yet the arrival rate is around 1-2 messages per second. Comparing the relative costs of the individual pipeline states with those in the Olympics dataset (Figure 12a) shows how the cost has shifted: The cost of Tf-Idf computation is significantly higher now, as the larger documents contain more terms for computation, which also increases the number of documents with the same term that need to be considered for adaptation. Considering the overall sparsity of the data, clustering and provenance computation are now relatively cheaper.

In terms of working sets (figure 20), a similar effect can be seen. The growth in size levels off quickly for all contributors, starting from active documents to non-singleton clusters. Note that the total number of clusters is very close to number of active documents, about a quarter. A significant difference exists to the Twitter dataset: the inverted index now dominates the space consumption. The number of terms in the inverted index (dictionary) already comes close to the number of similarity pairs, while the number of entries in the document/score lists now exceeds them. Considering that storing a term requires considerably more memory than a similarity score, the actual memory consumption is even higher. This growth can be attributed to the much larger number of tokens per message and the much higher overlap of terms between the messages.

The differences in the data distribution also affect the effectiveness of the optimizations for similarity computation. Indexing has very limited benefits for the news feed dataset, providing only minor improvements for similarity computation or none at all. The reasoning is the following: In contrast to the Twitter dataset many documents share common terms, as the overall number of terms per document is much higher. A

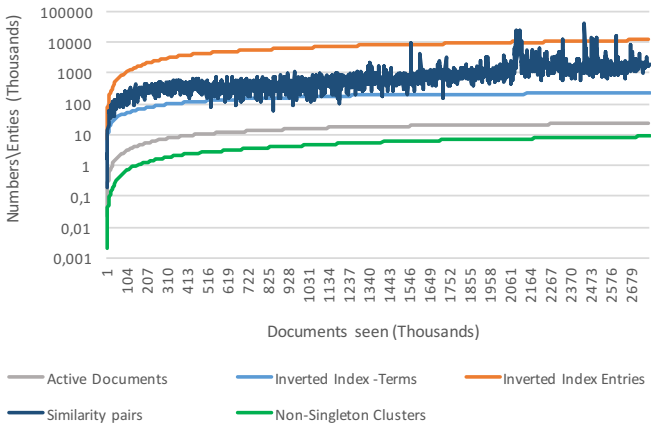


Fig. 20: Set Sizes over time for Newsfeed dataset

a result, when retrieving all the other documents containing the terms in the currently added or updated document, on average about 90% of documents are retrieved.

Early stop, however, does provide more significant benefits, reducing the number of documents for comparison to 35%. As outlined in Section 3.4, our strategy is to stop when the remaining terms will no longer be sufficient to lift the document above θ, so the similarity pair is irrelevant. Considering that the number of terms is much higher, their score decreases more gradually and each term will generate more candidate documents. The threshold for stopping was set more aggressively and an additional fixed threshold to clip the long tail of low-score terms is introduced.

Should high rates of long documents require more optimizations, we are currently considering the following options: (1) More elaborate strategies to determine the early stop such as gradient estimation or more aggressive pruning (2) Means to reduce the number of terms that are considered for similarity, thus reducing both the number of index entries and computation steps. Possible strategies could be more extensive text analysis (e.g., focusing only on certain types of words like nouns or extracted entities) or utilizing different similarity models.

Yet, given these results, we can conclude that our revised algorithm indeed provides the means to scale up our provenance reconstruction approach from small static datasets, to real-world – dynamic and large – datasets, without compromising precision.

5 Related Work

Provenance is a a very broad field ranging from provenance in databases [10], workflows [12], data streams [17], file systems [27] and the Web [29]. Here, we refer to works that are relevant for our methodology and scoping. The work presented in this paper spans several overlapping research domains, such as *origin tracing*, *clustering*,

provenance on social media, and *capturing implicit information diffusion*. In particular, it includes approaches with regard to reconstructing and predicting information diffusion paths, as well as systems where provenance is explicitly or implicitly reconstructed (e.g., through clustering). These works focus on varying application domains, which may be online or offline.

Meme Tracing A significant amount of research exists on tracing online content – often referred to as 'memes' – through similarity and by studying its diffusion properties. While these works focus on a very specific type of diffusion, without outputing provenance in an interoperable form, they often apply methods that rely on content similarity and clustering, which makes them relevant. For example, the work by [26] tracks memes that stay unchanged over time, clusters their variations and studies their temporal and structural properties. It was followed up by NIFTY [36], which can cluster and track information flow on a larger scale. Finally, [35] also rely on clustering methods to identify and group memes, but focus more on how these memes – and especially quotes – change as they propagate, and analyse the properties and temporal variants of the sources where they were observed.

News Clustering While there has been plenty of work for document clustering and its evolution, most of these approaches target an offline scenario and a close-world dataset such as a news archive. For example, the approach by [3] clusters news articles with k-means by computing their similarity and identifying groups of news clusters. The clustering is extended to work in an incremental environment where new documents are integrated into existing clusters, and frozen clusters represent inactive clusters. However, the paper makes many strict assumptions in the initial number of clusters, supports no cluster re-organization, considers an ad-hoc lower bound for similarity and excludes documents that belong to more than one topical category. The largest dataset tested was 26M documents for the period of one year. Another work that uses k-means to cluster news is that of [22], which also considers incremental updates and deletions of documents. It entails a refined method for documents expiry based on forgetting factors and their datasets consists of 64M documents. More specific to provenance, the – non-incremental, less scalable – foundations for the reconstruction algorithm in this paper were laid by [13], and applied to a small subset of a closed news archive. More recently, [1] presented a multi-funneling approach to provenance reconstruction for offline data. More precisely, they apply three techniques: one based on *IR techniques* similar to [13], one based on the *machine learning and topic modelling*, and one based on *matching the longest common subsequence*.

Provenance on Social Media A number of provenance-related works leverage the specific metadata and structure of social media. The work of [20] assigns provenance through profile information collected from different social media. While profile information reveals the popularity, and possibly trustworthiness of the contributor of particular messages, it misses the sources and intermediate steps that information took. More elaborate are approaches that leverage social connections to discover provenance information, under the assumption that these connections exert a significant influence. [16] recover the sub-graphs of information recipients given a small

fraction of them. This work relies solely on the social graph without knowing if any of these connections have shared similar information in the past. The goal is to identify possible diffusion paths that information might have taken. Similarly, [19] assume that information is most likely to flow from popular and central nodes, which means having a high degree and closeness propensity.

The work of [6] provides a reconstruction method through social connections based on well established information diffusion models. It considers information from individual messages in combination with information from the network to find intermediate forwarders. It leverages the assumption of frequent pattern propensity supporting that there are edges in the network which are more likely to propagate information than randomly selected edges. Finally, the work of [37] reconstructs explicit diffusion paths – i.e., retweets on Twitter – by leveraging social graph connections. It is this work that, through the combination with the offline approach by [13], led to the conceptual multi-level provenance reconstruction approach described by [14], which we lift to a Web scale in this paper.

Capturing Information Diffusion As a final note, we mention the research by [7], which also computes influence in information diffusion that official mechanisms cannot capture in Twitter. In more detail, the authors reconstruct diffusion paths with message similarity under the following hypothesis: users are being influenced by the last 100 messages from their friends' timelines. In order to compute provenance for a user's messages, all the last 100 messages from their friends should be collected and checked for possible similarity. On the one hand, this is a strict assumption and it is not always true according to our observations. On the other hand, a priori knowledge of friends' past messages is needed in a real-time set up, which stresses the limitations of Twitter's crawling API. Scale also becomes an issue, given the large amount of connections that users maintain (median value: 209, from a 2009 dataset by 4).

6 Conclusion and Future Work

The main contributions of the paper can be summarized in two lines. First, we highlight the importance of implicit interactions and influence that our algorithm can capture by a deep qualitative analysis. Second, we show that is feasible to reconstruct fine grained provenance on social media in an incremental, web-scale way. Our results demonstrate stable computational costs over time, while not affecting the quality of results.

Our analysis indicates that the reconstructed provenance generated by our algorithm is sound and unravels a significant share of online interactions and influence. By doing that, we offer a more complete picture of information provenance by implicit means which provides insights of how (provenance paths) and why (human interactions) information propagated in particular ways. Moreover, such analysis is implemented in an online fashion, allowing the end user to trace the provenance of data streams in a timely way. Identifying the provenance of information promptly is crucial for online journalists because it enables them to also assess the relevance and trustworthiness of particular messages and news in a timely manner.

Such analysis will open the path for streaming provenance reconstruction on social media and rule-based systems that incorporate observations from our deep qualitative analysis on human interactions. For future work, we plan to further investigate the impact of external influence on provenance reconstruction, and how it can be captured. For our system's optimization, we will exploit techniques for parallelism (possible for many parts of our pipeline) in order to elicit faster computations.

References

1. Aierken A, Davis DB, Zhang Q, Gupta K, Wong A, Asuncion HU (2014) A multi-level funneling approach to data provenance reconstruction. In: IEEE 10th International Conference on e-Science, IEEE, vol 2, pp 71–74
2. Al Hasan M, Salem S, Zaki MJ (2011) Simclus: an effective algorithm for clustering with a lower bound on similarity. Knowledge and information systems 28(3):665–685
3. Azzopardi J, Staff C (2012) Incremental clustering of news reports. Algorithms 5(3):364–378
4. Bakshy E, Hofman JM, Mason WA, Watts DJ (2011) Everyone's an influencer: quantifying influence on twitter. In: Proceedings of the fourth ACM international conference on Web search and data mining, pp 65–74
5. Baños RA, Borge-Holthoefer J, Moreno Y (2013) The role of hidden influentials in the diffusion of online information cascades. EPJ Data Science 2(1):1–16
6. Barbier G, Feng Z, Gundecha P, Liu H (2013) Provenance data in social media. Synthesis Lectures on Data Mining and Knowledge Discovery 4(1):1–84
7. Barbosa S, Cesar-Jr RM, Cosley D (2015) Using text similarity to detect social interactions not captured by formal reply mechanisms. In: e-Science (e-Science), 2015 IEEE 11th International Conference on, IEEE, pp 36–46
8. Blei DM, Lafferty JD (2006) Dynamic topic models. In: Proceedings of the 23rd international conference on Machine learning, pp 113–120
9. Cha M, Haddadi H, Benevenuto F, Gummadi PK (2010) Measuring user influence in twitter: The million follower fallacy. ICWSM 10(10-17):30
10. Cheney J, Chiticariu L, Tan WC, et al (2009) Provenance in databases: Why, how, and where. Foundations and Trends® in Databases 1(4):379–474
11. Comarela G, Crovella M, Almeida V, Benevenuto F (2012) Understanding factors that affect response rates in twitter. In: Proceedings of the 23rd ACM conference on Hypertext and social media, pp 123–132
12. Davidson SB, Boulakia SC, Eyal A, Ludäscher B, McPhillips TM, Bowers S, Anand MK, Freire J (2007) Provenance in scientific workflow systems. IEEE Data Eng Bull 30(4):44–50
13. De Nies T, Coppens S, Van Deursen D, Mannens E, Van de Walle R (2012) Automatic discovery of high-level provenance using semantic similarity. In: IPAW
14. De Nies T, Taxidou I, Dimou A, Verborgh R, Fischer PM, Mannens E, Van de Walle R (2015) Towards multi-level provenance reconstruction of information diffusion on social media. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp 1823–1826

15. De Nies T, Mannens E, Van de Walle R (2016) Reconstructing human-generated provenance through similarity-based clustering. In: International Provenance and Annotation Workshop, Springer, pp 191–194
16. Feng Z, Gundecha P, Liu H (2013) Recovering information recipients in social media via provenance. In: ASONAM, pp 706–711
17. Glavic B, Sheykh Esmaili K, Fischer PM, Tatbul N (2013) Ariadne: Managing fine-grained provenance on data streams. In: Proceedings of the 7th ACM international conference on Distributed event-based systems, pp 39–50
18. Gundecha P, Liu H (2012) Mining social media: a brief introduction. In: New Directions in Informatics, Optimization, Logistics, and Production, Informs, pp 1–17
19. Gundecha P, Feng Z, Liu H (2013) Seeking provenance of information using social media. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, ACM, pp 1691–1696
20. Gundecha P, Ranganath S, Feng Z, Liu H (2013) A tool for collecting provenance data in social media. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp 1462–1465
21. Jaho E, Tzoannos E, Papadopoulos A, Sarris N (2014) Alethiometer: a framework for assessing trustworthiness and content validity in social media. In: Proceedings of the 23rd International Conference on World Wide Web, ACM, pp 749–752
22. Khy S, Ishikawa Y, Kitagawa H (2008) A novelty-based clustering method for on-line documents. World Wide Web 11(1):1–37
23. Kovács F, Legány C, Babos A (2005) Cluster validity measurement techniques. In: 6th International symposium of hungarian researchers on computational intelligence, Citeseer
24. Kranen P, Assent I, Baldauf C, Seidl T (2011) The clustree: indexing micro-clusters for anytime stream mining. Knowledge and information systems 29(2):249–272
25. Kwon S, Cha M, Jung K, Chen W, Wang Y (2013) Prominent features of rumor propagation in online social media. In: Data Mining (ICDM), 2013 IEEE 13th International Conference on, IEEE, pp 1103–1108
26. Leskovec J, Backstrom L, Kleinberg J (2009) Meme-tracking and the dynamics of the news cycle. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 497–506
27. Magliacane S (2012) Reconstructing provenance. In: Proceedings of the 11th international conference on The Semantic Web-Volume Part II, Springer-Verlag, pp 399–406
28. Metaxas PT, Mustafaraj E (2012) Social media and the elections. Science 338(6106):472–473
29. Moreau L (2010) The foundations for provenance on the web. Foundations and Trends in Web Science 2(2–3):99–241
30. Moreau L, Missier (Eds) P, W3C Provenance Working Group (2013) PROV-DM: The PROV Data Model. W3C
31. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics 20:53–65

32. Sakaki T, Okazaki M, Matsuo Y (2010) Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web, ACM, pp 851–860
33. Salton G, McGill MJ (1986) Introduction to modern information retrieval. McGraw-Hill, Inc.
34. Sharma S (1995) Applied multivariate techniques. John Wiley & Sons, Inc.
35. Simmons MP, Adamic LA, Adar E (2011) Memes online: Extracted, subtracted, injected, and recollected. In: Fifth International AAAI Conference on Weblogs and Social Media
36. Suen C, Huang S, Eksombatchai C, Sosic R, Leskovec J (2013) NIFTY: a system for large scale information flow tracking and clustering. In: Proceedings of the 22nd international conference on World Wide Web, ACM, pp 1237–1248
37. Taxidou I, Fischer PM (2014) Online analysis of information diffusion in twitter. In: Proceedings of the 23rd International Conference on World Wide Web, WWW '14 Companion
38. Taxidou I, Alzoghbi A, Fischer PM, Schöller C (2015) Towards real-time lifetime prediction of information diffusion. In: Proceedings of the ACM Web Science Conference, ACM, WebSci '15, pp 60:1–60:2
39. Taxidou I, De Nies T, Verborgh R, Fischer P, Mannens E, Van de Walle R (2015) Modeling information diffusion in social media as provenance with W3C PROV. In: Proceedings of the 6th International Workshop on Modeling Social Media, pp 819–824
40. Taxidou I, Fischer PM, De Nies T, Mannens E, Van de Walle R (2016) Information diffusion and provenance of interactions in twitter: Is it only about retweets? In: Proceedings of the 25th International Conference Companion on World Wide Web, pp 113–114
41. Webber W, Moffat A, Zobel J (2010) A similarity measure for indefinite rankings. ACM Trans Inf Syst 28(4):20:1–20:38
42. Yang J, Leskovec J (2010) Modeling information diffusion in implicit networks. In: 2010 IEEE International Conference on Data Mining, pp 599–608