

The Evolution of Content Analysis for Personalized Recommendations at Twitter

Ajeet Grewal¹ and Jimmy Lin²

¹ Twitter, Inc.

² David R. Cheriton School of Computer Science, University of Waterloo
ajeet@twitter.com, jimmylin@uwaterloo.ca

ABSTRACT

We present a broad overview of personalized content recommendations at Twitter, discussing how our approach has evolved over the years, represented by several generations of systems. Historically, content analysis of Tweets has not been a priority, and instead engineering efforts have focused on graph-based recommendation techniques that exploit structural properties of the follow graph and engagement signals from users. These represent “low hanging fruits” that have enabled high-quality recommendations using simple algorithms. As deployed systems have grown in maturity and our understanding of the problem space has become more refined, we have begun to look for other opportunities to further improve recommendation quality. We overview recent investments in content analysis, particularly named-entity recognition techniques built around recurrent neural networks, and discuss how they integrate with existing graph-based capabilities to open up the design space of content recommendation algorithms.

ACM Reference Format:

Ajeet Grewal and Jimmy Lin. 2018. The Evolution of Content Analysis for Personalized Recommendations at Twitter. In *SIGIR '18: 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, July 8-12, 2018, Ann Arbor, MI, USA*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3209978.3210206>

1 INTRODUCTION

Twitter recommendation services notify users about what’s happening in real time, connecting them with relevant Tweets, accounts, and other content in a personalized manner. These recommendations are delivered via a variety of mechanisms, including email digests and push notifications to users’ mobile devices. At a high level, these products help Twitter users stay informed about the world and their interests.

Historically, content analysis of Tweets has not been a priority for building recommendation products. Other than simple tokenization and language identification, Twitter has not developed natural language processing capabilities until relatively recently. The reason for this is that structural signals (i.e., the follow graph) and engagement signals (Retweets, likes, replies, etc.) represent much “lower hanging fruit” in terms of value to user-facing products. In particular, they have two advantages:

- First, such signals are explicit. For example, users actively curate the list of accounts they follow, both adding *and removing* users over time. Engagement events are discrete, easy to extract from logs, and easy to aggregate.
- Second, such signals are easy to interpret and hence actionable: for example, we refer to the group of accounts that a user follows as the user’s “influencers”—because of the user’s active curation efforts, quite clearly these are accounts that the user wishes to receive information from. Similarly, a user replying to a Tweet clearly indicates an attempt to engage in conversation. The semantics of these actions is clear, making them powerful signals feeding any recommendation product. This stands in contrast to the generic click in web search, which is a weak relevance signal and much more difficult to interpret.

Compared to the two advantages above, there are a multitude of challenges in analyzing social media text: the brevity, informality, and idiosyncratic conventions used in Tweets mean that standard off-the-shelf NLP techniques perform quite poorly. As a result, in the early days of product development at Twitter, we believed that engineering effort was much more productively directed at exploiting structural and engagement signals, as opposed to tackling the many challenges associated with Tweet content analysis.

2 GRAPH-BASED RECOMMENDATIONS

Structural and engagement signals are naturally modeled as directed graphs, and hence Twitter’s first foray into recommendations took the form of a graph processing engine called Cassovary, built in 2010 [2]. The system generated user account recommendations based primarily on random walks over the static follower graph. With Cassovary, we demonstrated that explicit and interpretable signals yield high-quality recommendations using simple algorithms. Preliminary explorations concluded that a graph-based formulation yielded more promising results than an alternative approach based on analyzing users’ Tweet content.

The RealGraph [1, 2] represented the next major development of graph-based recommendations at Twitter, incorporating evidence from engagement signals and deploying machine-learned models to refine an initial set of candidates. After that came MagicRecs [3], which focused on signals that were gathered in real time—making recommendations based on temporally-correlated actions in the accounts that a user followed. This could be viewed as an instance of online motif detection in large dynamic graphs, where the goal is to detect the formation of certain configurations of user interactions to “trigger” candidate recommendations. MagicRecs paved the way for two real-time graph processing engines that are still in production today: GraphJet [5], which models a bipartite user-content

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SIGIR'18, July 8–12, 2018, Ann Arbor, MI, USA
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5657-2/18/07.
<https://doi.org/10.1145/3209978.3210206>

interaction graph, and RecService, a distributed graph processing engine designed to make recommendations based on real-time events incident on a user's social context.

Today, we have a much better understanding of the problem structure and the design space of solutions for graph-based recommendations. Our systems serve tens of thousands of queries per second, powering a broad range of recommendation products. One lesson learned is that due to the strength of the structural and engagement signals, simple recommendation algorithms are quite effective, particularly if the signals are gathered in real time. In fact, the systems-oriented challenges of building scalable, robust data pipelines and graph processing engines overshadow difficulties in developing recommendation algorithms themselves.

Nevertheless, as the result of significant engineering efforts over the past several years, many of the low hanging fruits associated with structural and engagement signals have already been picked. In the quest to further improve recommendation quality, we have recently turned to content analysis.

3 CONTENT ANALYSIS

Inside Twitter, content analysis of Tweets is implemented in what is commonly known as a pub-sub architecture, built around an internal service similar to Apache Kafka. We have developed a general framework where the contents of the Twitter Firehose (i.e., the stream of all public posts) are consumed from a queue by an annotation layer to produce an enriched stream of Tweets. These results are published to another queue, which can itself have multiple subscribers, thus allowing different parts of the organization to share annotations in a loosely-coupled manner.

Within this architecture, we have developed many capabilities. There are a number of deployed classifiers, for example, to identify NSFW (not safe for work) content. Of particular interest from the perspective of content analysis is the deployment of named-entity recognition (NER) techniques using recurrent neural networks [4], which have been in production since early 2017. We adopt a standard formulation of NER based on sequence labeling, using a custom tagset that includes persons, organizations, products, etc. Our model uses a standard bidirectional LSTM architecture, followed by a fully-connected layer and a softmax layer to generate the output labels. Training data comes from human-labeled Tweets sampled from the Firehose.

The annotated Tweets are made available to a number of downstream consumers, either deposited in our data warehouse to support batch analytics or published to a Kafka-like queue to be consumed by clients in real time. These annotated Tweets power a number of user-facing products. Here, we describe two:

Trends. Named entities power Twitter's "trends" product, which identifies topics that are popular at a particular point in time, and are personalized with respect to a user's interest, social context, and location. Whereas before trends were algorithmically derived based on hashtags and simple n -grams (with heuristic filtering), Twitter trends now also take advantage of the results of named-entity recognition to identify more semantically-coherent topics.

Real-time content recommendations. Named entities provide an important signal for real-time content recommendations. With MagicRecs [3], we have discovered that temporally-correlated events

provide strong signals to make timely, high-quality recommendations. For example, if many users reply to a Tweet (indicating an emerging conversation), followers of those users might be notified to join the conversation. This idea can be generalized to hashtags and named entities as well.

In more detail, a system called RecService maintains a real-time, in-memory graph of user-entity associations that is updated based on the stream of annotated Tweets described above. One common way of exposing this information is via what we call "social proof", which is an attempt to contextualize the relevance of algorithmic output. For example, if the named entity "Sarah Huckabee Sanders" was identified as a personalized trend for a particular user, RecService would be able to "explain" that this was due to the prevalence of mentions of that entity by accounts the user follows.

These two examples illustrate how named entities can serve as a method of aggregating signals at the semantic level. Broadly, named entities have another important use across all our products: to bootstrap recommendations for users who have poor structural and engagement signals. This is in essence the cold start problem for new users. We can rely on historical engagements with entities to estimate the types of content a user would be interested in. Given this history, we can generate new recommendations by finding high quality content about those entities, even in the absence of explicit structural and engagement signals—for example, a new user interested in the Golden State Warriors but hasn't "discovered" the online community of its fans yet.

Looking ahead, we have only begun to scratch the surface of NLP capabilities focused on content analysis. However, we continue to adopt a pragmatic, application-driven approach where investments are guided by product needs—as opposed to academic curiosity. Due to our significant experience in building graph-based recommendation products, most of our present efforts focus on how to best integrate results of context analysis with existing structural and engagement signals.

4 ACKNOWLEDGMENTS

We would like to acknowledge the contributions of current and past members of the Recommendations team and other engineers at Twitter, including Praveen Bommanavar, Naz Erkan, Daniel Hasegan, Pankaj Gupta, Ashish Goel, Siva Gurumurthy, Jerry Jiang, Tristan Jung, Gary Lam, Aaditya Landge, Brian Larson, Quannan Li, Ali Mollahosseini, Mahdi Namazifar, Venu Satuluri, Aneesh Sharma, Lohith Vuddemarri, Dong Wang, Zhijun Yin, Reza Zadeh, and Volodymyr Zhabiuk.

REFERENCES

- [1] Ashish Goel, Aneesh Sharma, Dong Wang, and Zhijun Yin. 2013. Discovering Similar Users on Twitter. In *Proceedings of the Eleventh Workshop on Mining and Learning with Graphs*.
- [2] Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Zadeh. 2013. WTF: The Who to Follow Service at Twitter. In *Proceedings of the 22nd International World Wide Web Conference (WWW 2013)*. 505–514.
- [3] Pankaj Gupta, Venu Satuluri, Ajeet Grewal, Siva Gurumurthy, Volodymyr Zhabiuk, Quannan Li, and Jimmy Lin. 2014. Real-Time Twitter Recommendation: Online Motif Detection in Large Dynamic Graphs. *Proceedings of the VLDB Endowment* 7, 13 (2014), 1379–1380.
- [4] Mahdi Namazifar. 2017. Named Entity Sequence Classification. *arXiv:1712.02316*.
- [5] Aneesh Sharma, Jerry Jiang, Praveen Bommanavar, Brian Larson, and Jimmy Lin. 2016. GraphJet: Real-Time Content Recommendations at Twitter. *Proceedings of the VLDB Endowment* 9, 13 (2016), 1281–1292.