Creator Context for Tweet Recommendation

Spurthi Amba Hombaiah[1] **Tao Chen**[1] **Mingyang Zhang**[1]
Michael Bendersky[1] **Marc Najork**[2] **Matt Colen**[3]
Sergey Levi[1] **Vladimir Ofitserov**[3] **Tanvir Amin**[3]

[1]Google Research [2]Google DeepMind [3]Google
{spurthiah,taochen,mingyang,bemike,najork,mcolen,sergeyle,vofitserov,tanviramin}@google.com

Abstract

When discussing a tweet, people usually not only refer to the content it delivers, but also to the person behind the tweet. In other words, grounding the interpretation of the tweet in the context of its creator plays an important role in deciphering the true intent and the importance of the tweet.

In this paper, we attempt to answer the question of how creator context should be used to advance tweet understanding. Specifically, we investigate the usefulness of different types of creator context, and examine different model structures for incorporating creator context in tweet modeling. We evaluate our tweet understanding models on a practical use case – recommending relevant tweets to news articles. This use case already exists in popular news apps, and can also serve as a useful assistive tool for journalists. We discover that creator context is essential for tweet understanding, and can improve application metrics by a large margin. However, we also observe that not all creator contexts are equal. Creator context can be time sensitive and noisy. Careful creator context selection and deliberate model structure design play an important role in creator context effectiveness.

1 Introduction

Linguists and philosophers have long recognized the importance of the interplay between utterance semantics and its context (Levinson, 1983). For instance, the meaning of a statement such as *I am standing here now* can only be interpreted in the context of its speaker. As a more media-related example, news stories often implicitly refer to recent events, assuming common context among their readers (e.g. *Latest news on Ukraine*). While context is important for all media content, its importance is even more pronounced on short-form content platforms like Twitter[1]. On Twitter, the 140 (and later 280) characters tweet word limits have long inspired idiosyncratic forms of communication (Westman and Freund, 2010). Therefore, applying computational natural language understanding to tweets alone can be challenging (e.g. Zampieri et al. (2019); Nguyen et al. (2020)).

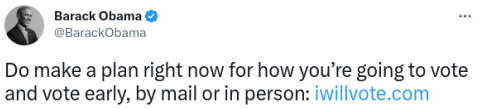


Figure 1: Tweet from Barack Obama

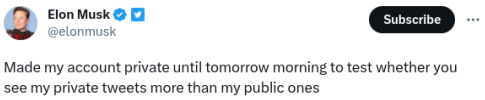


Figure 2: Tweet from Elon Musk

In this paper, we particularly focus on the context of the creator of the tweet as a key to facilitate tweet understanding. Creator context refers to the information about the creator which might not be present in the tweet itself. Take the tweet by Barack Obama in Figure 1 as an example. Without knowing that the creator is a former US President, it is hard to estimate the relationship and the importance of this tweet with regards to the 2020 US presidential election. In another example (Figure 2), the tweet mentions the pronoun "my", however, it is hard to tell who the "poster" is, and what event this tweet relates to, from the tweet content alone. Incorporating the creator context can easily address both of these problems. Moreover, knowing the creator is useful for understanding the authoritativeness and news-worthiness of the post, which is highly beneficial for downstream applications like tweet search and recommendation.

Accordingly, in this work, we investigate the importance of creator context for tweet understanding. As Twitter is known as a channel for users to

[1]Twitter and tweets were rebranded to "X" and "posts" respectively in July 2023. However, we use the former names in the paper as this work was carried out prior to July 2023.

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 353–363
December 6-10, 2023

post real-time commentary on world events (Suarez et al., 2018), we focus on the task of linking tweets to new stories. This task has many practical applications, as news publishers and aggregator services (Google News, Flipboard, Techmeme, Inkl, SmartNews, etc.) often provide Twitter integration in their products. Moreover, this task can assist journalists in composing news articles as they often use tweets as a cited source (Kapidzic et al., 2022).

The main contributions of this work are as follows:

- We propose a simple yet effective way to mine creator context from an account's metadata.
- We explore different architectures to incorporate creator context for news-tweet retrieval, and discuss the trade-off between efficiency and effectiveness.
- We propose a simple yet effective methodology to mine a large scale high-quality corpus of 8M news articles containing embedded tweets, without requiring expensive human annotation for training the models.
- Our proposed creator context and the retrieval models show strong performance on both the curated dataset and the general tweet stream.

2 Related Work

Tweet News Recommendation. Twitter users are known as heavy news consumers (Kwak et al., 2010). To facilitate this, a line of work generates personalized news article recommendation to Twitter users based on their interests (e.g. Abel et al. (2013)). These works often adopt content recommendation techniques, leveraging user's historical posts to build a profile, and comparing it against news articles. Another line of work attempts to perform recommendation at a post level, aiming at linking related news to a specific tweet. The seminal work by Guo et al. (2013) uses a graph-based latent variable model to capture tweet-news similarity.

In our work, we study the reverse task of recommending related tweets to news articles. Earlier work by Krestel et al. (2015) formulates this as a classification task (relevant or irrelevant). On a human labeled dataset of 340 ⟨*news, tweet*⟩ pairs, they build an AdaBoost model using tweet-news token-level similarity from a document likelihood model, and topic-level similarity from the Latent Dirichlet Allocation model, along with 16 other handcrafted features such as publication time, tweet length, and follower count of the Twitter user. In another work, Suarez et al. (2018) build lexical retrieval models to measure the lexical similarity between tweet textual content and query news. They curate and release a human labeled dataset consisting of 100 news articles and 6230 ⟨*news, tweet*⟩ pairs. With this dataset, a recent work by Thakur et al. (2021) directly apply deep retrieval models (most of them trained on the MS-MARCO dataset) in a zero-shot setup, in order to assess the generalization ability of deep models. Unlike the small datasets used by prior works, we collect a large set of 8M ⟨*news, tweet*⟩ positive pairs from news articles with embedded tweets, which enables us to train a deep retrieval model. Moreover, all the prior works use only the tweet text as opposed to our work of using tweet and creator context together, which brings significant gains as seen in our experiments.

Twitter User Modeling. Twitter user modeling is often studied in the context of personalized recommendation. Prior works largely leverage the Twitter users' authored posts for user profile modeling. Most of them aggregate the posts and then build user profiles with various semantic granularity, such as token-based, entity-based, topic-based or category-based (e.g. Abel et al. (2011b); Piao and Breslin (2016)). As tweets are short, some researchers attempt to use external resources, e.g. linked URLs or mined related articles (Abel et al., 2011b) to enrich the semantics of tweets. The prior works also find that the user interest is dynamically changing over time (Abel et al., 2011a; Piao and Breslin, 2016). Short-term profiles built on recent tweets do not capture the complete user interests well, and all the historical tweets are needed to build a long-term profile (Piao and Breslin, 2016). This is consistent with our findings (detailed in Section 3.1). However, obtaining all the historical tweets is often technically impractical, especially at large scale and for real-time applications. In our work, we turn to more stable sources of creator context that can be obtained efficiently to approximate the long-term interest.

3 Methodology

In this section, we first explore potential sources to mine creator context. We then introduce the task of recommending related tweets to news articles, and discuss how the creator context could augment tweet content for this task.

3.1 Creator Context

The metadata of an account is the most accessible information to represent a creator. We list five topically relevant creator attributes that we have explored below. We do not include other creator metadata like the number of followers/followees or the age of the account, as they are not topically related.

- Screen handle: The unique identifier of the creator (up to 15 characters).
- Display name: The full name of the creator as seen on their page (up to 50 characters).
- Bio: The full text of the creator's bio/profile description (up to 160 characters).
- Website: The URL of the creator's website (up to 100 characters). This often encodes creator's affiliation which is helpful to understand this person.
- Location: Geographical location of the creator (up to 30 characters). This is a key information to understand tweets about local events.

Another important attribute to understand a creator is their previous tweets. However, there are two major challenges to utilize historical tweets for creator modeling. On one hand, a creator could write posts on diverse topics which requires access to all the historical tweets to comprehensively model the topicality (Piao and Breslin, 2016). In practice, it is expensive to obtain a large number of tweets given the Twitter API pricing. On the other hand, creators are actively generating new tweets, and their interests are shifting over time (Abel et al., 2011a; Piao and Breslin, 2016); see Appendix A for examples and further discussion. This means that, in real world systems, to keep the creator context up to date, ideally we would need to constantly obtain creators' new tweets. This also suggests that, for the task of recommending tweets to news articles, we need access to tweets that could match the time frames of the news articles.

In our study, we were only able to obtain a few recent tweets crawled from each Twitter creator's home page. We explored several strategies of utilizing recent tweets in our experiments for creator context modeling, including using all recent tweets, drawing a random sample of recent tweets and using the tweet most similar to the creator bio. However, none of the methods is effective and they are even detrimental to the model performance in news tweet recommendation task. We posit that this is due to the time mismatch between the recent tweets and news articles. Twitter users' interests shift quickly over time, and thus modeling creators using historical posts requires temporal adaptation to retain performance. This is in line with other works on modeling dynamic content (Amba Hombaiah et al., 2021; Jin et al., 2022).

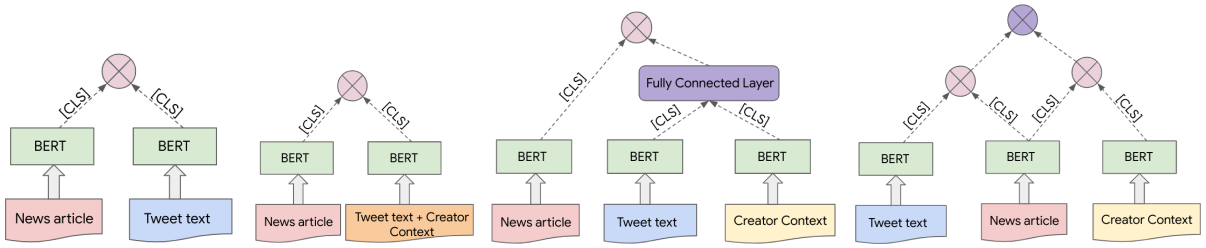
In contrast, creators' metadata tend to be stable over time. In our crawled data, we compared creator metadata from two snapshots which are 3.5 months apart. We found that 90% of creators have the exact bio (verbatim) and other metadata are also generally static. In the later part of this work, we concentrate on utilizing creator metadata for tweet understanding, for its stability and accessibility.

3.2 Recommending Related Tweets to News

Tweets are a valuable source of real-time news and have been used successfully for news dissemination, and for early detection of breaking events (Weng and Lee, 2011). Both news applications and search engines directly surface related tweets for various events and news stories. Moreover, journalists embed tweets in news articles to add depth, authenticity and perspectives to the narrative of their story. Building a model to recommend related tweets to a given news article is an important research task. Such a model could both be beneficial in user-facing applications and as an assistive writing tool for journalists.

We formulate this recommendation task as a retrieval problem: given a news article, we aim at identifying semantically relevant tweets from a large tweet pool. We adopt a dual encoder model for the task as it is a widely used retrieval architecture, and has demonstrated strong performance in many retrieval tasks (Thakur et al., 2021). Without creator context, the **Base model** simply consists of two BERT encoders, encoding news article and tweet textual content respectively. The two [CLS] embeddings from the top BERT layer are then used to compute cosine similarity, indicating the semantic relevance between the input news article and the tweet. During serving, we perform nearest neighbor search to obtain top tweets as final results.

Based on the dual encoder framework, we further propose creator context enhanced retrieval models by augmenting tweets with the proposed creator context. For creator context, we combine each attribute with a prefix ("screen", "display", "bio", "website", "location" respectively) as one text sequence. Twitter has length limitations for



(a) Base (b) Early Fusion (c) Intermediate Fusion (d) Late Fusion

Figure 3: An illustration of different Retrieval Models. A theoretical analysis of the best and worst case complexities of the models can be found in Section 3.2.

these attributes and the combined creator context sequences are generally shorter than 391 characters (max length for all attributes and their prefixes put together). Below, we discuss how creator context could be incorporated in different fusion stages (illustrated in Figure 3).

Early Fusion. The most straightforward way to augment tweets is to concatenate tweet text and creator context as one single input sequence for the tweet encoder. This allows the powerful cross-attention mechanism of BERT to model the interaction between a tweet and its creator context and generate a creator-aware tweet embedding. However, the time complexity of the attention mechanism is quadratic to the size of its input, making early fusion computationally expensive since the combined input is much longer than the tweet alone.

Intermediate Fusion. Unlike the early fusion, we use a separate BERT encoder to learn creator context embedding. We concatenate the two [CLS] embeddings from the tweet and the creator encoders and feed them to a fully connected layer, which generates the final creator-aware tweet embedding. During model serving, the creator embedding is computed on a per-creator basis, and could potentially be pre-computed. Compared to early fusion, this helps to significantly reduce computational cost, as embeddings of popular creators only need to be computed once.

Late Fusion. We further push the fusion to a later stage. We directly use the [CLS] embedding from the creator encoder to compute a $\langle$*news, creator*$\rangle$ cosine similarity score, and linearly combine it with a $\langle$*news, tweet*$\rangle$ cosine similarity score with a weight. We first tried to learn this weight as a model parameter. However, the learned weight is biased towards the creator encoder and did not work well on the development set. This is due to the lack of hard negatives in training (we use in-batch negatives) and the creator context alone becomes a distinctive feature. Instead, we consider this weight as a hyperparameter, and tune it on the development set via grid search (detailed in Section 4.3). Similar to the intermediate fusion, this model can reduce the computational cost of creator embeddings. The tuned weight can also provide better interpretability to indicate the contribution from the tweet and creator encoders. However, this model requires two retrievals and a score combining procedure, and thus is less efficient in serving.

Model Complexity. We use n and m to denote the tweet and creator context lengths, respectively. The best and worst case complexities for the Base model is $O(n^2)$ and the Early Fusion model is $O((n+m)^2)$. For Intermediate and Late Fusion models, the worst case complexity is $O(n^2+m^2)$ when the system is just deployed and the best case complexity is $O(n^2)$ when the system has pre-computed embeddings for all the creators after the system has been running for a sufficient period of time. Among our proposed models, the most efficient ones are the Intermediate and Late Fusion models (excluding the Base model).

4 Experiments & Analysis

In this section, we evaluate our creator context enhanced retrieval models for recommending related tweets to a news article. We compare the performance of different types of creator context and different model structures for incorporating it.

4.1 Dataset

There are two prior works which curated very small news-tweet datasets via human annotation. Krestel et al. (2015) selected 17 news articles and annotated 20 tweets (from top results of their model) per article, and Suarez et al. (2018) selected 100 news articles derived from Signal1M (Corney et al.,

2016) and collected human ratings for 62 tweets on average per article. Both datasets are too small to train a deep retrieval model.

Unlike prior work relying on expensive human annotation, we collect positive labels from existing news articles which have embedded tweets published during 2006 – 2022. Those tweets are carefully selected by the journalists when composing the articles. To be more specific, we obtained a large crawled news article dataset (similar to Liu et al. (2021)) and filtered out articles that do not have any embedded tweets or have more than 20 (likely spammy). We crawled the metadata from the Twitter creators' profile pages, including the screen handle (91% coverage), display name (91%), bio (87%), website (73%) and location (69%). In total, we collected 8M ⟨*news article, embedded tweet*⟩ pairs, of which there are 5.3M unique news articles (see Appendix B for some examples).

To further validate the relevance of the embedded tweets, we performed a small scale annotation. We sampled 100 news articles along with the top five tweets retrieved by our best performing retrieval model and one tweet originally embedded in the article (if it was not retrieved). Four annotators (authors of this paper) were asked to select the most relevant tweet to the article in question among these tweets (in shuffled order). From the annotation results, we see that for 90% of cases, the original embedded tweet is selected as the most relevant one. This study, albeit at a small scale, demonstrates the validity of using embedded tweets as a surrogate for relevance.

4.2 Experimental Setup

For all the models, we use a 12-layer BERT base model (Devlin et al., 2019) for each encoder, which is initialized using an "i18n" checkpoint pre-trained on a large news dataset (Liu et al., 2021). We also adopt their vocabulary which consists of $\sim$ 500K wordpiece tokens. Following Liu et al. (2021), we concatenate news title and body text (additionally remove embedded tweets to avoid information leak), and truncate the content over 512 wordpiece tokens. We allow up to 128 wordpiece tokens for both tweet and creator encoders, as tweet text (up to 280 characters) and creator context are short (up to 391 characters over all the contextual attributes and their prefixes). In the early fusion model, the max sequence length for the combined tweet and creator context encoder is 256 wordpiece tokens. We split ⟨*news, embedded tweet*⟩ pairs for training / development / testing in a ratio of 8:1:1.

We use the Wordpiece tokenizer (Wu et al., 2016) to tokenize articles. For tweets and creator context, we first extract intact tokens from hashtags and user mentions based on the following two rules (similar to Amba Hombaiah et al. (2021)):

- Split by camelcase and underscore whenever possible.
- If the above doesn't work, segment the compound word using a dictionary of unigrams constructed from Google n-grams[2] such that the probability of segmentation is maximized (by assuming conditional independence between the different segments).

As our dataset only contains positive pairs, we use in-batch negatives for model training. We optimize the model with sigmoid cross entropy loss and in-batch loss. We froze the parameters of the news encoder, as this pre-trained checkpoint has been well trained on news documents, while fine-tuning parameters for other encoders. For all the models, we carefully tuned hyperparameters including learning rate, batch size and training steps on the development set. We used a batch size of 256 and a learning rate of 1e-5 for training our models. All models were trained for 200k steps. For other hyperparameters, we use the same values as those used for training the standard BERT model (Devlin et al., 2019).

As this is a retrieval task, we primarily focus on the recall metric (recall@1K), but also report precision at top positions (1 and 5) and Mean Reciprocal Rank (MRR). We retrieve 1k tweets for every news article from the test set tweets for evaluation. For the Late Fusion model, we first retrieve the top 20k tweets using tweet and creator embeddings separately for each article. Then, we compute a combined cosine similarity score using a weight (chosen via grid search on the development set), and re-rank tweets based on the combined score.

For comparison, we adopt the Terrier implementation (Ounis et al., 2005) of BM25, a classical lexical retrieval model, as a baseline. For news-tweet lexical retrieval, Corney et al. (2016) found that using article title as query has comparable performance to using a longer query from article body.

[2] https://console.cloud.google.com/marketplace/product/bigquery-public-data/google-books-ngrams-2020

Hence, we use only news article titles as queries and retrieve 1k tweets for each article.

4.3 Retrieval Results

Table 1: Tweet recommendation results. *, #, †, ‡ indicate statistically significant (via paired t-test with $p < 0.05$) improvements over "BM25", "Base", "Late Fusion" and "Intermediate Fusion" models respectively.

Model	P@1	P@5	R@1000	MRR
BM25	0.116	0.181	0.439	0.153
Base	0.311*	0.461*	0.801*	0.391*
Early	0.362*#†‡	0.527*#†‡	0.875*#†‡	0.449*#†‡
Intermediate	0.354*#†	0.519*#†	0.871*#†	0.441*#†
Late	0.228*	0.379*	0.814*#	0.309*

Table 1 shows the results for the five models on the test set. The deep retrieval models perform significantly better than the lexical retrieval model BM25. Overall, the proposed models where we use creator context perform better than the Base model for which the creator context is not used, especially when considering recall. This proves the effectiveness of creator context for tweet modeling. Out of the three models which use creator context, the Early Fusion model performs the best, significantly improving the Base model performance by a relative measure of 9.2% on Recall@1K, 16.4% on Precision@1 and 14.8% on MRR. The Intermediate Fusion model is a close second. The additional improvement of the Early Fusion model is probably from the lower layer interactions between tweet and creator context. On the other hand, the Intermediate Fusion model, though slightly less effective, is much more efficient than Early Fusion as it allows using pre-computed creator embeddings. This could lead to significant cost and latency savings, especially for large scale user facing applications.

The Late Fusion model performs the worst among the three models using creator context. Compared to the Base model, its recall is better but other ranking focused metrics are worse. The improvement on recall is again an indication of the usefulness of creator context, but the low precision is a strong indication that a weighted sum of similarity scores fails to capture the needed interactions between tweet and creator context.

Model Recommendation. Based on the results, we recommend using the **Early Fusion** model if optimal effectiveness is desired. However, for an industry setup where serving cost and latency are mission-critical, using the **Intermediate Fusion** model would be highly beneficial. The Intermediate Fusion model allows pre-computation of creator embeddings unlike the Early Fusion model where they are recomputed for each tweet. As an additional practical benefit, decoupling computation of creator embeddings from tweet embeddings can enable them to be used separately in other applications like predicting creator similarities, matching creators to queries etc.

4.4 Creator Context Importance Analysis

We also investigate the importance of different creator attributes (discussed in Section 3.1) for modeling tweets. We conduct an ablation experiment by leaving out one attribute at a time, training our best performing Early Fusion model on the training set, and reporting its average loss on the test set.

As we see from Table 2, creator bio is the most useful creator context, as ablating it leads to the largest increase in loss. Creators often mention their interests and professions in their bio. This contextual information helps models to better understand tweets. Other creator context types are not as helpful as creator's bio, but they still bring benefits to the task of recommending tweets.

Table 2: Importance of different creator context. Lower loss denotes better performance.

Creator Attribute	Loss
All	0.064
w/o Screen handle	0.067
w/o Display name	0.068
w/o Bio	0.077
w/o Website	0.065
w/o Location	0.066
No author context	0.100

5 Discussion

Our dataset is curated from news articles and their embedded tweets. Those tweets are chosen by journalists and might be biased towards certain creators (e.g. popular figures/celebrities). We thus explore whether our model can generalize to the general population of tweets from arbitrary creators and if creator context can be useful for modeling tweets about local, rare and special-interest events.

To this end, we obtain a large set of public tweets from the Internet Archive (detailed in Appendix C). As news events are highly time sensitive, for each article, we restrict its candidate tweets to all tweets posted no earlier than one week of the article posting time (unlike the experiments in Section 4 which used all tweets without any time restriction as

retrieval candidates). We use our best performing Early Fusion model to retrieve relevant tweets and perform a small scale annotation for a quality check. To be specific, we randomly pick three dates (07/31/2017, 02/27/2018 and 06/24/2019), and then randomly sample 100 articles published on the three dates. We (three authors of this paper) annotated the relevance of top five retrieved tweets for each article. We find that our model achieves a high precision – 91% of articles have at least one relevant tweet in the top five results. This demonstrates that our model could perform well on the general tweet population. Please see Appendix D for examples.

We also find that the creators of the top retrieved results are diverse, consisting not only of popular figures but also ordinary people like local residents. See Figure 9 for an example, where, for a news article about an Olympic athlete returning to his home ground Wisconsin, the top retrieved tweet comes from a local resident. The creator's Twitter page mentions "Kimberley, Wisconsin" as the geo location making the creator context very useful in this context.

Moreover, we observe that creator understanding is particularly useful for modeling tweets about local and less popular events. Figure 10 shows an example of the top tweet retrieved by our model (which is also embedded) for a news article about a local news of a vegetation fire in San José, California. The creator's display name "San José Fire Dept." and their geo location "San José, California" are critical in matching and recommending this particular tweet for the news article.

6 Conclusion & Future Work

In this paper, we investigated the problem of using creator context for tweet understanding. Different types of creator context, including creators' bio, screen handle, display name, website and location are explored. We also proposed and examined three different model structures to incorporate creator context for the task of recommending tweets to news articles. We demonstrated that, with creator context, significant quality improvements can be achieved. We also showed that not all creator contexts are equal, and they have different effectiveness. For example, creators' tweets, as their topics shift quickly, when used as creator context, can adversely affect performance, especially when the tweets and news articles are not temporally aligned.

As future work, we would like to explore how non-topical creator features like followers count and creator social graph could be incorporated into tweet recommendation. We also plan to investigate the usefulness of creator context for other tasks (e.g., event detection) and other platforms (e.g., TikTok, YouTube Shorts).

Ethical Considerations

To the best of our knowledge, our work is ethical and has a positive impact on society and well-being of humans. We treat our data with utmost care and confidentiality. The dataset of news articles, tweets and creator information we use is encrypted and access protected. We periodically update the data to ensure we do not use any stale creator information which can compromise what creators want to reveal about themselves in the public.

In this paper, we focused on the usefulness of creator context but did not check if creator context (which is usually self-reported) is reliably truthful. Some form of creator context quality assurance may be warranted when using the proposed methods in real-world systems, where creators may be incentivized to "game the system" by inflating their biography.

References

Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2011a. Analyzing temporal dynamics in Twitter profiles for personalized recommendations in the social web. In *Proceedings of the 3rd International Web Science Conference*, WebSci '11.

Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2011b. Semantic enrichment of Twitter posts for user profile construction on the social web. In *The Semanic Web: Research and Applications*, ESWC '11, pages 375–389.

Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. 2013. Twitter-based user modeling for news recommendations. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 2962–2966.

Spurthi Amba Hombaiah, Tao Chen, Mingyang Zhang, Michael Bendersky, and Marc Najork. 2021. Dynamic language models for continuously evolving content. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pages 2514–2524.

David Corney, Dyaa Albakour, Miguel Martinez, and Samir Moussa. 2016. What do a million news articles look like? In *Proceedings of the First Interna-*

tional Workshop on Recent Trends in News Information Retrieval, pages 42–47.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, NAACL-HLT '19, pages 4171–4186.

Weiwei Guo, Hao Li, Heng Ji, and Mona Diab. 2013. Linking tweets to news: A framework to enrich short text data in social media. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '13, pages 239–249.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2022. Lifelong pretraining: Continually adapting language models to emerging corpora. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 1–16.

Sanja Kapidzic, Christoph Neuberger, Felix Frey, Stefan Stieglitz, and Milad Mirbabaie. 2022. How news websites refer to twitter: A content analysis of Twitter sources in journalism. *Journalism Studies*, 23(10):1247–1268.

Ralf Krestel, Thomas Werkmeister, Timur Pratama Wiradarma, and Gjergji Kasneci. 2015. Tweet-recommender: Finding relevant tweets for news articles. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 53–54.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is Twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600.

Stephen Levinson. 1983. *Pragmatics (Cambridge Textbooks in Linguistics)*. Cambridge University Press.

Jialu Liu, Tianqi Liu, and Cong Yu. 2021. NewsEmbed: Modeling news through pre-trained document representations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pages 1076–1086.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, EMNLP '20, pages 9–14.

Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. 2005. Terrier information retrieval platform. In *Advances in Information Retrieval*, ECIR '05, pages 517–519.

Guangyuan Piao and John G. Breslin. 2016. Exploring dynamics and semantics of user interests for user modeling on Twitter for link recommendations. In *Proceedings of the 12th International Conference on Semantic Systems*, SEMANTiCS '16, pages 81–88.

Axel Suarez, M-Dyaa Albakour, David Corney, Miguel Martínez, and José Esquivel. 2018. A data collection for evaluating the retrieval of related tweets to news articles. In *Proceedings of the 40th European Conference on Information Retrieval*, ECIR '18, pages 780–786.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the 35th Conference on Neural Information Processing Systems: Datasets and Benchmarks Track (Round 2)*, NeurIPS '21.

Jianshu Weng and Bu-Sung Lee. 2011. Event detection in Twitter. In *Proceedings of the 5th International AAAI Conference on Web and Social Media*, ICWSM '11, pages 401–408.

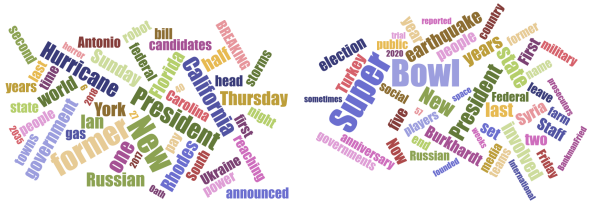
Stina Westman and Luanne Freund. 2010. Information interaction in 140 characters or less: Genres on Twitter. In *Proceedings of the Third Symposium on Information Interaction in Context*, IIiX '10, pages 323–328.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv:1609.08144*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, SemEval '19, pages 75–86.

A Shift in Creator Context Tweets Over Time

To illustrate the shift in creators' interests and the topics they tweet about, we show word clouds for two Twitter creators, Associated Press (https://twitter.com/AP) and Yann LeCun (https://twitter.com/ylecun) over a sample of recent tweets from the respective creators during two different time periods, November 2022 and February 2023. From Figure 4, for Associated Press, we see that some of the main topics during Nov'22 were around "Hurrican Ian", "Russia-Ukraine war" etc. However, the main topics are around "Turkey & Syria earthquake", "Super Bowl" etc during Feb'23. For the Twitter creator, Yann LeCun, from Figure 5, though most important topics largely concern deep learning in both time periods, trending topics "LLMs" and "ChatGPT" make an appearance in Feb'23.



(a) Nov 2022 (b) Feb 2023

Figure 4: Word clouds for Associated Press.

(a) Nov 2022 (b) Feb 2023

Figure 5: Word clouds for Yann LeCun.

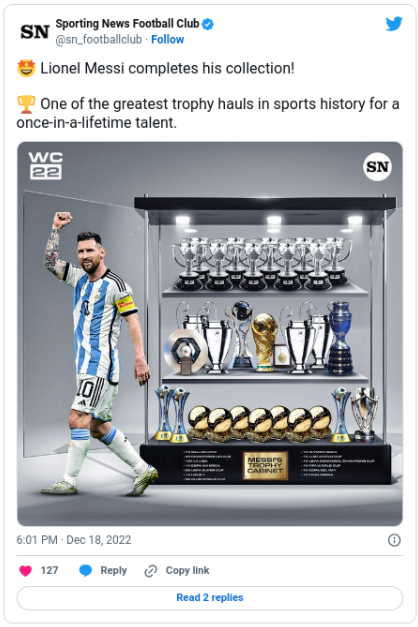
B Dataset Examples

Figure 6 shows an example of a snippet of a news article in English about the FIFA 2022 World Cup with an embedded tweet about Messi's trophies over the years.

Figure 7 shows an example of a snippet of a news article in Polish about the French politician Christine Lagarde's appointment as the President of the European Central Bank with a related embedded tweet from Lagarde in English. This example also illustrates that our model is trained to capture cross-lingual relations.

With penalties looming, Messi found the answer Argentina so badly wanted. The talisman completed a slick move, with France's appeals failing as they desperately tried to claim his finish had not crossed the line, yet still it was not enough to see off France.

Mbappe completed his hat-trick with a ruthless penalty two minutes from time, only to see his team fail from the spot twice as they surrendered their trophy by the narrowest of margins.



FIFA 2022 World Cup final goalscorers

After Messi's early opener, Di Maria scored in the 36th minute to cause France head coach Didier Deschamps to make two substitutions before the break.

France still looked limp until Mbappe scored in the 80th and 81st minutes. Messi scored his second in the 108th minute, with Mbappe converting a penalty 10 minutes later.

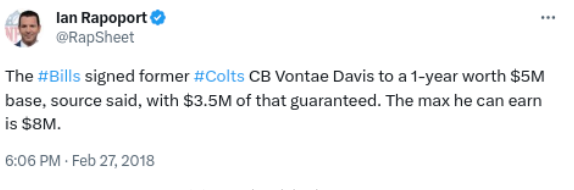
Figure 6: Snippet of a news article in English titled "Who won the 2022 FIFA World Cup? Final score, result and highlights from Qatar title decider" with an embedded tweet in English. URL: https://www.sportingnews.com/uk/football/news/who-won-2022-fifa-world-cup-final-score-result-highlights-qatar/aar5gfviuuapvkmcq3tf7dep.

C Internet Archive Dataset

To construct a tweet dataset to verify model generalization, we use the public crawl of tweets from the Internet Archive[3]. As we sample articles from three dates (07/31/2017, 02/27/2018 and 06/24/2019), we collect tweets posted no earlier than one week for the corresponding date (i.e., 07/24/2017 - 07/30/2017, 02/19/2018 - 02/26/2018 and 06/17/2019 - 06/23/2019). This results in 4.9M, 5.5M and 4.8M unique original tweets (no retweets) respectively. We extract the tweet text and the creator context information from the tweets for inference. Each week of tweets are used as retrieval candidates for their corresponding articles.

[3] https://archive.org/details/twitterstream

Figure 7: Snippet of a news article in Polish titled "Christine Lagarde nową szefową Europejskiego Banku Centralnego. Kim jest?" with an embedded tweet in English. URL: https://polskieradio24.pl/42/273/Artykul/2334907.



(a) Embedded tweet

(b) Retrieved tweet

Figure 8: Article original embedded tweet (chosen by the journalist) and top retrieved tweet by our model for a news article titled "Bills sign former Colts CB Vontae Davis". URL: https://coltswire.usatoday.com/2018/02/27/buffalo-bills-sign-indianapolis-colts-vontae-davis.

D Model Generalization Examples

We show a top retrieved tweet by our model from the Internet Archive dataset, along with the article and its original embedded article (chosen by the journalist who composed the article). In Figure 8, the news article is about the "Buffalo Bills signing a former Indianapolis Colts cornerback Vontae Davis" and our top retrieved tweet is highly topically related to the news article.

Figure 9 shows an article titled "Olympic gold medalist Matt Hamilton returns to Wisconsin". One of the top retrieved tweets by our model is from a local Wisconsin resident (a non-celebrity; the creator's Twitter page mentions "Kimberley, Wisconsin" as their location) and very topical as the article concerns a Wisconsin athlete. While the original embedded tweet is about the medals won during the "Curling - Men's event" (posted by the official account of Gangwon 2024 Winter Youth Olympic Games), our retrieved tweet not only is more relevant to the overall topic of the article, but also offers perspectives from a local resident. This demonstrates that our models, although having been trained on tweets embedded in news articles, generalize well over the general tweet population.

Figure 10 shows a local news article titled "Vegetation fire contained in San Jose" about a vegetation fire in San José, California . The top tweet retrieved by our model which is also the embedded tweet in the article is from the official account of the "San José Fire Department". The creator's display name, "San José Fire Dept." and their location "San José, California" are particularly useful as creator context. This demonstrates that creator context can be useful for modeling tweets for rare and local events.

(a) Embedded tweet

(b) Retrieved tweet

Figure 9: Article original embedded tweet (chosen by the journalist) and top retrieved tweet by our model for a news article titled "Olympic gold medalist Matt Hamilton returns to Wisconsin". URL: https://www.channel3000.com/features/olympic-gold-medalist-matt-hamilton-returns-to-wisconsin/article_bb055482-99ee-5f12-9cfd-aa3a7af7310a.html.

(a) Embedded and Retrieved tweet

Figure 10: Article original embedded tweet (chosen by the journalist) and top retrieved tweet by our model for a news article titled "Vegetation fire contained in San Jose". URL: https://www.ctinsider.com/california-wildfires/article/Vegetation-fire-breaks-out-in-San-Jose-14037622.php.