TimeLMs: Diachronic Language Models from Twitter

Daniel Loureiro*♠, Francesco Barbieri*♣,
Leonardo Neves♣, Luis Espinosa Anke◇, Jose Camacho-Collados◇
♠ LIAAD - INESC TEC, University of Porto, Portugal
♣ Snap Inc., Santa Monica, California, USA
◇ Cardiff NLP, School of Computer Science and Informatics, Cardiff University, UK
♠ daniel.b.loureiro@inesctec.pt, ♣ {fbarbieri,lneves}@snap.com,
◇ {espinosa-ankel,camachocolladosj}@cardiff.ac.uk

Abstract

Despite its importance, the *time* variable has been largely neglected in the NLP and language model literature. In this paper, we present TimeLMs, a set of language models specialized on diachronic Twitter data. We show that a continual learning strategy contributes to enhancing Twitter-based language models' capacity to deal with future and out-of-distribution tweets, while making them competitive with standardized and more monolithic benchmarks. We also perform a number of qualitative analyses showing how they cope with trends and peaks in activity involving specific named entities or concept drift. TimeLMs is available at https://github.com/cardiffnlp/timelms.

1 Introduction

Neural language models (LMs) (Devlin et al., 2019; Radford et al., 2019; Liu et al., 2019) are today a key enabler in NLP. They have contributed to a general uplift in downstream performance across many applications, even sometimes rivaling human judgement (Wang et al., 2018, 2019), while also bringing about a new paradigm of knowledge acquisition through pre-training. However, currently, both from model development and evaluation standpoints, this paradigm is essentially static, which affects both the ability to generalize to future data and the reliability of experimental results, since it is not uncommon that evaluation benchmarks overlap with pre-training corpora (Lazaridou et al., 2021). As an example, neither the original versions of BERT and RoBERTa are up to date with the current coronavirus pandemic. This is clearly troublesome, as most of the communication in recent years has been affected by it, yet these models would barely know what we are referring to when we talk about *COVID-19* or *lockdown*, to name just a few examples. The lack of *diachronic specialization* is especially concerning in contexts such

as social media, where topics of discussion change often and rapidly (Del Tredici et al., 2019).

In this paper, we address this issue by sharing with the community a series of *time-specific LMs specialized to Twitter data* (TimeLMs). Our initiative goes beyond the initial release, analysis and experimental results reported in this paper, as models will periodically continue to be trained, improved and released.

2 Related Work

There exists a significant body of work on dealing with the time variable in NLP. For instance, by specializing language representations derived from word embedding models or neural networks (Hamilton et al., 2016; Szymanski, 2017; Rosenfeld and Erk, 2018; Del Tredici et al., 2019; Hofmann et al., 2021). Concerning the particular case of LMs, exposing them to new data and updating their parameters accordingly, also known as continual learning, is a promising direction, with an established tradition in machine learning (Lopez-Paz and Ranzato, 2017; Lewis et al., 2020; Lazaridou et al., 2021; Jang et al., 2021). Other works, however, have proposed to enhance BERT-based topic models with the time variable (Grootendorst, 2020). With regards to in-domain specialization, there are numerous approaches that perform domain adaptation by pre-training a generic LM on specialized corpora. A well-known case is the biomedical domain, e.g., BioBERT (Lee et al., 2020), SciBERT (Beltagy et al., 2019) or PubMedBERT (Gu et al., 2021). In addition to these approaches to specialize language models, there have been similar temporal adaptation analyses to the one presented in our paper (Agarwal and Nenkova, 2021; Jin et al., 2021). In particular, these works showed that training language models in recent data can be beneficial, an improvement that was found to be marginal in Luu et al. (2021) in a different setting. In terms of continual lifelong learning, which is tangential to our

---

Authors marked with an asterisk (*) contributed equally.

main goal, Biesialska et al. (2020) provide a detailed survey on the main techniques proposed in the NLP literature.

More relevant to this paper, on the other hand, are LMs specialized to social media data, specifically Twitter, with BERTweet (Nguyen et al., 2020), TweetEval (Barbieri et al., 2020) and XLM-T (Barbieri et al., 2021) being, to the best of our knowledge, the most prominent examples. However, the above efforts barely address the diachronic nature of language. Crucially, they do not address the problem of specializing LMs to social media *and* putting the time variable at the core of the framework. Moreover, it is desirable that such time-aware models are released alongside usable software and a reliable infrastructure. Our TimeLMs initiative, detailed in Section 3, aims to address the above challenges.

3 TimeLMs: Diachronic Language Models from Twitter

In this section, we present our approach to train language models for different time periods.

3.1 Twitter corpus

For the training and evaluation of language models, we first collect a large corpus of tweets. In the following we explain both the data collection and cleaning processes.

Data collection. We use the Twitter Academic API to obtain a large sample of tweets evenly distributed across time. In order to obtain a sample which is representative of general conversation on that social platform, we query the API using the most frequent stopwords[1], for a set number of tweets at timestamps distanced by 5 minutes - for every hour of every day constituting a particular yearly quarter. We also use specific flags supported by the API to retrieve only tweets in English and ignore retweets, quotes, links, media posts and ads.

For our initial base model (2019-90M henceforth), we used an evenly time-distributed corpus from the API, for the period between 2018 and 2019, supplemented with additional tweets from Archive.org which cover the same period but are not evenly distributed.

Data cleaning. Before training any model, we filter each model's training set of tweets using the procedure detailed in this section. Starting with the assumption that bots are amongst the most active

users, we remove tweets from the top one percent of users that have posted most frequently. Additionally, following the recommendation of Lee et al. (2021), we remove duplicates and near-duplicates. We find near-duplicates by hashing the texts of tweets after lowercasing and stripping punctuation. Hashing is performed using MinHash (Broder, 1997), with 16 permutations. Finally, user mentions are replaced with a generic placeholder (@user), except for verified users.

3.2 Language model training

Once the Twitter corpus has been collected and cleaned, we proceed to the language model pre-training. This consists of two phases: (1) training of a base model consisting of data until the end of 2019; and (2) continual training of language models every three months since the date of the base model.

Base model training. Our base model is trained with data until 2019 (included). Following Barbieri et al. (2020), we start from the original RoBERTa-base model (Liu et al., 2019) and continue training the masked language model on Twitter data. The model is trained using the same settings as Barbieri et al. (2020), namely early stopping on the validation split and a learning rate of $1.0e^{-5}$. This initial 2019-90M base model converged after around fifteen days on 8 NVIDIA V100 GPUs.

Continuous training. After training our base model, our goal is to continue training this language model with recent Twitter corpora. At the time of writing, for practical and logistical reasons, the decision is to train a new version of each language model every three months. The process to train this updated language model is simple, as it follows the same training procedure as the initial pre-training of the language model explained above. Our commitment is to keep updating and releasing a new model every three months, effectively enabling the community to make use of an up-to-date language model at any period in time.

3.3 TimeLMs release summary

In Table 1 we include a summary of the Twitter corpora collected and models trained until the date of writing. Models are split in four three-month quarters (Q1, Q2, Q3 and Q4). Our base 2019-90M model consists of 90 million tweets until the end of 2019. Then, every quarter (i.e., every three months) 4.2M additional tweets are added, and the model gets updated as described above. Our latest

---

[1]We use the top 10 entries from: google-10000-english.txt

252

released models, which are 2021-Q4 and 2021-124M (the latter was re-trained only once with all the data from 2020 and 2021), are trained on 124M tweets on top of the original RoBERTa-base model (Liu et al., 2019). All models are currently available through the Hugging Face hub at `https://huggingface.co/cardiffnlp`.

Models	Additional	Total
2019-90M	-	90.26M
2020-Q1	4.20M	94.46M
2020-Q2	4.20M	98.66M
2020-Q3	4.20M	102.86M
2020-Q4	4.20M	107.06M
2021-Q1	4.20M	111.26M
2021-Q2	4.20M	115.46M
2021-Q3	4.20M	119.66M
2021-Q4	4.20M	123.86M
2021-124M	33.60M	123.86M

Table 1: Number of tweets used to train each model. Showing number of tweets used to update models, and total starting from RoBERTa-base by Liu et al. (2019).

In addition to these corpora for training language models, we set apart a number of tweets for each quarter (independent from the training set, with no overlap). These sets are used as test sets on our perplexity evaluation (see Section 4.2), and consist of 300K tweets per quarter, which were sampled and cleaned in the same way as the original corpus.

4 Evaluation

In this section, we aim at evaluating the effectiveness of time-specific language models (see Section 3) on time-specific tasks. In other words, our goal is to test the possible degradation of older models over time and, accordingly, test if this can be mitigated by continuous training.

Evaluation tasks. We evaluated the released language models in two tasks: (1) TweetEval (Barbieri et al., 2020), which consists of seven downstream tweet classification tasks; and (2) Pseudo-perplexity on corpora sampled from different time periods. While the first evaluation is merely aimed at validating the training procedure of the base language model, the second evaluation is the core contribution of this paper in terms of evaluation, where different models can be tested in different time periods.

4.1 TweetEval

TweetEval (Barbieri et al., 2020) is a unified Twitter benchmark composed of seven heterogeneous tweet classification tasks. It is commonly used to evaluate the performance of language models (or task-agnostic models more generally) on Twitter data. With this evaluation, our goal is simply to show the general competitiveness of the models released with our package, irrespective of their time periods.

Evaluation tasks. The seven tweet classification tasks in TweetEval are emoji prediction (Barbieri et al., 2018), emotion recognition (Mohammad et al., 2018), hate speech detection (Basile et al., 2019), irony detection (Van Hee et al., 2018), offensive language identification (Zampieri et al., 2019), sentiment analysis (Rosenthal et al., 2017) and stance detection (Mohammad et al., 2016).

Experimental setting. Similarly to the TweetEval original baselines, only a moderate parameter search was conducted. The only hyper-parameter fine-tuned was the learning rate ($1.0e^{-3}$, $1.0e^{-4}$, $1.0e^{-5}$). The number of epochs each model is trained is variable, as we used early stopping monitoring the validation loss. The validation loss is also used to select the best model in each task.

Comparison systems. The comparison systems (SVM, FastText, BLSTM, RoBERTa-base and TweetEval) are those taken from the original TweetEval paper, as well as the state-of-the-art BERTweet model (Nguyen et al., 2020), which is trained over 900M tweets (posted between 2013 and 2019). All the language models compared are based on the RoBERTa-base architecture.

Results. TweetEval results are summarized in Table 2. BERTweet, which was trained on substantially more data, attains the best averaged results. However, when looking at single tasks, BERTweet outperforms both our latest released models, i.e., TimeLM-19 and TimeLM-21, on the irony detection task[2] only. It is also important to highlight that TweetEval tasks include tweets dated until 2018 at the latest (with most tasks being considerably earlier). This suggests that our latest released model (i.e. TimeLM-21), even if trained up to 2021 tweets, is generally competitive even on past tweets. Indeed, TimeLM-21 outperforms the most similar TweetEval model, which was trained following a

---

[2]We note that the irony dataset was created via distant supervision using the #irony hashtag, and there could be a "labels" leak since BERTweet was the only model trained on tweets of the time period (2014/15) of the irony dataset.

	Emoji	**Emotion**	**Hate**	**Irony**	**Offensive**	**Sentiment**	**Stance**	**ALL**
SVM	29.3	64.7	36.7	61.7	52.3	62.9	67.3	53.5
FastText	25.8	65.2	50.6	63.1	73.4	62.9	65.4	58.1
BLSTM	24.7	66.0	52.6	62.8	71.7	58.3	59.4	56.5
RoBERTa-Base	30.8	76.6	44.9	55.2	78.7	72.0	70.9	61.3
TweetEval	31.6	79.8	55.5	62.5	81.6	72.9	72.6	65.2
BERTweet	33.4	79.3	56.4	**82.1**	79.5	73.4	71.2	67.9
TimeLM-19	33.4	**81.0**	**58.1**	48.0	**82.4**	73.2	70.7	63.8
TimeLM-21	**34.0**	80.2	55.1	64.5	82.2	**73.7**	**72.9**	66.2
Metric	M-F1	M-F1	M-F1	$F^{(i)}$	M-F1	M-Rec	AVG (F1)	TE

Table 2: TweetEval test results of all comparison systems.

similar strategy (in this case trained on fewer tweets until 2019), in most tasks.

4.2 Time-aware language model evaluation

Once the effectiveness of the base and subsequent models have been tested in downstream tasks, our goal is to measure to what extent the various models released are sensitive to a more time-aware evaluation. To this end, we rely on the pseudo perplexity measure (Salazar et al., 2020).

Evaluation metric: Pseudo-perplexity (PPPL). The pseudo log-likelihood (PLL) score introduced by Salazar et al. (2020) is computed by iteratively replacing each token in a sequence with a mask, and summing the corresponding conditional log probabilities. This approach is specially suited to masked language models, rather than traditional left-to-right models. Pseudo-perplexity (PPPL) follows analogously from the standard perplexity formula, using PLL for conditional probability.

Results. Table 3 shows the pseudo-perplexity results in all test sets. As the main conclusion, the table shows how more recent models tend to outperform models trained when evaluated older data in most test sets (especially those contemporaneous). This can be appreciated by simply observing the decreasing values in the columns of the Table 3. There are a few interesting exceptions, however. For instance, the 2020-Q1 and 2020-Q2 test sets, which corresponding to the global start of the coronavirus pandemic, are generally better suited for models trained until that periods. Nonetheless, models trained on more contemporary data appear to converge to the optimal results.

Degradation over time. How long does it take for a model to be outdated? Overall, PPPL scores tend to increase almost 10% after one year. In general,

PPPL appears to decrease consistently every quarterly update. This result reinforces the need for updated language models even for short time periods such as three-month quarters. In most cases, degradation on future data is usually larger than on older data. This result is not completely unexpected since newer models are also trained on more data for more time periods. In Section 6.1 we expand on this by including a table detailing the relative performance degradation over language models over time.

5 Python Interface

In this section we present an integrated Python interface that we release along with the data and language models presented in this paper. As mentioned in Section 3.3, all language models will be available from the Hugging Face hub and our code is designed to be used with this platform.

Our interface, based on the Transformers package (Wolf et al., 2020), is focused on providing easy single-line access to language models trained for specific periods and related use cases. The choice of language models to be used with our interface is determined using one of four modes of operation: (1) 'latest': using our most recently trained Twitter model; (2) 'corresponding': using the model that was trained only until each tweet's date (i.e., its specific quarter); (3) custom: providing the preferred date or quarter (e.g., '2021-Q3'); and (4) 'quarterly': using all available models trained over time in quarterly intervals. Having specified the preferred language models, there are three main functionalities within the code, namely: (1) computing pseudo-perplexity scores, (2) evaluating language models in our released or customized test sets, and (3) obtaining masked predictions.

Users can measure the extent to which the cho-

Models	2020-Q1	2020-Q2	2020-Q3	2020-Q4	2021-Q1	2021-Q2	2021-Q3	2021-Q4	Change
Barbieri et al., 2020	9.420	9.602	9.631	9.651	9.832	9.924	10.073	10.247	N/A
2019-90M	4.823	4.936	4.936	4.928	5.093	5.179	5.273	5.362	N/A
2020-Q1	4.521	4.625	4.699	4.692	4.862	4.952	5.043	5.140	-
2020-Q2	4.441	4.439	4.548	4.554	4.716	4.801	4.902	5.005	-4.01%
2020-Q3	4.534	4.525	4.450	4.487	4.652	4.738	4.831	4.945	-2.15%
2020-Q4	4.533	4.524	4.429	4.361	4.571	4.672	4.763	4.859	-2.81%
2021-Q1	4.509	4.499	4.399	4.334	4.439	4.574	4.668	4.767	-2.89%
2021-Q2	4.499	4.481	4.376	4.319	4.411	4.445	4.570	4.675	-2.83%
2021-Q3	4.471	4.455	4.335	4.280	4.366	4.394	4.422	4.565	-3.26%
2021-Q4	4.467	4.455	4.330	4.263	4.351	4.381	**4.402**	**4.463**	-2.24%
2021-124M	**4.319**	**4.297**	**4.279**	**4.219**	**4.322**	**4.361**	4.404	4.489	N/A

Table 3: Pseudo-perplexity results (lower is better) of all models in the Twitter test sets sampled from different quarters (each quarter correspond to three months. Q1: Jan-Mar; Q2: Apr-Jun; Q3: Jul-Sep; Q4: Oct-Dec). The last column reports difference in pseudo-perplexity, comparing the value obtained for each quarter's test set, between the model trained on the previous quarter and the model updated with data from that same quarter.

sen pretrained language models are aligned (i.e., familiar) with a given list of tweets (or any text) using pseudo-perplexity (see Section 4.2 for more details), computed as shown in Code 1.

```
from timelms import TimeLMs
tlms = TimeLMs(device='cuda')

tweets = [{'text': 'Looking forward to watching
    Squid Game tonight !'}]

pseudo_ppls = tlms.get_pseudo_ppl(tweets,
    mode='latest') # loads 2021-Q4 model
```

Code 1: Computing Pseudo-PPL on a given tweet using the most recently available model.

For a more extensive evaluation of language models using pseudo-perplexity, we provide a random subset of our test data across 2020 and 2021.[3] To evaluate other models from the Transformers package, we provide the 'eval_model' method (`tlms.eval_model()`) to compute pseudo-perplexity on any given set of tweets or texts (e.g., the subset we provide) using other language models supported by the Transformers package. Both scoring methods not only provide the pseudo-perplexity scores specific to each model (depending on specified model name, or TimeLMs specified mode), but also the PLL scores assigned to each tweet by the different models.

Finally, predictions for masked tokens of any given tweet or text may be easily obtained as demonstrated in Code 2.

```
tweets = [{"text": "So glad I'm <mask> vaccinated.",
    "created_at": "2021-02-01T23:14:26.000Z"}]

preds = tlms.get_masked_predictions(tweets, top_k=3,
```

---

[3] Limited to 50K tweets, the maximum allowed by Twitter. IDs for all test tweets are available on the repository.

```
    mode='corresponding')  # loads 2021-Q1 model
```

Code 2: Obtaining masked predictions using model corresponding to the tweet's date. Requires tweets or texts with a `<mask>` token.

Note that while the examples included in this paper are associated with specific dates (i.e., the `created_at` field), these are only required for the 'corresponding' mode.

6 Analysis

To complement the evaluation in the previous section, we perform a more detailed analysis in three important aspects: (1) a quantitative analysis on the degradation suffered by language models over time; (2) the relation between time and size (Section 6.2); and (3) a qualitative analysis where we show the influence of time in language models for specific examples (Section 6.3).

6.1 Degradation analysis

Table 4 displays the relative performance degradation (or improvement) of TimeLMs language models with respect to the test sets whose time period is the latest where they have been trained on (diagonals in the table). The table shows how models tend to perform worse in newer data sets, with a degradation of performance up to 13.68% of the earlier 2020-Q1 model on the latest 2021-Q4 model (with data almost two years later than the latest data the language model was trained on).

In order to compare the effect of continuous training with respect to single training, Figure 1 shows the PPPL performances of 2021-124M (trained on all 2020-2021 data at once) and the

Models	2020-Q1	2020-Q2	2020-Q3	2020-Q4	2021-Q1	2021-Q2	2021-Q3	2021-Q4
2020-Q1	0.00%	2.29%	3.94%	3.78%	7.52%	9.52%	11.53%	13.68%
2020-Q2	0.04%	0.00%	2.46%	2.59%	6.24%	8.16%	10.42%	12.75%
2020-Q3	1.87%	1.67%	0.00%	0.82%	4.53%	6.47%	8.54%	11.10%
2020-Q4	3.95%	3.74%	1.57%	0.00%	4.82%	7.14%	9.22%	11.43%
2021-Q1	1.58%	1.37%	-0.89%	-2.36%	0.00%	3.05%	5.16%	7.39%
2021-Q2	1.21%	0.82%	-1.55%	-2.83%	-0.77%	0.00%	2.83%	5.19%
2021-Q3	1.12%	0.75%	-1.95%	-3.20%	-1.26%	-0.61%	0.00%	3.25%
2021-Q4	0.10%	-0.17%	-2.97%	-4.47%	-2.51%	-1.83%	-1.37%	0.00%

Table 4: Difference across quarterly models and test sets comparing the pseudo-perplexity observed at the quarter corresponding to each model, against the pseudo-perplexity observed for that same model on both previous and future test sets. Highlights model degradation on future data, as well as how models fare on past data.
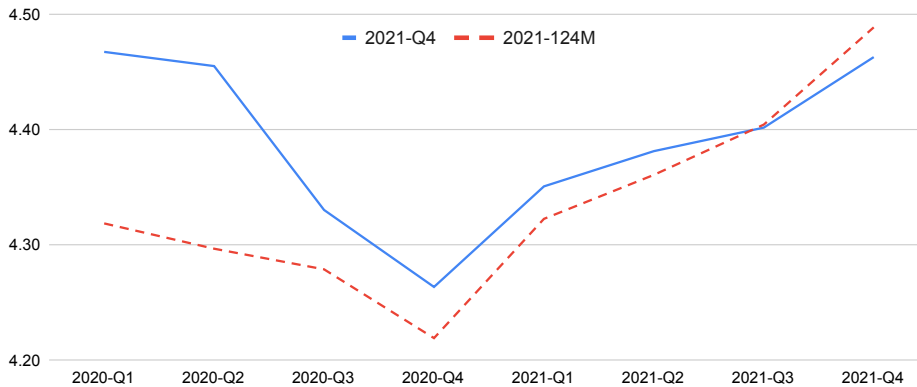

Figure 1: Performance (PPPL) of 2021-124M and 2021-Q4 models across the test sets.

2021-Q4 (updating 2021-Q3) models. Note how 2021-124M shows improved performance generally, with the largest differences being attained on the first two quarters of 2020, but not for the latest quarters where continuous training seems to work slightly better. While more analysis would be required, this result suggests that a single training is beneficial for earlier periods, while a quarterly training seems to be better adapted to the most recent data. However, there does not seem to be any meaningful catastrophic forgetting in the quarterly-updated model, as the differences are relative small.

6.2 Time and size control experiment

Given the results presented earlier, one may naturally wonder whether the improvement may be due to the increase in training size or the recency of additional data. While this question is not easy to answer (and probably the answer will be in-between these two reasons), we perform a simple control experiment as an initial attempt. To this end, we trained an additional language model with twice the training data of the third quarter of 2021 (2021-Q3). This way, the total number of training tweets

Models	2021-Q2	2021-Q3	2021-Q4
2021-Q2	4.445	4.570	4.675
2021-Q3	4.394	4.422	4.565
2021-Q3-2x	**4.380**	**4.380**	4.534
2021-Q4	4.381	4.402	**4.463**

Table 5: Results of the control experiment comparing quarterly models where the 2021-Q3 model is trained with twice the data from that quarter (2021-Q3-2x).

is exactly the same as the model trained until the fourth quarter of 2021 (2021-Q4).

Considering the results on Table 5, we find that the model trained on twice the data for Q3 outperforms the model trained with the default Q3 data in all tested quarters. This confirms the assumption that increasing training data leads to improved language model performance. When comparing with the model trained until 2021-Q4, results show this 2021-Q3-2x model is only slightly better in the 2021-Q2 and 2021-Q3 test sets. However, as we could expect, the model trained in more recent data (i.e., until 2021-Q4) gets the best overall results on the more recent test set (i.e., 2021-Q4).

Model	So glad I'm \<mask\> vaccinated.	I keep forgetting to bring a \<mask\>.	Looking forward to watching \<mask\> Game tonight!
2020-Q1	not	bag	the
	getting	purse	The
	self	charger	this
2020-Q2	not	mask	The
	getting	bag	the
	fully	purse	End
2020-Q3	not	mask	the
	getting	bag	The
	fully	purse	End
2020-Q4	not	bag	the
	getting	purse	The
	fully	charger	End
2021-Q1	getting	purse	the
	not	charger	The
	fully	bag	End
2021-Q2	fully	bag	the
	getting	charger	The
	not	lighter	this
2021-Q3	fully	charger	the
	getting	bag	The
	not	purse	This
2021-Q4	fully	bag	Squid
	getting	lighter	the
	not	charger	The

Table 6: Masked token prediction over time using three example tweets as input (using mode='quarterly'). For each quarterly model, the table displays the top-3 predictions ranked by their prediction probability.

6.3 Qualitative analysis

In this section we illustrate, in practice, how models trained on different quarters perceive specific tweets. First, we use their masked language modeling head to predict a `<mask>` token in context. Table 6 shows three tweets and associated predictions from each of our quarterly models. The model belonging to the most pertinent quarter exhibits background knowledge more aligned to the trends of that period. In the two COVID-related examples, we observe increasing awareness of the general notion of being *fully vaccinated* (as opposed to *not vaccinated*, the top prediction from the 2020-Q1 model) in the former, and, in the latter, two instances where *forgetting a mask* is more likely than forgetting other apparel less related to a particular period, such as *a charger*, *a lighter* or *a purse*. Finally, note how, in the last example, "Looking forward to watching `<mask>` Game tonight!", it is only in 2021-Q4 that predictions change substantially, when the model has been exposed to reactions to the "Squid Game" show, overlapping in time with its global release.

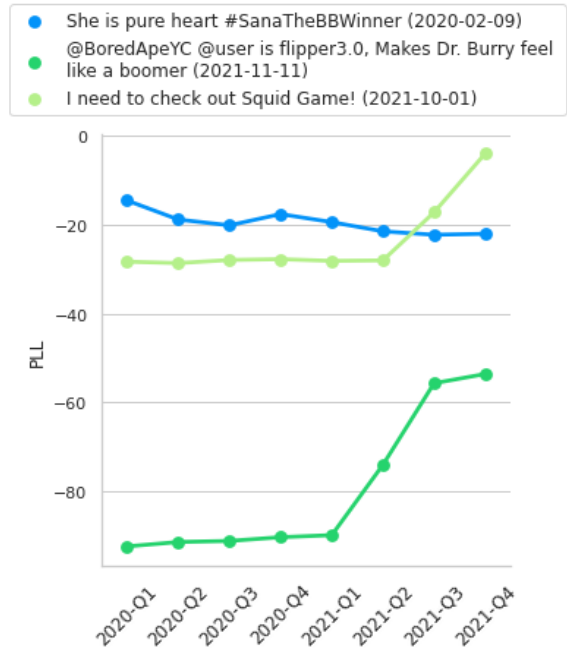Our second piece of analysis involves the visu-

Figure 2: PLL scores of TimeLMs language models trained over different periods for three selected tweets.

alization of pseudo log-likelihood (PLL) scores for tweets requiring awareness of a trend or event tied to a specific period (Figure 2). Indeed, more recent models are better at predicting tweets involving popular events, such as NFTs or, again, the show "Squid Game". Conversely, we observe a stagnation (or even degradation) of the PLL scores for a tweet about a contestant of an older reality show.

7 Conclusion

In this paper we presented TimeLMs, language models trained on Twitter over different time periods. The initiative also includes the future training of language models every three months, thus providing free-to-use and up-to-date language models for NLP practitioners. These language models are released together with a simple Python interface which facilitates loading and working with these models, including time-aware evaluation. In our evaluation in this paper, we have shown how time-aware training is relevant, not only from the theoretical point of view, but also the practical one, as the results demonstrate a clear degradation in performance when models are used for future data, which is one of the most common settings in practice.

As future work, we are planning to explicitly integrate the time span variable in the language models, i.e., introducing string prefixes, along the lines of Dhingra et al. (2022) and Rosin et al. (2022).

References

Oshin Agarwal and Ani Nenkova. 2021. Temporal effects on pre-trained models for language processing tasks. *arXiv preprint arXiv:2111.12790*.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2021. Xlm-t: A multilingual language model toolkit for twitter. *arXiv preprint arXiv:2104.12250*.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Francesco Barbieri, Jose Camacho-Collados, Francesco Ronzano, Luis Espinosa-Anke, Miguel Ballesteros, Valerio Basile, Viviana Patti, and Horacio Saggion. 2018. SemEval 2018 task 2: Multilingual emoji prediction. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 24–33, New Orleans, Louisiana. Association for Computational Linguistics.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciB-ERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6523–6541.

A.Z. Broder. 1997. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171)*, pages 21–29.

Marco Del Tredici, Raquel Fernández, and Gemma Boleda. 2019. Short-term meaning shift: A distributional exploration. 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)*, pages 2069–2075, Minneapolis, Minnesota. Association for Computational Linguistics.

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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-Aware Language Models as Temporal Knowledge Bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Maarten Grootendorst. 2020. BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Dynamic contextualized word embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6970–6984, Online. Association for Computational Linguistics.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. 2021. Towards continual knowledge learning of language models. *arXiv preprint arXiv:2110.03215*.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2021. Lifelong pretraining: Continually adapting language models to emerging corpora. *arXiv preprint arXiv:2110.08534*.

Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, et al. 2021. Pitfalls of static language modelling. *arXiv preprint arXiv:2102.01951*.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language

representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2020. Question and answer test-train overlap in open-domain question answering datasets. *arXiv preprint arXiv:2008.02637*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

David Lopez-Paz and Marc' Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A Smith. 2021. Time waits for no one! analysis and challenges of temporal misalignment. *arXiv preprint arXiv:2111.07408*.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana. Association for Computational Linguistics.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

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*, pages 9–14, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Alex Rosenfeld and Katrin Erk. 2018. Deep neural models of semantic shift. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 474–484, New Orleans, Louisiana. Association for Computational Linguistics.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Guy D. Rosin, Ido Guy, and Kira Radinsky. 2022. Time masking for temporal language models. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 833–841, New York, NY, USA. Association for Computing Machinery.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Terrence Szymanski. 2017. Temporal word analogies: Identifying lexical replacement with diachronic word embeddings. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 448–453, Vancouver, Canada. Association for Computational Linguistics.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 task 3: Irony detection in English tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.