Proceedings of the of the 17th Conference on Computer Science and Intelligence Systems pp. 121–128

DOI: 10.15439/2022F174
ISSN 2300-5963 ACSIS, Vol. 30

Temporal Language Modeling for Short Text Document Classification with Transformers

Jakub Pokrywka, Filip Graliński
Adam Mickiewicz University,
Faculty of Mathematics and Computer Science,
Uniwersytetu Poznańskiego 4
61-614 Poznań, Poland
Email: {firstname.lastname}@amu.edu.pl

***Abstract*—Language models are typically trained on solely text data, not utilizing document timestamps, which are available in most internet corpora. In this paper, we examine the impact of incorporating timestamp into transformer language model in terms of downstream classification task and masked language modeling on 2 short texts corpora. We examine different timestamp components: day of the month, month, year, weekday. We test different methods of incorporating date into the model: prefixing date components into text input and adding trained date embeddings. Our study shows, that such a temporal language model performs better than a regular language model for both documents from training data time span and unseen time span. That holds true for classification and language modeling. Prefixing date components into text performs no worse than training special date components embeddings.**

I. Introduction

Most language models are trained solely on text data. Leveraging text domain, such as language [12] or style [10] into a language model may have a positive effect on it. Time of text authorship may be also considered as an input feature, but this poses specific challenges (and opportunities) as:

- time is continuous, whereas language is discrete, at any time moment, an event might change a language irreversibly and not trivial to combine time and language units both from the mathematical and practical standpoint;
- texts might reflect natural and social cycles (days, weeks, years, cyclical sport and political events);
- text content might be correlated with extralinguistic features, themselves correlating with time (e.g. air temperature).

Recently, the NLP community has started to use time as a feature in training and/or fine-tuning large neural models ([1], [16], [19]). Here, we analyze temporal language modeling in the context of two classification tasks in different timescales: Ireland News Headlines and Twitter Sentiment Analysis. We also incorporate date components other than year. We focus on examining different approaches to date incorporation (learnable embeddings, prefixing text) using periodic and non-periodic time features under a downstream classification task.

The contributions of this paper are as follows:

- two classification datasets were redefined in a common setup in which three time-related tasks are introduced: classification (possibly) using temporal metadata, predicting temporal metadata (as a regression task) and temporal language-modeling task (as a cloze task).
- we compared three methods for introducing temporal information into neural language models;
- we considered not only linear time, but also cycles such as years, weeks, and months;
- we measured the performance of RoBERTa [14] models in several setups on the two datasets (using different parts of the temporal information, and both fine-tuning and training from scratch);
- the relations between the temporal metadata, the texts and the results obtained were analyzed.

The datasets and source of our code are publicly available.

Generally, utilizing a date does not cost much effort, because many internet documents are available with a timestamp and it is possible to adapt existing models to new domain. Such temporal language models may contribute to:

- e-commerce search engines, e.g. users intention with short phrase "umbrella" may refer to umbrella protecting from a rain in the autumn or sun umbrella in the summer;
- other types of search engines, e.g. historical newspapers;
- OCR for historical documents.

II. Datasets

Usually, text classification tasks do not incorporate time and other metadata. We suppose its impact is stronger for short texts due to shorter texts carrying less information. The time impact may be stronger for text, which may depend on people's mood or different interests. We carried out experiments with two large short-text classification datasets, where every sample is assigned a time stamp. One is spread over more than 20 years, the other ones — only 80 days. Both datasets are in English.

A. Ireland News

The dataset is available at Kaggle[1], its creator is Rohit Kulkarni. It consists of article headlines posted by the Irish

[1] https://www.kaggle.com/therohk/ireland-historical-news

TABLE I: Categories count in datasets.

category	item			
	train	dev	test	test 20/21
news	603996	75963	75783	30278
business	162550	20330	20034	14477
sport	195384	24543	24346	13447
opinion	91697	11572	11528	8086
culture	67260	8525	8424	5643
life&style	65120	8093	8084	7188

Times newspaper. Each headline is accompanied by a timestamp and article category (text of an article is not included). There are six main categories: news, sport, opinion, business, culture, life&style. The datasets statistics are described in Table I. There are more fine-grained subcategories provided in the original dataset, but they vary over time, so we didn't make use of them in our experiments.

Timestamps range from 1996-01-01 to 2021-06-30. There are 1,611,495 such headlines in total.

We employed the date range from 1996-01-01 to 2019-12-31 for most of our experiments and created an additional test set, which consists of 2020-2021 years, which dates are non-overlapping with the rest of the dataset. We refer to this test set as **Ireland News 2020-2021**. The test set **Ireland News**, without year annotation, refers to time span from training data (1996-2019). Since train/dev/test split is not determined at the original dataset site, we assign each sample randomly to train/dev/test using the 80%/10%/10% split. This resulted in the 1,186,898 / 149,134 / 148,308 train/dev/test split. The average number of words in the dataset is 7.1 per headline.

B. *Sentiment140*

This sentiment analysis dataset is obtained and described in [2]. Since in the original dataset the train set contains 1,600,000 items (positive and negative tweets) and test set only 498 (positive, negative, and neutral tweets), we made significant modifications: neutral tweets were deleted from the test set, 100,000 random items were added to the test set, also a dev set was created by randomly selecting 100,000 samples from the train set. This resulted in the 1,400,000 / 100,000 / 100,359 train/dev/test split. Timestamps range from 2009-04-06 to 2009-06-25. The datasets set are balanced (∼50% positive and ∼50% negative tweets). The average number of words is 13.8 per item. Tweets are from users in different time zones. We take time local to the author of a tweet.

III. Datasets analysis

The number of items per category differs in time. The distribution over days of month, months, years, weekdays in train datasets are presented in Figures 1 and 2 for, respectively, Sentiment140 and Ireland News. For the Sentiment140 dataset distribution over a year is not presented, since all items are from 2009. Mutual Information between presented factors and the class is given in Table V. In Ireland News, mutual information related to days of month and months is much lower than those of years and weekdays. In Sentiment140 mutual information is similar for days of month, months, and weekdays.

In both datasets, there are dependencies, which may be helpful for model performance. E.g. in Ireland News there are more sports texts on Friday and in Sentiment140 there are more negative texts on Wednesdays and Thursdays.

IV. Tasks

We created three tasks for each dataset: classification, 'fractional' year prediction, word gap prediction. Our main objective was to examine the impact of incorporating timestamps on text classification tasks. Fractional year prediction and word gap prediction tasks are mainly for analysis of the results in classification tasks.

We added timestamps in fractional-year form, which can be described by the following code:

```
days_in_year =
366 if year_is_leap_year else 365

fractional_year =
(year + (day_in_year-1+day) /
days_in_year )
```

Each item in our tasks is associated with a text, timestamp (day precision), fractional year, and category. Sample data is described in Table IV.

Each challenge for a given dataset uses the same train/dev/test split. The challenges are publicly available, courtesy of the site's owners, via the Gonito evaluation platform [3]. Source code of the challenge is available via the platform as well.

A. *Classification*

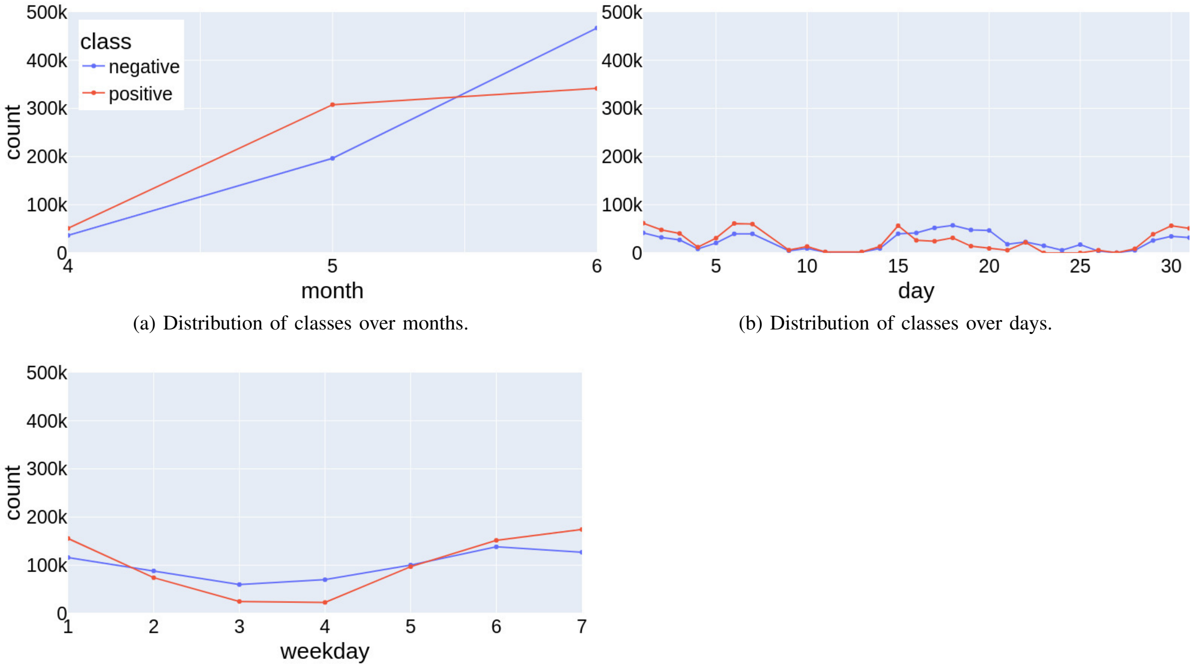
The task objective is to predict the headline category given text, date, and fractional year. The evaluation metric is simple accuracy. The challenges are available at: https://gonito.net/challenge/ireland-news-headlines (Ireland News) and https://gonito.net/challenge/sentiment140 (Sentiment140). Dataset download and submission instructions are under the "How To" tab, source code is under the "All Entries" → catalog icon in each submission row.

B. *Year prediction*

The objective is to predict the year given the text. The metric is Root Mean Square Error (RMSE). The challenges are available at: https://gonito.net/challenge/ireland-news-headlines-year-prediction (Ireland News) and https://gonito.net/challenge/sentiment140-year-prediction (Sentiment140).

C. *Word gap filling*

The task objective is to predict a masked word, like in Masked Language Modeling, given text, date, fractional year. Word is defined by characters split by spaces. There is always exactly one masked word in each sample to



(a) Distribution of classes over months.

(b) Distribution of classes over days.

(c) Distribution of items over weekdays.

Fig. 1: Distribution of classes over date factors in Sentiment140 dataset. Distribution over year is not presented, since all items come from one year.

TABLE II: Samples from the Ireland News dataset. To check article-id visit www.irishtimes.com/article-id The article ID is not provided in the challenge.

fractional year	timestamp	text	category	article ID
2004.5082	20040705	Sudan claims it is disarming militias	news	1.1147721
2008.4426	20080611	Bluffer's guide to Euro 2008	sport	1.1218069
2017.1068	20170209	Gannon offers homes in Longview near Swords	life&style	1.2966726

predict. The metric is PerplexityHashed implemented in the GEval evaluation tool [4], which is a modified version of LogLossHashed as described by [5]. This metric ensures fair assessment disregarding model vocabulary. The challenges are available at: https://gonito.net/challenge/ireland-news-headlines-word-gap (Ireland News) and https://gonito.net/challenge/sentiment140-word-gap (Sentiment140).

V. Methods

We used the RoBERTa model in the base version [14]. All models are described in this section. All code is publicly available via git commit hashes given in result tables.[2]

A. Regular Transformer as a baseline

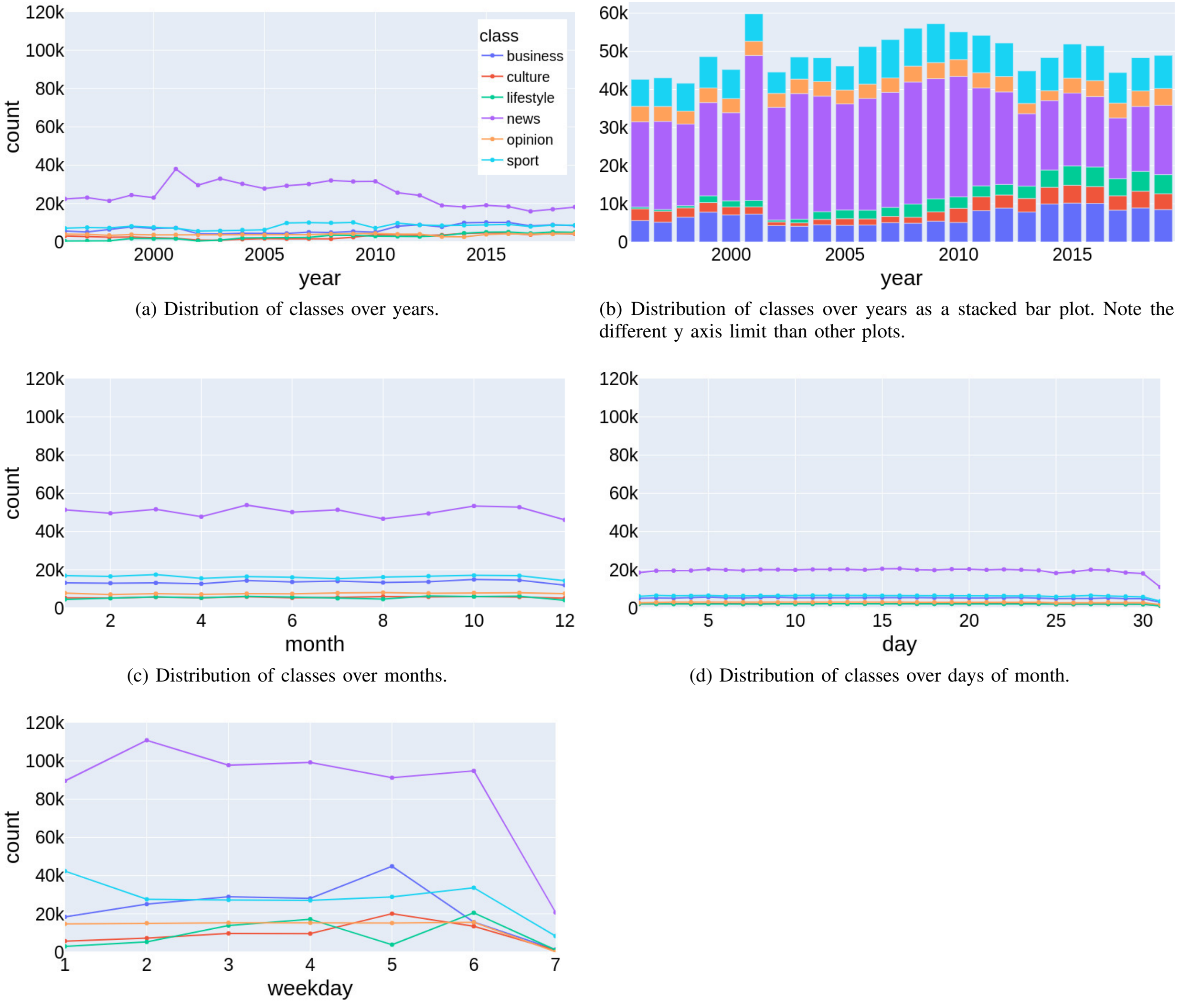
The baseline is a regular RoBERTa with no temporal information. We refer to this method as noDate in result tables.

B. Temporal Transformer

We selected the following temporal information: year, month, day of the month (day), weekday. All of them are incorporated in our temporal models. We experimented with 3 ways of including temporal information into RoBERTa models. The first two involve slight RoBERTa model architecture changes and training new embeddings during RoBERTa training. The third one is only input data modification. They are described below.

1) Date as embeddings added to every input token: Temporal embeddings are added to every input token as: $embedding = token\_emb + pos\_emb + year\_emb + month\_emb + monthday\_emb + weekday\_emb$
for each $token\_pos$. We refer to this method as addedEmbDate in result tables.

[2] Reference codes to repositories stored at Gonito.net [3] are given in curly brackets. Such a repository may be also accessed by going to http://gonito.net/q and entering the code there.



(a) Distribution of classes over years.

(b) Distribution of classes over years as a stacked bar plot. Note the different y axis limit than other plots.

(c) Distribution of classes over months.

(d) Distribution of classes over days of month.

(e) Distribution of classes over weekdays.

Fig. 2: Distribution of items over date factors in Ireland News dataset.

2) Date as stacked embeddings: Temporal embeddings are stacked at the beginning of the input sequence, as:

$$emb = \begin{cases} year\_emb & if \text{ token\_pos = 1} \\ month\_emb & if \text{ token\_pos = 2} \\ month\_emb & if \text{ token\_pos = 3} \\ weekday\_emb & if \text{ token\_pos = 4} \\ token\_emb+ & \\ pos\_emb & otherwise \end{cases}$$

Where all tokens are shifted 4 positions to the right, so first text token is on $token\_pos = 5$ We refer to this method as stackedEmbDate in result tables.

3) Date as regular text: We only modify text input of model by adding temporal information with prefixes, so item with date *20040705* and text `Sudan claims it is disarming militias` is combined to text `year: 2004 month: 7 day: 5 weekday: 1 Sudan claims it is disarming militias`.

VI. Experiments

A. Classification

We carried out experiments with text classification using all presented models. RoBERTa was finetuned and trained from pretrained checkpoints (which we refer to as pretrained) and with randomly initialized weights (which we refer to as ‘from scratch‘). The only training objective is the classification task. We report the results in Table IV.

We examined the impact on classification by each date factor. Since all temporal data incorporation methods yield similar results, we chose the regular text date incorporation method due to ease of its use (only text modification with no architecture changes). The results are presented in Table V. To examine this model conditioned by different prefixes we

TABLE III: Model roberta-pretrained-textDate predictions depending on a given date in a development dataset. If a date is represented by a dash, it is not prefixed to the model, bolded dates are as they occur actually in the dataset, not bolded are random. The examples are cherry-picked. To check article-id visit www.irishtimes.com/article-id The article ID is not provided in the challenge.

text	article ID	timestamp	actual	prediction
New bridge for Calzaghe to cross	1.914946	**20080419 Sat.**	sport	sport
New bridge for Calzaghe to cross	1.914946	20130307 Thu.	-	life&style
New bridge for Calzaghe to cross	1.914946	-	-	news
Sydney stereotypes	1.1102371	**20000913 Wed.**	sport	sport
Sydney stereotypes	1.1102371	20110422 Fri.	-	opinion
Sydney stereotypes	1.1102371	-	-	sport
Róisín Meets... comedian Mario Rosenstock	1.2463531	**20151212 Sat.**	life&style	life&style
Róisín Meets... comedian Mario Rosenstock	1.2463531	20040725 Sun.	-	news
Róisín Meets... comedian Mario Rosenstock	1.2463531	-	-	news

TABLE IV: Classification results. Different date incorporation into model. Acc stands for accuracy. The bold results are best in its category (without and with external data).

	Ireland News		Sentiment140	
method	acc	gonito	acc	gonito
most frequent from train	51.10	{161712}	49.88	{b4b180}
roberta-pretrained-noDate	82.35	{daaaf9}	89.27	{a8d1b7}
roberta-pretrained-stackedEmbDate	87.65	{9e041f}	**91.16**	{252c0c}
roberta-pretrained-addedEmbdate	86.82	{cede76}	91.04	{aa28dc}
roberta-pretrained-textDate	**87.84**	{7c52ed}	91.13	{688320}
roberta-scratch-noDate	77.88	{0798d5}	83.38	{e984db}
roberta-scratch-stackedEmbDate	**83.24**	{74efba}	**86.18**	{e3ff3e}
roberta-scratch-addedEmbdate	81.96	{587033}	85.47	{1c122b}
roberta-scratch-textDate	83.16	{413f72}	86.02	{d969ca}

TABLE V: Classification accuracy results. Different date elements included. Acc stands for accuracy. MI stands for Mutual Information between a class and a date factor. MI for Sentiment140 between year and class equals 0, because there is only 2009 year in the dataset.

	Ireland News			Sentiment140		
method	Acc	Gonito	MI(1e-5)	Acc	Gonito	MI(1e-3)
roberta-pretrained-noDate	82.35	{daaaf9}	-	89.27	{a8d1b7}	-
roberta-pretrained-textDate	87.84	{7c52ed}	-	91.13	{688320}	-
roberta-pretrained-textDay	82.66	{ca5340}	9	90.16	{2c2d07}	58
roberta-pretrained-textMonth	82.72	{3d5bb6}	61	89.59	{64cc1b}	16
roberta-pretrained-textYear	85.90	{893bbe}	3354	89.32	{be6d55}	0
roberta-pretrained-textWeekday	84.46	{daf69a}	3127	89.60	{8abd71}	19

TABLE VI: Roberta-pretrained-textDate classification on development set result. All results comes from the same model, the only difference is the prefix construction. Prefix is a standard model mode, no-prefix is a mode where no date is prefixed, and random-prefixed stands for a mode, where the date prefix comes from random date 1996-01-01 to 2021-06-30.

model	dev acc
prefix	87.97
no-prefix	78.38
random-prefix	73.97

checked its performance with no prefix and random prefix settings. Results are in Table VI and Table VII. The samples from different prefix settings are provided in Table IV.

To check model degradation, we made an inference on Ireland News test set from years 2020-2021. This is a time span later than training data, which comes from 1996-2019. The results are in Table VIII.

The impact of train dataset size is presented in Figure 3.

B. Year prediction

We choose two methods for year prediction. The first is a baseline using term frequency-inverse document frequency (TF-IDF) with logistic regression. The second is averaging all output embeddings of RoBERTa and feeding to linear regression (LR) layer. Both RoBERTa and linear regression weights are tuned during training. In both methods, the minimum (maximum) output is limited to the minimum (maximum)

TABLE VII: Classification improvement due to prefixing on roberta-pretrained-textDate model. All results comes from the same model, naming convention comes from Table VI.

dev set percentage	
accurate on both prefix and no-prefix	75.14
accurate on prefix, but not on no-prefix	12.83
accurate on no-prefix, but not on prefix	3.19
not accurate on prefix, nor on no-prefix	9.84

TABLE VIII: Classification accuracy results. Test set (years 2020-2021) comes from other time span than training set (years 1996-2019).

	Ireland News (2020/21)	
method	acc	gonito
most frequent	38.27	{953311}
roberta-pretr.-noDate	85.99	{e684b3}
roberta-pretr.-textDate	**87.79**	{5fba22}
roberta-pretr.-textYear	87.49	{8d5ad4}

fractional year found in the datasets. The results are presented in Table IX, along with a null-model baseline using the mean fractional year from the training set as the prediction for each data point.

C. Word gap filling

RoBERTa was finetuned and trained from a pretrained checkpoint and with randomly initialized weights. The training objective is Masked Language Modeling. Only prepending data to the input was considered as a method for introducing the data. See Table X.

VII. Discussion

For both datasets including dates into RoBERTa models raises the accuracy score. This stands true for pretrained and randomly initialized models. Stacked embedding and date incorporation as a text give a similar result and both are slightly better than the method of adding embeddings to every input token. It's easier to modify input text than modify model architecture, hence we recommend embedding date by prefixing input texts. The greater mutual information is between each factor and class factor, the more the model gains in accuracy score. The model trained with a date prefix performs well, only when the prefix is provided. There is no gain from date prefixing for a 1k documents train dataset and the gain is constant over 100k documents train dataset. Predicting fractional year is difficult in both datasets because all models perform not much better than baseline. We hypothesize this is a reason why classification benefits from date metadata, since adding strongly correlated factors (like a date to text in this case) would not bring information gain.

The temporal models perform better also for test sets from unseen years. To our surprise, day of the month, month, weekday, year incorporation into model performs only marginally better than incorporation only year for Ireland News 2020-2021 dataset.

In pretrained models, date incorporation slightly lowers perplexity. Models with randomly initialized weights benefit hugely from date incorporation.

VIII. Related work

There are several studies concerning language model degradation over time and adaptation to newer data [13], [17], [6]. [7] focused especially on text classification. They considered years as well as cyclical intervals (e.g., January-March). Their method was to train separate models for different time spans. [8] proposed method based on using discrete multiple temporal word embeddings based on time domains for document classification using recurrent neural networks. [9] developed model-agnostic timed dependent embedding representation for time and evaluated on recurrent neural networks across various tasks. [1] introduced temporal T5 language model, where a year was prefixed into text input and finetuned on temporal data. The experiments focused on knowledge extraction from language models and showed their method performs better in terms of language modeling and question answering than T5 language model with no prefixed year. [19] incorporated both geographical and time data into a transformer model for a QA task employing year as well as month and day. [16] prefixed year for semantic change detection. Additionally, the authors proposed the training objective of masking year information during model training. However, both [1], [16] use only year metadata, in contrast to our study, where we also days of month, months, weekdays are taken into consideration. [18] trained an SVM model to predict the date of text as a classification problem and [11] use approach of neologism based approach. Very recently [15] released temporal NLP challenges based on a large corpus of historic texts but didn't include downstream tasks, such as classification. The corpus consists of texts covering over 100 years. They trained from scratch and fine-tuned temporal RoBERTa models based on day of month, months, weekdays, and year as a prefixed text. They proved that temporal language models perform better than standard language models.

IX. Conclusion

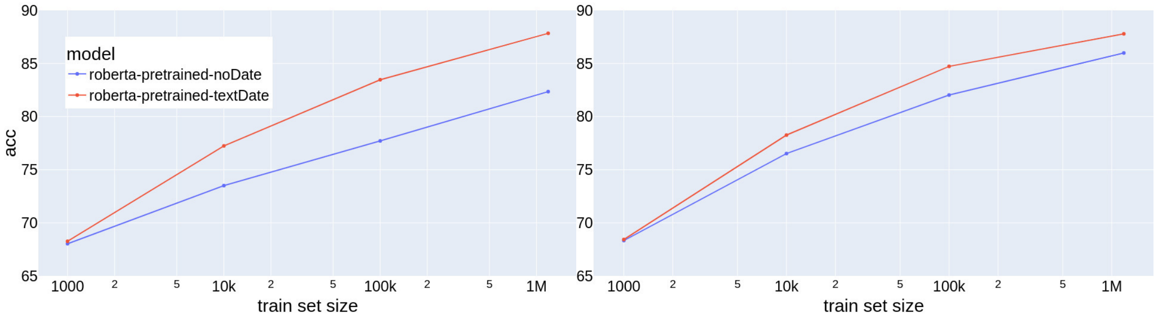
Transformer models benefit from temporal information data in classification tasks for short texts. We have proved that it's not only true for a year, but also other date factors, such as weekday, day of the month, and month. The greater the mutual information between a factor and a class, the greater the benefit. The result is important, because day of the month, month, weekday factors don't outdate after model training

TABLE IX: Fractional year prediction results, RMSE is for root-mean-square error, MAE – mean absolute error, LR – linear regression.

	Ireland News			Sentiment140		
method	RMSE	MAE	Gonito	RMSE	MAE	gonito
mean from train	6.76426	5.80722	{0b0e9c}	0.04674	0.03396	{4856c5}
TF-IDF + LR	5.32491	4.27185	{2226fb}	0.04917	0.03635	{579c8f}
RoBERTa + LR head	4.53676	3.38758	{632b5d}	**0.04469**	0.03289	{349e5b}
RoBERTa from scratch + LR head	**4.51179**	**3.35951**	{be0106}	0.04526	**0.03222**	{b672ee}

TABLE X: Word gap prediction results. Ppl hashed stands for perplexity hashed.

	Ireland News		Sentiment140	
method	ppl hashed	gonito	ppl hashed	gonito
equal probability	1024.0	{6bd5a8}	1024.0	{3de230}
RoBERTa from scratch	90.8	{9ac479}	51.0	{f0f343}
RoBERTa from scratch with time	**46.0**	{dc75a7}	**46.1**	{ddf16f}
RoBERTa no fine-tuning	51.0	{f0f343}	66.2	{e625c6}
RoBERTa fine-tuned	23.3	{42793a}	34.6	{a365da}
RoBERTa fine-tuned with time	**21.6**	{cfaf6c}	**33.6**	{37bd6e}



(a) Ireland News test (1996-2019) accuracy.

(b) Ireland News test (2020/21) accuracy.

Fig. 3: Test set accuracy varying on train dataset size for model with and without date incorporation.

due to its cyclical nature, differently to year, which is linear. The best and simplest method for temporal data incorporation seems to be input text modification.

References

[1] B. Dhingra, J. R. Cole, J. M. Eisenschlos, D. Gillick, J. Eisenstein, and W. W. Cohen. Time-aware language models as temporal knowledge bases, 2021.

[2] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *Processing*, 150, 2009.

[3] F. Graliński, R. Jaworski, Ł. Borchmann, and P. Wierzchoń. Gonito.net – open platform for research competition, cooperation and reproducibility. In A. Branco, N. Calzolari, and K. Choukri, editors, *Proceedings of the 4REAL Workshop: Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language*, pages 13–20. 2016.

[4] F. Graliński, A. Wróblewska, T. Stanisławek, K. Grabowski, and T. Górecki. GEval: Tool for debugging NLP datasets and models. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 254–262, Florence, Italy, 2019. Association for Computational Linguistics.

[5] F. Graliński. (Temporal) language models as a competitive challenge. In Z. Vetulani and P. Paroubek, editors, *Proceedings of the 8th Language & Technology Conference*, pages 141–146. Fundacja Uniwersytetu im. Adama Mickiewicza w Poznaniu, 2017.

[6] S. A. Hombaiah, T. Chen, M. Zhang, M. Bendersky, and M. Najork. Dynamic language models for continuously evolving content. *ArXiv preprint*, abs/2106.06297, 2021.

[7] X. Huang and M. J. Paul. Examining temporality in document classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 694–699, Melbourne, Australia, 2018. Association for Computational Linguistics.

[8] X. Huang and M. J. Paul. Neural temporality adaptation for document classification: Diachronic word embeddings and domain adaptation models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4113–4123, Florence, Italy, 2019. Association for Computational Linguistics.

[9] S. M. Kazemi, R. Goel, S. Eghbali, J. Ramanan, J. Sahota, S. Thakur, S. Wu, C. Smyth, P. Poupart, and M. A. Brubaker. Time2vec: Learning a vector representation of time. *ArXiv*, abs/1907.05321, 2019.

[10] N. S. Keskar, B. McCann, L. R. Varshney, C. Xiong, and R. Socher. Ctrl: A conditional transformer language model for controllable generation. *ArXiv*, abs/1909.05858, 2019.

[11] V. Kulkarni, Y. Tian, P. Dandiwala, and S. Skiena. Simple neologism based domain independent models to predict year of authorship. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 202–212, Santa Fe, New Mexico, USA, 2018. Association for Computational Linguistics.

[12] G. Lample and A. Conneau. Cross-lingual language model pretraining. In *NeurIPS*, 2019.

[13] A. Lazaridou, A. Kuncoro, E. Gribovskaya, D. Agrawal, A. Liska, T. Terzi, M. Gimenez, C. de Masson d'Autume, S. Ruder, D. Yogatama,

K. Cao, T. Kociský, S. Young, and P. Blunsom. Pitfalls of static language modelling. *ArXiv*, abs/2102.01951, 2021.

[14] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *ArXiv preprint*, abs/1907.11692, 2019.

[15] J. Pokrywka, F. Graliński, K. Jassem, K. Kaczmarek, K. Jurkiewicz, and P. Wierzchoń. Challenging America: Modeling language in longer time scales. *Findings of North American Chapter of the Association for Computational Linguistics*, 2022. forthcoming.

[16] G. D. Rosin, I. Guy, and K. Radinsky. Time masking for temporal language models, 2021.

[17] P. Röttger and J. Pierrehumbert. Temporal adaptation of BERT and performance on downstream document classification: Insights from social media. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2400–2412, Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics.

[18] T. Szymanski and G. Lynch. UCD : Diachronic text classification with character, word, and syntactic n-grams. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 879–883, Denver, Colorado, 2015. Association for Computational Linguistics.

[19] M. Zhang and E. Choi. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics.