Keeping in Time: Adding Temporal Context to Sentiment Analysis Models

Notebook for the LongEval Lab at CLEF 2023

Dean Ninalga[1],*

[1] Toronto, Canada

Abstract

This paper presents a state-of-the-art solution to the LongEval CLEF 2023 Lab Task 2: *LongEval-Classification* [1]. The goal of this task is to improve and preserve the performance of sentiment analysis models across shorter and longer time periods. Our framework feeds *date-prefixed* textual inputs to a pre-trained language model, where the timestamp is included in the text. We show *date-prefixed* samples better conditions model outputs on the temporal context of the respective texts. Moreover, we further boost performance by performing self-labeling on unlabeled data to train a student model. We augment the self-labeling process using a novel augmentation strategy leveraging the *date-prefixed* formatting of our samples. We demonstrate concrete performance gains on the LongEval-Classification [1] evaluation set over non-augmented self-labeling. Our framework achieves a 2nd place ranking with an overall score of 0.6923 and reports the best *Relative Performance Drop* (RPD) [2] of -0.0656 over the short evaluation set (see Alkhalifa et al. [3]).

Keywords

Self-Labeling, Sentiment Analysis, Temporal Misalignment, Date-Prefixing

1. Introduction

The application of language models such as BERT [4], RoBERTa [5] and XLM-RoBERTa [6] to textual data is a core component in many natural language processing (NLP) pipelines. However, a notable limitation of most language models is their lack of temporal awareness, as they typically encode text into fixed representations. Conversely, the nature of textual data is inherently dynamic and subject to change over time. Where traditional meanings of words, phrases, and concepts are constantly evolving [7, 8]. Furthermore, significant events can alter the factual basis of the text [9]. Although metadata of well-known text corpora includes timestamps, timestamps are almost never used within many NLP pipelines. A sentiment analysis model trained today could interpret the phrase: "you are just like X" as positive sentiment. However, an issue can arise once people consider a comparison to 'X' as a non-positive comparison. Subsequently, the model becomes *misaligned* if this flip in public opinion occurs. The model becomes less performative, especially on statements including comparisons to 'X'. Hence, it is

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

*Corresponding author.

djninalga@gmail.com (D. Ninalga)
0009-0008-4246-1936 (D. Ninalga)

hard to train a model that can generalize to future data without a sense of temporal context and awareness [10].

Mitigating *temporal misalignment* [11] between the facts and general sentiments of the current world and those found in text corpora is an active area of focus in various areas of research in nlp. In particular, work in NER (named-entity-recognition) [12, 9, 13] and question-and-answering [14, 15, 12, 16] often directly address temporal misalignment as they are considered *knowledge-intensive* tasks [10].

A common and straightforward way to address temporal misalignment in textual data is to create new models (or update old ones) with the most recent data available [17, 18, 10]. However, continually growing datasets incur an increase in computational costs for data acquisition and training models which also contributes to an ever-increasing environmental cost [19, 20]. Therefore, finding a solution outside of continuous retraining that preserves model performance over time is desirable.

In this paper, we follow Dhingra et al. [7] who use an alternative approach that modifies the textual input with its timestamp. Thus, we can take advantage of text-only pre-trained language models used for classification in addition to conditioning the models with the temporal context for the input.

We will outline our system, which is aligned with some of the recent works in NER and temporal misalignment, and evaluate it on the *LongEval-Classification* benchmark [1].

Our contribution is two-fold: (1) We show that date-prefixing the input text with its timestamp conditions the outputs of a language model on the temporal context of the input. (2) We utilize an augmentation strategy that leverages the date-prefixing by randomly modifying the timestamp of unlabeled inputs. We show that this augmentation strategy improves the performance benefits of semi-supervised learning on unlabeled data.

2. Background and Related Work

Recently, *TempLama* [7] showed that directly placing the year of the timestamp as a prefix in the text is performative in the context of named-entity-recognition. They, then feed the date-prefixed inputs to a T5 [21] model to directly model the temporal context. Cao and Wang [22] directly compares a date-prefixing approach to an embedding approach where the date is numerically embedded with a linear projection. [22] in the context of text generation, found that linear projection was less sensitive to the timestamps while date-prefixing is better at generating more temporally sensitive facts.

Self-labeling (or self-distillation) is a semi-supervised learning strategy that typically involves learning from pseudo-labels for unlabeled data. Self-labeling is demonstrated to add performance gains across a variety of domains including text classification [23]. Agarwal and Nenkova [9] found that self-labeling performs better than specialized pre-training objectives such as domain-adaptive pretraining [24] across several tasks including sentiment analysis. However, it is important to note that recently Ushio et al. [25] have shown that self-labeling, as presented in [9], is not as effective for NER when compared to models trained for specific time periods.

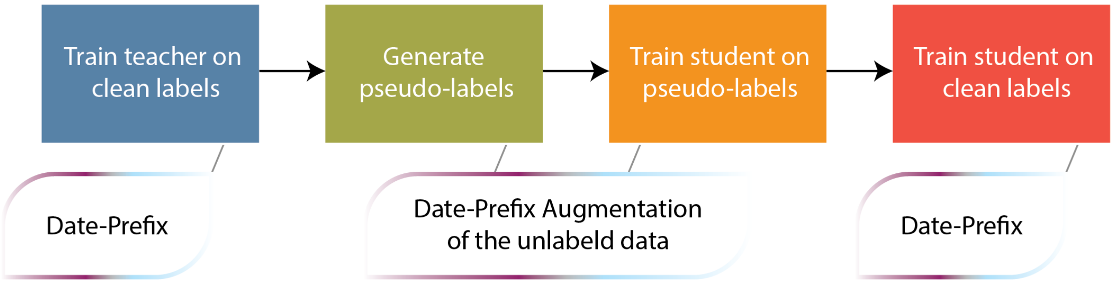
Train teacher on clean labels
Generate pseudo-labels
Train student on pseudo-labels
Train student on clean labels
Date-Prefix
Date-Prefix Augmentation of the unlabeld data
Date-Prefix

Figure 1: Method Overview: (top-row) summarization of our semi-supervised learning training pipeline stages, (bottom-row): modifications we made to the pipeline and at what stage they apply

3. Methodology

Figure 1 provides an overview of our system. Following Agarwal and Nenkova [9], we first train a teacher model on the full labeled dataset to create pseudo-labels for the unlabeled data. During this training phase, every sample in the labeled dataset is date-prefixed, meaning that the year of the timestamp is included as part of the input text. We use a novel augmentation strategy on the date prefixes (see Section section 3.3) to condition the pseudo-labels on the temporal context learned by the teacher. A new student model is then trained for 22000 training steps on the generated pseudo-labels and is subsequently trained on the original labeled data that was used for the teacher. Finally, we use the resulting student model for inference. For simplicity, both the teacher and student models share the same architecture. We provide further detail on the individual components of our system in the following sections.

3.1. Pre-Trained Model

Using a pre-trained language is generally much better than training a new model from scratch. However, it is not always clear which pre-training works best for any particular task. Here we use Bernice [26] a variant of XLM-RoBERTa [6] specialized for Twitter data. We train a single model for inference on the test set and we do not rely on ensembling techniques. We train using the cross-entropy classification loss.

3.2. Date-Prefixing

Consistent with Dhingra et al. [7] we prefix each input text with the year of the given time-stamp followed by the text itself (e.g. “year: 2023 text: I really do enjoy drinks with friends”). As we observe from Table 1 training on this data conditions the model outputs with the temporal context found in the data using date-prefixing. Table 1 provides real input and output examples based on a trained model across various years. We do not modify the architecture of the language model to take the timestamp as a vector input. By maintaining the use of textual-only input we are able to leverage any existing pre-trained models that have text-embedding only input.

Table 1
Date Prompt Conditioning: A demonstration of the date-prompting and subsequent model outputs conditioned on the prefix year. The model output is between 0 and 1, where the input is considered positive only if the output is above 0.5. The example input text is taken from the *LongEval-Classification* dataset [1].

Example input	Output	Label	Orginal Year	Prefix Year
"year: 2013 text: I really do enjoy being single"	0.503	positive	2018	2017
"year: 2018 text: I really do enjoy being single"	0.510	positive	2018	2018
"year: 2023 text: I really do enjoy being single"	0.495	negatuve	2018	2023

3.3. Date-Prefix Augmentation

When creating pseudo-labels to train a student model we use an augmentation strategy that takes advantage of our date-prefixing. Namely, given an unlabeled sample and its timestamp we randomly replace the year in the timestamp with a year between 2013 and 2021. Where, the years 2013 and 2021 are the earliest and latest years found in the labeled datasets, respectively. We perform an ablation experiment (see Section 4) demonstrating that this augmentation strategy outperforms non-augmented self-labeling on the evaluation set.

3.4. Training and Evaluation

We use a single model trained using both the training and development sets for two epochs for inference on the test set. Model parameters using the Adam optimizer [27] with a constant learning rate of 1e-5 using the binary-cross-entropy loss. Performance is measured using the macro-averaged F1 score of the future samples.

4. Experiments

4.1. Experimental Setup

In this section, we will compare the performance of models trained with and without the proposed augmentation strategies for pseudo-label generation. Namely, we will use a trained teacher model to generate labels with and without date-prefix augmentation. Subsequently, we a student models on each of the two sets of pseudo labels for 6000 training steps. Finally, then compare the downstream performance of each model.

Models will only be provided labels for the training set and trained until saturation on the interim evaluation set. For our experiments, we report the macro-averaged F1 scores for each subset of the evaluation set. We will also report the Relative Performance Drop (RPD) [2] for comparison between short and long-term time differences with respect to model performance.

$$\text{RPD} = \frac{f_{t_j}^{score} - f_{t_0}^{score}}{f_{t_0}^{score}} \tag{1}$$

Table 2
Abalation Results: Results on the evaluation set, testing our self-labeling augmentation strategy. (*baseline*: only using gold labels, *+sl*: trained on pseudo-labels generated from *baseline*, *+ft*: fine-tuned on gold labels, *(aug)*: date-prefix augmentation, *(no-aug)*: no augmentation applied) We report the macro F1 score alongside the RPD between the various evaluation sets. The best results are highlighted

Method	F1 Within	F1 Short	F1 Long	RPD Within-Short	RPD Within-Long
baseline	0.7266	0.6725	0.6595	-0.0744	-0.0924
+sl(no-aug)	0.7213	0.6747	**0.6916**	-0.0646	**-0.0411**
+sl(no-aug)+ft	**0.7355**	0.6728	0.6728	-0.0852	-0.0852
+sl(aug)	0.7278	0.6749	0.6648	-0.0727	-0.0865
+sl(aug)+ft (ours)	0.7210	**0.6833**	0.6719	**-0.0532**	-0.0681

4.2. Results

We report the evaluation results of our experiments in Table 2. Indeed, we see an overall improvement in performance especially when we observe the ‘short’ evaluation set results when using our full framework. Additionally, the model using date-prefix augmentation gives by far the best RDP of -0.0532 with respect to the ‘within‘ and ‘short’ evaluation sets. Note that the non-augmented models gives the best RDP of -0.0411 with respect to the ‘within‘ and ‘long‘ evaluation sets. However, when finetunning this same model on the gold labels, the RPD more than doubles to -0.0852 and is much worse than our full framework with -0.0681. A similar drop in performance can be seen when observing the F1 score on the ‘long’ evaluation set. It appears that fine-tuning the non-augmented model with clean data incurs a significant drop in performance. However, it is clear that our proposed augmentation strategy can leverage the older labeled data and attain significant performance gains.

5. Conclusion

In this paper, we introduce a competitive framework for preserving the performance of sentiment analysis models across various temporal periods. We promote date-prefixing, as a straightforward solution to condition the output of pre-trained language models with the temporal context of input text. Furthermore, we build on the self-labeling framework developed by Agarwal and Nenkova [9]. Namely, given our date-prefix formatting, we can generate pseudo-labels conditioned on the temporal context of the input text. We verify the performance gains of our proposed system against self-labeling without our augmentation strategy in our ablation experiments. Altogether, our system yields competitive performance in overall score and attains the best RPD for the short evaluation set [3].

References

[1] R. Alkhalifa, I. Bilal, H. Borkakoty, J. Camacho-Collados, R. Deveaud, A. El-Ebshihy, L. Espinosa-Anke, G. Gonzalez-Saez, P. Galuščáková, L. Goeuriot, E. Kochkina, M. Liakata, D. Loureiro, H. Tayyar Madabushi, P. Mulhem, F. Piroi, M. Popel, C. Servan, A. Zubiaga,

Longeval: Longitudinal evaluation of model performance at clef 2023, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2023.

[2] R. Alkhalifa, E. Kochkina, A. Zubiaga, Opinions are made to be changed: Temporally adaptive stance classification, Proceedings of the 2021 Workshop on Open Challenges in Online Social Networks (2021).

[3] R. Alkhalifa, I. Bilal, H. Borkakoty, J. Camacho-Collados, R. Deveaud, A. El-Ebshihy, L. Espinosa-Anke, G. Gonzalez-Saez, P. Galuščáková, L. Goeuriot, E. Kochkina, M. Liakata, D. Loureiro, H. T. Madabushi, P. Mulhem, F. Piroi, M. Popel, C. Servan, A. Zubiaga, Overview of the clef-2023 longeval lab on longitudinal evaluation of model performance, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), Lecture Notes in Computer Science (LNCS), Springer, Thessaliniki, Greece, 2023.

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, ArXiv abs/1810.04805 (2019).

[5] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, ArXiv abs/1907.11692 (2019).

[6] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Annual Meeting of the Association for Computational Linguistics, 2019.

[7] B. Dhingra, J. R. Cole, J. M. Eisenschlos, D. Gillick, J. Eisenstein, W. W. Cohen, Time-aware language models as temporal knowledge bases, Transactions of the Association for Computational Linguistics 10 (2021) 257–273.

[8] K. Margatina, S. Wang, Y. Vyas, N. A. John, Y. Benajiba, M. Ballesteros, Dynamic benchmarking of masked language models on temporal concept drift with multiple views, in: Conference of the European Chapter of the Association for Computational Linguistics, 2023.

[9] O. Agarwal, A. Nenkova, Temporal effects on pre-trained models for language processing tasks, Transactions of the Association for Computational Linguistics 10 (2021) 904–921.

[10] A. Lazaridou, A. Kuncoro, E. Gribovskaya, D. Agrawal, A. Liska, T. Terzi, M. Gimenez, C. de Masson d'Autume, T. Kociský, S. Ruder, D. Yogatama, K. Cao, S. Young, P. Blunsom, Mind the gap: Assessing temporal generalization in neural language models, in: Neural Information Processing Systems, 2021.

[11] K. Luu, D. Khashabi, S. Gururangan, K. Mandyam, N. A. Smith, Time waits for no one! analysis and challenges of temporal misalignment, in: North American Chapter of the Association for Computational Linguistics, 2021.

[12] J. Pustejovsky, R. Knippen, J. Littman, R. Saurí, Temporal and event information in natural language text, Language Resources and Evaluation 39 (2005) 123–164.

[13] M. J. Zhang, E. Choi, Mitigating temporal misalignment by discarding outdated facts, 2023.

[14] J. Jang, S. Ye, S. Yang, J. Shin, J. Han, G. Kim, S. J. Choi, M. Seo, Towards continual knowledge learning of language models, ArXiv abs/2110.03215 (2021).

[15] D. Chen, A. Fisch, J. Weston, A. Bordes, Reading wikipedia to answer open-domain questions, in: Annual Meeting of the Association for Computational Linguistics, 2017.

[16] A. Livska, T. Kovcisk'y, E. Gribovskaya, T. Terzi, E. Sezener, D. Agrawal, C. de Masson d'Autume, T. Scholtes, M. Zaheer, S. Young, E. Gilsenan-McMahon, S. Austin, P. Blunsom, A. Lazaridou, Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models, in: International Conference on Machine Learning, 2022.

[17] D. Loureiro, F. Barbieri, L. Neves, L. E. Anke, J. Camacho-Collados, Timelms: Diachronic language models from twitter, in: Annual Meeting of the Association for Computational Linguistics, 2022.

[18] J. Jang, S. Ye, C. K. Lee, S. Yang, J. Shin, J. Han, G. Kim, M. Seo, Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models, in: Conference on Empirical Methods in Natural Language Processing, 2022.

[19] E. Strubell, A. Ganesh, A. McCallum, Energy and policy considerations for deep learning in nlp, ArXiv abs/1906.02243 (2019).

[20] G. Attanasio, D. Nozza, F. Bianchi, D. Hovy, Is it worth the (environmental) cost? limited evidence for temporal adaptation via continuous training, 2022.

[21] C. Raffel, N. M. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, ArXiv abs/1910.10683 (2019).

[22] S. Cao, L. Wang, Time-aware prompting for text generation, in: Conference on Empirical Methods in Natural Language Processing, 2022.

[23] R. Shams, Semi-supervised classification for natural language processing, ArXiv abs/1409.7612 (2014).

[24] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, N. A. Smith, Don't stop pretraining: Adapt language models to domains and tasks, ArXiv abs/2004.10964 (2020).

[25] A. Ushio, L. Neves, V. Silva, F. Barbieri, J. Camacho-Collados, Named entity recognition in twitter: A dataset and analysis on short-term temporal shifts, in: AACL, 2022.

[26] A. DeLucia, S. Wu, A. Mueller, C. A. Aguirre, P. Resnik, M. Dredze, Bernice: A multilingual pre-trained encoder for twitter, in: Conference on Empirical Methods in Natural Language Processing, 2022.

[27] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, CoRR abs/1412.6980 (2014).