Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 7352–7359
Marseille, 20-25 June 2022

Attention-Focused Adversarial Training for Robust Temporal Reasoning

Lis Kanashiro Pereira[1], Kevin Duh[2], Fei Cheng[3], Masayuki Asahara[4], Ichiro Kobayashi[5]
Ochanomizu University[1,5], John Hopkins University[2], Kyoto University[3]
National Institute for Japanese Language and Linguistics (NINJAL) and Tokyo University of Foreign Studies [4]
kanashiro.pereira@ocha.ac.jp, kevinduh@cs.jhu.edu, feicheng@i.kyoto-u.ac.jp
masayu-a@ninjal.ac.jp, koba@is.ocha.ac.jp

Abstract

We propose an enhanced adversarial training algorithm for fine-tuning transformer-based language models (i.e., RoBERTa) and apply it to the temporal reasoning task. Current adversarial training approaches for NLP add the adversarial perturbation only to the embedding layer, ignoring the other layers of the model, which might limit the generalization power of adversarial training. Instead, our algorithm searches for the best combination of layers to add the adversarial perturbation. We add the adversarial perturbation to multiple hidden states or attention representations of the model layers. Adding the perturbation to the attention representations performed best in our experiments. Our model can improve performance on several temporal reasoning benchmarks, and establishes new state-of-the-art results.

Keywords: adversarial training, robustness, temporal reasoning

1. Introduction

Although recent pre-trained language models such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019b) have achieved great success in a wide range of natural language processing (NLP) tasks, these models may still perform poorly on temporal reasoning scenarios. Ribeiro et al. (2020) has shown that such models often fail to make even simple temporal distinctions, for example, to distinguish the words *before* and *after*, resulting in degraded performance.

Following best practices from recent work on enhancing model generalization and robustness, we propose a model that effectively leverages pre-trained representations (i.e. RoBERTa) and adversarial training for robust temporal reasoning. More specifically, our main contributions are: 1) we propose an enhanced adversarial training algorithm for fine-tuning transformer-based language models that boosts the fine-tuning performance of RoBERTa. Our algorithm generates and adds the perturbation to a combination of layers during adversarial training. We propose to add the adversarial perturbation to multiple hidden states or attention vectors of the model layers. We hypothesize this might further encourage the model to generate more stable and better adversarial examples, and improve model generalization capability. Common adversarial training approaches for NLP add the perturbation only to the embedding layer, ignoring the other layers of the model (Zhu et al., 2019; Jiang et al., 2019; Liu et al., 2020a; Pereira et al., 2020); 2) we apply our model to several temporal reasoning tasks, as they often suffer from having limited training, and are challenging as they require temporal knowledge usually not explicitly stated in text; 3) we improve their state-of-the-art results on challenging temporal datasets such as MC-TACO, MATRES, and Story Cloze Task.

2. Model

In this paper, we focus on fine-tuning RoBERTa models (Liu et al., 2019b) in our investigation of adversarial training, as RoBERTa has proven very effective for a wide range of NLP tasks.

Adversarial training works as an online data augmentation method and can help improve model performance, especially in low-resource scenarios. It can also help improve model performance without increasing the model size, which is helpful in scenarios where computational resources are limited. Adversarial training has proven effective in improving model generalization and robustness in computer vision (Madry et al., 2017; Goodfellow et al., 2014) and more recently in natural language processing (NLP) (Zhu et al., 2019; Jiang et al., 2019; Cheng et al., 2019; Liu et al., 2020a; Pereira et al., 2020). It works by augmenting the input with a small perturbation that maximizes the adversarial loss:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D}[\max_{\delta} l(f(x+\delta;\theta),y)],$$

where the inner maximization can be solved by projected gradient descent (Madry et al., 2017). The $f(;\theta)$ represents the model parameterized by θ, x denotes the input, y are the associated labels, and δ represents the small perturbation added to x. Recently, adversarial training has been successfully applied to NLP as well (Zhu et al., 2019; Jiang et al., 2019; Pereira et al., 2020). The approaches to estimate δ can be divided into two categories: adversarial training that uses the label y (Zhu et al., 2019) and adversarial training that uses the model prediction $f(x;\theta)$, i.e. a "virtual" label (Miyato et al., 2018; Jiang et al., 2019).

In our work, we propose to enhance the ALICE (Pereira et al., 2020) algorithm. ALICE combines the two approaches to estimate the perturbation δ: one that uses the label y (Zhu et al., 2019) and another that uses the model prediction $f(x;\theta)$, i.e., a "virtual" label (Miyato et al., 2018; Jiang et al., 2019). The first one is to improve the robustness of our target label, by avoiding an increase in the error of the unperturbed inputs, while the second term enforces the smoothness of the model, encouraging the output of the model not to change much, when injecting a small perturbation to the input. The formula of ALICE is shown below:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D}[\max_{\delta_1} l(f(x+\delta_1;\theta),y)+ \alpha \max_{\delta_2} l(f(x+\delta_2;\theta),f(x;\theta))], \quad (1)$$

where δ_1 and δ_2 are two different perturbations, bounded by a general l_p norm ball, estimated by a fixed K steps of the gradient-based optimization approach and $p = \infty$. Effectively, the second term encourages smoothness in the input neighborhood, and α is a hyperparameter that controls the trade-off between standard errors and adversarial errors. ALICE has been originally proposed for the commonsense reasoning task, however, it is a general algorithm that can be applied to other tasks as well.

In our experiments, we show the applicability of ALICE to the temporal reasoning tasks described in Section 3.1 and Appendix A. In addition, we investigate which combination of layers is best for adding the perturbation during training. ALICE originally adds the perturbation only to the embedding layer. We show that adding the perturbation to a combination of the transformer's layers instead leads to better results. For all tasks in this work, an input text sequence is divided into sub-word units w_t, $t = 1,\ldots,T$. The tokenized input sequence is then transformed into embeddings, $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T \in \mathbb{R}^n$, through a token encoder, which combines a token embedding, a (token) position embedding and a segment embedding (*i.e.* which text span the token belongs to) by element-wise summation. The embedding layer is used as the input to multiple transformer layers (Vaswani et al., 2017) to generate the contextual representations, $\boldsymbol{z}_1^{layer},\ldots,\boldsymbol{z}_T^{layer} \in \mathbb{R}^d$, which are the representations of the hidden state or attention representations of an intermediate layer of the RoBERTa model. By attention representations, we mean the output of the layer normalization of the attention block, which is composed by multi-head attention, residual connection, and layer normalization (Kobayashi et al., 2021). We first set a maximum layer (among all RoBERTa layers, including the embedding layer) where the adversarial perturbation can be added. In each epoch, for each mini-batch selected, we first sample noise vectors δ_1 and δ_2 from $\mathcal{N}(0,\sigma^2 I)$, with mean 0 and variation of σ^2. A layer among the embedding layer and the maximum layer previously set is randomly chosen and the model performs adversarial steps from this layer by K gradient steps. The perturbed inputs are then constructed by adding the perturbations δ_1 and δ_2 to the hidden state vector or the attention vector of the randomly chosen layer. Specifically, the model first performs a forward pass up to the chosen layer, then the perturbations δ_1 and δ_2 are separately added to its hidden states or attention representations, generating two different perturbed inputs. The best layer combination is chosen by using a development set. We name our enhanced model **ML-ALICE** (**M**ulti-**L**ayer **ALICE**). The algorithm of ML-ALICE is depicted in Algorithm 1.

3. Experiments

We compare ML-ALICE with standard training and state-of-the-art adversarial training methods such as virtual adversarial training (SMART) (Jiang et al., 2019) and FreeLB (Zhu et al., 2019). We use the standard uncased RoBERTa$_{\text{BASE}}$ model (Devlin et al., 2018), unless noted otherwise. Due to the additional overhead incurred during training, adversarial methods are somewhat slower than standard training. Like SMART and FreeLB, ML-ALICE requires an additional K adversarial steps compared to standard training. In practice, $K = 1$ suffices for ML-ALICE and SMART, so they are just slightly slower (roughly 2-3 times compared to standard training). FreeLB, by contrast, typically requires 2-5 steps to attain good performance, so it is significantly slower.

3.1. Datasets and Evaluation Metrics

We evaluate our model on the following tasks: temporal ordering prediction task, temporal entailment, event duration prediction, temporal commonsense reasoning, and story cloze task. We use the following datasets, respectively: MATRES (Ning et al., 2018), TEA (Kober et al., 2019), TimeML (Pan et al., 2006), MC-TACO (Zhou et al., 2020), and Story Cloze Task (Mostafazadeh et al., 2017). An example from the MC-TACO dataset is shown below. The correct answer is in **bold**.

> *Paragraph*: Growing up on a farm near St. Paul, L. Mark Bailey didn't dream of becoming a judge.
>
> *Question*: How many years did it take for Mark to become a judge?
>
> a) 63 years & b) 7 weeks & c) **7 years**
>
> d) 7 seconds & e) 7 hours &

A detailed description of all datasets can be found on Appendix A. We evaluate the performance on TEA and MATRES in terms of accuracy and F1-score, and TimeML and Story Cloze Task in terms of accuracy. For the MC-TACO dataset, we report the exact match (EM) and F1 scores, following Zhou et al. (2019). EM

Algorithm 1 ML-ALICE : We explore adding the small adversarial perturbation to the hidden representations or self-attention representations of a layer of the model.

Input: T: the total number of iterations, $\mathcal{X} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$: the dataset, $f(x;\theta)$: the machine learning model parametrized by θ, σ^2: the variance of the random initialization of perturbation δ_1 and δ_2, δ_{1_r} and δ_{2_r}: the perturbations added to the hidden states or self-attention representations of layer r , ϵ: perturbation bound, K: the number of iterations for perturbation estimation, L: the number of transformer based model's layers, f^{layer}: the function that computes the hidden representations or attention representations of a given layer, $\boldsymbol{z}$: the hidden representations or self-attention representations of a layer of the model, η: the step size for updating perturbation, τ: the global learning rate, α: the smoothing proportion of adversarial training in the augmented learning objective, Π: the projection operation, and max_layer: the number of the maximum layer where the noise can be added during training.

for $t = 1, .., T$ **do**
  for $(x, y) \in \mathcal{X}$ **do**
    $\delta_1 \sim \mathcal{N}(0, \sigma^2 I)$
    $\delta_2 \sim \mathcal{N}(0, \sigma^2 I)$
    Generate a random integer $r \in \{1, ...,$max_layer$\}$
    for $m = 1, .., K$ **do**
      // x : forward pass to the last layer of the model
      for $layer = 1, 2, ..., L$ **do**
        $\boldsymbol{z} \leftarrow f^{layer}(\boldsymbol{z})$
        if $layer$ is r **then**
          $g_{adv} \leftarrow \nabla_{\delta_{1_r}} l(f(\boldsymbol{z} + \delta_{1_r}; \theta), f(\boldsymbol{z}; \theta))$
          $\delta_{1_r} \leftarrow \Pi_{\|\delta_{1_r}\|_\infty \leq \epsilon}(\delta_{1_r} + \eta g_{adv})$
          $g_{adv} \leftarrow \nabla_{\delta_{2_r}} l(f(\boldsymbol{z} + \delta_{2_r}; \theta), y_t)$
          $\delta_{2_r} \leftarrow \Pi_{\|\delta\|_\infty \leq \epsilon}(\delta_{2_r} + \eta g_{adv})$
        end if
      end for
    end for
    $g_\theta \leftarrow \nabla_\theta l(f(x + \delta_{1_r}; \theta), y)$
      $+\alpha \nabla_\theta l(f(x + \delta_{2_r}; \theta), f(x; \theta))$
    $\theta \leftarrow \theta - \tau g_\theta$
  end for
end for

Output: θ

	TEA		TimeML	MC-TACO		SCT	MATRES	
Model	Acc	F1	Acc	EM	F1	Acc	Acc	F1
Human	-	-	87.70	75.80	87.10	-		
STD (RoBERTa_BASE)	95.20	89.40	80.86	39.79	68.63	92.95	90.54	87.80
FreeLB (Zhu et al., 2019)	95.75	90.62	82.75	44.37	71.52	92.68	90.54	87.78
SMART (Jiang et al., 2019)	95.32	89.77	82.25	46.77	73.07	92.89	91.26	88.77
ALICE (Pereira et al., 2020)	95.63	90.35	82.35	47.00	73.04	93.43	89.54	86.80
ML-ALICE (hidden)	95.69	90.52	83.35	49.25	**74.78**	93.53	90.26	87.59
ML-ALICE (attention)	**96.72**	**92.80**	**83.94**	**49.77**	73.93	**94.07**	**91.40**	**88.97**
STD (RoBERTa_LARGE)	95.99	91.11	81.06	**51.05**	**76.85**	**96.37**	91.12	88.93
T5-3B (Kaddari et al., 2020)	-	-	-	59.08	79.46	-	-	-
TacoML (Zhou et al., 2020)	-	-	81.70	-	-	-	-	-
SYMTIME (Zhou et al., 2021)	-		-	-	-	-	-	87.30
GDIN (Tian et al., 2020)	-	-	-	-	-	91.90	-	-

Table 1 Test results of TEA, TimeML, Story Cloze Task (SCT), MC-TACO, and MATRES. The best results are in **bold**. STD denotes the standard fine-tuning procedure where we fine-tune RoBERTa on each task specific temporal reasoning dataset. ML-ALICE denotes our proposed models. ML-ALICE (hidden states) denotes the model that uses the best layer combination to add the perturbation to the hidden state vectors, and ML-ALICE (attention) denotes the model that uses the best layer combination to add the perturbation to the attention weight vectors. Note that STD, FreeLB, SMART, ALICE, and all ML-ALICE models use $\text{RoBERTa}_{\text{BASE}}$ as the text encoder unless stated otherwise, and for a fair comparison, all these results are produced by ourselves. All values marked in bold are significantly higher compared to the best overall baseline, ALICE, measured by t-test at p-value of 0.05.

measures how many questions a system correctly labeled all candidate answers, while F1 measures the av-

erage overlap between one's predictions and the ground truth.

3.2. Implementation Details

Our model implementation is based on the MT-DNN framework (Liu et al., 2019a; Liu et al., 2020b). We use RoBERTa$_{\text{BASE}}$ (Liu et al., 2019b) as the text encoder. We used ADAM (Kingma and Ba, 2015) as our optimizer with a learning rate in the range $\in \{9\times10^{-6}, 1\times10^{-5}\}$ and a batch size in the range $\in \{16, 32, 64\}$. The maximum number of epochs was set to 10. A linear learning rate decay schedule with warm-up over 0.1 was used unless stated otherwise. To avoid gradient exploding, we clipped the gradient norm within 1. All the texts were tokenized using WordPiece and were chopped to spans no longer than 512 tokens. We also set the dropout rate of all the task-specific layers as 0.3. During adversarial training, we follow (Jiang et al., 2019) and set the perturbation size to 1×10^{-5}, the step size to 1×10^{-3}, and to 1×10^{-5} the variance for initializing perturbation. We search the regularization weight α in {0.01, 0.1, 1}. We set the number of projected gradient steps to 1. For the posterior regularization, we use the Jensen-Shannon divergence, as in (Cheng et al., 2021).

3.3. Main Results

We present our results in Table 1. We compare our model, ML-ALICE , with other state-of-the-art models. Overall, the adversarial methods, i.e., FreeLB, SMART, ALICE and ML-ALICE , were able to outperform the standard fine-tuning approach (STD) and the other baselines, without using any additional knowledge source, and without using any additional dataset other than the target task datasets. These results suggest that adversarial training leads to a more robust model and helps generalize better on unseen data.

All ML-ALICE models were able to outperform the other baselines. ML-ALICE consistently outperforms both standard training and the strong adversarial training methods of FreeLB and SMART. This indicates that adding the adversarial perturbation to the other layers of the model in addition to the embedding layer can improve the model generalization capability. Overall, ML-ALICE (attention) obtained better performance. We hypothesize this is because the layer normalization has larger impact in the internal representation, as also shown by Kobayashi et al. (2021). Remarkably, RoBERTa$_{\text{BASE}}$ with ML-ALICE performs on par with RoBERTa$_{\text{LARGE}}$ with standard training overall, and outperforms the latter on most datasets.

In all of our experiments, adding the adversarial perturbation to the lower layers of the model (i.e., layers 0 to 2) performed best. We hypothesize this is because they are more close to the original input and learn more more robust representations compared to the embedding layer, being less sensitive to noise.

Model	**TimeML**	**MC-TACO**		**SCT**
	Acc	EM	F1	Acc
Standard	49.35	10.59	32.27	87.28
FreeLB	41.87	12.53	24.59	88.99
SMART	42.77	9.38	27.11	86.21
ALICE	43.77	**15.02**	21.11	91.24
ML-ALICE	**55.23**	11.48	**43.64**	**91.98**

Table 2Comparison of standard and adversarial training in zero-shot evaluation on various temporal datasets, where the standard RoBERTa$_{\text{BASE}}$ model is fine-tuned on the CosmosQA dataset. All values marked in bold are significantly higher compared to the best overall baseline, ALICE, measured by t-test at p-value of 0.05.

3.4. Zero-Shot Learning on Commonsense Reasoning

Next, we compare standard and adversarial training in generalizability to out-domain datasets. Specifically, we fine-tune RoBERTa$_{\text{BASE}}$ on a commonsense reasoning dataset, CosmosQA (Huang et al., 2019), and evaluate on the temporal datasets. Since the commonsense reasoning task commonly involves reasoning about temporal events, e.g. what event(s)might happen before or after the current event, we hypothesize that those tasks might highly benefit from it. It has 35,888 questions on 21,886 distinct contexts taken from blogs of personal narratives. Each question has four answer candidates, one of which is correct. An example from this dataset is below. The correct answer is in **bold**.

> *Paragraph*: Did some errands today. My prime objectives were to get textbooks, find a computer lab, find career services, get some groceries, turn in payment plan application, and find out when KEES money kicks in. I think it acts as a refund at the end of the semester at Murray, but I would be quite happy if it would work now.
>
> *Question*: What happens after I get the refund?
>
> *Option 1*: **I can pay my bills.**
>
> *Option 2*: I can relax.
>
> *Option 3*: I can sleep.
>
> *Option 4*: None of the above choices.

See Table 2 for the results. ML-ALICE outperforms standard training and state-of-the-art adversarial training methods. Interestingly, the gains are particularly pronounced on the smallest dataset, TimeML, which has only 1,248 training samples. This provides additional evidence that ML-ALICE is especially effective in enhancing generalizability in data-constrained settings.

4. Conclusion

We present an enhanced adversarial training algorithm for fine-tuning transformer-based language mod-

els (i.e., RoBERTa) and apply it to the temporal reasoning task. Our ML-ALICE algorithm is simple yet effective in improving model generalization for various temporal reasoning tasks, especially in zero-shot learning setting. Our experiments demonstrated that it achieves competitive results without relying on any additional resource other than the target task dataset. Future directions include: applying ML-ALICE in pretraining and other NLP tasks, e.g., sequence labeling.

Acknowledgements

We thank the reviewers for their helpful feedback. This work has been supported by the project KAKENHI ID: 18H05521 and by project KAKENHI ID: 21K17802.

5. Bibliographical References

Cheng, Y., Jiang, L., and Macherey, W. (2019). Robust neural machine translation with doubly adversarial inputs.

Cheng, H., Liu, X., Pereira, L., Yu, Y., and Gao, J. (2021). Posterior differential regularization with f-divergence for improving model robustness. In *In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1078–1089. June 6–11, 2021.*

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). In *Bert: Pre-training of deep bidirectional transformers for language understanding*.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Graff, D. (2002). The aquaint corpus of english news text:[content copyright] portions© 1998-2000 new york times, inc.,© 1998-2000 associated press, inc.,© 1996-2000 xinhua news service. Linguistic Data Consortium.

Huang, L., Bras, R. L., Bhagavatula, C., and Choi, Y. (2019). Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2391–2401.

Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Zhao, T. (2019). Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv preprint arXiv:1911.03437*.

Kaddari, Z., Mellah, Y., Berrich, J., Bouchentouf, T., and Belkasmi, M. G. (2020). Applying the t5 language model and duration units normalization to address temporal common sense understanding on the mctaco dataset.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *ICLR (Poster) 2015*.

Kobayashi, G., Kuribayashi, T., Yokoi, S., and Inui, K. (2021). Incorporating residual and normalization layers into analysis of masked language models. *arXiv preprint arXiv:2109.07152*.

Kober, T., de Vroe, S. B., and Steedman, M. (2019). Temporal and aspectual entailment.

Liu, X., He, P., Chen, W., and Gao, J. (2019a). Multi-task deep neural networks for natural language understanding. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Liu, X., Cheng, H., He, P., Chen, W., Wang, Y., Poon, H., and Gao, J. (2020a). Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*.

Liu, X., Wang, Y., Ji, J., Cheng, H., Zhu, X., Awa, E., He, P., Chen, W., Poon, H., Cao, G., and Gao, J. (2020b). The microsoft toolkit of multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:2002.07972*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. (2018). Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Mostafazadeh, N., Roth, M. abd Louis, A., Chambers, N., and Allen, J. (2017). Lsdsem 2017 shared task: The story cloze test. In *In Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics (pp. 46-51).*

Ning, Q., Wu, H., and Roth, D. (2018). A multi-axis annotation scheme for event temporal relations. In *arXiv preprint arXiv:1804.07828*.

Pan, F., Mulkar-Mehta, R., and Hobbs, J. R. (2006). Extending timeml with typical durations of events. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 38–45.

Pereira, L., Liu, X., Cheng, F., Asahara, M., and Kobayashi, I. (2020). Adversarial training for commonsense inference. *arXiv preprint arXiv:2005.08156*.

Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., et al. (2003). The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

Ribeiro, M. T., Wu, T., Guestrin, C., and Singh, S. (2020). Beyond accuracy: Behavioral testing of nlp models with checklist. In *arXiv preprint arXiv:2005.04118*.

Saurí, R., Littman, J., Knippen, B., Gaizauskas, R., Set-

zer, A., and Pustejovsky, J. (2006). Timeml annotation guidelines. volume 1, page 31.

Tian, Z., Zhang, Y., Liu, K., Zhao, J., Jia, Y., and Sheng, Z. (2020). Scene restoring for narrative machine reading comprehension.

UzZaman, N., Llorens, H., Derczynski, L., Allen, J., Verhagen, M., and Pustejovsky, J. (2013). Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Zhou, B., Khashabi, D., Ning, Q., and Roth, D. (2019). "going on a vacation" takes longer than" going for a walk": A study of temporal commonsense understanding. 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 3363–3369.

Zhou, B., Ning, Q., Khashabi, D., and Roth, D. (2020). Temporal common sense acquisition with minimal supervision. In *arXiv preprint arXiv:2005.04304*.

Zhou, B., Richardson, K., Ning, Q., Khot, T., Sabharwal, A., and Roth, D. (2021). Temporal reasoning on implicit events from distant supervision. In *In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1361–1371 June 6–11, 2021.*

Zhu, C., Cheng, Y., Gan, Z., Sun, S., Goldstein, T., and Liu, J. (2019). Freelb: Enhanced adversarial training for language understanding. In *arXiv preprint arXiv:1909.11764*.

A. Evaluation Datasets

In this section, we describe the temporal reasoning tasks we tackle in this work. All tasks are challenging since they require deep understanding of the temporal properties of language.

Event Ordering Prediction Task: This task involves predicting the temporal relationship between a pair of input events in a span of text. We use the MATRES dataset (Ning et al., 2018). It originally contains 13,577 pairs of events annotated with a temporal relation (BEFORE, AFTER, EQUAL, VAGUE). The temporal annotations are performed on 256 English documents (and 20 more for evaluation) from the TimeBank (Pustejovsky et al., 2003), AQUAINT (Graff, 2002) and Platinum (UzZaman et al., 2013) datasets. An example of a sentence with two events (in bold) that hold the BEFORE relation is below:

> At one point, when it **(e1:became)** clear controllers could not contact the plane, someone **(e2:said)** a prayer.

We follow zhou2021distant, and we train and evaluate only the instances with a label of either "BEFORE" or "AFTER".

Temporal Entailment: this task requires models to correctly determine the internal and external temporal structure of predications when performing natural language inference. We use the Temporal and Aspectual entailment (TEA, (Kober et al., 2019)). This dataset contains pairs of short sentences with the same argument structure that differ in tense and aspect of the main verb, and follows a binary label annotation scheme (entailment vs. non-entailment). Examples from this dataset where are shown below:

> Steve married Elizabeth. Steve is proposing to Elizabeth. Label: *not entailment*
>
> Steve married Elizabeth. Steve was proposing to Elizabeth. Label: *entailment*

Event Duration Prediction Task: This task consists of deciding whether a given event has a duration longer or shorter than a day. We use TimeML (Saurí et al., 2006; Pan et al., 2006), a dataset with event duration annotated as lower and upper bounds. An example of a sentence with an event (in bold) that has a duration shorter than a day is shown below:

> In Singapore, stocks **hit** a five year low.

Temporal Commonsense Reasoning Task: This task focuses on temporal commonsense reasoning. We use the MC-TACO (Zhou et al., 2019) dataset. It considers five temporal properties: (1) duration (how long an event takes), (2) temporal ordering (typical order of events), (3) typical time (when an event occurs), (4) frequency (how often an event occurs), and (5) stationarity (whether a state is maintained for a very long time or indefinitely). It contains 13k tuples, each consisting of a sentence, a question, and a candidate answer, that should be judged as plausible or not. The sentences are taken from different sources such as news, Wikipedia, and textbooks. An example from the dataset is below. The correct answer is in **bold**.

> *Paragraph*: Growing up on a farm near St. Paul, L. Mark Bailey didn't dream of becoming a judge.
>
> *Question*: How many years did it take for Mark to become a judge?
>
> a) 63 years & b) 7 weeks & c) **7 years**
>
> d) 7 seconds & e) 7 hours &

Story Cloze Task (SCT): This task involves choosing an ending to a story. We use the Story Cloze Task dataset (Mostafazadeh et al., 2017), where the task is to choose the correct ending, among two choices, to a 4-sentence story. It captures a rich set of causal and temporal commonsense relations between daily events. An example from the dataset is below. The correct answer is in **bold**.

> *Story*: Danny bought a boat. His nearby marina was having a race. He decided to enter. Danny and his best friend manned the boat.
>
> a) Danny decided to go to sleep.
>
> b) **They prepared for the start of the race.**

The English datasets used in our experiments are summarized in Table 3.

For TimeML, we follow the train and test splits as in (Zhou et al., 2020). For MCTACO, we follow zhou2019going. For the MATRES dataset, we follow ning2018multi. Moreover, following (Zhou et al., 2021), we train and evaluate only the instances with a label of either "before" or "after", which accounts for about 80% of all instances. For the Story Cloze Task, we use the 2016 and 2018 data releases after removing duplicates. We set 20% of the TimeML, and Story Cloze Task training data as the development set to tune the hyperparameters. For the MC-TACO dataset, no training set is available. Following zhou2019going, we use the dev set for fine-tuning the model. We use 20% of this data for fine-tuning the parameters.

Dataset	#Train	#Test	#Label	Metrics
TEA	10,540	1,646	2	Accuracy & F1-score
TimeML	1,248	1,003	2	Accuracy
SCT	1,571	1,871	2	Accuracy
MATRES	10,906	698	2	Accuracy & F1-score
MC-TACO	3,783	9,442	2	F1-Score & Exact Match (EM)

Table 3 Summary of the five English evaluation datasets: TEA, TimeML, MATRES, Story Cloze Task (SCT), and MC-TACO.