On Data Augmentation and Consistency-based Semi-supervised Relation Extraction

Komal K. Teru
Vanguard AI Research (VAIR)
The Vanguard Group
komal_teru@vanguard.com

Abstract

To improve the sample efficiency of the Relation extraction (RE) models, semi-supervised learning (SSL) methods aim to leverage unlabelled data in addition to learning from limited labelled data points. Recently, strong data augmentation combined with consistency-based semi-supervised learning methods have advanced the state of the art in several SSL tasks. However, adapting these methods to the RE task has been challenging due to the difficulty of data augmentation for RE. In this work, we leverage the recent advances in controlled text generation to perform high quality data augmentation for the RE task. We further introduce small but significant changes to model architecture that allows for generation of more training data by interpolating different data points in their latent space. These data augmentations along with consistency training result in very competitive results for semi-supervised relation extraction on four benchmark datasets.

1 Introduction

Relation extraction (RE) task aims to discover the semantic relation between a given head entity and tail entity based on the context in the input sentence. For example, given a sentence "*The battle led to panic on the frontier, and settlers in the surrounding counties fled.*", the goal is to extract the `Cause-Effect` relation between the head entity '**battle**' and the tail entity '**panic**'. The models for RE task typically require large amounts of labelled data to give production-ready performance. A common strategy to improve the sample efficiency of machine learning models is semi-supervised learning methods which leverage easily accessible unlabelled data to improve the overall performance. While there are several paradigms of semi-supervised learning methods, consistency training based methods have advanced the state of the art in several SSL tasks Ghosh and Thiery (2021). These methods can typically reach performances that are comparable to their fully supervised counterparts while using only a fraction of labelled data points.

Recently, strong data augmentation combined with consistency training algorithms have shown great success, even surpassing fully supervised models, in low-data settings of various tasks Xie et al. (2020). Adapting these methods to the task of relation extraction has been challenging due to the difficulty of data augmentation for RE task. This is because, in addition to the input sentence, each data point also consists of a head entity and a tail entity contained in the input sentence. Typical data augmentation techniques used in NLP such as back-translation, synonym-replacement, language-model based augmentation, etc. Feng et al. (2021) can not be easily applied to such 'structured' input as they do not guarantee the integrity of either a) the entities in the input sentence or, b) the meaning of the input sentence itself. Figure 1 in the Appendix shows that using synonym-replacement and vanilla back-translation (BT) methods Sugiyama and Yoshinaga (2019) the entities themselves could be paraphrased or replaced. Matching the new and the old entities is a whole problem in itself. In the language-model based augmentation method Anaby-Tavor et al. (2020), the semantic meaning of the

INTERPOLATE workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Input sentence: The **battle** led to **panic** on the frontier, and settlers in the surrounding counties fled.

Synonym-replacement: The struggle cause scare on the frontier, and settlers in the surrounding counties fly.

LM-based augmentation: The **battle** reduced **panic** on the frontier, and settlers in the surrounding counties relaxed.

Vanilla BT: The war caused **panic** at the border, and residents of the nearby counties fled.

Our constrained BT: The **battle** sparked **panic** at the border, with residents fleeing in surrounding counties.

Figure 1: Different data augmentation techniques applied to a sample datapoint from SemEval dataset. Existing methods replace the head/tail entities, change the original meaning or do not give very fluent paraphrases.

input sentence changes altogether, which makes it difficult to employ consistency training. **Present work**. In this work, we leverage the recent advances in controlled text generation to perform high quality data augmentation for the relation extraction task that not only keeps the meaning and the head/tail entities intact but also produces fluent and diverse data points. In particular, we modify back-translation to leverage lexically constrained decoding strategies Post and Vilar (2018); Hu et al. (2019) in order to obtain paraphrased sentences while retaining the head and the tail entities. We further propose novel modifications to the widely popular relation extraction model architecture, that allows for generation of more samples by interpolating different data points in their latent space, a trick that has been very successful in other domains and tasks Berthelot et al. (2019); Chen et al. (2020b,a). Additionally, we leverage the entity types of the head and the tail entities, when available, in a way that effectively exploits the knowledge embedded in pre-trained language models. These data augmentations, when applied to unlabelled data, let us employ consistency training techniques to achieve very competitive results for semi-supervised relation extraction on four benchmark datasets. To the best of our knowledge, this is the first study to apply and show the merit of data augmentation and consistency training for semi-supervised relation extraction task.

2 Proposed approach

In this work, we focus on the sentence-level relation extraction task, i.e., given a *relation statement* $\mathbf{x} : (\mathbf{s}, e_h, e_t)$ consisting of a sentence, $\mathbf{s}$, a head entity, e_h, and a tail entity, e_t (both the entities are mentioned in the given sentence $\mathbf{s}$), the goal is to predict a relation $r \in \mathcal{R} \cup \{\text{NA}\}$ between the head and the tail entity, where $\mathcal{R}$ is a pre-defined set of relations. If the sentence does not express any relation from the set $\mathcal{R}$ between the two entities, then the relation statement $\mathbf{x}$ is accordingly labelled NA. This is typically done by learning a relation encoder model $\mathcal{F}_\theta : \mathbf{x} \mapsto \mathbf{h}_r$ that maps an input relation statement, $\mathbf{x}$, to a fixed length vector $\mathbf{h}_r$ that represents the relation expressed in $\mathbf{s}$ between e_h and e_t. This relation representation, $\mathbf{h}_r$, is then classified to a relation $r \in \mathcal{R} \cup \{\text{NA}\}$ via an MLP classifier. In our approach we build on the base model architecture described in CITEand implement two different data augmentation techniques and employ consistency training to obtain state-of-the-art performance.

Constrained back-translation. Back-translation Edunov et al. (2018) generates diverse and fluent augmentations while retaining the global semantics of the original input. Specifically, one translates a given text into an intermediate language, say, German, and translates it back to the source language, say English. Applying this back-translation technique in a vanilla fashion is not possible for RE task because one has little control over the retention of the head and tail entities (Figure 1). Thus, when translating back to the source language from the intermediate language we perform lexically-constrained decoding Hu et al. (2019), i.e., force the inclusion of pre-specified words and phrases–positive constraint set–in the output. In our case the original head and tail entity words/phrases make up this positive constraints set. [1]

[1] We use German and Russian as intermediate languages and use the pre-trained WMT'19 English-German and English-Russian translation models (in both directions) and their implementations provided by Ott et al. (2019).

Latent-space interpolation. Here, we adapt a mixup-based data augmentation technique to the RE task. As done in previous works (Chen et al., 2020b,a), we sample two random data points: $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$, where $\mathbf{x}$ and $\mathbf{y}$ denote the relation statement and the corresponding relation label. We then follow Chen et al. (2020b) to linearly interpolate latent representations of $\mathbf{x}$ and $\mathbf{x}'$ and obtain the encoder output representation of a *virtual* input $\tilde{\mathbf{x}}$, i.e., $\tilde{\mathbf{h}}^L = \mathcal{F}_\theta(\tilde{\mathbf{x}})$. We denote this whole mix-up operation as $\tilde{\mathbf{h}}^L := \text{MixRE}(\mathbf{x}, \mathbf{x}')$.

Now, for the RE task we need to extract a fixed-length relation representation from the encoder output representation of the entire input sequence. The traditional way to do this for RE task is by concatenating the encoder output representations of the entity marker tokens `[E1]` and `[E2]`. However, it is challenging to do this for a virtual sample, $\tilde{\mathbf{x}}$, as the entity markers are not clearly defined in this case. We thus modify the relation representation to be the encoder output representation of the `[CLS]` token. However, Baldini Soares et al. (2019) have shown this choice to be sub-optimal compared to concatenation of marker tokens. This is because the marker token representations provide direct access to the contextual information of the respective entities. Although the `[CLS]` token, in theory, has access to the entire context of the sentence, it might be difficult to capture the nuances like the head entity type, tail entity type, and the contextual information around the two entities all in a single vector.

On the other hand, entity type information is easily accessible in most RE benchmarks[2]. So, to compensate for the sub-optimal choice of using `[CLS]` token representation as the relation representation, we modify how we represent the entity spans in the input token to more effectively use the easily accessible entity type information. In particular, we note that the entity type labels can trivially be mapped to tokens from any pre-trained language model's vocabulary. For example, entity types like `PERSON` and `STATE_OR_PROVINCE` can be tokenized into a word/phrase like 'person' and 'state or province', respectively. Instead of using special marker tokens like `[E1]` and `[E2]`, we prepend the entity spans in the input sequence with the word/phrases corresponding to their respective types and enclose these 'type-words' in punctuation marks Zhou and Chen (2021).

We train our model with these mixup-based augmentations, back-translated augmentations, and consistency training following the same training procedure as described in Chen et al. (2020b). Detailed description of the whole training algorithm is provided in Appendix A.

3 Experiments

We perform experiments on four benchmark datasets for sentence-level RE: SemEval 2010 Task 8 (SemEval) Hendrickx et al. (2010), the TAC Relation Extraction Dataset (TACRED) Zhang et al. (2017), RE-TACRED Stoica et al. (2021), and KBP37 Zhang and Wang (2015). We compare MixRE with three state-of-the-art models that are representative of the existing class of methods for SSRE: MRefG Li et al. (2021), MetaSRE Hu et al. (2021a), and GradLRE Hu et al. (2021b). MetaSRE and GradLRE are two of the strongest methods in the widely adapted *self-training* methods for SSRE. Detailed description of the datasets and baselines is provided in Appendix B.

Implementation details. We follow the established setting to use stratified sampling to divide the training set into various proportions of labelled and unlabelled sets so that the relation label distribution remains the same across all subsets. Following existing work, we sample 5%, 10%, and 30% of the training set as labelled data for the SemEval and KBP37 datasets, and 3%, 10%, and 15% of the training set as labelled data for TACRED and RE-TACRED datasets. For all datasets and experiments, unless otherwise specified, we sample 50% of the training set as the unlabelled set. For TACRED and SemEval datasets we take the performance numbers of all baseline models reported by Hu et al. (2021b). For other datasets, we re-run the models with their best configuration as provided in their respective implementations. Full implementation details can be found in Appendix B.

3.1 Main Results

Table 1 shows F1 results of all baseline models and our proposed model, MixRE, on the four datasets when leveraging various amounts of labelled data and 50% unlabelled data. We report the mean and

[2] From new datasets/applications viewpoint, when entities are identified in a piece of text it is safe to assume that their types would also be identified.

Table 1: F1 score with various amounts of labelled data and 50% unlabelled data. Mean and standard deviation of 5 different runs is reported. Best performance on each configuration is bolded and second best is underlined.

	TACRED			KBP37		
%labelled Data	3%	10%	15%	5%	10%	30%
MRefG	43.81± 1.44	55.42± 1.40	58.21± 0.71	-	-	-
MetaSRE	46.16± 0.74	56.95± 0.33	58.94± 0.31	59.29± 0.92	61.83± 0.21	63.51± 0.69
GradLRE	47.37± 0.74	58.20± 0.33	59.93± 0.31	59.98± 0.37	62.67± 0.54	**66.41**± 0.28
MixRE(ours)	**55.80**± 1.33	**61.30**± 0.70	**63.07**± 0.93	**60.84**± 0.40	**63.82**± 0.71	**66.46**± 0.69

	RE-TACRED			SemEval		
%labelled Data	3%	10%	15%	5%	10%	30%
MRefG	-	-	-	75.48± 1.34	77.96± 0.90	83.24± 0.71
MetaSRE	44.42± 3.02	58.71± 1.70	61.71± 3.70	78.33± 0.92	80.09± 0.78	84.81± 0.44
GradLRE	61.22± 0.58	74.03± 1.74	**79.46**± 0.82	**79.65**± 0.68	**81.69**± 0.57	**85.52**± 0.34
MixRE(ours)	**71.33**± 1.22	**77.94**± 0.59	**79.76**± 0.47	77.58± 0.59	81.13± 0.82	**85.51**± 0.38

standard deviation of 5 different runs (with different seeds) of training and testing. MixRE gives state-of-the-art performance on 10 out of 12 different configurations across all four datasets. This reinforces the importance of consistency regularization beyond the currently popular self-training methods for SSRE. Interestingly, the performance gains are significantly higher for TACRED and RE-TACRED datasets–we see an average improvement of as much as 17% when trained on 3% labelled data. This can be attributed to the fact that entity type information is available for these datasets and entity type markers are very effective in exploiting the knowledge embedded in the pre-trained language models. We revisit this observation in our ablation studies (Section 3.2) where we concretely establish the benefits of using entity type markers.

3.2 Analysis and discussion

We first conduct experiments to empirically demonstrate the effectiveness of three components of our proposed model: i) data augmentation by latent space interpolation (Mix-DA), ii) data augmentation by constrained back-translation (BT-DA), and iii) entity type markers (ET). In Table 2, we report the mean F1 score of five different runs for different variations of our model by removing a certain combination of these components. As can be seen from Table 2, each of these components, contributes to the overall success of MixRE. For contribution of just the Mix-DA: we compare i) row `a` v/s row `c`, and ii) row `b` v/s row `d`. All comparisons show positive improvement. For contribution of just the BT-DA: we compare i) row `a` v/s row `b`, and ii) row `c` v/s row `d`. We note that BT-DA results only in marginal improvements in most cases. Upon closer inspection we note that the constrained-decoding algorithms we implement for BT-DA are actually not perfect, especially when combined with translation models. It sometimes misses the constraints and sometimes falls into repetitive loops in an attempt to satisfy the constraint. With the ever-improving language generation capabilities, we believe the quality of data augmentation will only improve with time and result in more significant performance improvements. For contribution of both DA techniques together: we compare row `a` v/s row `d`. All comparisons show significant improvements with data augmentation. The contribution of entity type markers can be noted in TACRED and RE-TACRED datasets. We see an average drop of 5.4% in F1 score across all 8 comparisons.

Next, we examine the effect of using different amounts of unlabelled data. In Figure 2, we report the average F1 score for different models trained with different amounts of unlabelled data and 10% labelled data. MixRE outperforms the baselines in all settings except on SemEval dataset, and, interestingly, the performance only marginally changes with the change in the amount of unlabelled data. Note that we train the models until the performance on the validation set stops improving for more than 5 epochs. Hence, MixRE generates, in principle, an infinite amount of unlabelled data via the mixup strategy. Coupled with the fact that the label distribution remains the same in all settings, adding more unlabelled data does not seem to add a lot of new information. This explains why the model performance is relatively insensitive to changing amounts of unlabelled data. This also implies that MixRE can leverage low amounts of unlabelled data better than the baselines.

Table 2: Ablation results on all datasets using 10% labelled set and 50% unlabelled set. Mean and standard deviation of 5 different runs is reported.

	Mix-DA	BT-DA	ET	TACRED	RE-T	KBP37*	SemEval*
a)	✓	✓	✓	61.30± 0.70	77.94± 0.59	63.82± 0.71	81.13± 0.82
			✗	56.82± 0.64	75.11± 1.16		
b)	✓	✗	✓	60.81± 1.31	77.77± 0.96	63.48± 0.53	79.71± 0.83
			✗	56.35± 0.97	74.67± 1.04		
c)	✗	✓	✓	59.65± 0.92	76.80± 0.98	62.64± 0.69	79.17± 1.64
			✗	55.52± 0.89	73.78± 1.34		
d)	✗	✗	✓	58.96± 1.21	77.25± 0.70	63.14± 0.90	79.20± 0.32
			✗	55.25± 1.53	74.58± 0.91		

* these datasets do not have entity type information

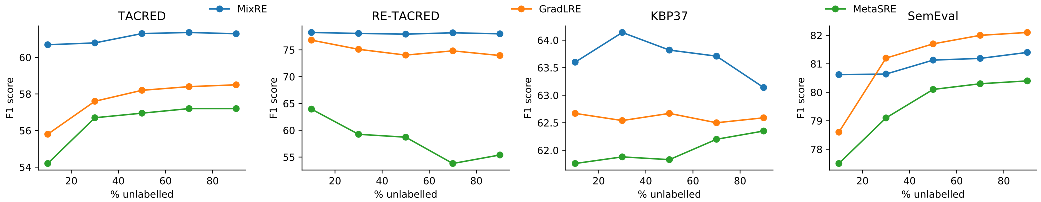

Figure 2: F1 Performance with various unlabelled data and 10% labelled data

Next, in Figure 3 we show how the performance of MixRE changes with a change in the mean of the Beta distribution from which λ is sampled on each iteration. Note that a value near 0 and 1 for λ means the augmented *virtual* data point will be closer to one of the underlying data points. As we get closer to 0.5 the virtual data points get further from the original data manifold and become more 'novel'. On TACRED and RE-TACRED datasets the performance peaks at $\mathrm{E}(\lambda) = 0.15$(or 0.85) and drops in the mid-values. This can be interpreted as: adding datapoints far from the original data manifold is detrimental for these datasets. Interestingly, on KBP37 and SemEval the pattern inverts, i.e., the performance increases as $\mathrm{E}(\lambda)$ towards 0.5, implying that more 'novel' augmentations help for these datasets.

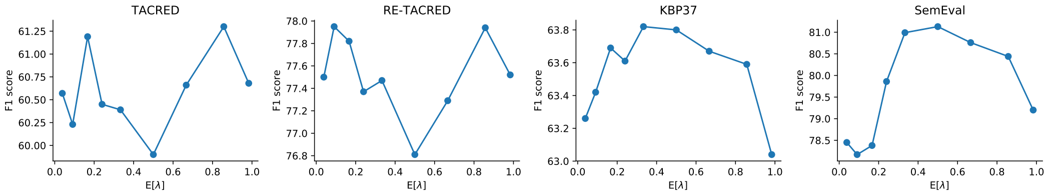

Figure 3: F1 Performance of MixRE with 50% unlabelled and 10% labelled data with changing mixing coefficient λ

4 Conclusion

In this paper, we propose a consistency training based semi-supervised algorithm for relation extraction and empirically show the merit of this class of methods in comparison to the current state-of-the-art *self-training* class of methods. This shows there is promise in a potential future direction where one could bootstrap the self-training methods with consistency training as done in some previous works on vision tasks Pham et al. (2021). Additionally, we show how the entity type information, when available, can result in massive performance boosts in the semi-supervised scenario. This is important because in most practical use cases when entities have already been identified, the entity type information is easy available and could be effectively leveraged in the proposed fashion.

Acknowledgements

The author would like to thank the reviewers for their thoughtful feedback, the entire VAiR team at Vanguard for their support, and especially David Landsman for their valuable input and discussion on the initial drafts of this work.

Disclaimer

This material is provided for informational purposes only and is not intended to be investment advice or a recommendation to take any particular investment action.

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7383–7390.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 32.

Jiaao Chen, Zhenghui Wang, Ran Tian, Zichao Yang, and Diyi Yang. 2020a. Local additivity based data augmentation for semi-supervised NER. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1241–1251, Online. Association for Computational Linguistics.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020b. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. 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.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Atin Ghosh and Alexandre H. Thiery. 2021. On data-augmentation and consistency-based semi-supervised learning. In *International Conference on Learning Representations*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved lexically constrained decoding for translation and monolingual rewriting. 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 839–850, Minneapolis, Minnesota. Association for Computational Linguistics.

Xuming Hu, Chenwei Zhang, Fukun Ma, Chenyao Liu, Lijie Wen, and Philip S. Yu. 2021a. Semi-supervised relation extraction via incremental meta self-training. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 487–496, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xuming Hu, Chenwei Zhang, Yawen Yang, Xiaohe Li, Li Lin, Lijie Wen, and Philip S. Yu. 2021b. Gradient imitation reinforcement learning for low resource relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2737–2746, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Wanli Li, Tieyun Qian, Xu Chen, Kejian Tang, Shaohui Zhan, and Tao Zhan. 2021. Exploit a multi-head reference graph for semi-supervised relation extraction. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, pages 1–7. IEEE.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V. Le. 2021. Meta pseudo labels. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11552–11563.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. 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 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

George Stoica, Emmanouil Antonios Platanios, and Barnabas Poczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13843–13850.

Amane Sugiyama and Naoki Yoshinaga. 2019. Data augmentation using back-translation for context-aware neural machine translation. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 35–44, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45.

Wenxuan Zhou and Muhao Chen. 2021. An improved baseline for sentence-level relation extraction. *ArXiv*, abs/2102.01373.

A Proposed approach

Task formulation. In this work, we focus on the sentence-level relation extraction task, i.e., given a *relation statement* $\mathbf{x} : (\mathbf{s}, e_h, e_t)$ consisting of a sentence, $\mathbf{s}$, a head entity, e_h, and a tail entity, e_t (both the entities are mentioned in the given sentence $\mathbf{s}$), the goal is to predict a relation $r \in \mathcal{R} \cup \{\text{NA}\}$ between the head and the tail entity, where $\mathcal{R}$ is a pre-defined set of relations. If the sentence does not express any relation from the set $\mathcal{R}$ between the two entities, then the relation statement $\mathbf{x}$ is accordingly labelled NA.

This is typically done by learning a relation encoder model $\mathcal{F}_\theta : \mathbf{x} \mapsto \mathbf{h}_r$ that maps an input relation statement, $\mathbf{x}$, to a fixed length vector $\mathbf{h}_r$ that represents the relation expressed in $\mathbf{s}$ between e_h and e_t. This relation representation, $\mathbf{h}_r$, is then classified to a relation $r \in \mathcal{R} \cup \{\text{NA}\}$ via an MLP classifier.

Base model architecture. Most recent methods for RE use a Transformer-based architecture Devlin et al. (2019); Vaswani et al. (2017) for the relation encoder model, $\mathcal{F}_\theta$. To represent the head and tail entities in the input to the encoder, the widely accepted strategy is to augment the input sentence $\mathbf{s}$ with entity marker tokens–`[E1], [/E1], [E2], [/E2]`–to mark the start and end of both entities. Concretely, an input sentence like "*Lebron James currently plays for LA Lakers team.*" when augmented with entity marker tokens becomes

`[E1]` **Lebron James** `[/E1]` currently
plays for `[E2]` **LA Lakers** `[/E2]` team.

This modified text is input to the Transformer-based sequence encoder. Next, the encoder output representations[3] of the tokens `[E1]` and `[E2]` are concatenated to give the fixed length relation representation, $\mathbf{h}_r = [\mathbf{h}_{[E1]} \oplus \mathbf{h}_{[E2]}]$. This fixed length vector is in turn passed through an MLP classifier, $p_\phi(\mathbf{h}_r)$, to give a probability vector, $\mathbf{y}$, over the relation set $\mathcal{R} \cup \{\text{NA}\}$.

In our approach we build on the base model architecture described above and introduce additional model design elements that are necessary to obtain an improved performance in semi-supervised relation extraction (SSRE) task. We first describe the two data augmentation techniques we perform, and the model architectural changes we introduce that facilitate these augmentations. Then, we describe the training procedure we follow to leverage unlabelled data and achieve state-of-the-art performance on three out of four benchmark datasets for SSRE.

A.1 Constrained back-translation

Back-translation Edunov et al. (2018) generates diverse and fluent augmentations while retaining the global semantics of the original input. Specifically, one translates a given text into an intermediate language, say, German, and translates it back to the source language, say English. Using different intermediate languages and temperature based sampling results in a diverse set of paraphrases. Applying this back-translation technique in a vanilla fashion is not possible for RE task because one has little control over the retention of the head and tail entities (Figure 1). Thus, when translating back to the source language from the intermediate language we perform lexically-constrained decoding Hu et al. (2019), i.e., force the inclusion of pre-specified words and phrases–positive constraint set–in the output. In our case the original head and tail entity words/phrases make up this positive constraints set. We use German and Russian as intermediate languages and use the pre-trained WMT'19 English-German and English-Russian translation models (in both directions) and their implementations provided by Ott et al. (2019). This methodology generates diverse data augmentations for a given sentence. For example, the sentence "*The battle led to panic on the frontier, and settlers in the surrounding counties fled.*" is converted to "*The battle sparked panic at the border, with residents fleeing in surrounding counties*" when back-translated via German, and to "*The battle caused panic on the border and settlers in nearby counties fled.*" when done via Russian. This strong data-augmentation technique for RE can be applied to both labelled and unlabelled data opening the doors to consistency training Xie et al. (2020) as we will see in Section A.3.

A.2 Latent-space interpolation

Here, we adapt a mixup-based data augmentation technique to the RE task by making necessary modifications to the base model architecture we described in Section 2. As done in previous works

[3] hidden state from the last layer of the Transformer model

Chen et al. (2020b,a), we sample two random data points–$(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$, where $\mathbf{x}$ and $\mathbf{y}$ denote the relation statement and the corresponding relation label–from the training data and separately compute the respective latent representations, $\mathbf{h}^m$ and $\mathbf{h}'^m$, upto the layer m of the relation encoder $\mathcal{F}_\theta$ as follows:

$$\mathbf{h}^l = \mathcal{F}_\theta^l(\mathbf{h}^{l-1}); \quad l \in [1, m],$$
$$\mathbf{h}'^l = \mathcal{F}_\theta^l(\mathbf{h}'^{l-1}); \quad l \in [1, m],$$

where $\mathbf{h}^l$ is the latent representation of all tokens in the sentence $\mathbf{x}$ at the l^{th} layer of the encoder. Next, the latent representations of each token in $\mathbf{x}$ at the m^{th} layer are linearly interpolated:

$$\tilde{\mathbf{h}}^m = \lambda \mathbf{h}^m + (1 - \lambda)\mathbf{h}'^m,$$

where λ is the mixing coefficient which is sampled from a Beta distribution, i.e., $\lambda \sim \text{Beta}(\alpha, \beta)$. Then, the interpolated latent representation is passed through the rest of the encoder layers:

$$\tilde{\mathbf{h}}^l = \mathcal{F}_\theta^l(\tilde{\mathbf{h}}^{l-1}); \quad l \in [m + 1, L].$$

This final encoder output representation, $\tilde{\mathbf{h}}^L$, can be interpreted as the encoder output representation of a *virtual* input $\tilde{\mathbf{x}}$, i.e., $\tilde{\mathbf{h}}^L = \mathcal{F}_\theta(\tilde{\mathbf{x}})$. We denote this whole mixup operation[4] as $\tilde{\mathbf{h}}^L := \text{MixRE}(\mathbf{x}, \mathbf{x}')$. The label for this augmented *virtual* sample is given by the linear interpolation of the respective labels, $\mathbf{y}$ and $\mathbf{y}'$, with the same mixing coefficient λ i.e., $\tilde{\mathbf{y}} := \text{mix}(\mathbf{y}, \mathbf{y}') = \lambda \mathbf{y} + (1 - \lambda)\mathbf{y}'$. This *virtual* data point, $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$, is the augmented data point and can be used as additional training data.

Proposed architecture change. Now, for the RE task we need to extract a fixed-length relation representation from the encoder output representation of the entire input sequence. The traditional way to do this for RE task is by concatenating the encoder output representations of the entity marker tokens `[E1]` and `[E2]`. However, it is challenging to do this for a virtual sample, $\tilde{\mathbf{x}}$, as the entity markers are not clearly defined in this case. We thus modify the relation representation to be the encoder output representation of the `[CLS]` token. However, Baldini Soares et al. (2019) have shown this choice to be sub-optimal compared to concatenation of marker tokens. This is because the marker token representations provide direct access to the contextual information of the respective entities. Although the `[CLS]` token, in theory, has access to the entire context of the sentence, it might be difficult to capture the nuances like the head entity type, tail entity type, and the contextual information around the two entities all in a single vector.

On the other hand, entity type information is easily accessible in most RE benchmarks[5]. So, to compensate for the sub-optimal choice of using `[CLS]` token representation as the relation representation, we modify how we represent the entity spans in the input token to more effectively use the easily accessible entity type information. In particular, we note that the entity type labels can trivially be mapped to tokens from any pre-trained language model's vocabulary. For example, entity types like `PERSON` and `STATE_OR_PROVINCE` can be tokenized into a word/phrase like 'person' and 'state or province', respectively. Instead of using special marker tokens like `[E1]` and `[E2]`, we prepend the entity spans in the input sequence with the word/phrases corresponding to their respective types and enclose these 'type-words' in punctuation marks Zhou and Chen (2021). The modified input to the transformer along with the `[CLS]` token looks as follows:

`[CLS]` @ * person * **Lebron James** @ plays
for & * organization * **LA Lakers** & team.

We use different punctuation symbols to distinguish between subject and object entities. This representation helps leverage the knowledge already contained in the pre-trained large-language model about the type of the entity and offset some of the downside of using a simplified relation representation in the `[CLS]` token. Zhou and Chen (2021) recently showed the success of this method in the fully supervised setting. Here we use it in conjunction with a simplified relation representation and show its merit in semi-supervised RE setting.

[4] MixRE entails the model architecture changes discussed below.

[5] From new datasets/applications viewpoint, when entities are identified in a piece of text it is safe to assume that their types would also be identified.

A.3 Consistency training for SSRE

Let the given limited labelled set be $\mathbf{X}_l = \{\mathbf{x}_1^l, ..., \mathbf{x}_n^l\}$, with their relation labels $\mathbf{Y}_l = \{\mathbf{y}_1^l, ..., \mathbf{y}_n^l\}$, where $\mathbf{y}_i^l \in \{0, 1\}^{|\mathcal{R} \cup \{\text{NA}\}|}$ is a one-hot vector and $\mathcal{R}$ is the set of pre-defined relations. Let $\mathbf{X}_u = \{\mathbf{x}_1^u, ..., \mathbf{x}_m^u\}$ be a large unlabelled set. The goal is to apply both the data augmentation techniques described above and train a model with consistency loss to effectively leverage unlabelled data along with the limited labelled data.

We largely adapt the semi-supervised training techniques introduced by Chen et al. (2020b). For each $\mathbf{x}_i^u$ in the unlabelled set $\mathbf{X}_u$, we generate K augmentations $\mathbf{x}_{i,k}^a, k \in \{1, 2, ..., K\}$ using the constrained back translation technique with different intermediate languages[6]. These augmentations make up the set $\mathbf{X}_a = \{\mathbf{x}_{i,k}^a\}$. For a given unlabelled data point $\mathbf{x}_i^u$ and its K augmentations $\mathbf{x}_{i,k}^a$ the label is given by the average of current model's predictions on all $K + 1$ data points:

$$\mathbf{y}_i^u = \frac{1}{K+1}\left(p_\phi(\mathcal{F}_\theta(\mathbf{x}_i^u)) + \sum_{k=1}^{K} p_\phi(\mathcal{F}_\theta(\mathbf{x}_{i,k}^a))\right),$$

where $\mathbf{y}_i^u$ is a probability vector. This not only enforces the constraint that the model should make consistent predictions for different augmentations but also makes the predictions more robust by ensembling all the predictions. We merge the unlabelled set and the augmented set into $\mathbf{X}_{\text{ua}} = \mathbf{X}_u \cup \mathbf{X}_a$ and the corresponding labels are given by $\mathbf{Y}_{\text{ua}} = \mathbf{Y}_u \cup \mathbf{Y}_a$, where $\mathbf{Y}_u = \{\mathbf{y}_i^u\}$, $\mathbf{Y}_a = \{\mathbf{y}_{i,k}^a\}$, and $\mathbf{y}_{i,k}^a = \mathbf{y}_i^u \;\; \forall \;\; k \in \{1, 2, ..., K\}$, i.e., all the augmented data points share the same label as the original unlabelled data point.

Given this cumulative set $\mathbf{X}_{\text{ua}}$ and their generated labels $\mathbf{Y}_{\text{ua}}$ as additional training data, we employ the MixRE augmentation technique to generate arbitrary amounts of training data. In particular, we randomly sample two data points $\mathbf{x}_s^{\text{ua}}, \mathbf{x}_t^{\text{ua}} \in \mathbf{X}_{\text{ua}}$, and compute the encoder output representation of a new *virtual* data point with $\text{MixRE}(\mathbf{x}_s^{\text{ua}}, \mathbf{x}_t^{\text{ua}})$ and the corresponding target label with $\text{mix}(\mathbf{y}_s^{\text{ua}}, \mathbf{y}_t^{\text{ua}})$.

Additionally, while computing the final unsupervised loss in each training iteration we filter out the unlabelled data points with prediction confidence below a certain threshold γ Xie et al. (2020). Finally, to encourage low-entropy predictions on unlabelled data, we sharpen the predictions with a sharpening coefficient T:

$$\hat{\mathbf{y}}_i^{\text{ua}} = \frac{(\mathbf{y}_i^{\text{ua}})^{\frac{1}{T}}}{||(\mathbf{y}_i^{\text{ua}})^{\frac{1}{T}}||_1}.$$

Everything put together, the final unsupervised loss in each training iteration with mini-batch size B is computed as:

$$\mathcal{L}_{\text{unsp}} = \frac{1}{B}\sum_{\mathbf{x}_s^{\text{ua}}, \mathbf{x}_t^{\text{ua}} \sim \mathbf{X}_{\text{ua}}}^{B} \mathtt{m}(\mathbf{x}_s^{\text{ua}}, \mathbf{x}_t^{\text{ua}})\mathcal{L}_{\text{mix}}(\mathbf{x}_s^{\text{ua}}, \mathbf{x}_t^{\text{ua}}),$$

where

$$\begin{aligned}\mathcal{L}_{\text{mix}}(\mathbf{x}_s^{\text{ua}}, \mathbf{x}_t^{\text{ua}}) &= \text{CE}(\text{mix}(\hat{\mathbf{y}}_s^{\text{ua}}, \hat{\mathbf{y}}_t^{\text{ua}})|| \\ &\qquad p_\phi(\text{MixRE}(\mathbf{x}_s^{\text{ua}}, \mathbf{x}_t^{\text{ua}}))), \\ \mathtt{m}(\mathbf{x}_s^{\text{ua}}, \mathbf{x}_t^{\text{ua}}) &= I(\max \mathbf{y}_s^{\text{ua}} > \gamma) I(\max \mathbf{y}_t^{\text{ua}} > \gamma).\end{aligned}$$

Here, $I(.)$ is an indicator function and $\mathtt{m}(.)$ denotes the confidence masking function which filters out the low-confidence datapoints. In our implementation $p_\phi(.)$ is a two-layer MLP classifier on top of the relation encoder model. CE denotes the cross entropy loss function. Note that we only apply the augmentation techniques on the unlabelled data set. Initial experiments applying these to the labelled data set resulted in only marginal improvements and even performance deterioration in some cases, likely due to introduction of too much noise into an already limited labelled set.

This combined with the traditional supervised loss, $\mathcal{L}_{\text{sup}} = \sum_{\mathbf{x}_i \sim \mathbf{X}_l}^{B} \text{CE}(\mathbf{y}_i^l || p_\phi(\mathcal{F}_\theta(\mathbf{x}_i)))$, constitutes the total loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sup}} + \gamma_m \mathcal{L}_{\text{unsp}},$$

where γ_m is a parameter which controls the trade-off between supervised and unsupervised loss.

[6] In our specific implementation K = 2; with German and Russian as intermediate languages.

Table 3: Dataset statistics

Dataset	# rel.	examples	# no_relation
TACRED	42	106264	79.51%
RE-TACRED	40	91467	63.17%
KBP37	37	21046	10.33%
SemEval	19	10717	17.39%

B Experiment details

B.1 Datasets

We use two widely popular relation extraction benchmark datasets: SemEval 2010 Task 8 (SemEval) Hendrickx et al. (2010), and the TAC Relation Extraction Dataset (TACRED) Zhang et al. (2017). SemEval is a standard benchmark dataset for evaluating relation extraction models containing 10717 examples in total. Each sentence is annotated with a pair of untyped nominals (concepts; example in Figure 1) that are related via one of 19 semantic relation types (including `no_relation`). TACRED is a large-scale crowd-sourced relation extraction dataset with 106264 examples which is collected from all the prior TAC KBP relation schema. Unlike SemEval, sentences in TACRED are labelled with pairs of typed-entities that are related via one of 42 person- and organization-oriented relation types (including `no_relation`). In addition to these standard benchmark datasets, we also show results on two more datasets: RE-TACRED Stoica et al. (2021) and KBP37 Zhang and Wang (2015). RE-TACRED is a re-annotated version of the original TACRED dataset using an improved annotation strategy to ensure high-quality labels. Zhou and Chen (2021) provide a compelling analysis and recommend using this as the evaluation benchmark for sentence-level RE. KBP37 is another sentence-level RE dataset with 21046 total examples collected from 2010 and 2013 KBP documents as well as July 2013 dump of Wikipedia. In terms of size, this falls between SemEval and TACRED. Similar to SemEval the entity types are not available in this dataset, however the 37 relation types are person- and organization-oriented like in TACRED. This dataset is thus a good segue between the two standard benchmarks. The statistics of these datasets is given in Table 3. The sources of all the datasets are given in Table 4. We use the given train/validation/test splits for TACRED, RE-TACRED and KBP37 datasets. For SemEval dataset, we use the same splits as all the baselines, i.e., we split the original training set into 90% training set and 10% validation set.

B.2 Baseline models

We compare MixRE with three state-of-the-art models that are representative of the existing class of methods for SSRE: MRefG Li et al. (2021), MetaSRE Hu et al. (2021a), and GradLRE Hu et al. (2021b). MRefG leverages the unlabelled data by semantically or lexically connecting them to labelled data by constructing reference graphs, such as entity reference or verb reference. This approach heavily leverages the linguistic structure of the data and is the only existing method that falls outside the *self-training* class of methods. MetaSRE generates pseudo labels on unlabelled data by learning from the mistakes of the classification model as an additional meta-objective. GradLRE on the other hand generates pseudo label data to imitate the gradient descent direction on labelled data and bootstrap its optimization capability through trial and error Hu et al. (2021b). MetaSRE and GradLRE are two of the strongest methods in the widely adapted *self-training* methods for SSRE.

B.3 Implementation details

To be consistent with all the baselines we initialize the text encoder of MixRE with the `bert-base-cased` model architecture and pre-trained weights. Following Chen et al. (2020b),

Table 4: Dataset sources

Dataset	Source
TACRED	`https://catalog.ldc.upenn.edu/LDC2018T24`
RE-TACRED	`https://github.com/gstoica27/Re-TACRED`
KBP37	`https://github.com/zhangdongxu/kbp37`
SemEval	`https://semeval2.fbk.eu/semeval2.php?location=data`

we use {7, 9, 12} for the mixup layer set; this layer subset contains most of the syntactic and semantic information as suggested by Jawahar et al. (2019). We use the BERT tokenizer and set maximum sequence length as 256 to pre-process all datasets. We use the AdamW optimizer Loshchilov and Hutter (2019) with 5e-5 learning rate and 0.1 warmup ratio. We sweep over the following hyperparameters: sharpening coefficient T, confidence threshold γ, the Beta-distribution parameters (α, β)[7], and the unsupervised loss weight γ_m. We perform incremental grid search to get the best performing configuration based on the F1 score on validation set. Table 5 shows the set of values we use for each parameter. Table 6 shows the best parameter values on each dataset and configuration.

Table 5: Hyperparameter search values

Parameter	Values
T	{0.4, 0.6, 0.8, 1.0}
γ	{0, 0.15, 0.2, 0.25}
β	{1, 10, 30, 60, 120, 190, 300, 600}*
γ_m	{0.01, 0.1, 1}

* corresponding means of the sampled mixing coefficient, λ, are given by {0.04, 0.09, 0.17, 0.24, 0.33, 0.50, 0.67, 0.86, 0.98}

Table 6: Best hyperparameter values

	TACRED			KBP37		
	3%	10%	15%	5%	10%	30%
T	0.8	0.8	0.8	0.8	0.8	0.8
γ	0.15	0.15	0.90	0.25	0.25	0.70
β	10	10	10	120	120	190
γ_m	0.01	0.1	0.1	0.1	1.0	1.0
	RE-TACRED			SemEval		
	3%	10%	15%	5%	10%	30%
T	0.4	0.4	0.4	0.8	0.8	0.8
γ	0.0	0.0	0.9	0.2	0.2	0.7
β	10	10	10	60	60	60
γ_m	0.01	0.1	0.1	0.1	1.0	1.0

We train each model on a single NVIDIA Tesla T4 GPU with 16GB memory. We employ mixed precision training and gradient checkpointing techniques for faster and memory-efficient training. Note that we train the models until the performance on the validation set plateaus. The full MixRE model roughly takes about 6 hours to train on TACRED, 5 hours in RE-TACRED, 1 hour in KBP37, and about 30 minutes on SemEval. Note that the training time slightly vary (± 30 minutes) depending on the percentage of labelled and unlabelled data we use. The number of parameters in all our models are largely dominated by the `bert-base-cased` that we use as the text encoder. The relatively negligible varying component is the MLP classifier that varies with the varying number of relations in each dataset.

[7] α is fixed to be 60 and we change the values of β to control the mean of the distribution