

Unsupervised Text Style Transfer with Padded Masked Language Models

Eric Malmi
Google Research
emalmi@google.com

Aliaksei Severyn
Google Research
severyn@google.com

Sascha Rothe
Google Research
rothe@google.com

Abstract

We propose MASKER, an unsupervised text-editing method for style transfer. To tackle cases when no parallel source–target pairs are available, we train masked language models (MLMs) for both the source and the target domain. Then we find the text spans where the two models disagree the most in terms of likelihood. This allows us to identify the source tokens to delete to transform the source text to match the style of the target domain. The deleted tokens are replaced with the target MLM, and by using a *padded* MLM variant, we avoid having to predetermine the number of inserted tokens. Our experiments on sentence fusion and sentiment transfer demonstrate that MASKER performs competitively in a fully unsupervised setting. Moreover, in low-resource settings, it improves supervised methods’ accuracy by over 10 percentage points when pre-training them on silver training data generated by MASKER.

1 Introduction

Text-editing methods (Dong et al., 2019; Malmi et al., 2019; Awasthi et al., 2019; Mallinson et al., 2020), that target monolingual sequence transduction tasks like sentence fusion, grammar correction, and text simplification, are typically more data-efficient than the traditional sequence-to-sequence methods, but they still require substantial amounts of parallel training examples to work well. When parallel source–target training pairs are difficult to obtain, it is often still possible to collect non-parallel examples for the source and the target domain separately. For instance, negative and positive reviews can easily be collected based on the numerical review scores associated with them, which has led to a large body of work on unsupervised text style transfer, e.g., (Yang et al., 2018; Shen et al., 2017; Wu et al., 2019; Li et al., 2018).

The existing unsupervised style transfer methods aim at transforming a source text so that its style

matches the target domain but its content stays otherwise unaltered. This is commonly achieved via text-editing performed in two steps: using one model to identify the tokens to delete and another model to infill the deleted text slots (Li et al., 2018; Xu et al., 2018; Wu et al., 2019). In contrast, we propose a more unified approach, showing that both of these steps can be completed using a single model, namely a *masked language model* (MLM) (Devlin et al., 2019). MLM is a natural choice for infilling the deleted text spans, but we can also use it to identify the tokens to delete by finding the spans where MLMs trained on the source and the target domain disagree in terms of likelihood. This is inspired by the recent observation that MLMs are effective at estimating (pseudo) likelihoods of texts (Wang and Cho, 2019; Salazar et al., 2020). Moreover, by using a *padded* variant of MLM (Mallinson et al., 2020), we avoid having to separately model the length of the infilled text span.

To evaluate the proposed approach, MASKER, we apply it to two tasks: sentence fusion, which requires syntactic modifications, and sentiment transfer, which requires semantic modifications. In the former case, MASKER improves the accuracy of state-of-the-art text-editing models by more than 10 percentage points in low-resource settings by providing silver data for pretraining, while in the latter, it yields a competitive performance compared to existing unsupervised style-transfer methods.

2 Method

Our approach to unsupervised style transfer is to modify source texts to match the style of the target domain. To achieve this, we can typically keep most of the source tokens and only modify a fraction of them. To determine which tokens to edit and how to edit them, we propose the following three-step approach:

(1) Train padded MLMs on source domain data (Θ_{source}) and on target domain data (Θ_{target}). (§2.1)

- (2) Find the text spans where the models *disagree* the most to determine the tokens to delete. (§2.2)
- (3) Use Θ_{target} to replace the deleted spans with text that fits the target domain.

2.1 Padded Masked Language Models

The original MLM objective in BERT (Devlin et al., 2019) does not model the length of infilled token spans since each [MASK] token corresponds to one wordpiece token that needs to be predicted at a given position. To model the length, it is possible to use an autoregressive decoder or a separate model (Mansimov et al., 2019). Instead, we use an efficient non-autoregressive *padded MLM* approach by Mallinson et al. (2020) which enables BERT to predict [PAD] symbols when infilling a fixed-length spans of n_p [MASK] tokens.

When creating training data for this model, spans of zero to n_p tokens, corresponding to whole word(s), are masked out after which the mask sequences are padded to always have n_p [MASK] tokens. For example, if $n_p = 4$ and we have randomly decided to mask out tokens from i to $j = i + 2$ (inclusive) from text W , the corresponding input sequence is:

$$W_{\setminus i:j} = (w_1, \dots, w_{i-1}, [\text{MASK}], [\text{MASK}], [\text{MASK}], [\text{MASK}], w_{i+3}, \dots, w_{|W|}).$$

The targets for the first three [MASK] tokens are the original masked out tokens, i.e. w_i, w_{i+1}, w_{i+2} , while for the remaining token the model is trained to output a special [PAD] token.

Similar to (Wang and Cho, 2019; Salazar et al., 2020), we can compute the pseudo-likelihood (\mathcal{L}) of the original tokens $W_{i:j}$ according to:

$$\begin{aligned} \mathcal{L}(W_{i:j} | W_{\setminus i:j}; \Theta) &= \prod_{t=i}^j P_{\text{MLM}}(w_t | W_{\setminus i:j}; \Theta) \\ &\times \prod_{t=j+1}^{i+n_p-1} P_{\text{MLM}}([\text{PAD}]_t | W_{\setminus i:j}; \Theta), \end{aligned}$$

where $P_{\text{MLM}}(*_t | W_{\setminus i:j}; \Theta)$ denotes the probability of the random variable corresponding to the t -th token in $W_{\setminus i:j}$ taking value w_t or [PAD]. Furthermore, we can compute the maximum pseudo-likelihood infilled tokens $\widehat{W}_{i:j} = \arg \max_{W_{i:j}} \mathcal{L}(W_{i:j} | W_{\setminus i:j}; \Theta)$ by taking the most likely insertion for each [MASK] independently, as done by the regular BERT. These maximum likelihood estimates are used both when de-

termining which spans to edit (as described in §2.2) as well as when replacing the edited spans.

In practice, instead of training two separate models for the source and target domain, we train a single conditional model. Conditioning on a domain is achieved by prepending a special token ([SOURCE] or [TARGET]) to each token sequence fed to the model.¹ At inference time, padded MLM can decide to insert zero tokens (by predicting [PAD] for each mask) or up to n_p tokens based on the bidirectional context it observes. In our experiments, we set $n_p = 4$.²

2.2 Where to edit?

Our approach to using MLMs to determine where to delete and insert tokens is to find text spans where the source and target model disagree the most. Here we introduce a scoring function to quantify the level of disagreement.

First, we note that any span of source tokens that has a low likelihood in the target domain is a candidate span to be replaced or deleted. That is, source tokens from index i to j should be more likely to be deleted the lower the likelihood $\mathcal{L}(W_{i:j} | W_{\setminus i:j}; \Theta_{\text{target}})$ is. Moreover, if two spans have equally low likelihoods under the target model, but one of them has a higher maximum likelihood replacement $\widehat{W}_{i:j}^{\text{target}}$, then it is safer to replace the latter. For example, if a sentiment transfer model encounters a polarized word of the wrong sentiment and an arbitrary phone number, it might evaluate both of them as unlikely. However, the model will be more confident about how to replace the polarized word, so it should try to replace that rather than the phone number. Thus the first component of our scoring function is:

$$\begin{aligned} \text{TargetScore}(i, j) &= \mathcal{L}(\widehat{W}_{i:j}^{\text{target}} | W_{\setminus i:j}; \Theta_{\text{target}}) \\ &\quad - \mathcal{L}(W_{i:j} | W_{\setminus i:j}; \Theta_{\text{target}}). \end{aligned}$$

This function can be used on its own without having access to a source domain corpus, but in some

¹The motivation for using a joint model instead of two separate models is to share model weights to give more consistent likelihood estimates. An alternative way of conditioning the model would be to add a domain embedding to each token embedding as proposed by Wu et al. (2019).

²In early experiments, we also tested $n_p = 8$, but this resulted in fewer grammatical predictions since each token is predicted independently. To improve the predictions, we could use SpanBERT (Joshi et al., 2020), which is designed to infill spans, or an autoregressive model like T5 (Raffel et al., 2019).

cases, this leads to undesired replacements. The target model can be very confident that, e.g., a rarely mentioned entity should be replaced with a more common entity, although this type of edit does not help with transferring the style of the source text toward the target domain. To address this issue, we introduce a second scoring component leveraging the source domain MLM:

$$\text{SourceScore}(i, j) = -\max \left[0, \mathcal{L} \left(\widehat{W}_{i:j}^{\text{target}} \mid W_{\setminus i:j}; \Theta_{\text{source}} \right) - \mathcal{L} \left(W_{i:j} \mid W_{\setminus i:j}; \Theta_{\text{source}} \right) \right]$$

By adding this component to $\text{TargetScore}(i, j)$, we can counter for edits that only increase the likelihood of a span under Θ_{target} but do not push the style closer to the target domain.³

Our overall scoring function is given by:

$$\text{Score}(i, j) = \text{TargetScore}(i, j) + \text{SourceScore}(i, j).$$

To determine the span to edit, we compute $\arg \max_{i,j} \text{Score}(i, j)$, where $1 \leq i \leq |W| + 1$ and $i - 1 \leq j \leq i + n_p - 1$. The case $j = i - 1$ denotes an empty source span, meaning that the model does not delete any source tokens but only adds text before the i -th source token.

The process for selecting the span to edit is illustrated in Figure 1, where the source text corresponds to two sentences to be fused. The source MLM has been trained on unfused sentences and the target MLM on fused sentences from the DiscoFuse corpus (Geva et al., 2019). In this example, the target model is confident that either the boundary between the two sentences or the grammatical mistake “in the France” should be edited. However, also the source model is confident that the grammatical mistake should be edited, so the model correctly ends up editing the words “. She” at the sentence boundary. The resulting fused sentence is: *Marie Curie was born in Poland and died in the France*.

Efficiency. The above method is computationally expensive since producing a single edit requires $\mathcal{O}(|W| \times n_p)$ BERT inference steps – although

³SourceScore(i,j) is capped at zero to prevent it from dominating the overall score. Otherwise, we might obtain low-quality edits in cases where the likelihood of the source span $W_{i:j}$ is high under the source model and low under the target model but no good replacements exist according to the target model. Given the lack of good replacements, $\widehat{W}_{i:j}^{\text{target}}$ may end up being ungrammatical, pushing SourceScore(i,j) close to 1 and thus making it a likely edit, although TargetScore(i,j) remains low.

	TargetScore					SourceScore					Score				
Marie	0.00	0.11	0.00	0.00	0.00	0.00	-0.22	-0.01	-0.02	0.00	0.00	-0.12	-0.01	-0.02	0.00
Curie	0.00	0.00	0.00	0.01	0.00	0.00	-0.03	-0.01	-0.00	0.00	0.00	-0.02	-0.01	0.01	0.00
was	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	-0.00	0.00	0.00	0.00	0.00	0.01
born	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	-0.00	-0.00	0.00	0.00	0.00	0.02	-0.00
in	0.00	0.00	0.04	0.02	0.07	0.00	0.00	-0.16	-0.01	0.00	0.00	0.00	-0.12	0.00	0.07
Poland	0.00	0.09	0.02	0.07	0.01	0.00	-0.19	-0.00	0.00	0.00	0.00	-0.10	0.02	0.07	0.01
.	0.00	0.16	0.49	0.06	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.49	0.06	0.02
She	0.00	0.01	0.01	0.01	0.01	0.00	0.00	-0.00	-0.00	-0.00	0.00	0.01	0.01	0.01	0.01
died	0.00	0.01	0.01	0.01	0.00	0.00	-0.02	-0.00	-0.00	-0.00	0.00	-0.01	0.00	0.01	0.00
in	0.00	0.00	0.50	0.07	0.00	0.00	0.00	-0.38	-0.02	0.00	0.00	0.00	0.11	0.05	0.00
the	0.00	0.46	0.09	0.00	0.00	0.00	-0.44	-0.02	0.00	0.00	0.00	0.01	0.07	0.00	0.00
France	0.02	0.10	0.00	0.00	0.00	-0.11	-0.12	0.00	0.00	0.00	-0.09	-0.01	0.00	0.00	0.00
.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
.	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
	#deleted words					#deleted words					#deleted words				

Figure 1: MASKER replaces span “. She” by “and [PAD] [PAD] [PAD]”, resulting in the following fused sentence: *Marie Curie was born in Poland and died in the France*.

these can be run in parallel. The model can be distilled into a much more efficient supervised student model without losing – and even gaining – accuracy as shown in our experiments. This is done by applying MASKER to the unaligned source and target examples to generate aligned silver data for training the student model.

3 Experiments

We evaluate MASKER on two different types of tasks: *sentence fusion* and *sentiment transfer*. For both experiments, we only apply MASKER once to edit a single span of at most four tokens, since the required edits are often local.⁴

3.1 Sentence Fusion

Sentence fusion is the task of fusing two (or more) incoherent input sentences into a single coherent sentence or paragraph, and DiscoFuse (Geva et al., 2019) is a recent parallel dataset for sentence fusion. We study both a fully unsupervised setting as well as a low-resource setting.

Unsupervised. First, we remove the alignment between unfused and fused examples in the training set of 4.5 million examples and finetune MASKER on the resulting, non-parallel source and target corpora. This model yields an Exact match accuracy (which is a standard metric for sentence fusion (Geva et al., 2019; Rothe et al., 2020)) of 12.65

⁴We tried running multiple iterations of MASKER, but this somewhat decreased the accuracy of the method. When parallel development data is available, it could potentially be used to optimize a threshold of $\text{Score}(i, j)$ so that the model could be called repeatedly until the $\text{Score}(i, j)$ falls below the threshold. Alternatively, it would be interesting to explore methods for simultaneously identifying multiple, not necessarily adjacent, spans to edit.

Method	Exact score
MASKER (unfused \rightarrow fused)	12.65
ablating SourceScore	2.14
MASKER (fused \rightarrow unfused)	23.18

Table 1: Unsupervised sentence fusion results.

on the development set. This is already on par with the Exact score of 12.32 obtained by a supervised LASERTAGGER model (Malmi et al., 2019) trained on 450 examples. We also test ablating SourceScore, which results in a decreased Exact score of 2.14, attesting to the importance of using the source model. Finally, we test our model on the reverse direction of going from a fused text to unfused sentences. Here, MASKER yields a significantly higher Exact score of 23.18. This direction is typically easier since it does not involve predicting discourse markers, which would require modeling the semantic relation between two sentences. The predictions of the reverse model are used in the low-resource experiments. The unsupervised results are summarized in Table 1.

Low resource. We use MASKER to generate noisy unfused sentences for 46K target fusions in the DiscoFuse development set. This silver data is used to pretrain three different model architectures, LASERTAGGER (Malmi et al., 2019), FELIXINSERT (Mallinson et al., 2020), and BERT2BERT (Rothe et al., 2020), which have previously been used for training fusion models under low-resource settings. The results on the test set (45K examples) without and with pretraining on MASKER outputs are shown in Table 2. On average, the silver data from MASKER improves the Exact score by 13.37 when 450 parallel training examples are available and still by 2.01 when 45 000 parallel examples are available.

3.2 Sentiment Transfer

In sentiment transfer, the task is to change a text’s sentiment from negative to positive or vice versa. We use a dataset of Yelp reviews (Li et al., 2018), containing 450K training, 4K development, and 1K test examples. Half of the test reviews are positive and half negative, and human annotators have written a reference review of the opposite sentiment for each test review. We use the same automatic evaluation metrics used in previous work: BLEU score and accuracy that a classifier trained to distinguish negative and positive reviews assigns to

Method	Parallel training examples			
	0	450	4500	45000
LASERTAGGER	0.00	12.32	25.74	38.46
+ MASKER silver data	19.61	25.97	34.20	42.41
FELIXINSERT	0.00	15.34	34.11	46.09
+ MASKER silver data	18.22	25.23	38.43	47.21
BERT2BERT	0.00	0.00	3.35	42.07
+ MASKER silver data	13.05	16.57	30.14	43.03
Average improvement	16.96	13.37	13.19	2.01

Table 2: Low-resource sentence fusion results. Using the predictions of MASKER as silver data to pretrain models improves the Exact score.

Method	BLEU	ACC (%)
DELETEANDRETRIEVAL	8.5	87.9
AC-MLM w/ frequency-ratio	13.2	37.9
AC-MLM w/ attention-based	15.7	53.4
AC-MLM w/ fusion-method	15.3	40.9
MASKER	14.5	40.9
LASERTAGGER w/ MASKER silver data	15.3	49.6

Table 3: Yelp review sentiment transfer results.

the modified reviews being of the target sentiment.

We finetune the MLMs on the training set and apply the resulting MASKER model to the test set. Additionally, we apply the MASKER model to the non-parallel training set to create parallel silver data and train a LASERTAGGER model. Interestingly, the latter setup outperforms MASKER alone (15.3 vs. 14.5 BLEU score; 49.6 vs. 40.9 sentiment accuracy). We think this happens because LASERTAGGER employs a restricted vocabulary of 500 most frequently inserted phrases, which prevents the model from reproducing every spurious infilling that the padded MLM may have produced, effectively regularizing MASKER. In Table 3, we report these results along with baseline methods developed specifically for the sentiment transfer task by Li et al. (2018) and Wu et al. (2019). Overall, MASKER yields a competitive performance although AC-MLM w/ attention-based (Wu et al., 2019) slightly outperforms it.

4 Related Work

Section 1 provides a high-level overview of the related work. Closest to this work is the AC-MLM sentiment transfer method by Wu et al. (2019). This method first identifies the tokens to edit based on n -gram frequencies in the source vs. target domain (as proposed by Li et al. (2018)) and based

on LSTM attention scores (as proposed by Xu et al. (2018)). Then it replaces the edited tokens using a conditional MLM. In contrast to their work, our approach leverages the same MLM for both identifying the (possibly empty) span of tokens to edit and for infilling the deleted span. Moreover, our padded MLM determines the number of tokens to insert without having to pre-specify it. In that sense, it is similar to the recently proposed Blank Language Model (Shen et al., 2020).

In addition to the two applications studied in this work, it would be interesting to evaluate MASKER on other style transfer tasks. Tasks for which unsupervised methods have recently been developed include formality transfer (Rao and Tetreault, 2018; Luo et al., 2019), lyrics style transfer (Nikolov et al., 2020; Lee et al., 2019), text simplification (Paetzold and Specia, 2016; Surya et al., 2019), and sarcasm generation (Mishra et al., 2019; Chakrabarty et al., 2020).

5 Conclusions

We have introduced a novel way of using masked language models for text-editing tasks where no parallel data is available. The method is based on training an MLM for source and target domains, identifying the tokens to delete by finding the spans where the two models disagree in terms of likelihood, and infilling more appropriate text with the target MLM. This approach yields a competitive performance in fully unsupervised settings and substantially improves over previous works in low-resource settings.

References

- Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. 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 4251–4261.
- Tuhin Chakrabarty, Debanjan Ghosh, Smaranda Muresan, and Nanyun Peng. 2020. R^3 : Reverse, Retrieve, and Rank for Sarcasm Generation with Commonsense Knowledge. *arXiv preprint arXiv:2004.13248*.
- 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.
- Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. Editnets: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402.
- Mor Geva, Eric Malmi, Idan Szpektor, and Jonathan Berant. 2019. Discofuse: A large-scale dataset for discourse-based sentence fusion. 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 3443–3455.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.
- Joseph Lee, Ziang Xie, Cindy Wang, Max Drach, Dan Jurafsky, and Andrew Y Ng. 2019. Neural text style transfer via denoising and reranking. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 74–81.
- Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. 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 1865–1874.
- Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Xu Sun, and Zhifang Sui. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5116–5122. AAAI Press.
- Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. Felix: Flexible text editing through tagging and insertion. *arXiv preprint arXiv:2003.10687*.
- Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. 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 5057–5068.
- Elman Mansimov, Alex Wang, Sean Welleck, and Kyunghyun Cho. 2019. A generalized framework of sequence generation with application to undirected sequence models. *arXiv preprint arXiv:1905.12790*.

- Abhijit Mishra, Tarun Tater, and Karthik Sankaranarayanan. 2019. A modular architecture for unsupervised sarcasm generation. 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 6146–6155.
- Nikola I Nikolov, Eric Malmi, Curtis G Northcutt, and Loreto Parisi. 2020. Conditional rap lyrics generation with denoising autoencoders. *arXiv preprint arXiv:2004.03965*.
- Gustavo H Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3761–3767.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Sudha Rao and Joel Tetreault. 2018. Dear Sir or Madam, May I Introduce the GYAFD Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer. 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 129–140.
- Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.
- Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *ACL*.
- Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pages 6830–6841.
- Tianxiao Shen, Victor Quach, Regina Barzilay, and Tommi Jaakkola. 2020. Blank language models. *arXiv preprint arXiv:2002.03079*.
- Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2058–2068.
- Alex Wang and Kyunghyun Cho. 2019. Bert has a mouth, and it must speak: Bert as a markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36.
- Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Mask and infill: applying masked language model to sentiment transfer. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5271–5277. AAAI Press.
- Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988.
- Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, pages 7287–7298.

A Examples of Model Outputs

To further illustrate how MASKER works, Table 7 shows all the input sequences and the output scores that go into computing Figure 1 in the main paper. Furthermore, Tables 5 and 6 present random samples of correct and incorrect outputs by MASKER for the DiscoFuse and Yelp datasets.

B Hyperparameter Settings

We did not perform any hyperparameter tuning, but used a fixed learning rate of $3e-5$ and a batch size roughly proportionate to the training set size (see Table 4 for the chosen values). The number of training steps was determined by running the training until convergence and choosing the checkpoint with the highest validation score, shown in Table 4.

C Other Experimental Details

Code. The padded MLM implementation is based on: <https://github.com/google-research/bert>. LASERTAGGER code is available at: <https://github.com/google-research/lasertagger>

Datasets. The DiscoFuse dataset (Geva et al., 2019) is available at: <https://github.com/google-research-datasets/discofuse>. The Yelp review dataset (Li et al., 2018) is available at: <https://github.com/lijuncen/Sentiment-and-Style-Transfer>.

Evaluation. To compute BLEU scores, we used the implementation of Wu et al. (2019): https://github.com/IIKES/MLM_transfer. The AC-MLM baseline predictions after 10 training epochs are taken from the directory. For the sentiment classification accuracy score, we trained a BERT

Method	Dataset	Learning rate	Batch size	Exact score (validation)
Padded MLM	DiscoFuse	3e-5	512	44.03
LASERTAGGER	DiscoFuse, MASKER silver data	3e-5	256	27.09
LASERTAGGER	DiscoFuse, 450	3e-5	32	33.16
LASERTAGGER	DiscoFuse, 4500	3e-5	64	43.36
LASERTAGGER	DiscoFuse, 45000	3e-5	128	49.43
Padded MLM	Yelp	3e-5	2048	49.15
LASERTAGGER	Yelp, MASKER silver data (neg to pos)	3e-5	512	31.79
LASERTAGGER	Yelp, MASKER silver data (pos to neg)	3e-5	512	31.15

Table 4: Hyperparameter settings for the proposed method in Table 1 and 2, along with the Exact scores on validation set. For Padded MLM, the validation score refers to the accuracy of predicting all four masked tokens correctly.

model, which yields an accuracy of 98.4% on the development set (slightly higher than the CNN classifier used by Shen et al. (2020) which has an accuracy of 97.7%). The Exact scores reported in the paper were computed after lowercasing the predictions and the targets.

Padded MLM pretraining. The padded masked language model used in our experiments uses the uncased BERT-base architecture (Devlin et al., 2019) with 110M parameters. It is pretrained with the maximum pad length of $n_p = 4$ on the Wikipedia and books corpora that the original BERT was also trained on. When creating MLM finetuning data for the source and the target domain, we always mask out only a single span of zero to four input tokens so that the masked span corresponds to whole word(s). The accuracy of the MLM at filling the masked span correctly is 44% for sentence fusion and 49% for sentiment transfer as shown in Table 4.

Computing infrastructure. The models were trained using Tensor Processing Units (TPUs). Inference was distributed to multiple CPUs using Apache Beam and Google Cloud.

Runtime. Inference time increases with the sequence length. For the example in Figure 1 of the main paper, prediction takes 52 seconds when running BERT inference on CPU. Using GPUs or TPUs can significantly reduce the runtime, but we chose to use CPUs to be able to distribute the computation more effectively. Moreover, after distilling the model into a LASERTAGGER model (the autoregressive variant) as done in our experiments, inference takes only 535 milliseconds on GPU (Malmi et al., 2019).

Random Sample of Correct MASKER Predictions	
Source	the boat was hoisted aboard the carpathia along with other titanic lifeboats . the boat was brought to new york .
Prediction	the boat was hoisted aboard the carpathia along with other titanic lifeboats and brought to new york .
Source	beausoleil was a good - looking and rebellious youth . by 15 , beausoleil was sent to reform school .
Prediction	beausoleil was a good - looking and rebellious youth . by 15 , he was sent to reform school .
Source	it is believed that in terms of antiquity , this temple pre-dates the srirangam temple , . the name aadi vellarai .
Prediction	it is believed that in terms of antiquity , this temple pre-dates the srirangam temple , hence the name aadi vellarai .
Source	john was in charge of the roads north of kapunda . ben had yorke peninsula and the southern routes .
Prediction	john was in charge of the roads north of kapunda , while ben had yorke peninsula and the southern routes .
Source	in early 2018 , the central bank re-released the 1 - qiaif regime . it could replicate the section 110 spv .
Prediction	in early 2018 , the central bank re-released the 1 - qiaif regime so that it could replicate the section 110 spv .
Source	he also set up trade schools . girls could earn their living .
Prediction	he also set up trade schools so that girls could earn their living .
Source	the prime minister supplied the reason why : she said it was . we depend for our liberty and freedom upon the united states .
Prediction	the prime minister supplied the reason why : she said it was because we depend for our liberty and freedom upon the united states .
Source	long - distance routes are listed here . those entirely or mostly within one vdot district are at the following pages .
Prediction	long - distance routes are listed here , while those entirely or mostly within one vdot district are at the following pages .
Random Sample of Incorrect MASKER Predictions	
Source	she claimed that she did not consult the party . she believed the outcome had been assured .
Prediction	she claimed that she did not consult the party . however , she believed the outcome had been assured .
Target	she claimed that she did not consult the party because she believed the outcome had been assured .
Source	the 1927 german football championship was won by 1 . the 1927 german football championship is the 20th edition of the competition .
Prediction	the 1927 german football championship was won by 1 . the 1927 german football championship was the 20th edition of the competition .
Target	the 1927 german football championship , the 20th edition of the competition , was won by 1 .
Source	many other fourteeners surround elbert in all directions . it is very close to central colorado 's collegiate peaks .
Prediction	many other fourteeners surround elbert because it is very close to central colorado 's collegiate peaks .
Target	many other fourteeners surround elbert in all directions , and it is very close to central colorado 's collegiate peaks .
Source	the others want to return to centralville . davidson orders them back to fight the athsheans .
Prediction	the others want to return to centralville , but davidson orders them back to fight the athsheans .
Target	although the others want to return to centralville , davidson orders them back to fight the athsheans .
Source	in march 2015 , she suffered a brain aneurysm . in march 2015 , she is expected to fully recover after surgery in the summer of 2015 .
Prediction	in march 2015 , she suffered a brain aneurysm . however , she is expected to fully recover after surgery in the summer of 2015 .
Target	in march 2015 , she suffered a brain aneurysm but is expected to fully recover after surgery in the summer of 2015 .
Source	johann adam weishaupt died in gotha in 1830 . johann adam weishaupt is the founder of the order of the illuminati .
Prediction	johann adam weishaupt died in gotha in 1830 . johann weishaupt is the founder of the order of the illuminati .
Target	johann adam weishaupt , the founder of the order of the illuminati , died in gotha in 1830 .
Source	ohio state senator . ohio state served a six - year term as a justice of the ohio supreme court .
Prediction	ohio state senator . ohio secretary of state served a six - year term as a justice of the ohio supreme court .
Target	ohio state senator and served a six - year term as a justice of the ohio supreme court .
Source	as minister he was responsible for this event . there was considerable controversy surrounding the disaster he did not step down .
Prediction	as minister he was responsible for this event . however , there was considerable controversy surrounding the disaster he did not step down .
Target	as minister he was responsible for this event , although there was considerable controversy surrounding the disaster he did not step down .

Table 5: Random samples of correct and incorrect prediction on the DiscoFuse development set by the MASKER model going from unfused to fused sentences (Exact score 12.65).

Random Sample of Correct MASKER Predictions	
Source	so far i 'm not really impressed .
Prediction	so far i 'm really impressed .
Source	either way i would never recommend buying from camping world .
Prediction	either way i would recommend buying from camping world .
Source	this is a horrible venue .
Prediction	this is a great venue .
Source	this place is a terrible place to live !
Prediction	this place is a great place to live !
Source	i 'm not one of the corn people .
Prediction	i 'm one of the corn people .
Source	this is easily the worst greek food i 've had in my life .
Prediction	this is easily the best greek food i 've had in my life .
Source	the sandwich was not that great .
Prediction	the sandwich was great .
Source	its also not a very clean park .
Prediction	its also a very clean park .
Random Sample of Incorrect MASKER Predictions	
Source	also , could they not bring a single pack of cheese or red peppers ?
Prediction	also , could they bring a single pack of cheese or red peppers ?
Target	they had plenty of cheese packets and red pepper.
Source	service was average but could not make up for the poor food and drink .
Prediction	service was good but could not make up for the poor food and drink .
Target	service was above average as well as the food and drink .
Source	the only saving grace was the black beans .
Prediction	the saving grace was the black beans .
Target	one of several saving graces was the black beans
Source	the rest of their food is edible but their employees and service are horrible .
Prediction	the rest of their food is edible and their employees and service are horrible .
Target	the food is great but the employees werent moving fast enough
Source	crab cakes , salt , no other flavor .
Prediction	crab cakes , salt , no other .
Target	crab cakes were salty and delicious.
Source	i was very disappointed with this place .
Prediction	i love this place .
Target	i wasn't disappointed with this place at all.
Source	my pad thai tasted like thai rice noodles with barbeque sauce .
Prediction	the thai tasted like thai rice noodles with barbeque sauce .
Target	the pad thai tasted like the best i have had in a while
Source	they also have lost sight of what good deli food is .
Prediction	they also have lost sight of how good deli food is .
Target	the know what good deli food is

Table 6: Random samples of correct and incorrect prediction on the Yelp review test set by the MASKER model going from negative to positive reviews.

