SegMix: A Simple Structure-Aware Data Augmentation Method

Anonymous ACL submission

Abstract

Many Natural Language Processing tasks involve predicting structures, such as Syntax Parsing and Relation Extraction (RE). One central challenge in supervised structured prediction is the lack of high-quality annotated data. The recently proposed interpolation-based data augmentation (DA) algorithms (i.e. *mixup*) augment the training set via making convex interpolation between training data points (Zhang et al., 2018). However, current algorithms (e.g. SeqMix (Zhang et al., 2020), LADA (Chen et al., 2020a)) that apply *mixup* to language structured prediction tasks are not aware of the syntactic or output structures of the tasks, making their performance unstable and requiring additional heuristic constraints. Furthermore, SeqMix-like algorithms expect a linear encoding scheme of the output structure, such as BIO-Scheme for Named Entity Recognition (NER), restricting its applicability.

To this end, we propose **SegMix**, a simple framework of interpolation-based algorithms that can adapt to both the syntactic and output structures, making it robust to hyper-parameters and applicable to different tasks. We empirically show that SegMix consistently improves performance over several strong baseline models on two structured prediction tasks (NER and RE). SegMix is a flexible framework that unifies existing rule-based language DA methods, creating interesting mixtures of DA techniques. Furthermore, the method is easy to implement and adds negligible overhead to training and inference.

1 Introduction

Data augmentation (DA), which introduces unobserved data based on the observed data (van Dyk and Meng, 2001), is a common strategy used in machine learning to deal with data-scarcity problems. Recently DA has received increasing attention in Natural Language Processing (NLP) due to the emergence of tasks in low-resource languages and

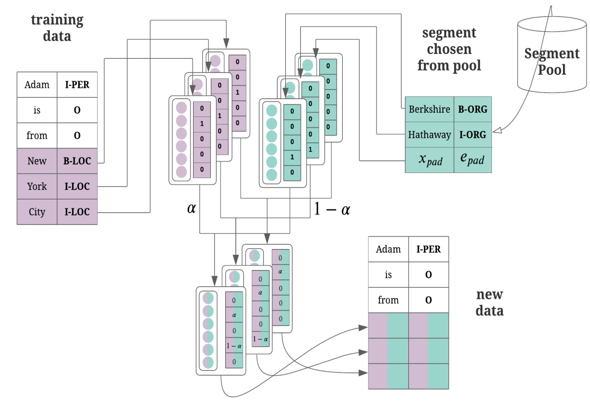

Figure 1: Example of SegMix for NER. On the left is the training sentence with NER tags; the colored block is the chosen segment (entity in this case). On the right is a segment randomly chosen from a predefined pool. A new segment is produced by performing a linear interpolation between the two segments. Then finally, the augmented data is generated by replacing the original segment with the mixed one.

large-scale models that require large amounts of data (Feng et al., 2021). Existing DA for NLP can be categorized into rule-based, interpolation-based, and model-based (Feng et al., 2021).

We focus on interpolation-based DA on structured prediction tasks, which interpolates the inputs and labels of two or more training examples (Feng et al., 2021). Proposed in *mixup* (Zhang et al., 2018), the interpolation DA method is initially used in computer vision (CV) tasks. Zhang et al. 2018 argues that *mixup* regularizes the model to favor simple linear behavior in-between training examples. Driven by the success of *mixup* on CV tasks, several attempts have been made to apply similar interpolations in language tasks (Chen et al. 2020b, Cheng et al. 2020, Miao et al. 2020).

A challenge to perform *mixup* in NLP task its requirements for continuous inputs and outputs (Feng et al., 2021) since both need to be linearly interpo-

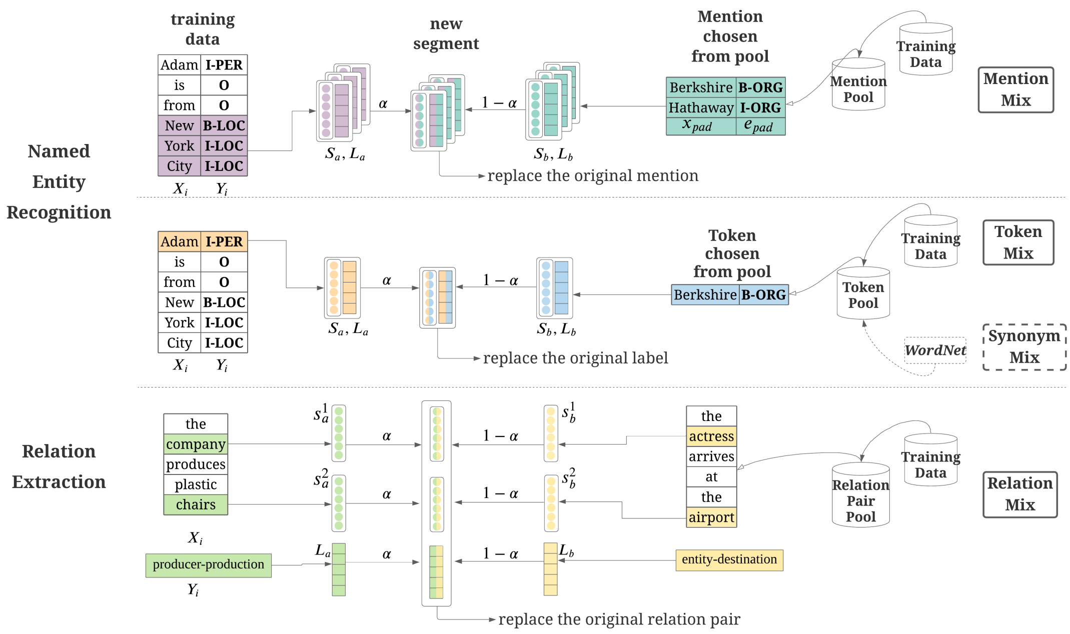

Figure 2: Different variations of SegMix (MentionMix, SynonymMix, and RelationMix). The left is the original training sequence. The colored blocks are the segments to be mixed. Segments on the right are returned randomly from the predefined pool. Mention Pool and Relation Pair Pool are constructed from the training data, while the Synonym Pool is constructed with a pretrained WordNet and returns a synonym of the chosen token.

lated to create augmented example. For instance, SeqMixS[1] (Guo et al., 2020) proposed to interpolate sentence embeddings under Seq2Seq settings.

However, the proposed embedding-mix solution does not solve structured prediction tasks (predicting a predefined target structure extracted from an unstructured input (Smith, 2011)). For example, in Named Entity Recognition (NER), which aims to recognize mentions from text belonging to predefined semantic types such as person, location, organization etc (Nadeau and Sekine, 2007). Mixing two sentences without a matching target structure will generate unsensible output structures (examples provided in Fig. 4), potentially confusing the model. LADA (Chen et al., 2020a) validated this through experiments: when applying SeqMixS directly to NER task, they found that the generated data was too "noisy". SeqMixS sometimes breaks the syntactic and output structure, which is important for structured prediction tasks.

Another example is Relation Extraction (RE) tasks, which aims to classify the relation type between two predefined nominals in the sentence. Unlike BIO tagging scheme commonly used in NER tasks, most existing methods in RE do not have a linear encoding scheme. Thus it is not straightforward to apply SeqMixS directly to RE.

Even in applicable tasks, existing work uses extra heuristic constraints to ensure high-quality augmented data. For example, LADA mixes sentences with a similar embedding only, SeqMix (Zhang et al., 2020) uses an additional discriminator to filter out "noisy" data. These constraints add complexity to the methods and limit the explorable data space. Empirically, we also find that these methods are sensitive to hyperparameters like augmentation rates ($\frac{\text{\#of augmented data}}{\text{\#of training data}}$). A bad augmentation rate sometimes harms model performance, leading to worse scores than baseline.

To address these problems, we propose **Segment Mix (SegMix)**, a DA method that performs linear interpolations on meaningful, task-related segments to preserve the syntactic and output structures. The segments are randomly replaced with the interpolation of the original segment and another segment drawn from a predefined segment pool. Specifically, we explore two popular structured prediction tasks: Named Entity Recognition (NER) and Relation Extraction (RE). We empirically show

[1]Originally named SeqMix, we use SeqMixS to avoid confusion with the other SeqMix (Zhang et al., 2020). "S" stands for Seq2Seq.

that SegMix improves model performance consistently on different experimental setups and hyperparameters, demonstrating its robustness. Furthermore, SegMix imposes few constraints on the original data or the mixing pairs, potentially allowing it to explore a much larger data space. The method can also be extended flexibly into other structured prediction tasks by defining task-related segments.

SegMix connects several existing DA methods. The replacement-based DA methods are a "hard" version of SegMix which replaces the segments completely. The original SeqMixS is a variation with a segment defined as the whole sequence.

2 Related Work

Rule-based DA. Rule-based DA specifies rules to insert, delete, or replace part of the text (van Dyk and Meng, 2001). Easy Data Augmentation (Wei and Zou, 2019) proposed a set of token-level random perturbation operations (insertion, deletion, and swap) (Dai and Adel, 2020). SwitchOut (Wang et al., 2018) randomly replaces words in the sentence with other random words. WordDrop (Sennrich et al., 2016a) drops tokens at random. These methods explore the vicinity area around the data point and assume they share the same label.

Interpolation-based DA. Originally proposed for image classification tasks, *mixup* (Zhang et al., 2018) performs convex combinations between a pair of data points and their labels. *mixup* improves the performance in image classification tasks by regularizing the neural network to favor simple linear behavior in-between training examples (Zhang et al., 2018). There have been several adaptations of *mixup* on NLP tasks. TMix (Chen et al., 2020b) performs an interpolation of text in hidden space on text classification tasks. Snippext (Miao et al., 2020) mixes up BERT encodings and passes them through a classification layer for sentiment analysis tasks. AdvAug (Cheng et al., 2020) mixes adversarial examples as an adversarial augmentation method for Neural Machine Translation.

However, direct application of whole sequence level *mixup* yields little improvement in structured prediction tasks. As shown empirically in LADA (Chen et al., 2020a) on NER, direct mixing of two sentences changes both local token representation and the context embeddings required to identify the mention entity (Chen et al., 2020a). Thus LADA adds additional constraints by mixing the sequences only with its k-nearest neighbors to reduce the noises (Chen et al., 2020a). SeqMix (Zhang et al., 2020) scans both sequences with a fixed-length sliding window and mixes the sub-sequence within the windows. However, this approach does not eliminate the problem of generating low-quality data — extra constraints are needed ensure the quality of generated data. These constraints complicate the method and constrain the explorable data space.

Structured Prediction. In structured prediction tasks, a predefined target structure is extracted from the input sequences (Smith, 2011). Common tasks include POS tagging, Named Entity Recognition (NER), and Relation Extraction (RE). There have been several attempts applying *mixup*-like algorithms to NER (Chen et al., 2020a; Zhang et al., 2020). Unlike NER, RE models typically do not use a linear encoding scheme (i.e. BIO). Thus it is not straightforward to apply SeqMix. To the best of our knowledge, interpolation-based DA methods have not been applied to RE tasks.

Model-based DA Model-based DA uses pre-trained models to generate augmented data. Back-translation translates the input sequence into another language and back to the original (Sennrich et al., 2016b). G-DAUG[c] (Yang et al., 2020) generates synthetic examples using pretrained language models. Although useful for some sequence classification tasks, it is not straightforward to apply similar techniques to structured prediction tasks since the output structure is hard to be reconstructed after replacement of the whole sequence. Unsupervised Data Augmentation (Xie et al., 2019) noises unlabeled examples produced by advanced DA methods under the same consistency training framework. Hu et al. 2019 proposes to learn different DA schemes with the same gradient-based algorithm, which adapts a reward learning algorithm from Reinforcement Learning for joint data manipulation learning and model training. These algorithms assume extra models or change the model structure, while this work focuses on simple DA methods by combining rule-based and interpolation based methods.

3 Method

Consider a training dataset $\mathcal{D} = \{(X_i, Y_i) | i \in N\}$ of size N, where each input X_i is a sequence of tokens $X_i = (X_i^1, X_i^2, \ldots,)$ and a task-dependent structured output Y_i, a structured prediction algorithm generally encodes the output Y_i using a task-dependent scheme. For example, NER labels are

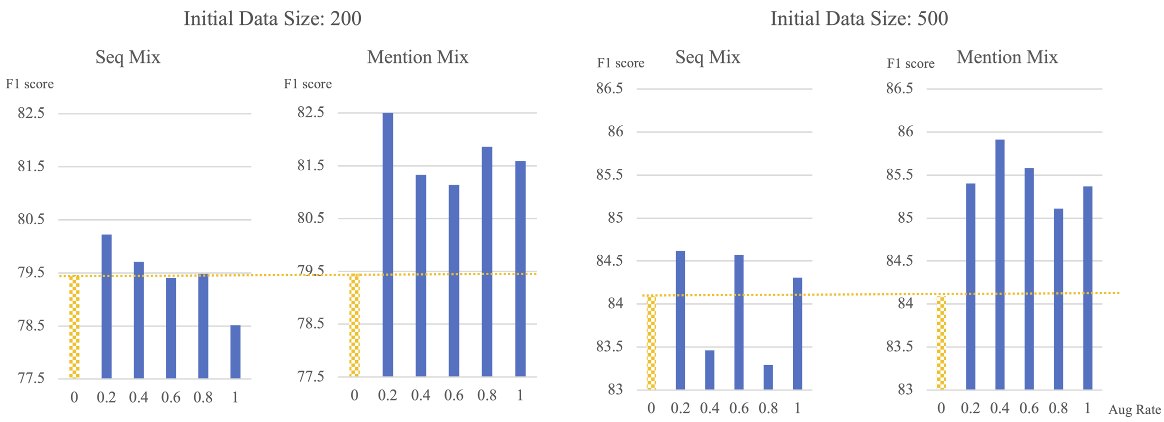

Figure 3: F1 score with variant augmentation rates with MentionMix and SeqMix. The dashed line represents the baseline performance. MentionMix constantly outperforms the baseline performance, while SeqMix is unstable and sometimes oscillates below the baseline and less overall improvement.

often encoded with the BIO-scheme, such that each token in X_i is associated with a label. In Relation Extraction, a label is associated with a pair of nominal phrases. SegMix is flexible to adapt to different encoding schemes by designing task-dependent segments, easily applicable to different tasks.

Formally, given a training instance, a segment $s(u, v)$ is a continuous sequence of tokens $(X_i^u, X_i^{u+1}, \ldots, X_i^v)$, a segment list S_j is a list of segments from the instance. We choose segment lists that are meaningful to the task. For example, in Relation Extraction, we use segment lists of length 2, containing the pair of nominals of a relation. We further associated each segment list with an appropriate label list L_j (more details below).

Segment Pool: A segment pool of size M: $\mathcal{P}^k = \{(S_j, L_j) | j \in M\}$ is generated by collecting all segment lists S_j available for mixing. The pool can be constructed from the training data or an external resource. Here, k refers to the length of segment list, which is a constant for a specific task.

Segment Mix: SegMix performs linear interpolation on a task-dependent segment lists. As demonstrated in Algo.1, with training data set $\mathcal{D}$, Segment Pool $\mathcal{P}^k$, mix rate r, `SegMix` $(\mathcal{D}, \mathcal{P}^k, r)$ returns an augmented data set $\mathcal{D}_A$ of size $r \cdot N$. For each data point (X_i, Y_i) drawn from the training set, we randomly pick a segment list S_a and the corresponding label list L_a. We then draw the other pair (S_b, L_b) from the segment pool.

Let $\mathbf{Emb}$ be an embedding function on $\mathbb{R}^V \mapsto \mathbb{R}^D$, here V is size of the vocabulary, and D is the embedding dimension. Let $\mathbf{OHE}$ be a function that returns the one-hot encoding of a label. For all $s_a, s_b = S_a[i], S_b[i], 1 \leq i \leq \text{len}(S_a)$, and $l_a, l_b = L_a[j], L_b[j], 1 \leq j \leq \text{len}(L_a)$. Define $e_a, e_b = \mathbf{Emb}(s_a), \mathbf{Emb}(s_b), o_a, o_b = \mathbf{OHE}(l_a), \mathbf{OHE}(l_b)$. The embeddings and one-hot encodings are then padded according to sequence length. Let $\tilde{e}_a, \tilde{e}_b, \tilde{o}_a, \tilde{o}_b$ be the padded version of the embeddings and one-hot encodings. Finally, we perform a linear interpolation between $\tilde{e}_a, \tilde{e}_b$ and $\tilde{o}_a, \tilde{o}_a$ with a mix rate λ chosen randomly from a Beta distribution (see specifications in 4.1):

$$e'_a \leftarrow \tilde{e}_a \cdot \lambda + \tilde{e}_b \cdot (1 - \lambda) \quad (1)$$

$$o'_a \leftarrow \tilde{o}_a \cdot \lambda + \tilde{o}_b \cdot (1 - \lambda) \quad (2)$$

In Eq.1, 2, $\cdot$ is a scalar multiplication, and $+, -$ are vector element-wise operations. When $\lambda = 1$, the augmented data falls back to the original one. When $\lambda = 0$, the segments are completely replaced by the segments drawn from the pool, equivalent to replacement-based DA techniques.

Finally, the augmented data point is generated by copying the original data and replacing the chosen segment and labels with the mixed version.

We present 3 variations of SegMix for NER and 1 for RE with different types of Segment Pool $\mathcal{P}^k$.

MentionMix Inspired by Mention Replacement (MR), MentionMix performs linear interpolations on a mention level (a contiguous segment of tokens with the same entity label). A mention pool $\mathcal{P}^1$ is constructed by scanning through the training data set and extracting all mention segments and their corresponding labels. Thus each segment list is composed of a single mention and a list of entity

Algorithm 1 SegMix $(\mathcal{D}, \mathcal{P}^k, r)$

$\mathcal{D}_A \leftarrow \{\}$
$\mathcal{D}_S \leftarrow$ sample$(\mathcal{D}, len(\mathcal{D}) \cdot r)$
for (X_i, Y_i) in $\mathcal{D}_S$ **do**
  $E_i, O_i \leftarrow$ **Emb**(X_i), **OHE**(Y_i)
  $\lambda \leftarrow Beta(\alpha, \alpha)$
  $S_a, l_a \leftarrow$ random k segment lists in X_i, Y_i
  $S_b, l_b \leftarrow$ random k segment lists in $\mathcal{P}$
  $X_i', Y_i' \leftarrow X_i$.copy(), Y_i.copy()
  for s_a^j, s_b^j in S_a, S_b **do**
    $e_a, e_b =$ **Emb**(s_a), **Emb**(s_b)
    $start, end \leftarrow$ index range of s_a^j in X_i
    $\tilde{e}_a^j, \tilde{e}_b^j \leftarrow$pad_to_longer$(e_a^j, e_b^j)$
    $E_i[start : end] \leftarrow \tilde{s}_a^j \cdot \lambda + \tilde{s}_b^j \cdot (1 - \lambda)$
  end for
  for l_a^j, l_b^j in l_a, l_b **do**
    $o_a, o_b =$ **OHE**(l_a), **OHE**(l_b)
    $start, end \leftarrow$ index range of l_a^j in Y_i
    $\tilde{o}_a^j, \tilde{o}_b^j \leftarrow$pad_to_longer$(o_a^j, o_b^j)$
    $O_i[start : end] \leftarrow \tilde{o}_a^j \cdot \lambda + \tilde{o}_b^j \cdot (1 - \lambda)$
  end for
  $\mathcal{D}_A$.add((E_i, O_i))
end for
Output $\mathcal{D}_A$

labels encoded with BIO-scheme.

TokenMix Inspired by Label-wise Token Replacement (LwTR), TokenMix performs linear interpolations on a token level. We use tokens with entity labels in BIO-scheme from the training datasets as the Token Pool $\mathcal{P}^1$. Each segment list is composed of a single token and the label.

SynonymMix Inspired by Synonym replacement (SR), we construct the Synonym Pool $\mathcal{P}^1$ from an external resource. Specifically, the pool returns a synonym of the token in the original sequence based on *WordNet* (Miller, 1995). We assume the two synonyms share the same label, thus interpolation only happens within input.

RelationMix We also study RE as an example where SeqMix is not directly applicable. Since each relation is composed of two possibly non-adjacent nominals in a sentence, we construct a pool $\mathcal{P}^2$ with groups of two nominals and a relation label[2]. During mixing phase, the two nominals and their corresponding relation labels is mixed with another pair of nominals from $\mathcal{P}^2$.

4 Experiments

We conduct experiments on two structured prediction tasks: Name Entity Recognition (NER) and relation Extraction (RE). The NER experiments are on two datasets in different languages: CoNLL-2003 (Sang and Meulder, 2003) in English with 4 entity types and GermEval (Benikova et al., 2014) in German with 12 entity types. Given an input sequence, the task is to identify all entities positions and their types, such as location, organization, and person. We use the BIO-tagging scheme so that I-XXX denotes the word inside an entity and B-XXX denotes the word at the beginning.

The RE experiment is on SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals (Hendrickx et al., 2019). Given a sequence with two predefined nominals, the task is to determine the semantic relations between the pair. For example, in the sentence "The actress arrives at the airport", nominal "actress" and "airport" have an entity-destination relation. There are 9 relation types in total, such as Cause-Effect, Product-Producer, Entity-Destination, etc.

In order to create a data-scarce setting, we randomly sample $5\%, 10\%, 30\%$[3] of the original training data as training set. The validation dataset and test dataset are unchanged.

To compare with existing interpolation-based methods, we also run experiments on the best model in LADA (Inter+Intra LADA, code available on Github[4]) without extra unlabeled data. To compare with rule-based techniques, we implement Mention Replacement, Synonym Replacement, Label Replacement, and Relation Replacement as special cases of SegMix - setting the mix rate λ to 1 so that the segment is entirely replaced.

Label Smoothing(LS), assigning data with a soft "label" instead of 0/1 values is a common technique used to prevent the network from becoming over-confident (Müller et al., 2019). To show that SegMix can provide additional benefits on top of LS, we also compare the results of the baseline model with LS only and with both LS and SegMix.

[2]The order of nominals is contained in the labels. For example, the label list contain both producer-product(e1,e2) and producer-product(e2,e1)

[3]700, 1400, 4200 for CoNLL-2003; 1200, 2400, 7200 for GermEval; 400, 800, 2400 for SemEval-2010 Task 8

[4]https://github.com/GT-SALT/LADA

	CoNLL-2003			GermEval		
	5%	10%	30%	5%	10%	30%
BERT	83.28	86.85	89.28	70.28	75.64	79.63
BERT + LADA (Chen et al., 2020a)	84.85	87.85	89.87	71.32	77.51	81.95
BERT + Mention Replacement	85.69	87.37	89.00	74.51	75.98	80.83
BERT + Synonym Replacement	86.09	87.95	89.25	73.77	73.26	75.52
BERT + Label Replacement	85.69	87.37	89.00	73.26	79.49	79.20
BERT + **MentionMix** †	86.81	**88.78**	**90.14**	**76.06**	**80.32**	**83.48**
BERT + **SynonymMix** †	**87.07**	88.39	89.87	75.07	78.64	80.89
BERT + **TokenMix** †	84.51	87.08	88.08	74.48	77.07	80.99
BERT + Label Smoothing	84.86	86.66	88.25	71.32	77.51	81.95
BERT + **MentionMix** †+ Label Smoothing	**87.07**	88.39	89.87	75.07	79.99	82.31

Table 1: F1 scores on CoNLL 2003 and GermEval under different training data size settings ($5\%, 10\%, 30\%$) compared with LADA and replacement-based augmentation methods. SegMix consistently outperforms other methods under various initial data sizes, especially under data-scarce setting (around 3% improvement on the baseline with 5% of training data and 2% improvement with 10% of training data). †denotes our methods.

	5%	10%	30%
BERT	56.68	73.42	82.33
BERT + **Replacement**	55.98	67.57	79.72
BERT + **RelationMix** †	**60.32**	**73.75**	**82.44**

Table 2: F1 scores of RelationMix on SemEval-2010 under different training data size settings compared with replacement-based augmentation.

4.1 Implementation Details

Throughout our experiments, we adopt the pre-trained *bert−base−uncased*[5] (Vaswani et al., 2017) model for CoNLL-2003 and SemEval-2010, *bert−base−multilingual−uncased* for GermEval as the encoder, a linear layer to make prediction, and a soft cross-entropy loss. We train all the models for 100 epochs in maximum and take the checkpoint with the maximum validation score as the final model. The initial learning rate is set to $5e-5$, 0.1 for weight decay, and 8 for the α in the beta distribution from which we generate the mix rate[6].

4.2 Results

We conduct experiments under various numbers of training data ($5\%, 10\%, 30\%$ of original training data) and compare them with existing DA methods. The results for NER are shown in Table 1. On both CoNLL-2003 and GermEval, MentionMix has the best performance, exceeding the performance of sequence-level mix and replacement. SegMix is particularly useful under data-scarce situations - improving the baseline architecture by 3% on CoNLL and 6% on GermEval in terms of absolute F1 scores, under the 5% data settings. However, we notice that performance of TokenMix is not as stable as MentionMix and SynonymMix on NER - yielding around the same results as interpolation-based and rule-based methods. We hypothesize that mixing on a token level might break the original mention structure (e.g. a token with label I-ORG might be mixed with another with label B-PER).

On SemEval, we compare RelationMix (mixing pairs of nominals and corresponding relation labels) with baseline and Relation Replacement (replacing nominal pairs). We find that simple replacement worsens the baseline performance, while RelationMix improves the baseline, especially under data-scarce situation - A 4% absolute F1 improvement under the 5% setting.

Overall, SegMix methods consistently outperform their replacement-based counterparts and sequence-level mix (e.g. LADA, SeqMix). This result is consistent with our hypothesis that "soft" mix of data points on structure-aware segments yields better results than "hard" replacement or mixing on a whole-sequence level.

[5]https://github.com/huggingface/transformers

[6]We perform ablation study on α in Appendix A.1 and find that α has no significant impact on the performance.

Augmentation Rate	1%	3%	5%	10%	30%	Average
0 (Baseline)	79.46	84.15	83.28	86.85	89.28	+0
0.1	82.10	85.57	85.93	88.04	89.92	+1.41
0.2	**82.57**	85.40	**86.61**	88.67	89.52	**+1.68**
0.3	81.45	85.73	86.47	88.60	**90.14**	+1.54
0.4	81.33	**85.91**	86.03	88.45	89.85	+1.34
0.5	81.00	85.57	86.32	88.12	89.85	+1.27
0.6	81.14	85.58	86.21	88.03	89.79	+1.24
0.7	81.61	85.80	86.55	**88.78**	89.25	+1.45
0.8	81.86	85.11	86.35	88.05	89.71	+1.40
0.9	81.02	85.53	86.24	88.30	89.25	+1.17
0.1	81.59	85.37	86.06	87.98	89.90	+1.31
Average	81.57	85.43	86.27	88.30	89.72	+1.38

Table 3: F1 scores of MentionMix on CoNLL 2003 with variant augmentation rates ($\frac{\text{\#of augmented data}}{\text{\#of training data}}$) under different initial data sizes. SegMix consistently improves over the baseline, demonstrating its stability and robustness over varying augmentation rates. The last row is the averaged improvement score for each augmentation rate over different initial data sizes. The last column is the averaged score for each initial data size over different augmentation rates.

Robustness with respect to augmentation rate. A restriction we find in previous attempts on SeqMix is that the model performance tends to drop below the baseline as the augmentation rate rises above a certain value (Zhang et al., 2020). As demonstrated in Fig.3, the F1 scores for SeqMix sometimes get below the baseline score. Such an unstable performance could add a significant burden in hyperparameter tuning. Furthermore, the optimal augmentation rate varies for different initial data settings. A good augmentation rate for 200 data size might not be good for 500 data size. Through experiments on varying augmentation rates under 5 different data-scarcity settings, we show that MentionMix consistently improves the baseline performance under different augmentation rates and data usage settings, making it more applicable in practical contexts. The specific scores are presented in Table 3.

Original: **Swedish** [MISC] options and derivatives exchange **OM Gruppen AB** [ORG] said on Thursday it would open an electronic bourse for forest industry products in **London** [LOC] in the first half of 1997.
MentionMix: **Swedish** [MISC] options and derivatives exchange **Javier Gomez de** [PER/ORG] said on Thursday it would open an electronic bourse for forest industry products in **London** [LOC] in the first half of 1997.
SeqMixS: **Sweden** [MISC/ORG] **option** [O/ORG] but [unused33] transfer **. . [unused10]** [O/ORG] saying to Friday them might closed his electronics . with woods companies Products of **Paris** [O/LOC] of a second three in 1995.

Figure 4: Mixed sentence samples recovered by mapping embeddings to the nearest token (L2 distance). [A/B] represents the linear interpolation of the one-hot encodings of the two labels A and B.

4.3 Analysis

We argue that SegMix, which linearly interpolates data points on segments meaningful to the task, keeps the syntactic and output structure intact. To help understand the mixed instances, we choose some sample sequence in CoNLL 2003, and visualize it in Fig. 4 by mapping the mixed embeddings to the nearest word in the vocabulary.

The mixed example generated by MentionMix preserves the syntactic and entity structures while achieving linear interpolation between each mention. On the other hand, the example generated by SeqMixS is not semantically meaningful. Specifically, due to the high proportion of non-entity phrases in the dataset, SeqMix tends to mix entity mentions with non-entity segments (label [O]). The resulting sentences often contain non-meaningful entities (e.g. *option* and *. . [unused10]* in Table), but are being perceived as entities (with non-[O] label). The non-entity phrases in the sentence would

Ex. 1	Baseline	**English** [MISC] county sides and another against **British Universities** [MISC]
	MentionMix	**English** [MISC] county sides and another against **British Universities** [ORG]
Ex. 2	Baseline	May 22 First one-day international at **Headingley** [ORG]
	MentionMix	May 22 First one-day international at **Headingley** [LOC]
Ex. 3	Baseline	July 9 v **Minor Counties** [MISC] XI
	MentionMix	July 9 v **Minor Counties** [ORG] XI

Table 4: Examples of cases predicted by the baseline model and MentionMix from validation dataset. The bold segments represent an entity mention, blue segments represent an misclassified mention.

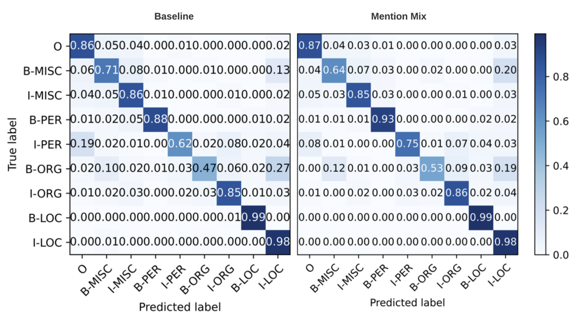

Figure 5: Confusion Matrix on CoNLL 2003 with and without SegMix with 5% of training data.

also be mixed, producing semantically incorrect context phrases like *second three in 1995*.

We also examine the model's confidence calibration – how well the model is predicting probability estimates representative of the true correctness likelihood (Guo et al., 2017). We use Expected Calibration Error (Naeini et al., 2015) (ECE) - a weighted average of accuracy/confidence difference as a metric to examine calibration and find that MentionMix is better calibrated. We observe that the ECE score drops from 3.2% to 1.2% after applying MentionMix. We also find that MentionMix continues to improve the model with Label Smoothing (Table 1). We argue that linear interpolation of both inputs and labels explores a larger data space than a simple soft perturbation in the label, thus leading to further improvement. We leave the theoretical analysis to future work.

Error Analysis We compare the confusion matrix of the baseline model and MentionMix for each classes for 5% of CoNLL 2003 data in Fig. 5. There is an overall improvement in the accuracy for each class, especially for PER and ORG. Before SegMix, the model tends to mistakenly predict [LOC] for [ORG] (27% $\rightarrow$ 19%), and [O] for [PER] (19% $\rightarrow$ 8%). MentionMix introduces more variations of meaningful entities into training, preventing the model from predicting a fixed label.

We also list some improved cases in Table 4, Ex. 1 and 2 is a case of correction between for ORG, while Ex. 3 is a case where the entity label is correct, but the mention range remains incomplete (both predicts *Minor Counties* as a mention instead of *Minor Counties XI*).[7]

Observing cases like Ex. 3, we hypothesize that SegMix mainly helps the model to distinguish between ambiguous types instead of span detection. To validate this claim, we convert all mentions to [B] and [I] during inference phase and find out that there is little difference between the models (both around 98%) in terms of span accuracy — confirming our hypothesis.

5 Conclusion

This paper proposes SegMix, a simple data augmentation technique that is effective in data-scarce situations for structured prediction tasks. By choosing task-dependent segments, the augmented examples still preserve reasonable syntactic and output structures while also exploiting the benefits of linearity of data space. Furthermore, it extends the application range of *mixup* in NLP tasks. We demonstrate its robustness by evaluating model performance under various settings on two NER datasets and one RE dataset. Our experiments indicate that SegMix consistently improves the model performance and outperforms other methods. SegMix is a framework that unifies several rule-based and interpolation-based methods, which puts little constraint on data structure and is straightforward to use. SegMix opens up several possibilities for further exploration. The flexibility of SegMix makes it possible to extend it to other NLP tasks. Besides supervised learning, we also plan to study SegMix under unsupervised and semi-supervised settings.

[7] We also list some cases for RE in Appendix.A.2

References

Darina Benikova, Chris Biemann, and Marc Reznicek. 2014. Nosta-d named entity annotation for german: Guidelines and dataset. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2524–2531, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jiaao Chen, Zhenghui Wang, Ran Tian, Zichao Yang, and Diyi Yang. 2020a. Local additivity based data augmentation for semi-supervised NER. *CoRR*, abs/2010.01677.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020b. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification.

Yong Cheng, Lu Jiang, Wolfgang Macherey, and Jacob Eisenstein. 2020. Advaug: Robust adversarial augmentation for neural machine translation. *CoRR*, abs/2006.11834.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. *CoRR*, abs/2010.11683.

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.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. *CoRR*, abs/1706.04599.

Demi Guo, Yoon Kim, and Alexander Rush. 2020. Sequence-level mixed sample data augmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5547–5552, Online. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2019. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *CoRR*, abs/1911.10422.

Zhiting Hu, Bowen Tan, Ruslan Salakhutdinov, Tom M. Mitchell, and Eric P. Xing. 2019. Learning data manipulation for augmentation and weighting. *CoRR*, abs/1910.12795.

Zhengjie Miao, Yuliang Li, Xiaolan Wang, and Wang-Chiew Tan. 2020. Snippext: Semi-supervised opinion mining with augmented data. *CoRR*, abs/2002.03049.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? *CoRR*, abs/1906.02629.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30.

Mahdi Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, page 2901–2907.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *CoRR*, cs.CL/0306050.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Noah A. Smith. 2011. *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

David A van Dyk and Xiao-Li Meng. 2001. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. 2019. Unsupervised data augmentation. *CoRR*, abs/1904.12848.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug

Downey. 2020. Generative data augmentation for commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1008–1025, Online. Association for Computational Linguistics.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization.

Rongzhi Zhang, Yue Yu, and Chao Zhang. 2020. Seqmix: Augmenting active sequence labeling via sequence mixup. *CoRR*, abs/2010.02322.

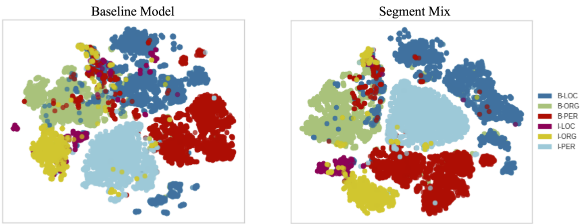

Figure 6: Visualization of label distribution by t-SNE of baseline model v.s. SegMix.

α	F1 score
1	86.79
2	86.75
4	86.79
8	86.81
16	86.45

Table 5: Ablation study on α in beta distribution, which is used to generate random mix rate.

A Appendix

A.1 Ablation Study on α

The mix rate λ (rate by which two segments are mixed) in our experiments is randomly drawn from a beta distribution $(beta(\alpha, \alpha)$. To determine if α matters, we vary a set of αs on ConLL-2003 dataset with 5% of initial data. As shown in Table 5, varying α has negligible influence on the performance.

A.2 Case Study for Relation Extraction

We also list some error cases for Relation Extraction in Table 6.

A.3 t-SNE Visualization

We also plot out the t-SNE of the baseline model and after MentionMix. as shown in Fig. 6, MentionMix is able to achieve a better separation across different distributions.

Ex. 1	the complete [**statue**]$_{e_1}$ topped by an imposing [**head**]$_{e_2}$was originally nearly five metres high
True Relation	Other
Baseline Prediction	Component-Whole(e2,e1)
MentionMix Prediction	other
Ex. 2	the [**slide**]$_{e_1}$which was triggered by an avalanche - control [**crew**] $_{e_2}$ damaged one home and blocked the road for most of the day
True Relation	Cause-Effect(e2,e1)
Baseline Prediction	Product-Producer(e1,e2)
MentionMix Prediction	Cause-Effect(e1,e2)

Table 6: Examples of correctly classified cases after MentionMix in validation dataset. The bold segments represents an entity mention, blue segments represent an misclassified mention.