No Syntaxation Without Representation: Syntactic Considerations for Neural Machine Translation Data Augmentation

Garyk Brixi
Harvard College
garykbrixi@college.harvard.edu

Jenna Landy
Harvard GSAS
jlandy@g.harvard.edu

Arpan Sarkar
Harvard GSAS
arpan_sarkar@g.harvard.edu

Jie Sun
Harvard School of Public Health
jie_sun@hsph.harvard.edu

Abstract

Data augmentation improves accuracy of ML models for natural language processing by increasing the amount and variety of training data. Augmentation approaches for NLP can be applied at the *token-level* (e.g. contextual replacement) or at the *embedding-level* (e.g. soft contextual replacement or mixing two sequences by averaging their embeddings with SeqMix). This paper extends existing augmentation methods to maintain part of speech (POS) and further extends SeqMix to combine similar or dissimilar sequences. While methods prior to this paper keep the semantic meaning of a sentence, a weakness is that they don't maintain syntax. We address this by matching POS in word replacement and token mixing, which shows up to a 1 point increase in BLEU. Further, in prior SeqMix methods, the sequences to be mixed are chosen at random, which we address by combining more similar or different sequences. We find that mixing sequences of similar length shows up to a 0.6 point improvement in BLEU.

1 Introduction

Machine translation (MT) models require a large amount and variety of data to train well. Data augmentation adds synthetic training data and improves NLP classification and translation. Common augmentation methods hinge on creating new, *realistic*, artificial data based on true data with a *small* amount of added noise.

Some data augmentation methods are implemented at the *token-level*, such as randomly removing a word from a sentence (drop) or using a separately trained language model (LM) to predict a good replacement word (LM sample), resulting in new, artificial training text that is readable. Others look at the *word embedding-level*, such as combining embeddings of two sentences at each position to capture the average meaning (SeqMix). In these cases, the augmented sequences are not readable, but still have a sense of meaning because similar embedding vectors indicate similar words. Generally, embedding-level augmentations have been shown to improve upon token-level methods.

A source of weakness in these methods is that while they maintain the semantic meaning of a sentence, they *don't maintain syntax, including part of speech* (POS). Even the LM-based methods put very little weight on words matching the original POS (<5%, besides pronouns). While this may be okay for classification tasks, in MT we expect translated sentences to be syntactically correct. To create new, *realistic*, artificial data for MT, a focus on syntax may be vital. In an attempt to keep proper syntax, this paper offers elegant extensions to existing techniques that maintain POS. On a MT task from German to English, the new methods improve BLEU score by up to 1 point (10.8 to 11.8).

Another weakness is that current token- and embedding-level sequence mixing methods *mix tokens based on position*, and the sequences to be mixed are *chosen at random*. Both of these decisions lead to extremely noisy training sequences that may not maintain semantics *or* syntax. Instead, this paper introduces methods mixing tokens by matching POS to keep realistic syntax, mixing sequences of the same length to keep realistic sentence length, and mixing semantically similar sequences to keep realistic meaning. The greatest improvement is in mixing sequences of the same lengths, increasing BLEU score by 0.6 (6.7 to 7.3).

1.1 Related Works

Data augmentation was originally developed in computer vision to improve the diversity of the training set by applying transformations of the existing data. For images, augmentation techniques such as crop, rotation or color normalization have been widely used (Krizhevsky et al., 2012).

Basic computer vision data augmentation approaches have been extended to natural language

processing in intuitive ways. Randomly erasing or replacing sections of an image is similar to randomly dropping words or replacing them with a synonym from a WordNet-sourced thesaurus (Zhang et al., 2015), a similar word as determined by k-nearest neighbors (Wang and Yang, 2015), a word sampled from the unigram distribution or a placeholder (Xie et al., 2017), or a word sampled from the output of a language model (LM) (Kobayashi, 2018). Geometric transformations of images – flipping, adjusting color channels, cropping or rotating, and shifting – has extended to different methods of rearranging words in a sentence (Artetxe et al., 2017). Combining images with Mixup extends to combining sentences, randomly selecting from which sentence to take the next token (Guo et al., 2020). Generating new images with translation between domains is akin to backtranslation: translating a sentence into a second language, then back to the original (Shorten and Khoshgoftaar, 2019).

Mixup, originally developed to combine images in computer vision (Zhang et al., 2018), has been extended to classification and translation tasks by interpolating sentences. Mixup linearly combines random pairs of sequences, combining source sentences into a single new source, and both target sentences into a single target. Different versions of mixup have been applied to word tokens (Guo et al., 2020), embeddings (Guo et al., 2020) (Jindal et al., 2020b), final hidden layers (Guo et al., 2019), all hidden layers (Guo et al., 2020) and specific hidden layers (Jindal et al., 2020a) (Yang et al., 2020). This paper specifically builds upon the SeqMix version of mixup, which averages the word embeddings of two sentences at each position. One constant between existing versions of mixup, including SeqMix, is that the pair of sentences to mix are chosen *at random* and, if done at the token or embedding-level, the pairs of words to mix between sentences are chosen *based on their position*.

Some of these methods implicitly attempt to maintain the syntax of a sentence: replacing a word with a synonym or otherwise similar word will likely result in a syntactically sound sentence. Others have explored the idea of explicit syntactic data augmentation, but in isolation from the semantic augmentation techniques described above. Min et al. (2020) switches the subject and object of a sentence to create augmented training data while retaining syntax, improving the generalization of BERT embeddings. Şahin and Steedman (2018) augment data by morphing dependency trees, maintaining local syntax.

The novelty of the part of speech based methods introduced in this paper is that they expand upon semantic augmentation while maintaining syntax.

2 Baseline Augmentation Methods

The augmentation methods developed in this paper are compared to eight baselines. The most basic has no augmentation (1. none), followed by randomly dropping tokens (2. drop, Iyyer et al. (2015)), randomly replacing tokens with a placeholder <UNK> (3. blank, Xie et al. (2017)), and randomly swapping words within a window size $k = 3$ (4. swap, Artetxe et al. (2017)). The next two replace tokens with a word sampled form the unigram frequency (5. smooth, Xie et al. (2017)) and from the output of an autoregressive language model (6. LM sample, Kobayashi (2018)). The latter is extend to randomly replacing a token's embedding with the weighted average of all embeddings using the probabilities from an autoregressive language model as weights (7. soft, Gao (2019)). Finally, position-based averaging of word embeddings between two sentences is considered (8. SeqMix Guo et al. (2020)).

The token-level augmentation methods (swap, drop, blank, smooth, and LM sample) are implemented before tokens are passed through the embedding layer of the encoder, meaning the original source sequence of tokens is altered. The embedding-level soft augmentation method is applied just after the embedding layer of the encoder, but before any other layer of the encoder. SeqMix is applied at the embedding-level to both source and target sequences; this is the only method discussed that changes the target. All non-SeqMix methods have a parameter $\gamma = 0.1$ as the probability that single token is augmented. SeqMix has a mixing parameter $\max(\lambda, 1 - \lambda)$, which determines how much weight is put on each sentence ($\lambda = 0.5$ would be equal mixing). The max operation guarantees that each sentence in the dataset will be weighted over 0.5 once per epoch. λ is sampled from a $Beta(0.01, 0.01)$ distribution for each sentence pair.

Although each epoch will contain the same number of sequences, each epoch will train on slightly different data because the augmentation is randomly applied to each batch. Note that augmenta-

tion is *only* performed in the training stage, and is never used in validation or generation.

3 New Augmentation Methods

This section proposes eight new data augmentation methods.

3.1 Smooth with part of speech

The baseline smooth method replaces a word in the source sequence with a token randomly sampled from the unigram distribution. In smooth with part of speech, the set of possible replacement words for word w_i with part of speech p_i is reduced to those that match the part of speech p_i. That is, the replacement w_j for word w_i is sampled from the conditional unigram distribution given p_i.

$$P_{smooth\_pos}(w_j|p_i) = \begin{cases} \frac{n_j}{\sum_{p_k=p_i} n_k} & \text{if } p_j = p_i \\ 0 & \text{if } p_j \neq p_i \end{cases}$$

3.2 LM sample with part of speech

The baseline LM sample method relies on a language model (LM) that has been separately trained on the training sentences in the source language. To replace a word w_i in a source sequence, previous tokens are passed into the LM as context, and the LM returns a probability distribution for its prediction of w_i. In the baseline LM sample method, the word is replaced by a word randomly sampled this probability distribution.

For LM sample with part of speech, the set of possible replacement words are limited to those that match part of speech of the original word. The replacement for word w_i with part of speech p_i is sampled from the conditional language model distribution given part of speech p_i. This filtered probability distribution is as follows, where $lm(w_j)$ is the probability distribution returned by the language model given the context corresponding to w_j.

$$P_{lm\_pos}(w_j|p_i) = \begin{cases} \frac{lm(w_j)}{\sum_{p_k=p_i} lm(w_k)} & \text{if } p_j = p_i \\ 0 & \text{if } p_j \neq p_i \end{cases}$$

3.3 Soft with part of speech

The baseline soft method uses the same probability distribution from a language model as LM sample. Rather than replacing a word with a single sampled token, the probabilities are used to create an expected word embedding as the weighted average of all embeddings. This expected embedding replaces the word's embedding in the sequence.

The soft method with part of speech instead uses the *conditional* probability distribution from section 2.3.2 to create the expected embedding of word w_i given its part of speech p_i.

$$\begin{aligned} \mathbb{E}_{soft\_pos}(e_i|p_i) &= \sum_{k \in |V|} e_k \times P_{lm\_pos}(w_k|p_i) \\ &= \frac{\sum_{p_k=p_i} e_k \times lm(w_k)}{\sum_{p_k=p_i} lm(w_k)} \end{aligned}$$

3.4 Wobbly SeqMix

Guo et al. (2020) proposed SeqMix which softly combines two sentence pairs to create a synthetic source and target sequence. Sentences are combined at the word embedding-level for both source and target sequences with a mixing parameter $\lambda \sim Beta(\alpha, \alpha)$ sampled for each mixing operation. Randomly sampled sentence pairs X and Y are combined to construct synthetic sentences Z:

$$\begin{aligned} Z_{\text{source}} &= \lambda X_{\text{source}} + (1-\lambda) Y_{\text{source}} \\ Z_{\text{target}} &= \lambda X_{\text{target}} + (1-\lambda) Y_{\text{target}} \end{aligned}$$

The ith word of each sentence, e.g. $X_{\text{target},i}$ and $Y_{\text{target},i}$, have their word embeddings combined into $Z_{\text{target},i}$ (with padding for extra words). The synthetic Z source and Z target are then passed into the encoder and decoder, respectively.

In wobbly SeqMix, random noise is added to λ for each position: $\lambda_i = \lambda + \mathcal{N}(0, 5\text{E-}3)$.

3.5 Wobbly SeqMix with part of speech-specific mixing parameter

In this case, the mixing parameter λ is independently sampled for each part of speech in each sentence pair, with the same λ_p for part of speech p applied to the source and target sequences.

3.6 SeqMix with part of speech matching

Instead of mixing words at each position i, this method mixes a word, e.g. $X_{\text{target},i}$, by matching part of speech with $Y_{\text{target},j}$ such that $p_{\text{target},i} = p_{\text{target},j}$. Again, the mixing is applied to the source and target sequences.

3.7 SeqMix with matching sentence length

Rather than selecting random pairs to mix together, sentence pairs are matched to have the same, or closest available, source length.

3.8 SeqMix with matching by K-means BERT similarity

Input sentences are clustered based on their hidden layer output of the SentenceTransformer, a BERT-based transformer model for capturing sentence-level meaning (Reimers and Gurevych, 2020). Instead of mixing at random, K-means is used to cluster sentences based on their Euclidean distances, and sentences are drawn within or across clusters to form similar or dissimilar pairs. K = 10 and K = 2 clusters were tested, where the bigger the K, the more similar the sentence pairs will be within each cluster.

4 Experiments

Eighteen total models are considered – eight baselines, five new part of speech-based methods, and five new sentence pairing-based methods. Beam search is used to translate the test corpus. BLEU (equal weights up to 4-gram) scores are compared. The candidate and target sequences are also converted to sequences of parts of speech (POS). The BLEU scores for these POS sequences are computed as a metric that loosely corresponds to how correct the syntax is.

4.1 Models

All augmentation methods are implemented on LSTM-based seq2seq model. All methods are also implemented on the transformer architecture, besides those based on SeqMix due to time constraints.

The LSTM architecture has a hidden size of 512 and embedding size 64. The transformer architecture replicates that introduced by Vaswani et al. (2017): N = 6 layers, each with h = 8 self-attention heads, a FFNN of dimension d_{ff} = 2048, embedding and hidden sizes of d_{model} = 512, and attention and feed-forward dropout $p_{att} = p_{ff} = 0.1$.

4.2 Data

The International Workshop on Spoken Language Translation (IWSLT) 2017 dataset (Cettolo et al., 2012) contains a multilingual TED Talks Machine Translation task from German to English, and has been used in the past to test the effectiveness of augmentation on translation tasks (Gao, 2019). These experiments use a 10% sample of the full IWSLT dataset due to compute power limitations. This subset has 20611 training, 89 validation, and 157 test sequences.

The `spacy` package was used to tag parts of speech for each source (using the German news pipeline) and target (using the English web pipeline) sequence (Honnibal and Montani, 2017).

4.3 Training

Translation models are trained using the Adam optimizer with learning rate 1E$-$3 and weight decay 1E-4 for 20 epochs. The learning rate was chosen by grid search using the base LSTM model to optimize best validation BLEU score over values ranging from 1E-2 to 1E-5. To avoid overfitting, the final model is selected as the epoch checkpoint with the best validation BLEU score. The LSTM model uses a batch size of 1 with gradient accumulation every 32 batches, while the transformer uses a batch size of 32.

The language model used in the LM sample and soft methods is a bidirectional, stacked (2 layers) LSTM with hidden size 64 where all final hidden and cell states are concatenated and passed through a linear and logsoftmax layer. It is trained the same way as the LSTM models, but selecting the final model with lowest validation loss. Matching the implementation in Gao (2019), the language model is trained only on the source sequences in the dataset. The language model is not changed while training the translation models.

All models were trained on Google Colab with a single NVIDIA Tesla P100-PCIE-16GB GPU.

4.4 Results and Conclusions

Table 1 shows test BLEU scores for the baseline and newly proposed token replacement augmentation methods, both when using LSTM and transformer architectures. The new methods to maintain part of speech show moderate improvements over their respective baselines in BLEU scores, up to a 0.4 point increase from soft to soft + POS with the LSTM, and up to 1.0 point increase from LM sample to LM sample + POS with the transformer. It is inconclusive whether maintaining part of speech improves the POS BLEU scores.

Table 2 shows test BLEU scores for the SeqMix syntax-based augmentation methods. SeqMix with same-length sentences achieved the highest BLEU score and offered an improvement over SeqMix. Wobbly SeqMix and wobbly + POS SeqMix did not substantially impact performance in BLEU score compared to SeqMix on its own. However, there is a slight improvement (0.2 points) when part of speech is added to wobbly SeqMix.

Augmentation Method *new methods italicized*	**LSTM BLEU**	**LSTM POS**	**Transformer BLEU**	**Transformer POS**
None	6.0	18.6	10.7	29.7
Drop	6.0	18.1	11.3	29.2
Blank	**6.6**	18.2	10.7	28.5
Swap	5.8	18.9	10.7	29.4
Smooth	6.2	18.7	11.5	**31.1**
Smooth + POS	6.3	18.9	**12.4**	30.4
LM	6.1	18.6	10.8	30.2
LM + POS	6.2	17.2	11.8	**31.1**
Soft	6.1	**19.1**	12.0	29.4
Soft + POS	6.5	18.0	**12.4**	30.9

Table 1: **Token replacement part of speech-based methods**, test BLEU scores and part of speech (POS) BLEU scores using LSTM and Transformer seq2seq. Scores are reported out of 100 BLEU. Comparing each new method to its respective baseline, and across all methods.

Augmentation Method *new methods italicized*	**LSTM BLEU**	**LSTM POS**
SeqMix	6.7	19.9
SeqMix + wobbly	6.6	19.8
SeqMix + wobbly + POS	6.8	19.0
SeqMix rearrangement POS	0.0	2.0
SeqMix + length match	**7.3**	**20.5**

Table 2: **SeqMix syntax-based methods**, test BLEU scores and part of speech (POS) BLEU scores using LSTM seq2seq. Scores are reported out of 100 BLEU.

Augmentation Method *new methods italicized*	**LSTM BLEU**
SeqMix + sim (K = 10)	2.4
SeqMix + diff (K = 10)	3.0
SeqMix + sim (K = 2)	4.9
SeqMix + diff (K = 2)	4.1

Table 3: **SeqMix sentence pairing by K-means BERT similarity methods**, test BLEU scores and part of speech (POS) BLEU scores using LSTM seq2seq. Similar (within cluster) and different (across cluster) pairs using $K = 2$ or $K = 10$ clusters. Scores are reported out of 100 BLEU.

The SeqMix rearrangement by POS method lead to complete loss of performance, suggesting that the rearrangement method was too dramatic a permutation to the mixed sentences.

Table 3 reports metrics considering similarity by k-means clustering on BERT-based embeddings. All of these methods showed worse performance compared to no augmentation and baseline SeqMix. However, comparing within these methods, $K = 2$ achieves better results than $K = 10$. This indicates that restricting matching to most similar clusters ($K = 10$) was worse than a more random method ($K = 2$), which is consistent with SeqMix ($K = 1$) being better than both.

In summary, results are promising for the smooth POS, LM POS, soft POS, and length matching SeqMix methods developed in this paper.

5 Future Research

The most direct next step would be applying methods developed here to the full dataset with a longer training time, to see how the performance is replicated. Aside from a more refined training regime, there are several avenues of future research.

The part of speech BLEU score was a basic attempt to measure the correctness of syntax, but future research into syntax-specific metrics and tasks would be beneficial.

Similar results to LM sample + POS and soft + POS may be obtained by applying LM sample and Soft using a language model trained with a loss function rewarding correct part of speech categorization. Alternatively, a language model could be trained on two tasks: next word prediction and part of speech classification.

Rather than positional or part of speech based mixing, one could look at matching dependency parse or semantic role labels (SRL). These methods would be relevant to both the selection of which sentences to mix and as alternatives to the position-based mixing of words.

6 Impact Statement

This paper used 10% of an existing dataset (IWSLT2017) specifically designed for machine translation tasks. This paper uses the German-to-English set, but researchers could apply the same methods to other language pairs.

The quality of the dataset has been validated, as it is produced by the International Conference on Spoken Language Translation, which is a premier scientific conference that specializes in the creation of data suites, benchmarks and metrics in the field.

We have proposed and experimented with 8 new methods of data augmentation for neural machine translation. They have covered both syntactic and semantic variations, based on a thorough survey of the field. Given the amount of data and the constraint on computing resources, not all methods showed improvement over the baseline, but they offered a useful road map for future explorations.

The methods not using part of speech could also be useful for translation of low-resource languages. However, our part of speech based methods are exclusive to languages with preexisting part of speech taggers. Because these methods would not be available to every language, if implemented in user-facing products, it would not benefit everyone equally. This inequity for low-resource languages may disproportionately affect populations that already experience marginalization.

We don't see any new bias issues with these data augmentation methods themselves, but all existing ethical pitfalls of NLP and particularly machine translation are still at play. This includes perpetuating bias that is present in training data, such as gender biases for occupations as described in Bolukbasi et al. (2016).

This project does not introduce any new datasets or involve any identity characteristics.

References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *ACL*, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Fei Gao. 2019. Soft contextual data augmentation for neural machine translation.

Demi Guo, Yoon Kim, and Alexander Rush. 2020. Sequence-level mixed sample data augmentation. *EMNLP*.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. *ACL*.

A. Jindal, D. Gnaneshwar, R. Sawhney, and R. R. Shah. 2020a. Augmenting nlp models using latent feature interpolations.

A. Jindal, D. Gnaneshwar, R. Sawhney, and R. R. Shah. 2020b. Leveraging bert with mixup for sentence classification. 34(10):13829–13830.

Sasuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. *NAACL*, pages 452–457.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. pages 1097–1105.

Junghyun Min, Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. *ACL*, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

Connor Shorten and Taghi Khoshgoftaar. 2019. A survey on image data augmentation for deep learning.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *NIPS*, pages 6000–6010.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. *EMNLP*, pages 2557–2563.

Ziang Xie, Ikbal Sida, Jiwei Li, Daniel Levy, Aiming Nie, Dan Jurafsky, and Andrew Ng. 2017. Data noising as smoothing in neural network language models.

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

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

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *NIPS*, pages 649–657.

Gözde Gül Şahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. *NIPS*, 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.