Controllable Macaronic Language Generation for Language Learning

Adithya Renduchintala and **Philipp Koehn** and **Jason Eisner**
Center for Language and Speech Processing
Johns Hopkins University
{adi.r,phi}@jhu.edu jason@cs.jhu.edu

Abstract

We introduce a new machine translation task for generating macaronic language target sentences. Our high-level goal is to serve adult students of a foreign language who cannot yet read fluently in the target language. Mixed-language translation can be used to obtain practice reading material that is engaging and appropriate for their skill level. Our method allows learners or teachers to control which parts of a source sentence should be translated.

We examine domain-adversarial training to overcome the inherent reluctance of an encoder-decoder model to switch languages within a sentence and devise a simple yet effective method to create synthetic macaronic language training, validation, and test data.

1 Introduction

Learning a foreign language is a daunting task. Even with the availability of numerous self-directed "gamified" learning applications such as Duolingo (von Ahn, 2013), students face an uphill challenge. The lack of engaging learning/practice materials is particularly challenging for adult beginner students. Students often have to undergo tedious memorization to acquire and retain basic vocabulary in order to advance to the next stage. Reading macaronic language documents is a way for students to instead acquire vocabulary in context. Such documents are used in the recent Swych (2015) and OneThirdStories (2018) apps.

We envision a macaronic language MT system being used in foreign language instruction. To create macaronic language materials for a unit on body parts, for example, the teacher may find a narrative of a medical checkup written in the student's native language, and mark words that are to be translated into the foreign language. Alternatively, the teacher could start with a foreign-language narrative and translate difficult parts back into the student's native language to make the text comprehensible. Or the student could select these difficult parts for themself, by clicking as they read the text; or they could be selected automatically by a personalized macaronic language based instructional framework (Labutov and Lipson, 2014; Renduchintala et al., 2016a,b).

Markup	That $\overset{de}{\underline{\text{shows}}}$ you that we have people who $\overset{de}{\underline{\text{are capable of doing great things.}}}$
Output	That *zeigt* you that we have people who *fähig sind, großartige Dinge zu bewirken.*

Table 1: Overview of our controllable macaronic language generation task. The user (a teacher or a student) should be able to "markup" portions of the text that get translated. The unmarked text remains in English.

The focus of this paper is the development and evaluation of NMT systems tailored to generate macaronic language target translations and respect source-side annotations, which indicate desired level of mixing. Table 1 gives an overview of the task. The markup indicates which portions of the sentence the system should translate. These portions need not be linguistically identifiable as phrases. Notice that the translations are not merely word replacements (glosses); the output translates the phrase with reordering that is consistent with the target language (German) word order.

2 Method

We propose two augmentations to the LSTM based encoder-decoder architecture for neural MT (Bahdanau et al., 2014; Luong et al., 2015; Hochreiter and Schmidhuber, 1997). The first seeks to endow the system with the flexibility to switch languages while generating a target sequence. The second makes it pay attention to input annotations that indicate the target language for each token.

2.1 Target Language Adversarial Training

The decoder of an NMT can be thought of as a language model that is conditioned on the source input. Sequential dependencies in the output sequence are explicitly encoded into the auto-regressive generation of each target token. Thus, in the usual setting where the decoder is trained on monolingual output sentences, it will learn to output such sentences. This is true even if the monolingual training sentences are not all in the same language.

Could we encourage the NMT model to output macaronic language sentences even when it is trained only on a mix of monolingual sentences? We propose applying domain-adversarial multi-task training (Ganin et al., 2016) to solve this issue. The auto-regressive property dictates that an output token y_j conditions on the decoder hidden state $\mathbf{s}_j$, where j is the output word index. We want our model to be auto-regressive in terms of the target tokens but agnostic as to the *language* of these tokens. That is, we want the decoder hidden states $\mathbf{s}_j$ to retain syntactic and semantic information about the previous tokens but to forget their target language.

When training the NMT system to produce a target sequence $\mathbf{y}$, we also know the language—$\texttt{A}$ or $\texttt{B}$—of each word in $\mathbf{y}$. Let $\mathbf{g} \in \{\texttt{A},\texttt{B}\}^{|\mathbf{y}|}$ denote this sequence of labels. We train a separate neural module that tries to predict the language label g_j from the NMT decoder's hidden state $\mathbf{s}_j$(Equation 1):

$$\mathcal{L}_{\text{dis}} = \log p(\mathbf{g}_{0:J} \mid \mathbf{s}_{0:J}; \boldsymbol{\phi}, \boldsymbol{\theta}) \qquad (1)$$

As the language discriminator depends on the decoder state, this objective depends on all of the NMT parameters $\boldsymbol{\theta}$, as well as on the discriminator's own parameters $\boldsymbol{\phi}$ (which are fewer).

The discriminator parameters $\boldsymbol{\phi}$ are trained to maximize $\mathcal{L}_{\text{dis}}$ (Equation 2), but the NMT parameters $\boldsymbol{\theta}$ are adjusted to minimize $\mathcal{L}_{\text{dis}}$ while maximizing translation accuracy. We follow Ganin et al. (2016) and apply the reverse gradient $\frac{-\partial \mathcal{L}_{\text{dis}}}{\partial \boldsymbol{\theta}}$ by using a gradient reversal layer (GRL). In short, we update the two sets of parameters via

$$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \lambda \frac{\partial \mathcal{L}_{\text{dis}}}{\partial \boldsymbol{\phi}} \qquad (2)$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \frac{\partial \mathcal{L}_{\text{NMT}}}{\partial \boldsymbol{\theta}} - \lambda \cdot \frac{\partial \mathcal{L}_{\text{dis}}}{\partial \boldsymbol{\theta}} \qquad (3)$$

2.2 Desired Language Source Features

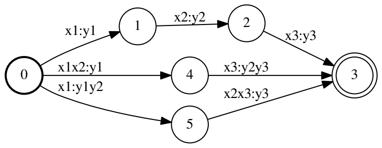

Figure 1: An example FST L which encodes phrase alignments between a source $[x_1, x_2, x_3]$ and target $[y_1, y_2, y_3]$. Path $0,1,2,3$ encodes single-word phrases, while $0,5,3$ for example enodes a one-to-many phrase $x_1 : y_1, y_2$ and a many-to-one phrase $x_2, x_3 : y_3$.

Our next goal is to give the end user of the macaronic language translation model the ability to control which words/phrases should be translated. We realize this by adding source features to indicate the desired language on the target side. Each token in the source is annotated with a feature to specify whether the token should be left in the source language $\texttt{A}$ or translated into language $\texttt{B}$. These are not hard constraints, however, as we want the model to retain enough flexibility to produce "fluent" macaronic language translations.

We augment the standard NMT model to accept an additional stream of features $\mathbf{f} \in \{\texttt{A},\texttt{B}\}^{|\mathbf{x}|}$ along with the source sequence $\mathbf{x}$. We follow the approach in Sennrich and Haddow (2016) and concatenate the word embedding and feature embedding and then project the concatenated vector back to the size of the original word embedding:

$$\mathbf{h} = \text{BiLSTM}([\text{e}(\mathbf{x}; \boldsymbol{\theta}_{\mathbf{e}}); \text{e}(\mathbf{f}, \boldsymbol{\theta}_{\mathbf{f}})]) \qquad (4)$$

where $\text{e}(.; \boldsymbol{\theta})$ is an embedding function with two sets of parameters: $\boldsymbol{\theta}_{\mathbf{e}}$ for source tokens and $\boldsymbol{\theta}_{\mathbf{f}}$ for source-language features.

3 Synthetic Data Creation

3.1 Multilingual Data

For each $([x_1, x_2, \ldots, x_n], [y_1, y_2, \ldots, y_m])$ training pair in the original dataset, we create two new training examples (see examples in Multilingual row in Table 2) with the desired language features in the input and language labels in the output. The "copy" example has input-output pair of the form $([x_1|\texttt{A}, x_2|\texttt{A}, \ldots, x_n|\texttt{A}], [x_1|\texttt{A}, x_2|\texttt{A}, \ldots, x_n|\texttt{A}])$ and the "translate" example has the form $([x_1|\texttt{B}, x_2|\texttt{B}, \ldots, x_n|\texttt{B}], [y_1|\texttt{B}, y_2|\texttt{B}, \ldots, y_m|\texttt{B}])$.

Our hope is that the modified NMT model will learn that whenever an input token x_j is associated with $f_j = \texttt{B}$, the corresponding output token(s) are a translation of the input, and when the feature is $f_j = \texttt{A}$ the source token appears in the target unchanged.

3.2 Macaronic Language Training Data

We also consider including synthetic training examples in which the target sequence is itself macaronic language. We first word-align training bitext using FastAlign (Dyer et al., 2013) and then apply the "grow-diagonalize-finalize" heuristic (Koehn et al., 2003). We then extract consistent phrase pairs from each source-target pair (Och and Ney, 2000; Koehn et al., 2003). The extracted phrase pairs are encoded as a Finite State Transducer (FST) P. Composing the phrase translation machine P, with a source FST X (on the input) and target FST Y on the output side results in a phrase-segmentation lattice FST L. An example lattice is shown in Figure 1.

$$L = X \circ P \circ Y \qquad (5)$$

Each path $\mathbf{p}$ through L defines a unique phrase-segmentation mapping between the source sentence and target sentence. We go over each arc in $\mathbf{p}$and construct a new source sequence (with a desired language feature) and a new macaronic language target sequence, which corresponds to the source sequences' desired language feature (see Algorithm 1) using a biased coin flip $r \sim B(1,\gamma)$. The training examples from the macaronic language creation procedure are shown in the "Mixed" section of Table 2.

Algorithm 1 Mixed-Lang. Extraction

```
Require: $L$                                   ▷ Phrase-segmentation Lattice
Require: A, B                                  ▷ Src-Tgt Language labels
$\mathbf{x},\mathbf{y} = [],[]$                ▷ initial source and mixed-target
$\mathbf{p} \sim L$                            ▷ sample path form Lattice
for $x:y \in$ SortedArcs($\mathbf{p}$) do
    $r \sim B(1,\gamma)$
    if $r = 1$ then
        $\mathbf{x} += [w|\mathtt{B} \;\; \forall \;\; w \in x]$    ▷ add B feature to input
        $\mathbf{y} += [w|\mathtt{B} \;\; \forall \;\; w \in y]$    ▷ place Tgt phrase in output
    else
        $\mathbf{x} += [w|\mathtt{A} \;\; \forall \;\; w \in x]$    ▷ add A feature to input
        $\mathbf{y} += [w|\mathtt{A} \;\; \forall \;\; w \in x]$    ▷ place Src phrase in output
return$\{\mathbf{x},\mathbf{y}\}$              ▷ new source and mixed target pair
```

4 Related Work

Our work is inspired by domain adaptation in NMT. Specifically the notion of domain mixing via adversarial training and target-side domain features introduced in Britz et al. (2017). Tars and Fishel (2018) also treat multi-domain models as multi-lingual models and examine different source and target featurization (Östling and Tiedemann, 2017; Johnson et al., 2017).

The use of source-side features in domain adaptation (Kobus et al., 2016; Zeng et al., 2018) and for NMT in general (Sennrich and Haddow, 2016) has also been studied recently. Using source-side features as explicit control has also been presented in Sennrich et al. (2016).

To the best of our knowledge, extensions of these methods to produce mixed language have not been attempted. Furthermore, in all previous applications, training data was freely available, and there was no requirement to create synthetic data.

5 Experiments

5.1 Data

We use the En-De dataset from the International Workshop on Spoken Language Translation (IWSLT) 2014 dataset, which contains approximately 167k training, 7.2k validation, and 6.7k test sentence pairs (Cettolo et al., 2014). We first convert the training data into 334k Multilingual training examples using the method described in Section 3.1. Next we synthetically create our macaronic language training data using the method described in Section 3.2. We set $\gamma = 0.5$ in Algorithm 1, Out of the 167k original training examples, we extract 145k macaronic language training examples. The reduction in examples is due to sentence pairs with "null" word alignments in either the source or target side and/or inadequate phrase size. Both these conditions result in a phrase-segmentation lattice (Equation 5) with no paths. The maximum phrase size in the phrase-extraction procedure was set to 10. A smaller phrase-limit drastically increases the number of "null" lattices. Furthermore, a large phrase size exposes the NMT model to more word reordering and will help the model retain performance for full translation. We also convert the IWSLT validation data into macaronic language examples and set $\gamma = 0.5$.

5.2 NMT Model

We used a 2-layer biLSTM encoder and a 2-Layer LSTM decoder model with input-feeding attention mechanism for all our experiments (Luong et al., 2015). The embedding size, encoder, and decoder recurrent size was set to 512 (256 in the encoder for each direction). Dropout of 0.2 was used between each computational block of the NMT model (Srivastava et al., 2014). We rescale the dimensions of our updates (2)–(3) using the Adam optimization method (Kingma and Ba, 2014).

	Source Text (Lang)	Target Text(Lang)
Original	That is great (en)	Das ist großartig
Multilingual	That|en is|en great|en That|de is|de great|de	That is great Das ist großartig
Macaronic Language	That|de is|de great|en That|de is|en great|de That|en is|en great|de . . .	Das ist great Das is großartig That is großartig . . .

Table 2: Synthesized training examples from the original source-target pair. For each training example there are two possible multilingual training examples, one that completely copies the input to the output and another that completely translates the input. Mixed-language examples on the other hand can be numerous. We show 3 example macaronic language examples for the sentences pair. Details on how we extract macaronic language examples are discussed in Section 3.2. The list of mixed examples in the table is not exhaustive.

Training	Model	
Data	Baseline	Adversarial
Multilingual	53.18	55.28
Mixed-language	56.48	56.75
Mixed+Multilingual	61.22	60.95
Lex	54.79	

Table 3: Lower-cased BLEU of NMT Models trained and tested on synthetic mixed-translations outputs.

5.3 Synthetic Validation Results

Unlike standard translation tasks, our task does not have any "gold" macaronic language references, which forces us to convert existing validation data into macaronic language data to tune our models. We use BLEU as our evaluation metric and treat the synthesized macaronic language targets as reference.

Table 3 shows ablation results of NMT models trained with only Macaronic Language data, only Multilingual data and both Mixed and Multilingual data on the synthetic validation data. We see that exposing the NMT system to macaronic language training is vital to performance. Using the target side adversarial loss (Adv) improves the BLEU score by ~ 2 points when no macaronic language training is used. However, when the models are exposed to macaronic language outputs during training, the adversarial loss does not make an impact, suggesting that the adversarial loss is most useful to counter the mismatch between training and testing.[1] For qualitative analysis, the inputs and model predictions from the Mixed+Multilingial model are shown in Table 4.

We also compare these models with a simple lexical look-up model, which picks the most likely German tokenif the English token has source feature B, or simply copies the English word to the output if it has source feature A. The BLEU evaluation metric rewards the models for predicting output tokens in the correct sequence, regardless of whether they were translations or merely copied.[2]

5.4 Synthetic Test Set Results

Again we use synthetic data for test set evaluation extracted from the IWSLT test data. For a fair evaluation, we use a word-alignment model trained on a separate dataset instead of the IWSLT training data (which was used to create our synthetic training data). This ensures that our test set does not implicitly find phrase segmentations, which the training data has seen. We use the first 100k sentence pairs from the WMT 2014 English-German data to extract word alignments (and from them phrase-segmentations) for the synthetic test set (Bojar et al., 2014). Furthermore, we analyze our models with test data created over the range of $\gamma = [0,1]$ in 0.1 increments. At each γ we generate 5 different test sets. We evaluate each data ablation model trained with adversarial loss on these test sets. The test results are shown in Figure 2. We find that the lexical look-up model which was a reasonable contender at small γ sharply deteriorates after 0.5. We see that multilingual trained model suffers in the middle ranges of γ when compared against the macaronic language trained models, but recovers again as γ is close to 1. We are encouraged to see that the combination of both data types sustains performance over the entire range of mixing, indicating that the

[1] We searched over five λ values 0.1,0.2,1.0,5.0,10.0 and settled on 1.0.

[2] For the lexical look-up model the English output is guaranteed to be correct making it very competitive at small γ values

Markup	He gave a talk about how education $\overset{de}{\underline{\text{and school kills creativity.}}}$
Prediction	He gave a talk about how education *und schulen kreativität tötet* .
Markup	$\overset{de}{\underline{\text{It was sombody who was trying}}}$ to ask a question about Javascript.
Prediction	*Es war jemand , der versuchte* , to ask a question about Javascript .
Markup	$\overset{de}{\underline{\text{We were standing on the edge}}}$ of thousands of acres of $\overset{de}{\underline{\text{cotton}}}$.
Prediction	*Wir standen am rande* of thousands of acres of *baumwolle* .
Markup	And we're building upon innovations of $\overset{de}{\underline{\text{generations who went before us.}}}$
Prediction	And we're building upon innovations of *generationen , die vor uns gingen* .

Table 4: Examples of inputs and predicted outputs by our NMT model trained on both mixed and multilingual data with adversarial loss. We see that the macaronic language translations are able to correctly order German portions of the sentences, especially at the sentence ending. The source-features have also been learned by the NMT model and translations are faithful to the markup. The case, tokenization and italics added in post.

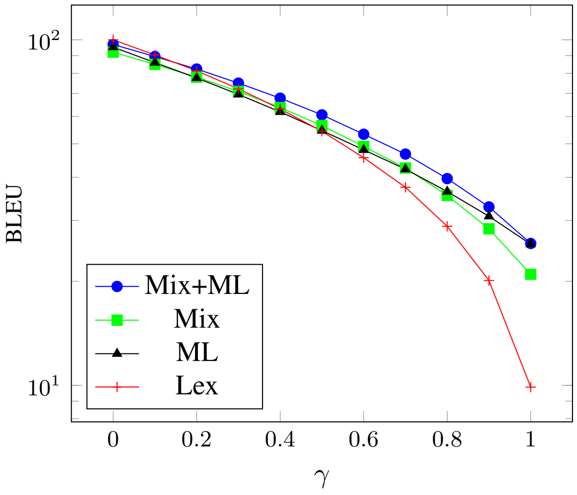

Figure 2: Lower-cased BLEU scores for different models on synthetic test sets with different γ values(See Algorithm 1). Each point is an average of 5 different test sets, each generated with the specific γ value.

different data types have a complementary effect.

6 Conclusion

We introduce a new task of macaronic language generation and propose a method to create synthetic training, validation, and test data to learn macaronic language generating models. Our approach substantially outperforms a word look-up based baseline (61.2 vs 54.7 BLEU points). Even when trained on parallel text without synthetic mixed data, we outperform the baseline with our language-agnostic adversarial loss.

We hope to use macaronic language generation to further the creation of personalized practice material for language learners. We leave this downstream evaluation for future work.

References

Luis von Ahn. 2013. Duolingo: Learn a language for free while helping to translate the web. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, pages 1–2.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pages 12–58.

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126.

M Cettolo, J Niehues, S Stüker, L Bentivogli, and M Federico. 2014. Report on the 11th IWSLT evaluation campaign, IWSLT 2014. In *IWSLT-International Workshop on Spoken Language Processing*, pages 2–17. Marcello Federico, Sebastian Stüker, François Yvon.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM Model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association of Computational Linguistics*, 5(1):339–351.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Catherine Kobus, Josep Crego, and Jean Senellart. 2016. Domain control for neural machine translation. *arXiv preprint arXiv:1612.06140*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Igor Labutov and Hod Lipson. 2014. Generating code-switched text for lexical learning. In *Proceedings of ACL*, pages 562–571.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421.

Franz Josef Och and Hermann Ney. 2000. A comparison of alignment models for statistical machine translation. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*, volume 2.

OneThirdStories. 2018. Onethirdstories. `https://onethirdstories.com/`. Accessed: 2019-02-20.

Robert Östling and Jörg Tiedemann. 2017. Continuous multilinguality with language vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 644–649.

Adithya Renduchintala, Rebecca Knowles, Philipp Koehn, and Jason Eisner. 2016a. Creating interactive macaronic interfaces for language learning. In *Proceedings of ACL (System Demonstrations)*.

Adithya Renduchintala, Rebecca Knowles, Philipp Koehn, and Jason Eisner. 2016b. User modeling in language learning with macaronic texts. In *Proceedings of ACL*.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, volume 1, pages 83–91.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Swych. 2015. Swych. `http://swych.it/`. Accessed: 2019-02-20.

Sander Tars and Mark Fishel. 2018. Multi-domain neural machine translation. *arXiv preprint arXiv:1805.02282*.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 447–457. Association for Computational Linguistics.